AVEVA™ Communication Drivers Pack – Standards - Gateway Driver

User Guide

Although precaution has been taken in the preparation of this documentation, AVEVA 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 AVEVA. 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. AVEVA, the AVEVA logo and logotype, OSIsoft, the OSIsoft logo and logotype, ArchestrA, Avantis, Citect, DYNSIM, eDNA, EYESIM, InBatch, InduSoft, InStep, IntelaTrac, InTouch, Managed PI, OASyS, OSIsoft Advanced Services, OSIsoft Cloud Services, OSIsoft Connected Services, OSIsoft EDS, PIPEPHASE, PI ACE, PI Advanced Computing Engine, PI AF SDK, PI API, PI Asset Framework, PI Audit Viewer, PI Builder, PI Cloud Connect, PI Connectors, PI Data Archive, PI DataLink, PI DataLink Server, PI Developers Club, PI Integrator for Business Analytics, PI Interfaces, PI JDBC Driver, PI Manual Logger, PI Notifications, PI ODBC Driver, PI OLEDB Enterprise, PI OLEDB Provider, PI OPC DA Server, PI OPC HDA Server, PI ProcessBook, PI SDK, PI Server, PI Square, PI System, PI System Access, PI Vision, PI Visualization Suite, PI Web API, PI WebParts, PI Web Services, PRiSM, PRO/II, PROVISION, ROMeo, RLINK, RtReports, SIM4ME, SimCentral, SimSci, Skelta, SmartGlance, Spiral Software, WindowMaker, WindowViewer, and Wonderware are trademarks of AVEVA and/or its subsidiaries. All other brands may be trademarks of their respective owners.

U.S. GOVERNMENT RIGHTS

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the license agreement with AVEVA Group Limited or its subsidiaries and as provided in DFARS 227.7202, DFARS 252.227-7013, FAR 12-212, FAR 52.227-19, or their successors, as applicable.

AVEVA Third Party Software Notices and Licenses: https://www.aveva.com/en/legal/third-party-software-license/

Publication date: Thursday, October 19, 2023

Publication ID: 868897

Contact information

AVEVA Group Limited
High Cross
Madingley Road
Cambridge
CB3 0HB. UK

https://sw.aveva.com/

For information on how to contact sales and customer training, see https://sw.aveva.com/contact.

For information on how to contact technical support, see https://sw.aveva.com/support.

To access the AVEVA Knowledge and Support center, visit https://softwaresupport.aveva.com.

Contents

Chapter 1

Introduction to the Gateway Communication Driver

About the Gateway Communication Driver

The Gateway Communication Driver acts as a communication protocol converter, to link clients and data sources that communicate using different protocols. The Gateway is hosted by the OI Server Manager, a Microsoft Management Console (MMC) snap-in, which is part of the Operations Control Management Console (OCMC) suite of utilities.

This user assistance publication covers only the information you need to configure and run the Gateway component. The documentation that accompanies the related components provide details on their operation. For a better understanding of this user guide, we recommend you to read the documentation for both the MMC and the Core. To read these documents in the OCMC, click **Action** and select the **Help** command from the menu. An Adobe Acrobat version of the Communication Drivers Pack Help 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, see the MMC Help.

Supported Data Sources and Clients

The Gateway Communication Driver allows Windows applications to access data from various data sources. The following matrix indicates supported source-client mappings.

Servers	**Clients**					
	OPC	**SuiteLink**	**DDE**	**FastDDE v2**	**FastDDE v3**	**MQTT**
OPC v2.05 Server	N/A	Yes	Yes	No	Yes	Yes

SuiteLink Server	Yes	N/A	Yes	No	No	Yes
ArchestrA	Yes	Yes	Yes	Yes	Yes	Yes
DDE Server	Yes	Yes	N/A	No	No	Yes
FastDDE v2 Server	Yes	Yes	No	N/A	No	Yes
FastDDE v3 Server	Yes	Yes	No	No	N/A	Yes
InTouch	Yes	Yes	Yes	No	No	Yes
OPC UA Server	Yes	Yes	Yes	No	No	Yes
MQTT Publisher	Yes	Yes	Yes	Yes	Yes	Yes

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

- To use ArchestrA data source, Gateway Communication Driver must be located on the same node as ArchestrA.
- To use DDE/FASTDDE data sources:
  - All versions of DDE must be local.
  - FastDDE v2 supports value data only.
  - FastDDE v3 supports VTQ (value, time, quality).
- InTouch v7.11 and higher is supported.

Note: NetDDE is not supported.

Supported Client Protocols

The client applications connect to Gateway Communication Driver using the following protocols:

- OPC
- SuiteLink
- DDE/FastDDE
- Message Exchange
- OPC UA
- MQTT
- PCS

OPC

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

OPC defines a common, high-performance interface that permits exchange of data with field devices, and be reused by HMI, SCADA, control and custom applications. Over a network, OPC uses Distributed COM (DCOM) 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.
- The performance monitor of the operating system allows access to extensive diagnostics of the data throughput, server loading, computer resource consumption, and network transport. 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 standard WinSock interface.

DDE/FastDDE

The DDE/FastDDE communication protocols allow communication between a client and a server. Dynamic Data Exchange (DDE) protocol is developed by Microsoft whereas FastDDE protocol is proprietary to AVEVA. For DDE/FastDDE communications the Communication Driver must be activated in Desktop mode (must start from command line).

DDE

DDE is a communications protocol that 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 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.

Note: On Windows Vista and later operating systems, Local DDE is supported only when the Communication Driver is activated from its executable file or launched from InTouch. Local DDE is not supported when the Communication Driver is activated from the OCMC.

FastDDE

FastDDE provides a means of packing multiple DDE messages into a single message. This packing improves efficiency and performance by reducing the total number of DDE transactions required between a client and a server.

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

Message Exchange

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

OPC UA

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

OPC UA is implemented as a client in Gateway Communication Driver.

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

Classic OPC	**OPC UA**
• 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 • Is dependent on Windows operating systems • Has limited security • No built-in capabilities to handle problems, such as lost messages	• Uses a services architecture to export data, which improves the ease of communication and connectivity • Is platform independent and can connect to a wide variety of devices and platforms • Has built-in security • 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 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.

- **MQTT Subscription:** If you want to subscribe to MQTT data, it is highly recommended you use the MQTT Communication Driver as it can subscribe MQTT payload in either JSON or Sparkplug format.
- **MQTT Publishing:** The Gateway Communication Driver provides MQTT publishing only in JSON format. If you want to publish data from System Platform, you should use the publishing functionality in the MQTT Communication Driver as it can publish MQTT payload in either JSON or Sparkplug format.

Chapter 2

Configuring the Gateway Communication Driver

Before Configuring the Gateway

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

- Windows Server operating system ICMP(v4) Echo Request may be disabled. You must enable this for an OPC client such as Gateway Communication Driver to connect to the server.
- To determine the status of ICMP (v4) Echo on the server, open a command prompt on the Gateway Communication Driver computer and type:

  ```
  ping SERVER_COMPUTER
  ```

  where,

  SERVER_COMPUTER: 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.

Setting up the Gateway Communication Driver for the First time

If you are setting up the Gateway Communication Driver for the first time, perform the following tasks in the order listed:

1. To install the Gateway Communication Driver, double-click the **Setup.exe** file.
2. Start the Operations Control Management Console (From the Windows **Start** menu, point to **Programs**, **AVEVA**, and then click the **Operations Control Management Console** icon).
3. In the OI Server Manager tree, under the Local node, navigate to the Gateway Communication Driver in the OCMC. For general information about working in this snap-in environment, see the Communication Drivers Pack Help.

4. Configure the global parameters.

   Note:
   - The default Poke Mode settings for Gateway Communication Driver 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. Global parameters that appear dimmed are either not supported or cannot be configured in Gateway Communication Driver. Simulation Mode is not supported.

   - The default value of the **Buffered Data (Maximum Queued Updates)** parameter is 10 for the Gateway Communication Driver.

   For more details, see Configuring Global Parameters in the Communication Drivers Pack Help.

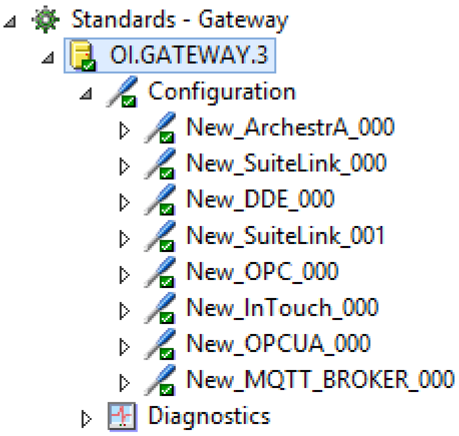
5. Before activating Gateway Communication Driver for connection, you must first build and configure a hierarchy of one or more data sources to establish communications between sources and clients. See Determining the Gateway Communication Driver Data Source Hierarchy.
6. Add one or more groups and topics object for each data source.
7. Add one or more device groups and items. See Configuring Device Item Definitions.
8. Activate the Gateway Communication Driver.See Activating/Deactivating the Gateway Communication Driver.
9. Troubleshoot any problems. See Troubleshooting.

The basic rules for configuring/activating the Gateway Communication Driver are:

- Only one instance of Gateway Communication Driver can run per node without license.
- Gateway Communication Driver can be activated and deactivated using the OI Server Manager snap-in.
- Gateway Communication Driver can be activated as a COM Server (OPC Server) using standard COM activation mechanisms.
- Gateway Communication Driver can be run only out-of-proc within OPC clients.
- Gateway Communication Driver can communicate only with ArchestrA data source components delivered with Application Server v2.0 and later. Earlier versions of Application Server are not supported.

Determining the Gateway Communication Driver Data Source Hierarchy

Before attempting to configure GatewayCommunication Driver , determine the hierarchical structure of the data sources you wish to use.The data source configuration part of Gateway Communication Driver hierarchy begins under the Configuration branch.



Standards - Gateway
OI.GATEWAY.3
Configuration
New_ArchestrA_000
New_SuiteLink_000
New_DDE_000
New_SuiteLink_001
New_OPC_000
New_InTouch_000
New_OPCUA_000
New_MQTT_BROKER_000
Diagnostics

When you are viewing the configuration hierarchy of Gateway Communication Driver and someone views the same 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 select a hierarchy away from the original hierarchy).

Object Naming Convention within a Hierarchy

- The format of the default name of an hierarchy object is in the format

  `New_<ObjectName>_###`

  where,

  - **<ObjectName>**: name of the object type
  - **###**: 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.

Chapter 3

Connecting to Data Sources

Connecting to an ArchestrA Data Source

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

Refer to Configuring the Gateway Communication Driver for a general overview about configuring data sources in Gateway Communication Driver.

Configuring an ArchestrA Data Source Object

To add an ArchestrA data source object to your Gateway Communication Driver hierarchy

1. Right-click **Configuration** in the hierarchy, and select **Add ArchestrA Connection**.
   - A new object is created in the hierarchy tree and is named **New_ArchestrA_000** by default (in "edit mode"). Rename it, if desired. You are allowed to add only one ArchestrA data source.

   The **New_ArchestrA_000 Parameters** configuration view (right pane) is displayed.
2. Configure the new ArchestrA object according to the following option definitions:
   - **Device Group Name**: Name of the topic to which DDE or SuiteLink clients of Gateway Communication Driver connect in order to access items in the ArchestrA data source.

     Default value is ArchestrA (this cannot be edited).
   - **Reconnect Attempts**: Number of times Gateway Communication Driver attempts to reconnect to the specified data source if a connection fails.

     Default value is 3.

Minimum/maximum range is -1 to 1,000,000. Entry of a value that is excessively out of the allowed range will display an error message about illegal format.

The value (-1) means no limit to the number of attempts. The value Zero (0) means no attempts.

- **Reconnect Period**: Delay (in ms) between reconnection attempts if a connection fails.

  Default value is 30000 ms. Minimum/maximum range is 10,000 to 300,000 ms (corresponding to the range of 10 sec to 5 min). Entry of a value beyond the allowed range will display an error message about illegal format.

- **Write Credentials**: User credentials created in ArchestrA for write qualifications.
- **Read Only**: Check this box for the Gateway Communication Driver to make all the items connected through the ArchestrA data source as 'read-only'. This is in addition to other read-only conditions that ArchestrA imposes. Clearing the selection removes the read-only condition for the item imposed by the Gateway Communication Driver. By default, the **Read-Only** check-box is selected.
- **Domain:** Enter the domain name to logon to ArchestrA. If the **Read Only** box is unchecked and ArchestrA has security enabled, enter the domain name as configured in ArchestrA.

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

- **User Name and Password:** These options, along with **Domain**, comprise the credentials used to logon to ArchestrA. If the **Read Only** box is unchecked and ArchestrA has security enabled, you must enter valid credentials as configured in ArchestrA.

  Note: ArchestrA user login data is not hot-configurable. Gateway Communication Driver must be restarted for the new values to take affect.

Configuring an ArchestrA Group Object

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

To add a group object to your ArchestrA data source hierarchy

1. Right-click the new data source object, and select **Add ArchestrAGroup Connection**.
   - A new object is created in the hierarchy tree and is named **New_ArchestrAGroup_000** by default (in "edit mode"). Rename it, if desired. You are allowed to add up to 100 new group objects.

   The **New_ArchestrAGroup_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 Gateway Communication Driver connect to in order to access items at the ArchestrA group. Default value is the concatenation of the names of the ArchestrA object and the group object. This cannot be edited.
   - **ArchestrA Item ID Prefix**: A string prefixed to item names added through this ArchestrA group. For instance, a prefix of "Blower_" is 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 for the Gateway Communication Driver to make all the items connected through the ArchestrA data source as 'read-only'. This is in addition to other read-only conditions that ArchestrA imposes. Clearing the selection removes the read-only condition for the item imposed by the Gateway Communication Driver. By default, the Read-Only check-box is selected.

Naming an ArchestrA Group

- Each ArchestrA group or topic must be uniquely named for the data source associated with it.
- The ArchestrA group name should not be identical (same in a case-insensitive manner) to an item prefix. This can cause a name clash and an unexpected behavior.
- The ArchestrA group name should not be identical to an item name, or the first part of the item name. This can cause an ambiguity in Gateway Communication Driver namespace. For example, do not name an ArchestrA group "Float" if a "Float.PV.Value" item exists in the Galaxy.

For more information, see Using Item Prefixes.

Configuring ArchestrA Device Items

ArchestrA data sources allow you to add items either at the data source branch of the hierarchy or through group objects. You can add items directly to the ArchestrA data source branch or in a group that allows you to group related ArchestrA tagnames together.

To add device items to your group, select the new group object and click the **Device Items** tab.

For more information about Device Items, see Device Item Definitions.

ArchestrA Item Names

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

Gateway Communication Driver supports writes to ArchestrA items configured as SecuredWrite. It does not support writes to an item whose security is configured as VerifiedWrite.

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

- To access an item in ArchestrA via Gateway Communication Driver through an OPC client, use the following syntax:

  Establish connection:
  ```
  OI.Gateway.3
  ```
  Reference item:
  ```
  ArchestrA.TIC101.PV
  ```
- To access an item in ArchestrA via Gateway Communication Driver 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 Gateway Communication Driver 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 Gateway Communication Driver 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 ArchestrA group. This prefix, which is added at runtime, can simplify item naming for ArchestrA groups in some situations.

Assume the item prefix for the ArchestrA 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

`ArchestrA.Blower_001.Temp1` (at the data source level)

`ArchestrA.Blower.001.Temp1` (at the group level)

DDE/SuiteLink Client Syntax

`Gateway|ArchestrA!Blower_001.Temp1` (at the data source level)

`Gateway|ArchestrA_Blower!001.Temp1` (at the group level)

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

ArchestrA Data Conversion

A key part of protocol conversion capabilities of the Gateway Communication Driver is its data type conversion between DDE, SuiteLink, OPC, and ArchestrA Message Exchange sources and clients. Each protocol has a set of supported data types for the values that can be accessed.

- If a client pokes an out-of-range value for any data type, Gateway Communication Driver does no clamping on the value, and passes the client request to the server.
- All pokes greater than 499 characters return Uncertain quality in the client and OCMC. The value is successfully poked to ArchestrA but it is truncated to 499 characters on the read-back. Additionally, all data below +/-1.5e-45 is rounded to 0.0.

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

ArchestrA – DDE/SuiteLink Mappings

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

ArchestrA to DDE/SuiteLink Conversions

In the case of the Gateway Communication Driver receiving data from an ArchestrA source and sending it to a DDE/SuiteLink client, the Gateway Communication Driver 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 Communication Driver receiving (write) data from a DDE/SuiteLink source and sending it to an ArchestrA client, it 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 GatewayCommunication Driver returns a failed write status to the client.

DDE/SuiteLink Type	**ArchestrA Type**	**Comments**

Discrete	Boolean	False = 0, True = 1
Real	Float	
Integer	Integer	Gateway Communication Driver 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 Communication Driver receiving data from an ArchestrA source and sending it to an OPC client, it 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 ArchestrA Conversions

In the case of the Gateway Communication Driver receiving (write) data from an OPC source and sending it to an ArchestrA client, it converts OPC 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 Communication Driver returns a failed write status to the client.

OPC Variant Type	**ArchestrA 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 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 Configuring the Gateway Communication Driver for a general overview about configuring data sources in Gateway Communication Driver.

Configuring a DDE Data Source Object

To add a DDE data source object to your Gateway Communication Driver hierarchy

1. Right-click **Configuration** in the hierarchy, and select either **Add DDE Connection.**

- A new object is created in the hierarchy tree and is named **New_DDE_000** by default.
- 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** configuration view (right pane) is displayed.

2. Configure the new DDE 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, 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 > : " / \ | , . ; ? ' [ ] { } ` ~ ! @ # $ % ^ & * ( ) _ + - =.
- **Reconnect Attempts:** Number of times Gateway Communication Driver 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 mins). 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 SuiteLink Data Source Object

To add a SuiteLink data source object to your Gateway Communication Driver hierarchy

1. Right-click **Configuration** in the hierarchy, and select **Add SuiteLink Connection**.

- A new object is created in the hierarchy tree and is named **New_SuiteLink_000** by default.
- 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_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, 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 Gateway Communication Driver. Default value is localhost. Use the browse button to select from a list of all nodes on your network.
- **Reconnect Attempts:** Number of times Gateway Communication Driver 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 and select **Add Topic Connection**.
   - 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 Gateway Communication Driver 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 Gateway Communication Driver-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.

Note: 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 Gateway Communication Driver, and its associated item name syntax:

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

  Establish connection:

  "OI.Gateway.3"

  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 Gateway Communication Driver 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 Gateway Communication Driver 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 Gateway Communication Driver's protocol conversion capabilities is its data type conversion between DDE, SuiteLink, OPC, and OPC sources and clients. Each protocol has a set of supported data types for the values that can be accessed. If a client pokes an out-of-range value for any data type, Gateway Communication Driver does no clamping on the value. It passes the client request to the server.

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

The following sections describe the data conversion mapping scheme applied by Gateway Communication Driver.

OPC–DDE/SuiteLink Mappings

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

DDE/SuiteLink to OPC Conversions

When the Gateway Communication Driver receives data from a DDE/SuiteLink source and sends it to an OPC client, it converts DDE/SuiteLink types to OPC types as follows:

DDE/SuiteLink Type	OPC Variant Canonical Mapping
Discrete	VT_BOOL

Float	VT_R4
Integer	VT_I4
String	VT_BSTR

Note: In case of conversion failures, Gateway Communication Driver returns Bad quality to the OPC client.

OPC to DDE/SuiteLink Conversions

When the Gateway Communication Driver receives (writes) data from an OPC client and sends it to a DDE/ SuiteLink data source, it 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 Configuring the Gateway Communication Driver for a general overview about configuring data sources in Gateway Communication Driver.

Configuring an InTouch Data Source Object

To add an InTouch data source object to your Gateway Communication Driver hierarchy

1. Right-click **Configuration** in the hierarchy, and select **Add InTouch Connection** 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.

    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 Gateway Communication Driver 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 Gateway Communication Driver-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 Gateway Communication Driver 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 Gateway Communication Driver 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. For information about how to use the Tag Browser, see

InTouch HMI documentation. 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 Connection** 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 Gateway Communication Driver 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 Gateway Communication Driver-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. For information about how to use the Tag Browser, see the InTouch HMI documentation . 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.

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.

Note: 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

For all topics within a book that are not **Chapter** topics.

Handling Time Zones with the Time Property

To share time stamp values across different time zones (Platforms), use the Time data type in every time zone location.

To share the time stamp as a string, note that when the Time data type is converted to a string (for example, in a script), it is automatically converted to local time. Hence, 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

The following configuration examples do not preserve the original time zone. That is, 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.

Example 1:

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

GalaxyB:Object1.TimeAttr shows the time adjusted to the local time zone of the GalaxyA Gateway Communication Driver and not the time zone of the PLC.

Example 2:

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

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

To preserve the Time Stamps

To avoid these problems and preserve the time stamps, subscribe to the GalaxyA:Object1.IntAttr value property. 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 the following 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 Gateway Communication Driver, and its associated item name syntax:

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

  Establish connection:

  "OI.Gateway.3"

  Reference item:

  "InTouch1.Pump1"

An InTouch data source is a special case of DDE and SuiteLink data source. Gateway Communication Driver 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 Gateway Communication Driver 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 Gateway Communication Driver 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.

Connecting to an MQTT Data Source

The MQTT data source is a top-level node that can be added to the Gateway Communication Driver 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 Configuring the Gateway Communication Driver for a general overview about configuring data sources in Gateway Communication Driver.

Configuring an MQTT Data Source Connection

Note: The MQTT subscriber within the Gateway has been superseded with a standalone MQTT Communication Driver. The MQTT Gateway subscriber will be phased out in a future release of the Communication Drivers. We recommend you to modify your configuration to start using the standalone MQTT Communication Driver instead.

To add an MQTT data source connection to your Gateway Communication Driver hierarchy

- 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 Gateway Communication Driver instance. If you need to connect to multiple brokers, you can create multiple instances of the GatewayCommunication Driver. See Instantiating the Gateway Communication Driver for additional information.

To configure the MQTT Connection

Follow the steps below to configure the MQTT Connection.

1. Configure Broker Connection
2. Enable TLS based secure connection.
3. Enable identity based username and password (at group level)
4. Enable Store and Forward (at group level)

Note: The steps 2, 3 and 4 are optional. The steps 3 and 4 are configured at the group level.

Step1: Configure Broker Connection

To configure the broker connection:

1. **Network Address**: Enter the IP address or host name of the MQTT Broker. The number of characters must not exceed 255. The field cannot be blank. The MQTT broker connection can be configured with an IPv4 or an IPv6 address space.
2. **Port Number:** Specify the **Port Number.** The default value is 1883 (without security-enabled). For a secure connection to the broker, the default port number is 8883. Edit the value if the MQTT broker uses a non-default port.
3. To verify that the MQTT Broker is accessible, click **Validate Address and Port**.

   The status of the test is displayed in a **Test Connection** 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.*" Ensure that the network address, and port number are correct.

Step2: (Optional) Enable TLS based secured connection

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 identify 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.

1. Select the **Secure Connection with Broker** checkbox.

   The **Port Number** in Step1 automatically changes to 8883. Edit the port number if necessary.

2. To set up encryption and privacy to the MQTT Broker, select the version of the TLS from the **Select Transport Layer Security (TLS) version**. The options available are **tlsv1**, **tlsv1.1**, and **tlsv1.2**. The TLS version depends on the configuration settings of the MQTT broker. Adding TLS in SuiteLink encrypts the data flow between SuiteLink Client and SuiteLink Server.

   If using a self-signed certificate, it is recommended to first verify the certificate. To verify the digital certificate:

   a. Click the **Download** button.

   b. Click the browse **(...)** button to view the self-signed certificate of the broker. Do not connect to a broker if you do not trust its self-signed certificate.

      Replace the existing certificates with your own certificates in the following file path:

      C:\ProgramData\Wonderware\CertStore\sample.pem

3. Click **Validate Security**, 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.**

     Either the green, yellow or red security icon is displayed, along with the corresponding description of the connection.

     - A green security icon with the status message: *Connection to the broker is secured and trusted* indicates that connection to the broker is encrypted and the broker certificate is issued by a trusted certification authority (CA).

     - A yellow security icon with the status message: *Connection to the broker is secured and untrusted* icon indicates that connection to the broker is encrypted, but the broker certificate is self-signed and is therefore untrusted.

     - A red security icon with the status message: *Connection to the broker is unsecured and untrusted* indicates that the identity of the broker cannot be verified (unknown and untrusted), and the connection is unencrypted.

   - If the security validation is not successful, the final status is "*Security Validation failed!*" Click **OK.** Edit the settings as necessary

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.

Step3: (Optional) Enable identity based username and password in each group.

You can enable the identity based username and password in each group. This step is configured at the group level, and does not required TLS to be enabled. For more information, see Configuring an MQTT Group Connection.

Configuring an MQTT Group Connection

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

- Select the MQTT Broker connection, right-click and select **Add MQTTGroup Connection** from 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.

To configure the MQTTGroup Connection

1. **Device Group Name**: The Device Group Name is used for accessing MQTT data from a DDE/SuiteLink client. It is automatically populated and cannot be modified.
2. **Client Id:** The client Id is used to uniquely identify a connection of the subscriber/publisher to the broker. Each group has a unique Client Id.
3. **Quality of Service (QoS):** The QoS level determines the message delivery parameters, with 0 as the lowest level of service, and 2 as the highest level. Select a level of **Quality of Service** (QoS) from the list.
   - **0 - At Most Once**: The message will be delivered no more than one time. This implied that it may not be delivered too. 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 the slowest message transfer mode.

     Note: If the MQTT publisher has a different QoS than the configured QoS, the configured QoS is used.
4. (Optional) Enter unique Username/Password for each group connection.
5. (Optional) Enable Store and Forward for each group connection.

Step 3: (Optional) Enter unique Username/Password for each group connection

1. Under **Identity**, select the **Enable** checkbox to enable the 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 **Validate Identity**, to verify that the MQTT Broker can be accessed on the configured MQTT channel.

Step 4: (Optional) Enable Store and Forward for each group connection

To enable the Store and Forward feature for a group:

- Under **Store and Forward,** select the checkbox **Enable Store and Forward for this Group**.

Each MQTT group has a maximum Store and Forward cache size of 4 GB. Once this limit is reached, depending on the following selection, the data is discarded or the collection of data is stopped.

- **Wrap-Around Data**

  The oldest data stored in the cache is discarded to allow storage space for new data.

- **Stop Collecting**

  If this option is selected, the system ceases to collect and store any new data. Once the connection is established again, the stored files are forwarded, which reduces the cache size, allowing storage space for new data.

For more information, see MQTT Store and Forward .

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.

Configuring an MQTT Publisher

The MQTT protocol allows the Communication Driver and associated software to communicate with edge devices that have published MQTT data to an MQTT broker. The publisher functionality allows the Communication Driver 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.

The MQTT publisher can be configured with all the Communication Driver data sources, such as ArchestrA, DDE, SuiteLink, OPC and OPC UA. 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). It is highly recommended to set up an MQTT Broker connection before configuring the MQTT Publisher. Refer Configuring an MQTT Data Source Connection section for steps to configure the MQTT Broker.

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

1. Connecting to Data Sources
2. Connecting to Data Sources
3. Configuring an MQTT Data Source Connection
4. Configuring an MQTT Group Connection

Workflow to configure an MQTT Publisher

Follow the workflow given below to configure an MQTT publisher.

1. Add a Data Source Object to your Gateway Communication Driver Hierarchy.
2. Add a Data Source Group Object
3. Add an MQTT Group Connection
4. Add an MQTT Reference Item

5. Publish MQTT Information

Example:

The example below describes the procedure to configure the OPC data source to publish data using the MQTT publisher. To use other data sources such as ArchestrA, DDE, SuiteLink or OPC UA, follow the configuration procedure for the respective connection.

Add an OPC Data Source Object to your Gateway Communication Driver Hierarchy

See Configuring an OPC Data Source Object.

Add an OPC Data Source Group Object

See Configuring an OPC Group Object.

Add an MQTT Group Connection

See Configuring an MQTT Group Connection.

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

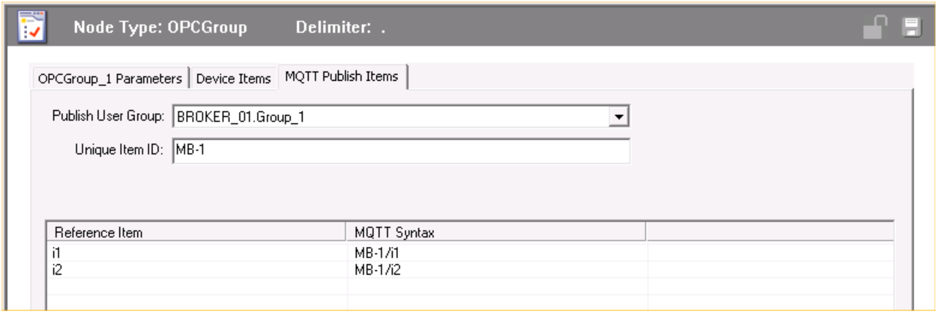
Add an MQTT Reference Item

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, and 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. To change the default name, double-click it and enter the new name.
    - Enter a unique name for the new device item (i1 in this example).
4. 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.
5. Multiple reference items can be added for the selected **Publish User Group**.



Publish User Group: BROKER_01.Group_1

Unique Item ID: MB-1

Reference Item	MQTT Syntax
i1	MB-1/i1
i2	MB-1/i2

6. Save the configuration.

Publish MQTT Information

1. To activate, right-click OI.Gateway.3 and click **Activate Server.**

   Note: In this example, the Simulation Communication Driver is configured and hence, is activated automatically.

2. The MQTT information for the reference items is published and can be viewed in the following paths
   - **OI.GATEWAY.3 > Diagnostics > Client Groups > MQTT Plugin**
   - **Wonderware – SIM > OI.SIM.1 > Diagnostics > Client group > OPC_001.OPCGroup_1**
3. 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.

Licensing for MQTT Connectivity

The Gateway Communication Driver licensing is applicable only for the MQTT connectivity. All other connection types (DDE, SuiteLink, OPC, OPC-UA, ArchestrA, and InTouch) do not need a license.

1. The MQTT connection requires a license (standard or professional). For an MQTT connection, a license is not required if the configuration has up to 32 subscribed references since startup. A license is required if the cumulative of the number of active registrations, and the previously registered and unregistered number exceeds the 32-tag limit. The server maintains a count of registered and unregistered references for licensing purposes.

   Example: If the connection has 25 active registrations, but has also previously registered and unregistered 25 additional unique ones, the connection has exceeded the 32-tag limit and thus require a license.

2. For professional functionality such as multi-instance configuration installation, the MQTT connection requires a professional Gateway Communication Driver license.

Note: With a Standard license, 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

The MQTT connection runs without a license in Demo mode for 120 minutes. At the end of the 120-minutes demo mode:

- MQTT stops updating items
- All non-system items have a Bad quality status
- New items are rejected

While in demo mode the MQTT checks for a license every 30 seconds. If a license is not found, it logs a warning. Once the Communication Driver finds a valid license, it logs a message, stops looking for a license, and begins running normally. For more information, see the *AVEVA Enterprise License Manager Help.*

License Status

You can determine the status of the license using the system item **$SYS$MQTTIsLicensed** or the icon of the active Gateway Communication Driver.

- The system item **$SYS$MQTTIsLicensed** indicates the status of the license, in both active and demo mode. A return of 1 indicates that it is licensed and 0 indicates that license is not present.

- The colour of the Gateway Communication Driver icon indicates the status of the license. For more information, see the Communication Drivers Pack Help.

MQTT Item 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:

- The topic name must not be left blank.
- 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 MQTT driver does not currently support usage of wild cards.

Syntax Example

```
site/area1/mixer4/valve/input
```

Using JSON Strings

Data messages should be in string format. The string can be formatted as a JSON key value pair, which is automatically detected. 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 uses 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 can 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

Using Embedded JSON Strings

The MQTT subscriber in the Gateway Communication Driver supports embedded JSON strings, as well as strings at multiple nested levels (objects and arrays). Refer the examples of possible JSON string forms and the referenced values below.

Example 1: If TagX receives a JSON string of form `{ “Value1”: 1, “Value2”:2}`

Values in the JSON string are referenced as follows:

Tag Reference	Value
TagX	{ “Value1”: 1, “Value2”:2}
TagX.Value1	1
TagX.Value2	2

Example 2: If TagA receives a JSON string of form `{ “Value1”: 1, “Value2”:{ “value3”: 2, “value4”: 3 }, “Value5”: [ 5, 6] }`

Values in the JSON string are referenced as follows:

Tag Reference	Value
TagA	{ “Value1”: 1, “Value2”:{ “value3”: 2, “value4”: 3 }, “Value5”: [5, 6]}
TagA.Value1	1
TagA.Value2	{ “value3”: 2, “value4”: 3 }
TagA.Value2.value3	2
TagA.Value2.value4	3
TagA.Value5	[5,6]
TagA.Value5[0]	5
TagA.Value5[1]	6

Example 3: If TagA receives a JSON string of form `[ “stringValue1”, “stringValue2”]`

Values in the JSON string are referenced as follows:

Tag Reference	Value
TagS	[“stringValue1”, “stringValue2”]
TagS[0]	stringValue1
TagS[1]	stringValue2

MQTT Item Syntax for VTQ Timestamp

Each payload in MQTT has an associated time value, which shows the time of occurrence of an event. Two time types are supported:

- Unix timestamp

  Example: 1507103430 = Wednesday October 4 2017 07:50:30 UTC)

- ISO 8601 timestamp string in UTC format

  Example 1: 2017-10-04T07:50:30Z = Wednesday October 4 2017 07:50:30 UTC)

  Example 2 (when time is in ms): 2017-10-04T07:50:30.134Z = Wednesday October 4 2017 07:50:30.134 UTC).

  The ISO 8601 timestamp string ends with Z.

Using &T& String

For an accurate timestamp value of a data field, the MQTT data source concatenates the data field and the timestamp field using the &T& string. If a value item is X and the time item is Y, the item X&T&Y allows the value X to be timestamped with the value of Y. The resulting item is read-only.

Prerequisites to use &T& String

The prerequisites for using the &T& string to timestamp the date field are:

- A time field must be available in the data record
- The time field must be in Unix or ISO time type.

Examples

Example 1:

Data field: **obj.temp_c**

Timestamp field: **obj.time**

The syntax for VTQ time stamp using the &T& token:

```
obj.temp_c&T&obj.time
```

If the timestamp field is in the same object as the data field, the VTQ timestamp is obtained using the syntax:

```
obj.temp_c&T&.time
```

Example 2:

Consider two device items in the MQTTGroup node: Tank01_ISO and Tank01_Unix

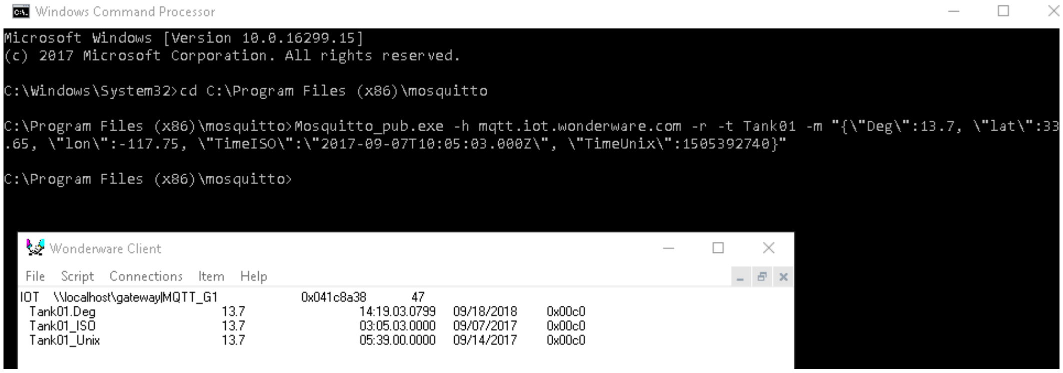
For the device item Tank01_ISO:

- Full syntax of the Item Reference is Tank01.Deg&T&Tank01.TimeISO
- Short syntax of the Item Reference is Tank01.Deg&T&.TimeISO

For the device item Tank01_Unix:

- Full syntax of the Item Reference is Tank01.Deg&T&Tank01.TimeUnix
- Short syntax of the Item Reference is Tank01.Deg&T&.TimeUnix

At the command: `Mosquitto_pub.exe -h mqtt.iot.wonderware.com -r -t Tank01 -m "{\"Deg\":13.7, \"lat\":33.65, \"lon\":-117.75, \"TimeISO\":\"2017-09-07T10:05:03.000Z\", \"TimeUnix\":1505392740}"`, the values are displayed in the Client.



```
Microsoft Windows [Version 10.0.16299.15]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\Program Files (x86)\mosquitto

C:\Program Files (x86)\mosquitto>Mosquitto_pub.exe -h mqtt.iot.wonderware.com -r -t Tank01 -m "{\"Deg\":13.7, \"lat\":33
.65, \"lon\":-117.75, \"TimeISO\":\"2017-09-07T10:05:03.000Z\", \"TimeUnix\":1505392740}"

C:\Program Files (x86)\mosquitto>
```

```
IOT   \\localhost\gateway|MQTT_G1           0x041c8a38      47
 Tank01.Deg                 13.7                  14:19.03.0799   09/18/2018    0x00c0
 Tank01_ISO                 13.7                  03:05.03.0000   09/07/2017    0x00c0
 Tank01_Unix                13.7                  05:39.00.0000   09/14/2017    0x00c0
```

Transferring Files using MQTT Broker

The MQTT Broker supports transferring files of different formats such as PDF, .xls, .doc, .jpeg across the nodes. You can transfer files from one instance of Gateway Communication Driver (initiated by System Platform/ InTouch) to a MQTT broker to be consumed by another instance of Gateway Communication Driver. The file to be transferred must be in the local file system, and not in a network/shared location.

To transfer a file, use the client to poke the complete file path to one of the MQTT topic names - wwData or wwDataDT.

- On the publisher side, the MQTT publisher in Gateway Communication Driver will send the file content to the subscriber side.
- On the subscriber side, if it receives an update on either of the two topics, it will create a file based on the name of the file used in the publisher side and provides the absolute path of the received path in the topic.

Note: The MQTT topic names **wwData** and **wwDataDT** are case sensitive.

To poke a file to the MQTT Topic Name:

1. In the SuiteLink client, click **Item**.
2. The **Item Dialog** appears:
    - In the **Item** field, enter wwData or wwDataDT
    - In the **Value** field, enter the path of the file to be transferred
3. Click **Poke**

   The files are transferred, and saved to the path.

Poking the file to wwData

When the file is poked to wwData, the file is transferred to the public folder - **C:\users\public\Wonderware\wwData**

The filename and timestamp are retained. Any subsequent transfer using the same file name will overwrite the file in the previous transfer. The format of the file name is:

`YYYYMMDD_<filename>`

Poking the file to wwDataDT

When the file is poked to wwDataDT, the file is transferred to the public folder - **C:\users\public\Wonderware\wwDataDT**

The file name is prepended with the transfer time (UTC) at the destination folder. The timestamp on the file is retained. Any subsequent transfer using the same file name will overwrite the file in the previous transfer. The format of the file name is:

```
YYYYMMDD_HHMMSSsss_<filename>
```

MQTT Store and Forward

The MQTT Store and Forward feature allows seamless data publish and transfer to the data consumer, MQTT broker. When the connectivity to the MQTT broker is not available, the published data items is stored. Once the connectivity is restored, the published data is forwarded to the broker. The published data items are stored in the .snf format. The .snf files follow the First-In-First-Out mechanism. That is, the files stored first at the time of network disruption are forwarded first on resuming the connectivity. The .snf files are stored in the following file path.

C:\ProgramData\Wonderware\OI-Server\$Operations Integration Supervisory Servers$ \OI.GATEWAY\OI.GATEWAY\NCC\SNF\

The .snf files are automatically named in numerical order, to indicate the order in which they were stored.

The Store and Forward feature is group specific. To enable this feature for a group:

- In the **Store and Forward** section of the MQTT Group Parameters configuration view, select the checkbox **Enable Store and Forward for this Group**.

Each MQTT group has a maximum Store and Forward cache size of 4 GB. Once this limit is reached, depending on the following selection, the data is discarded or the collection of data is stopped.

- **Wrap-Around Data**

  The oldest data stored in the cache is discarded to allow storage space for new data.

- **Stop Collecting**

  If this option is selected, the system ceases to collect and store any new data. Once the connection is established again, the stored files are forwarded, which reduces the cache size, allowing storage space for new data.

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 Configuring the Gateway Communication Driver for a general overview about configuring data sources in Gateway Communication Driver.

Configuring an OPC Data Source Object

To add an OPC data source object to your Gateway Communication Driver hierarchy

1. Right-click **Configuration** in the hierarchy, and select **Add OPC Connection** 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.

   The **New_OPC_001 Parameters** configuration view (right pane) is displayed.

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.3, 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 Gateway Communication Driver 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 Gateway Communication Driver 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, Gateway Communication Driver will attempt to start the external OPC Server Out-of-Proc. When not selected, it 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, Gateway Communication Driver adds the item to the OPC server with the requested type set to the type specified in the VT suffix. Gateway Communication Driver 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, Gateway Communication Driver 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. Gateway Communication Driver does not parse the item names looking for a VT suffix.

     For more information, see Using VT Item Suffixes.

   - **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 Connection** on the shortcut menu.

   A new object is created in the hierarchy tree and is named **New_OPCGroup_000** by default. Rename it, if desired. You are allowed to add up to 100 new group objects.

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

   The **New_OPCGroup_000 Parameters** configuration view (right pane) is displayed.

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

   a. **Device Group Name:** Name of the topic that DDE or SuiteLink clients of Gateway Communication Driver 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).

   b. **Update Rate:** Value (in ms) used by Gateway Communication Driver 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.

   c. **OPC Item ID Prefix:** String prefixed to all item names added to the OPC group. Default value is blank. The syntax to use the prefix is `<Prefix>.<Item>`. For example: If the item prefix is boiler1 and the Item name is temperature, the item requested from data source is boiler1.temperature

   d. **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.

   e. **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 Gateway Communication Driver -imposed read-only qualifications. In other words, items inherently read-only in the data source remain so. Default value is checked.

   f. **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.

   g. **Browse OPC Items**: Opens the OPC browser, in which you can select items directly from the OPC server.

3. Configure the Device Items by one of the following methods:

   a. Use the OPC Item Browser to browse OPC server items and populate in **Device Items** tab.

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

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.

To browse OPC server items and populate in Device Items tab

1. In the OPCGroup Parameters tab, click **Browse OPC Items.**

   The **OPC Item Browser** window appears.

2. In the **Branches** section, navigate and select the node to browse items. The available items populates in the **Available Items** section.

3. To add an item to the basket, do one of the following:
   - Select the item, and click the + icon on the top right corner.
   - Select the item, and drag-drop to the Basket section.
   - Select the item and press **Enter**.
   - Right-click on the item, and select **Add to Basket**.

4. To delete an item from the basket, do one of the following:
   - Select the item, and click the X icon on the top right corner.
   - Select the item and press **Delete.**
   - Right-click on the item, and select **Remove from Basket**.

5. To filter the items, click **Filter...**

   The **Specify Filter Criteria** dialog appears. Enter the filter criteria using Item Name, Item Data Type and Item Access Rights.

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.

OPC Connectivity, DCOM, Windows Firewall, and Anonymous Access

The OPC connectivity between two computers rely on compatible authentication mechanisms, which ensures that both computers can communicate with one another. In OPC, this configuration is stored in the DCOM settings. As per settings for Anonymous Access and full control, the configuration could be set to allow everyone, without restrictions. Our software does not support the Anonymous Access and full control. Hence, it is recommended that the two computers which need to communicate with one another, have the appropriate security access configuration that is common to both.

Note: Our software no longer supports Anonymous Logon setting in DCOM configuration for OPC connections.

Potential Issue:

To ensure successful connectivity to an OPC server on a remote machine, you must be aware of the security and configuration issue of OPC. The OPC connectivity employs the callback scheme where the OPC server may need to call-back to the OPC client. The following symptoms may be presented if the security and configuration on either or both of the machines are not set up correctly.

- The OPC Client application fails to create an OPC Group
- The OPC Client application does not display data updates. Consequently, data values remain unchanged or display “bad” quality

- The logger reports a COM error 0x80040202

Potential Solutions:

Depending on the configuration of the Operating System, apply one or more solutions below to resolve the issue.

Solution 1: Resolving Invalid Username/Password

Issue: When the OPC client receives a call-back from the OPC server, the OPC client authenticates the caller identity. The OPC Client fails to validate the username and password combination of the OPC Server.

Solution: DCOM Matching Identity: In the DCOM configuration of the OPC Server computer, the **User (Username and Password)** selected in the **Identity** tab of the **DCOM configuration** dialog, must match an existing user in OPC Client computer.

Solution 2: Resolving Guest-Only Access

Issue: In a workgroup environment, the Windows Operating System may force local users to authenticate as guest. However, the guest privilege is insufficient to access the OPC client computer.

Solution: Fall back the "**Network access: Sharing and security model for local accounts**" security policy on the computer to "**Classic – local users authenticate as themselves**"

Solution 3: Resolving Windows Firewall Blocks

Issue: The Windows Firewall blocks the call-backs from the OPC Server, while the OPC client would still be able to make the outgoing calls.

Solution: As an initial test, disable the firewall on either or both the OPC client and OPC server computers. If the problem is resolved by disabling the firewall, revert to enabling the setting and confirm that the DCOM port 135 inbound and outbound rules are configured in the firewall and are set to allow access. You may need to contact you system administrator to set the specific firewall settings.

Using VT Item Suffixes

This section describes how a connected client requests access to items (or attributes) of a particular OPC data source. You can configure OPC and OPC UA data sources to use the /VT item suffix to gain better control of data access in Gateway Communication Driver. The /VT item suffix determines the type of data the underlying OPC server reports to the Gateway Communication Driver.

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.

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.`

Similarly, you can apply RequestVT (/VT item suffix) to an item name using the following syntax: `/VT_<DataType>`

For example, 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 Gateway Communication Driver is subscribed as `<OPCUAServer>.Blower.Int2 /VT_I2.`

For information about specific VT data types, see OPC Data Conversion and OPC UA Data Conversion.

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
/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 Suffix	Description
/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 Gateway Communication Driver through a SuiteLink client, use the following syntax:

Establish connection:

```
   Application = Gateway
   Topic = OPCServer1_OPCGroup1
```

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 Gateway Communication Driver 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 name of the Device Group is generated as "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 same as the DDE/SuiteLink client. Access an item "R1" in an OPC data source, "ModbusOverOPC", via Gateway Communication Driver through InTouch, and specify a 4-byte signed Integer as follows:

Establish connection:

```
   Application = Gateway
   Topic = OPCServer1_OPCGroup1
```

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 Gateway Communication Driver's protocol conversion capabilities is its data type conversion between DDE, SuiteLink, and OPC sources and clients. Each protocol has a set of supported data types for the values that can be accessed.

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

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.

The following section describes the data conversion mapping scheme applied by Gateway Communication Driver.

DDE/SuiteLink-OPC Mappings

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

OPC to DDE/SuiteLink Conversions

When the Gateway Communication Driver receives (writes) data from an OPC client and sends it to a DDE/ SuiteLink data source, it 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

When the Gateway Communication Driver receives data from a DDE/SuiteLink source and sends it to an OPC client, it converts DDE/SuiteLink types to OPC types as follows:

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

Note: In case of conversion failures, Gateway Communication Driver returns Bad quality to the OPC client.

Connecting to an OPC UA Data Source

The OPC UA data source is a top-level node that can be added to the Gateway Communication Driver configuration. When the Gateway Communication Driver connects to the OPC UA server:

- One connection is created per OPC UA hierarchy.
- One subscription is created for each OPC UA group.

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 Configuring the Gateway Communication Driver for a general overview about configuring data sources in the Gateway Communication Driver.

Configuring an OPC UA Data Source Object

To add an OPC UA data source object to your Gateway Communication Driver Hierarchy

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

   A new object is created in the hierarchy tree.

2. The OPCUA Connection object is named "New_OPCUA_000" by default. Rename it, if required.

Step 1: To Configure the OPCUA Server Details

Use the **OPCUA Server Details** section of the OPC UA editor to configure the OPC UA Server. Ensure that the configuration on the OPCUA faceplate matches the configuration settings in the OPCUA Server that you are trying to connect.

Prerequisites:

Before configuring the OPC UA Server Details, ensure that you have the following details:

- If the OPC UA server can be directly connected:
  - **OPC UA Server Endpoint URL:** Required for the OPC UA client to connect to the OPC UA server. Note down the endpoint from your OPC UA server. At the least, you should know the IP address of the OPC UA server.

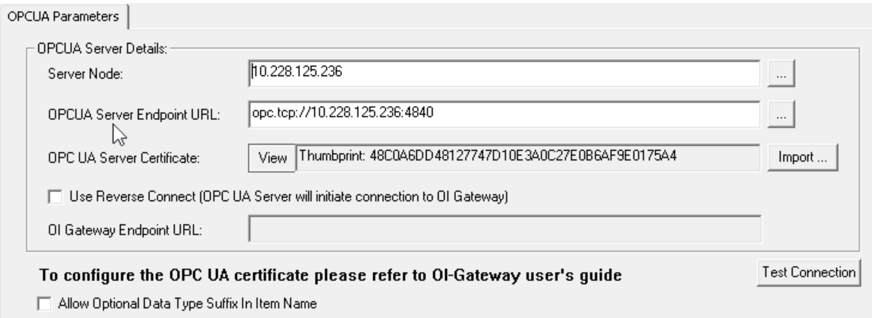
    The Gateway Communication Driver can also connect to an OPC UA server in an IPv6 network by entering the link-local IPv6 address of the machine on which the OPC UA server runs.

    The general syntax of the OPC UA Server Endpoint URL is:

    ```
    opc.tcp://[<IP address of the OPC UA Server>]:<Port number>/<Server Name>
    ```

  - **OPC UA Server Certificate:** Required to establish a trust relation between the OPC UA server and OPC UA client. You can import a digital certificate of the OPC UA directly. Alternatively the Gateway Communication Driver can download a certificate from the OPC UA. Review the certificate to ensure that it is genuine before trusting it.
  - **Security Policy:** Indicates the encryption policy used for the connection. Note down the security policy used by the OPC UA server.
  - **User Authentication:** If your OPC UA server is configured with the User name and Password, then you need to enter the same user credentials for the OPC UA connection. Note down the User name and Password that is used in your OPC UA server.
- If the OPC UA server is protected by a firewall and cannot be directly connected, you can use the OPC UA Reverse Connect feature where the OPC UA server can be configured to initiate the connection to the Gateway Communication Driver:
  - **Gateway Communication Driver Endpoint URL:** Required while using Reverse Connect to allow the OPC UA Server to initiate a connection with the Gateway Communication Driver. Note down the complete endpoint URL as mentioned in the OPC UA server for the reverse connect client URL.

To configure the OPC UA Server details



OPCUA Parameters

OPCUA Server Details:

Server Node: 10.228.125.236

OPCUA Server Endpoint URL: opc.tcp://10.228.125.236:4840

OPC UA Server Certificate: Thumbprint: 48C0A6DD48127747D10E3A0C27E0B6AF9E0175A4

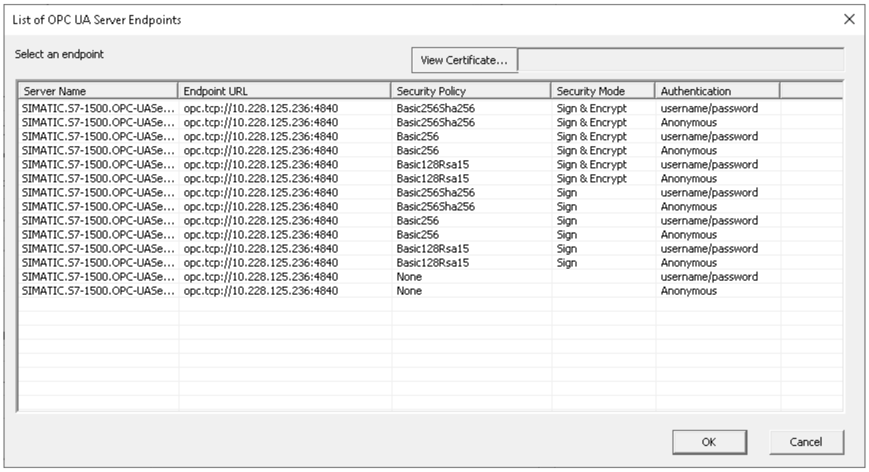
□ Use Reverse Connect (OPC UA Server will initiate connection to OI Gateway)

OI Gateway Endpoint URL:

To configure the OPC UA certificate please refer to OI-Gateway user's guide

□ Allow Optional Data Type Suffix In Item Name

1. In the **Server Node** field, browse the **Server Node** using the browse button (...), or enter the server node name or the IP address.
2. OPC UA **Server Endpoint URL field:** If you know of the endpoint exposed by the OPC UA server, you can enter it manually. If the OPC UA server supports local discovery through OPC UA defined port 4830, you can just leave this field blank. When you click on the ellipse (...) button, the end point selection dialog will be posted to allow you to select the appropriate endpoints supported by the OPC UA server. Be sure to click on the **View Certificate** button in the **Endpoint Selection** dialog to ensure the digital certificate reflects appropriately the corresponding OPC UA server that you are connecting to. You can select any endpoint in the list by selecting any entry in the list followed by the **OK** button or by simply double-clicking on any entry.



List of OPC UA Server Endpoints

Select an endpoint

Server Name	Endpoint URL	Security Policy	Security Mode	Authentication
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	Basic256Sha256	Sign & Encrypt	username/password
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	Basic256Sha256	Sign & Encrypt	Anonymous
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	Basic256	Sign & Encrypt	username/password
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	Basic256	Sign & Encrypt	Anonymous
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	Basic128Rsa15	Sign & Encrypt	username/password
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	Basic128Rsa15	Sign & Encrypt	Anonymous
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	Basic256Sha256	Sign	username/password
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	Basic256Sha256	Sign	Anonymous
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	Basic256	Sign	username/password
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	Basic256	Sign	Anonymous
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	Basic128Rsa15	Sign	username/password
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	Basic128Rsa15	Sign	Anonymous
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	None		username/password
SIMATIC.S7-1500.OPC-UASe...	opc.tcp://10.228.125.236:4840	None		Anonymous

3. In the **OPC UA Server Certificate** field, click the **Import** button to add the OPC UA server certificate, or enter the details manually. Select the certificate that you have downloaded from your OPC UA server and click **OK**. In the **Certificate Import** window that appears, click **View Certificate** to verify if you have selected the correct certificate, and then click **Accept** to trust the certificate. You can also click **View** if you wish to see the certificate details after trusting the certificate.

Important: Importing a certificate from an untrusted source may incur a security risk. It is recommended to review the certificate to ensure it is issued by a trusted authority before accepting it.

4. If the OPC UA Server is protected by a network firewall and that it also supports OPC UA Reverse Connect, select the **Use Reverse Connect (OPC UA Server will initiate connection to OI Gateway)** checkbox. The OPC UA Reverse Connect feature enables the OPC UA server to initiate a connection with the Gateway Communication Driver. In a secure environment, inbound connections to the OPC UA server are restricted as the server operates behind a firewall. By using Reverse Connect, the firewall restriction is alleviated as the

connection is being made from the OPC UA server to the client. This field is optional. If you do not select this option, you can skip to step 6.

5. If you select the **Use Reverse Connect (OPC UA Server will initiate connection to OI Gateway)** checkbox in the above step, the **OI Gateway Endpoint URL** field gets enabled. This is endpoint that the Gateway Communication Driver will listen to incoming connection from the OPC UA server. Make sure the port specified in this URL is not blocked by the local firewall of the Gateway Communication Driver computer. An inbound rule may need to be created in the firewall setting to open the specific local port for TCP connection. This must have the same value of the end point URL configured in the OPC UA server for Reverse Connect.

   The general convention for the URL is: `opc.tcp://<hostname>:<port>/<reference>`

   where,

   <hostname>: machine host name

   <port>: tcp port number that is available and opened at the firewall

   <reference>: a unique string that identifies this connection. Ensure that this string is unique across reverse connection strings in any instances and hierarchies of Gateway Communication Driver on the same machine.

6. Click **Test Connection** to establish the connection between the OPC UA Server and Gateway Communication Driver. When Reverse Connect is used to listen to any connection from the OPC UA server, a progress dialog will be displayed. The OPC UA server will need to connect to OI Gateway within 120 seconds to avoid any connection time-out in the progress dialog.

   Once the connection is established, the **Advanced Configuration** section is minimized and the updated **OPC UA Namespace** section is displayed.

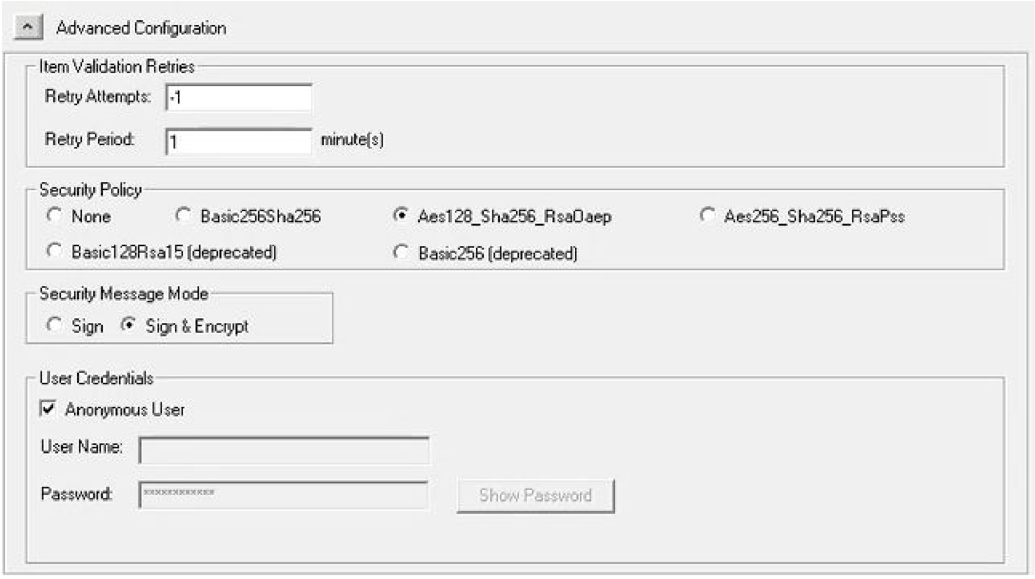
   Note: When using Reverse Connect, the Gateway Communication Driver has an internal time-out of 120 seconds while awaiting connection initiated from the OPC UA server. You can also cancel the test anytime by clicking on the **Cancel** button.

7. Select the **Allow Optional Data Type Suffix in Item Name** check-box to add data type as a suffix to an item name. This parameter is hot configurable. For more information about OPC UA data item names, access the OPC UA Tag Browser or see OPC UA Item Names and Syntax.

For errors while connecting to the OPC UA Server, see Connectivity with the OPC UA Server in the Troubleshooting chapter.

Step 2A: Advanced Configuration (If using secured OPC UA connection)

Use the **Advanced Configuration** section of the OPC UA editor to set the OPC UA connection security parameters: Security Policy, Security Message Mode, and User Credentials. To expand/collapse the **Advanced Configuration** section, click the arrow next to **Advanced Configuration**. You can click the arrow button to maximize or minimize the Advanced Configuration section.



Advanced Configuration

Item Validation Retries

Retry Attempts: -1

Retry Period: 1 minute(s)

Security Policy

○ None ○ Basic256Sha256 ● Aes128_Sha256_RsaOaep ○ Aes256_Sha256_RsaPss

○ Basic128Rsa15 (deprecated) ○ Basic256 (deprecated)

Security Message Mode

○ Sign ● Sign & Encrypt

User Credentials

☑ Anonymous User

User Name:

Password: ************

To configure the OPCUA Security

Before configuring the OPC UA security parameters, we recommend that you reference the OPCUA Server to match the security configuration.

- **Item Validation Retries**

  The Item Validation Retries allows to configure the retry details for item validation.

  - **Retry Attempts**: Indicates the number of Retry attempts for the validation. The valid range is between -1 to 10000000. The default value of -1 indicates an infinite retry.
  - Retry Period: Indicates the retry period (in minutes) for each retry of item validation. The valid range is 1 to 10000000.The default value is 1 minute.

- **Security Policy**

  The Security Policy indicates the encryption policy used for the connection.

  - **None**: No encryption is applied.
  - **Basic256Sha256:** Indicates that Basic256Sha256 security is applied.
  - **Aes128_Sha256_RsaOaep:** Indicates that Aes128_Sha256_RsaOaep is applied.
  - **Aes256_Sha256_RsaPss**: Indicates that Aes256_Sha256_RsaPss is applied.
  - **Basic128Rsa15 (deprecated):** Indicates that Basic128Rsa15 security is applied.
  - **Basic256 (deprecated):** Indicates that Basic256 security is applied.

  Note: It is strongly recommended to use the security policy (Basic256Sha256, Aes128_Sha256_RsaOaep or Aes256_Sha256_RsaPss) to encrypt communication with the server. Encryption is critically important when passing username and password to an OPC UA server. Note that legacy policies Basic128Rsa15 and Basic256 are no longer considered secure and are classified as deprecated. While those deprecated policies may be preferable to no encryption at all, it is recommended that a more secure policy is used when supported by targeted OPC UA servers.

- **Security Message Mode**

The **Security Message Mode** indicates the message mode of the connection. If the **Security Policy** is set to **None**, the **Security Message Mode** is **Not Applicable**. If the **Security Policy** is set to any options other than **None** (that is, Basic128Rsa15, Basic256, or Basic256Sha256), the **Security Message Mode** options - **Sign** and **Sign and Encrypt** populates for selection.

- **Sign:** All messages are signed but not encrypted.
- **Sign and Encrypt:** All message are signed and encrypted. This is the most secured option and is recommended to use this option.

- **User Credentials**

  This section allows you to configure the user credentials for the OPC UA connection. Select the **Anonymous User** checkbox to allow the OPC UA client to connect to the OPC UA server without credentials. 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. Clear the checkbox to provide a user name and password in their respective fields.

  Note: If the user name and password is configured for connection with OPCUA Server, it is recommended to configure the appropriate security policy Basis256Sha256, Aes128_Sha256_RsaOaep or Aes256_Sha256_RsaPss) to ensure user name and password values are fully encrypted.

- Once you configure all the fields, click the **Save** icon to save your configuration.

Step 2B: Trusting the Gateway Communication Driver OPC UA Certificates with the OPC UA Server

When using a secured connection to the OPC UA Server, the Gateway Communication Driver node must be trusted by the OPC UA Server. Apart from the procedure mentioned in **Step 1: To Configure the OPCUA Server Details** for trusting OPC UA server, you can also trust the certificate manually.

Follow the steps below to trust the Gateway Communication Driver node.

1. Open the OPC UA configuration.
2. Ensure that the configuration on the OPC UA faceplate matches the configuration settings in the OPC UA Server that you are trying to connect. Click **Test**.

   A new Gateway Communication Driver OPC UA certificate is automatically copied to the OPC UA Server.
3. Follow the OPC UA Server workflow to trust the new certificate that was published to it.
4. Return to the OPC UA Configuration screen and click **Test** to ensure that the operation is successful.

Note: If you are upgrading from a version older than 5.2, you must trust the certificate again as explained above. It is recommended to download the OPC UA Server Certificate directly, and import it while configuring the OPC UA Server details. This ensures that the certificate is genuine.

Step 2C (Optional): Specify your own certificate for the Gateway Communication Driver

If you want to use your own certificates (self-signed or CA-signed), as a certificate for the Gateway Communication Driver, follow the steps below:

1. Gateway Communication Driver uses a digital certificate named "OIGateway OPCUA". To use your own certificate, replace the existing Gateway Communication Driver certificates with your own certificates, using the existing "oiGateway OPCUA" name.
2. You must replace the public certificate and the private key in the respective folders:

- Public certificate (.der file): **C:\ProgramData\Wonderware\OI-Server\$Operations Integration Supervisory Servers$\OI.GATEWAY\CertificateStores\certs\OIGateway OPCUA.der**
- Private key (.pem file): **C:\ProgramData\Wonderware\OI-Server\$Operations Integration Supervisory Servers$\OI.GATEWAY\CertificateStores\private\OIGateway OPCUA.pem**

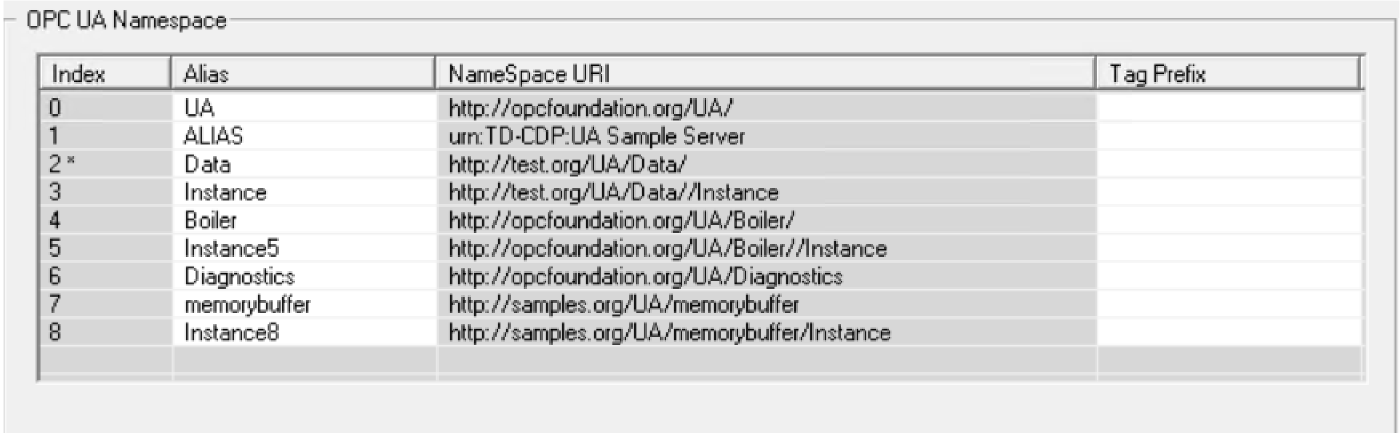
The self-signed certificate has a validity of ten years from the date it is generated and is encoded with the machine name according to the digital certificate requirement in the OPC UA specification.

3. Follow the steps in section 2B.

OPC UA Namespace

The **OPC UA Namespace** section of the OPC UA editor displays the Namespace alias table for the Namespace URIs present in the OPC UA Server to which the OPC UA Client is connected. 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 Communication Driver.
- 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.



OPC UA Namespace

Index	Alias	NameSpace URI	Tag Prefix
0	UA	http://opcfoundation.org/UA/	
1	ALIAS	urn:TD-CDP:UA Sample Server	
2 *	Data	http://test.org/UA/Data/	
3	Instance	http://test.org/UA/Data//Instance	
4	Boiler	http://opcfoundation.org/UA/Boiler/	
5	Instance5	http://opcfoundation.org/UA/Boiler//Instance	
6	Diagnostics	http://opcfoundation.org/UA/Diagnostics	
7	memorybuffer	http://samples.org/UA/memorybuffer	
8	Instance8	http://samples.org/UA/memorybuffer/Instance	

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.
Namespace URI	Displays the Namespace URI imported from the OPC UA Server. This information cannot be edited.
Tag Prefix	Here you can add the common prefix for an item reference. To add a prefix: Double-click anywhere in the cell under the Tag Prefix column and enter the desired prefix. A maximum of 250 characters is allowed.

To edit the OPC UA 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. **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 OPCUAGroup Connection** 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 maximum of 10 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 Interval**: Expressed in milliseconds, this parameter controls the frequency at which the Gateway Communication Driver requests for data updates from the OPC UA Server.

   c. **Data Sampling Interval**: Expressed in milliseconds, this parameter indicates the fastest rate at which the OPC UA server should sample its underlying source for data changes. If exception-based reporting is supported by the OPC UA server, this parameter may be set to 0 to request exception-based data changes.

      Since the **Update Interval** indicates how often the Gateway Communication Driver retrieves the data from the OPC UA server, you should set the **Data Sampling Interval** at a rate faster than the **Update Interval**.

      For example, if you set the **Update Interval** as 5000ms and the **Data Sampling Interval** as 1000ms, then the Gateway Communication Driver records 5 data samples every 5 seconds.

Note: Do not use this parameter if the **Demand Read** option is checked.

d. **Read Only**: When selected, all items connected through the OPC UA group are read only. This parameter is hot configurable.

e. **Demand Read**: If this option is checked, the value of the **Update Interval** is used to poll the OPC UA server for data.This option uses less resources on the OPC UA server but imposes limit on data throughput and hence, should only be used when low throughput condition is acceptable.

   For high performance data throughput in runtime, leave this option unchecked (default). Gateway Communication Driver will request the OPC UA server to monitor data changes in the data items. Whenever there are value changes in the data items, the OPC UA server will publish the changes to Gateway Communication Driver.

3. Configure the Device Items by one of the following methods:

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

   b. To browse OPC UA server items and populate in **Device Items** tab, click **Browse OPCUA Server** to browse OPC UA server items and populate in **Device Items** tab. For more information, see .

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.

Browsing the OPC UA Server Namespace

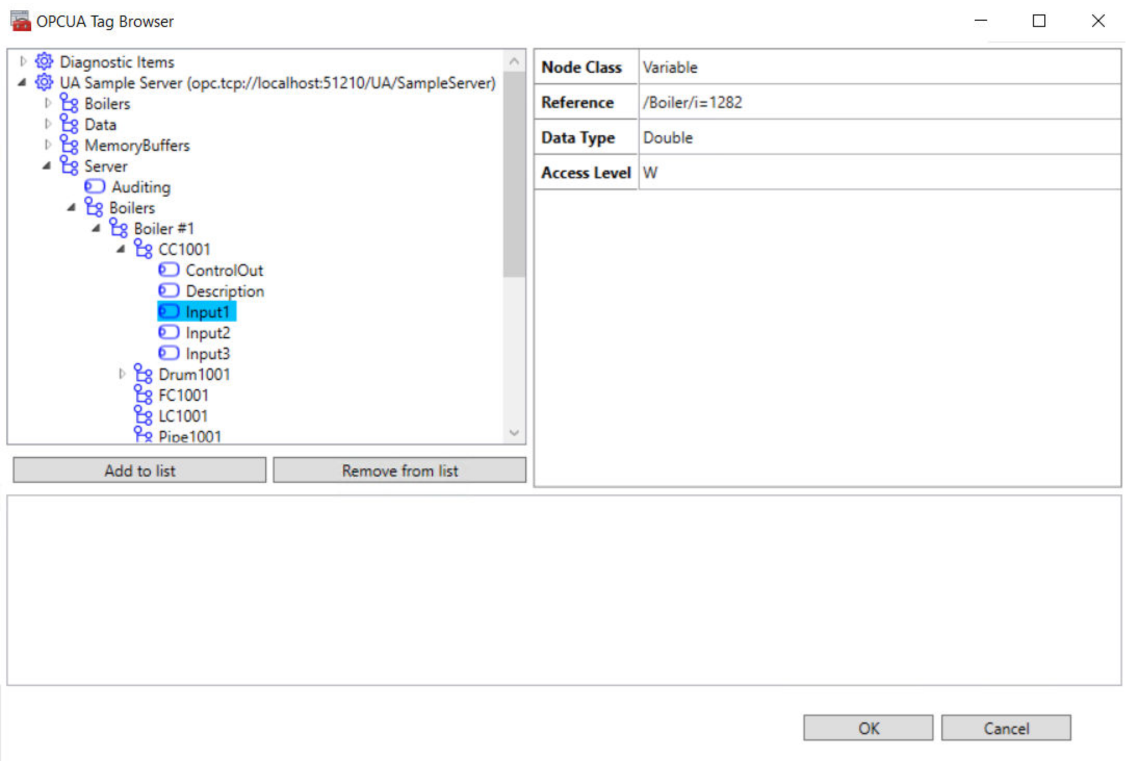
The OPC UA connection in the Gateway Communication Driver is acts as an OPC UA client. Thus, it can access the OPC UA server namespace to browse diagnostic items and server data items. Selected items are automatically added in the **Device Items** tab for easy mapping.

To browse the OPC UA server

1. In the **DeviceGroup Parameters** tab, select **Browse OPCUA Server.**

   The **OPCUA Tag Browser** window appears.

   Note: If you have enabled OPC UA reverse connect, it may take some time to open the **OPCUA Tag Browser** as the OPC UA server has to initiate the connection to the Gateway Communication Driver.



Node Class	Variable
Reference	/Boiler/i=1282
Data Type	Double
Access Level	W

For errors while connecting to the OPC UA Server, see Connectivity with the OPC UA Server.

2. Select the required diagnostic item tag or server-specific tag to view the properties of the tag in the right pane:
   - **Node Class**: Indicates the type of the node (object, function, or variable)
   - **Reference**: Indicates the unique item reference or Node ID of the tag.
   - **Data Type**: Indicates the data type of the tag.
   - **Access Level**: Indicates the access level of the tag (R (read),W (write) or R/W (read/write))

Note: Hold the **Ctrl** button to select multiple tags. If you select multiple tags, the tag properties will not be displayed. If you select a folder, all the items inside that folder will be moved to the lower grid.

3. Select **Add to list**. To remove an item, select the item and then select **Remove from list**.
4. Click **OK**.

   The **Device Items** tab is auto-populated with the selected items. You can modify the item reference if required.

Browsing the OPC UA Connection with multiple subscription device groups

For an OPC UA connection with multiple subscription device groups, the OPC UA Tag Browser launched from any of the device groups of an OPC UA connection displays the system items hierarchy ($SYS$DIAG) with all the device groups associated with that connection, in addition to the Global and UA Client items. Select the device group (Conn=<device group>) to view the corresponding system items in that device groups .

Example:

Consider two OPC UA Connections: New_OPCUA_000 and New_OPCUA_001.

- New_OPCUA_000 consists of three device groups: OPCUAGroup1, OPCUAGroup2, OPCUAGroup3
- New_OPCUA_001 consists of two device groups: OPCUAGroupA and OPCUAGroupB

Launching the OPCUA Tag browser from any of the device groups under New_OPCUA_000 displays the three device groups (OPCUAGroup1, OPCUAGroup2, and OPCUAGroup3) in the $SYS$DIAG hierarchy. Select each device group to view the corresponding system items.

Launching the OPCUA Tag browser from any of the device groups under New_OPCUA_001 displays the two device groups (OPCUAGroupA and OPCUAGroupB) in the $SYS$DIAG hierarchy. Select each device group to view the corresponding system items.

OPC UA Methods

Methods are executed in the OPC UA Server through the OPC UA Client in the Gateway Communication Driver.

You can invoke methods in an OPC UA Server to:

- Trigger certain operations or calculations in the OPC UA Server
- Load data to the OPC UA Server
- Retrieve data from the OPC UA Server

Methods are called from scripts in client applications that are natively integrated with Gateway. Currently, you can use the script functions provided in the Application Server script library to invoke methods.

Refer to the respective client application's documentation for information on how to call Methods and for the associated scripting syntax.

Method Call Parameters

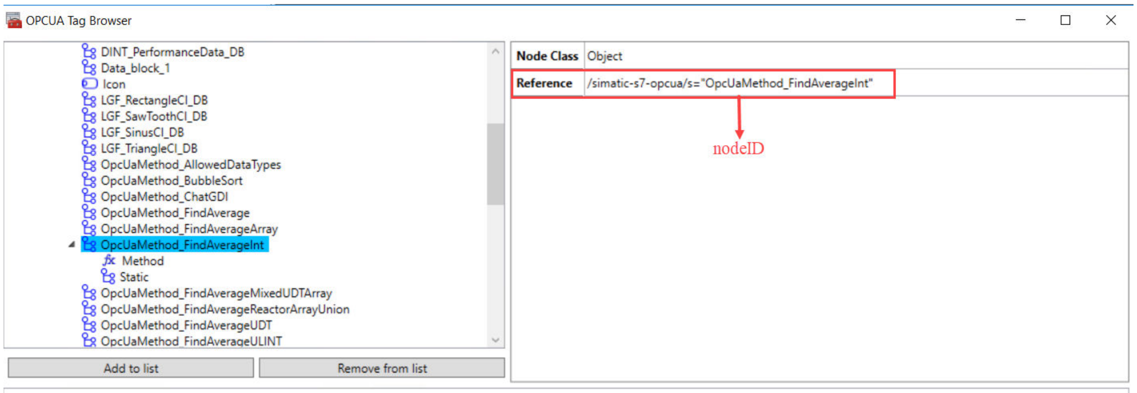
The following details may be required by a client application while scripting a method call to an OPC UA server:

To initialize a client to connect to the OPC UA client in the Gateway Communication Driver

- `string serverMachineName`: The name of the machine that hosts the Gateway Communication Driver.
- `string instanceName`: The PCS Scope Name configured in the Gateway Communication Driver.

To call a method

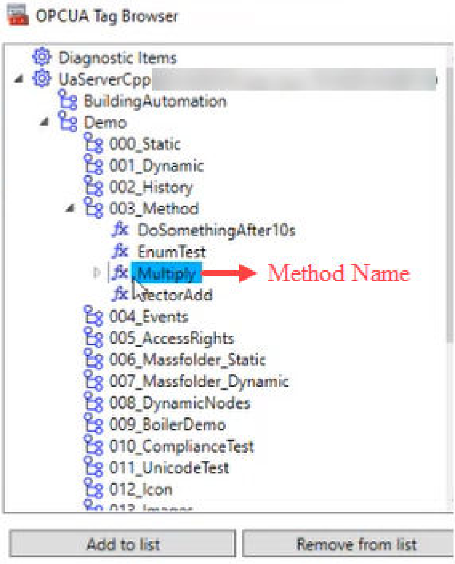
- `string sourceID`: The **Device Group Name** configured in the OPC UA client.
- `string nodeID`: The reference of the object hosting the method on the OPC UA server. You can access the `nodeID` in the OPC UA Tag Browser if it is exposed by the OPC UA server.



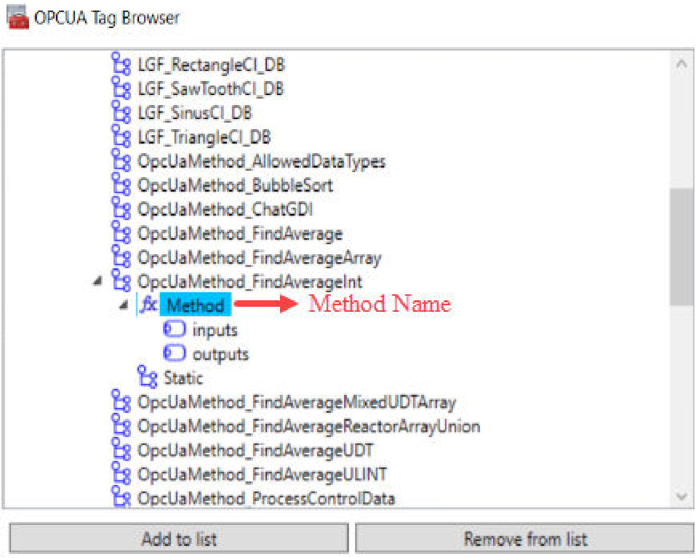
DINT_PerformanceData_DB
Data_block_1
Icon
LGF_RectangleCI_DB
LGF_SawToothCI_DB
LGF_SinusCI_DB
LGF_TriangleCI_DB
OpcUaMethod_AllowedDataTypes
OpcUaMethod_BubbleSort
OpcUaMethod_ChatGDI
OpcUaMethod_FindAverage
OpcUaMethod_FindAverageArray
OpcUaMethod_FindAverageInt
Method
Static
OpcUaMethod_FindAverageMixedUDTArray
OpcUaMethod_FindAverageReactorArrayUnion
OpcUaMethod_FindAverageUDT
OpcUaMethod_FindAverageULINT

Node Class	Object
Reference	/simatic-s7-opcua/s="OpcUaMethod_FindAverageInt"

- `string MethodName`: Name of the method as defined in the OPC UA server. You can access the name of the method in the OPC UA Tag Browser if it is exposed by the OPC UA server.



Diagnostic Items
UaServerCpp [redacted]
BuildingAutomation
Demo
000_Static
001_Dynamic
002_History
003_Method
DoSomethingAfter10s
EnumTest
Multiply
VectorAdd
004_Events
005_AccessRights
006_Massfolder_Static
007_Massfolder_Dynamic
008_DynamicNodes
009_BoilerDemo
010_ComplianceTest
011_UnicodeTest
012_Icon



LGF_RectangleCI_DB
LGF_SawToothCI_DB
LGF_SinusCI_DB
LGF_TriangleCI_DB
OpcUaMethod_AllowedDataTypes
OpcUaMethod_BubbleSort
OpcUaMethod_ChatGDI
OpcUaMethod_FindAverage
OpcUaMethod_FindAverageArray
OpcUaMethod_FindAverageInt
Method
inputs
outputs
Static
OpcUaMethod_FindAverageMixedUDTArray
OpcUaMethod_FindAverageReactorArrayUnion
OpcUaMethod_FindAverageUDT
OpcUaMethod_FindAverageULINT
OpcUaMethod_ProcessControlData

In the examples above, "Multiply" and "Method" are the Method names.

Supported Data Types

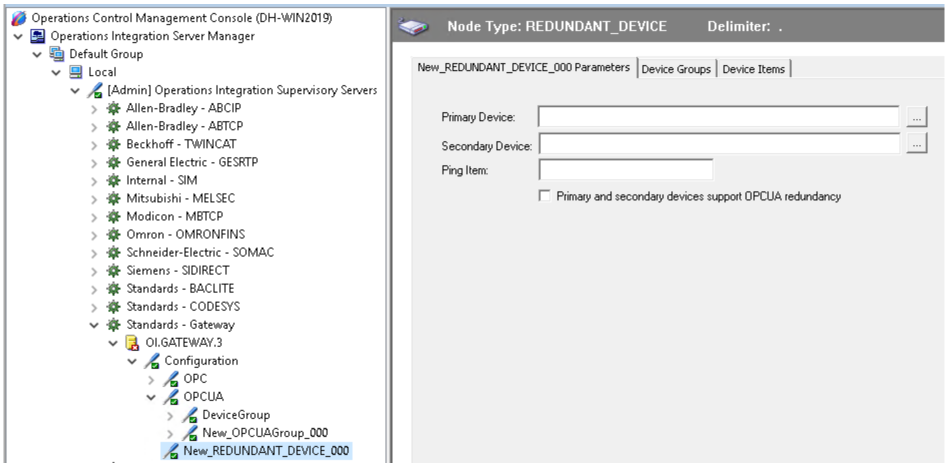
The following table lists the supported OPC UA data types while calling methods in the OPC UA server.

Supported OPC UA Data Types	Description
Boolean	Represents a true or false value.
SByte	Represents a signed 8-bit integer.
Byte	Represents an unsigned 8-bit integer.
Int16	Represents a signed 16-bit integer.
UInt16	Represents an unsigned 16-bit integer.
Int32	Represents a signed 32-bit integer.
UInt32	Represents an unsigned 32-bit integer.

Int64	Represents a signed 64-bit integer.
UInt64	Represents an unsigned 64-bit integer.
Float	Represents a 32-bit floating-point number.
Double	Represents a 64-bit floating-point number.
String	Represents a sequence of Unicode characters.
DateType	Represents a specific point in time.
LocalizedText	Represents a human-readable text with an optional locale identifier.
GenericStruct, GenericField	Represents a complex structured data type.

Configuring Redundant OPC UA Connection

As of Communications Driver Pack 2023, you can configure the Redundant Device Object in Gateway Communication Driver to leverage the Service Level exposed by the OPC UA Server.



Based on the OPC UA standard, an OPC UA server can provide a Service Level indicating its quality of service. The Service Level is divided into 4 tiers:

- 0 - Maintenance
- 1 - No Data
- 2 - 199 Degraded
- 200 - 255 Healthy

While the Gateway Communication Driver can fail over between primary and secondary connections when the configured ping item returns a bad quality, you can also check the box **Primary and secondary devices support OPCUA redundancy**. When the Service Level of active OPC UA server drops below the Healthy tier (value below 200), Gateway Communication Driver will fail over the active connection to the standby connection.

For more information refer to Configuring Device Redundancy topic under AVEVA™ Communication Drivers Pack Help.

OPC UA Item Names and 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	Description and Example
Default Namespace	`UAItemReference`	UAItemReference is the identifier of the item (complete path of the OPC UA Server item, followed by the NodeId) in OPC UA Server. It is resolved by the OPC UA Server, and the case is preserved. This is case sensitive Example: PLCport.PlCobject.N40:0
Namespace Alias	`/UANamespaceAlias/ UAItemReference`	UANamespaceAlias is the Namespace alias name used to identify the namespace in the OPC UA Server. This is configured in the OPC UA editor. It is replaced by the true namespace URI in UA Client, before it is sent to the server. This is case insensitive Example: /Data

For more information, see Using VT Item Suffixes.

Supported Node ID Types for OPC UA

The following table shows the different node ID types supported and the corresponding prefixes:

Node ID Type	Prefix
Integer	i/I
String	s/S
Opaque	b/B
GUID	g/G

In the item reference syntax, before the item name we need to add the node type.

Example:

/Data/i=10216

/ALIAS5/b=0xD31DF4DF8E7AEBBF3D

/ALIAS5/G={a7203570-3fef-4eff-8981-735d5646d1a9}

Chapter 4

Configuring Device Item Definitions

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, and 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.
2. To change the default name, double-click it and enter 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 apply the change.

To rename a device item from the list

1. Right-click on the device item to be renamed, and select the **Rename** command from the shortcut menu.
2. 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

- Right-click on the device item to be deleted, and 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 Gateway Communication Driver tree hierarchy, you are prompted to save the modifications to the configuration set.

To clear all device items

- Right-click anywhere in the **Device Items** configuration view, and 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 Gateway Communication Driver 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 Gateway Communication Driver 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, and select the **Export** command from the shortcut menu.
2. Select the folder into which the list is to be saved.
3. Name the list to be exported.
4. Click the **Save** button.

   The entire 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, and select the **Import** command from the shortcut menu.

2. Select the archived list (.csv file) to be imported, and click **Open**.

   The entire 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.

Resolving Item Names from Clients

Gateway Communication Driver resolves item names from its clients at runtime in the following order:

System items (those prefixed with SYS) > Device items (those defined in the **Device Items** configuration view) > All other items (validated directly from the PLC device)

Chapter 5

Managing the Gateway Communication Driver

Gateway Communication Driver Post-Configuration Steps

After you configure the Gateway Communication Driver, and before you can access data with your client application, follow the steps below.

1. Determine what kind of client applications are to be used with this Gateway Communication Driver.
2. Activate Gateway Communication Driver.

   Some client applications can programmatically activate the Gateway Communication Driver.

Archiving Configuration Sets

After your Gateway Communication Driver 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 the **Configuration** node under the Gateway Communication Driver hierarchy, and select **Archive Configuration Set**.
2. In the **Archive Configuration Set** configuration view, provide a Configuration Set Name.
3. Click **Archive**.

   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 Gateway Communication Driver is not running.
2. In the OI Server Manager, right-click the **Configuration** node under the Gateway Communication Driver hierarchy, and select **Use Another Configuration Set**.
3. Select a configuration set in the sub-menu.

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

Activating/Deactivating the Gateway Communication Driver

When you activate the Gateway Communication Driver, it starts communicating and accepting requests from client applications. Also, a Gateway Communication Driver can be activated by an OPC client connection request. There are three different modes of activating the gateway.

1. **Activate (Auto start after reboot)**: Activates the gateway. The Gateway Communication Driver is started and activated automatically when the computer starts up.
2. **Activate until reboot (Manual start after reboot)**: Activates the gateway. The Gateway Communication Driver gets deactivated after a reboot, and has to be activated manually.
3. **Desktop mode (Must start from command line)**: Activates the gateway from the command-line only, and not from the OCMC. This option is enabled for base instance of the Gateway Communication Driver only. For all cloned instances, this option is disabled.

Note: Use the desktop mode activation for DDE/FastDDE communications.

To activate the Gateway Communication Driver

1. In the OI Server Manager, navigate to the Gateway Communication Driver .
   a. Expand **Operations Integration Server Manager**, expand **Default Group**, and then expand **Local**.
   b. Expand **Standards- Gateway**.
2. Right-click **OI.Gateway.3** and select one of the modes to activate the server.

   Selecting any one mode of activation disables all other activation options in the menu. To activate the server in any other mode, you must deactivate the server first.

To deactivate the Gateway Communication Driver

1. In the OI Server Manager, navigate to the Gateway Communication Driver.
   a. Expand **Operations Integration Server Manager**, expand **Default Group**, and then expand **Local**.
   b. Expand **Standards- Gateway**.
2. Right-click **OI.Gateway.3** and then select **Deactivate (Must be activated to run again)**.
3. Read the warning message and click **Yes**.

   Deactivating your Communication Driver stops it from communicating with client applications.

Note: Communication Driver with active OPC clients does not stop until the last OPC client shuts down.

In-Proc/Out-of-Proc

The Communication Driver can run only as a stand-alone process (out-of-proc). If the option CLXCTX_ALL is chosen, out-of-proc activation for the Communication Driver is triggered. Explicitly starting as part of the client process (in-proc) is not currently Communication Driver supported. Activation using the CLSCTX_ACTIVATE_64_BIT_SERVER flag is also not supported.

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

Hot Configuration

Gateway Communication Driver is mostly hot-configurable, that is, changes to the parameter values take effect immediately while the Gateway Communication Driver is active.

The following parameters and features in the Gateway Communication Driver are hot-configurable:

- 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.

Note: If a configuration change is made when the Gateway Communication Driver is running, it is highly recommended to perform a reset on the corresponding hierarchy.

Chapter 6

Accessing Data in the Gateway Communication Driver

About Accessing Data in the Gateway Communication Driver

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

Creating device items in the Communication Driver is not required for the OPC client applications.

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 Gateway Communication Driver means.

To access an OPC item, the OPC client needs to connect to Gateway Communication Driver (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, Gateway Communication Driver 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 Gateway Communication Driver.

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 Gateway Communication Driver, the program name is OI.Gateway.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 Gateway Communication Driver under a specific data source for the common custom attributes between Gateway Communication Driver and the source, such as update interval. If not specified from the client, the default device group using the global configuration attribute values from Gateway Communication Driver 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 Gateway Communication Driver under the OI Server Manager, separated by delimiters.
- **item name**: A specific data element, the leaf of the hierarchy tree of Gateway Communication Driver, 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 Gateway Communication Driver. For more information, see Connecting to an OPC Data Source.

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 Gateway Communication Driver, the application name portion of the address is Gateway.
- **topic name**: Meaningful names are configured in Gateway Communication Driver 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 Gateway Communication Driver.

  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. For additional information, see MQTT Item Names.

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.3`
   b. Item Reference: `OPCUA1.Topic1.Device1.TankLevel`
2. A SuiteLink client accessing an item in an OPC UA data source:
   a. Connection:
      i. Application: `Gateway`
      ii. Topic: `OPCUA1_Topic1`
   Example: <New_OPCUA_000>_< New_OPCUAGroup_000>)
   b. Item Reference: `Device1.TankLevel`

Accessing Data Using PCS

The PCS item syntax for an IData client to communicate with the Gateway Communication Driver is:

`<OIServerScope>:<TopicName>.<ItemName>`

- **OIServerScope**: The syntax of the OI Server Scope is `OI$<OIServerInstanceName>$<ComputerName>`

where,

- **OI** uniquely identifies the scope name as belonging to the OI Product line
- **$** is a separator in case the parts of the scope need to be parsed. The OIServerInstanceName and ComputerName must not contain $.
- **OIServerInstanceName** is the instance of the Gateway Communication Driver as viewed in the OCMC.
- **ComputerName** is the name of the machine on which the Gateway Communication Driver is running.

- **TopicName:** A device group that is configured in the Gateway Communication Driver.
- **ItemName**: The full name of the item required by the specific protocol of the Gateway Communication Driver where the item is being subscribed.

Note: You can configure the PCS Scope Name of a Communication Driver instance to any name of your choice in the Configuration node. Refer to "Configuring Client Connectivity" in the AVEVA™ Communication Drivers Pack user guide for more details.

Chapter 7

Gateway Communication Driver Features

Instantiating the Gateway Communication Driver

Newer Communication Drivers that have been optimized for 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 Gateway Communication Driver. You can create a new instance, and clone an existing instance of Gateway Communication Driver.

Note: Creating multiple instances of the Communication Drivers requires a Professional level license. For more information, see Licensing for MQTT Connectivity.

To create a new instance of the Gateway Communication Driver

1. In the OI Server Manager, navigate to the Gateway Communication Driver.
   a. Expand **Operations Integration Server Manager**, expand **Default** Group, and then expand **Local**.
   b. Expand **Operations Integration Supervisory Servers**.
2. Right-click the **Standards - 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 instance is created.

To clone an existing Gateway Communication Driver instance

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

   c. Expand **Standards - 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.

Chapter 8

System Items

Standard System Items

System items provide you with easy access to Gateway Communication Driver'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 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.

In the ArchestrA 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 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.

For DDE/SuiteLink clients, you can access $SYS$Status always comes from the leaf level of the Gateway Communication Driver 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 ArchestrA data source, $SYS$Status is always good, even at the ArchestrA Group level.

Global System Items

The following global system items refer to information regarding the data source(s) Gateway Communication Driver is connected to.

System Item Name	Type/Access Rights	Description	Values
$SYS$Licensed	Boolean/ Read	Only MQTT Connection needs a license. All other connection types (DDE, SuiteLink, OPC, OPC-UA, ArchestrA, and InTouch) do not need a license.	The value is always true.
$SYS$ReadOnly	Boolean/Read	Binary status indication of the Read Only state of a Gateway Communication Driver. If TRUE, the Read/Write access of all items are overridden to read only and cannot be written. If an item is written, a line is logged in the OCMC Logger and the write request is rejected. If FALSE, the Communication Driver items are read/write or read only according to their individual configurations.	Range: 0, 1 1: Read only 0: Not read only

Device-Specific System Items

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

System Item Name	Type/Access Rights	Description	Values
$SYS$Status	Boolean/ Read	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. The status can be good even if individual items have errors.	RANGE: 0, 1 1: Gateway connection to the data source is intact. 0: Error communicating with the data source.

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 Gateway Communication Driver.

System Item Name	Type/ Access Rights	Description	Values
$SYS$UpdateInterval		Not used.	
$SYS$MaxInterval		Not used.	
$SYS$WriteComplete	Integer/ ReadWrite	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 If the value 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). If the	RANGE: -1, 0, 1 1: Write complete (no writes are pending – initial state). 0: Writes are pending. -1: Writes completed with errors.

System Item Name	Type/ Access Rights	Description	Values
		value of this item is zero, it cannot be poked.	
$SYS$ReadComplete	Integer/ ReadWrite	Accesses the state of initial reads on all items in the corresponding device group. Poking a 0 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.	RANGE: -1, 0, 1 1: Read complete (all values have been read). 0: Not all values have been read. -1: All values have been read but some have a non-good quality.
$SYS$ItemCount	DWord/ Read	Accesses the number of items in the corresponding device group. This item is read-only.	RANGE: 0… 2147483647 >=0: Number of active items.
$SYS$ActiveItemCount	DWord/ Read	Accesses the number of active items in the corresponding device group. This item is read-only.	RANGE: 0… 2147483647 >=0: Number of active items.
$SYS$ErrorCount	DWord/ Read	Accesses the number of all items (active and inactive) that have errors (non-good OPC quality) in the corresponding topic. If the communications status of a device group is bad, all items have errors. This item is read-only.	RANGE: 0… 2147483647 >=0: Number of all items (active and inactive) with errors.

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

Gateway Communication Driver-Specific System Items

The following system items refer to specific information regarding Gateway Communication Driver. The Gateway Communication Driver-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 Gateway Communication Driver has established a successful connection to the configured data source and topic (if any).	True, False True: Connected to the data source False: Disconnected
$SYS$Gateway ConnectionStatus String	String/ Read-Only	Indicates whether Gateway Communication Driver has established a successful connection to the configured data source and topic (if any).	Connected, Disconnected
$SYS$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.	True, False True: Triggers reconnect attempt. If data source is already connected, it is disconnected and then reconnected. False: Does nothing. Default value.

System Item Name	Type/ Access Rights	Description	Values
$SYS$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.	0...2147483647 If 0 is specified, then the fastest update interval supported by the OPC server is used.
$SYS$MQTTIsLicensed	Boolean/ Read	Indicates if a valid license is present for the MQTT connectivity.	0, 1 0: No valid license exists. 1: Valid license exists

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, client can conveniently switch to the secondary device.	TRUE, FALSE
$SYS$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

System Item Name	Type/ Access Rights	Description	Values
$SYS$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, as 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 as 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.

Note: 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 : diagnostic items for the entire OPC UA hierarchy.
- Subscription Diagnostic Items : diagnostic information for a specific subscription.
- OPC UA Client Diagnostic Items : diagnostic items for the entire OPC UA hierarchy.

The OPC UA diagnostic items follow a specific syntax. For syntax information, see Diagnostic Item Syntax .

Diagnostic Item Syntax

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

The following syntax illustrates the full path to a diagnostic item.

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

where,

$SYS$DIAG: special namespace containing diagnostic items in the service

<Object Path>: substituted with a one or more object qualifiers, each separated by '/'

<Item>: 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>: One of the diagnostic item names from the following table

<Item>	**Type**	**R/W**	**Description**
ActiveItemCount	Int32	R	Number of registered items active in all connections
ConnectionEvents	Int32	R	Number of connection adds or deletes since Gateway Communication Driver starts. This counter will rollover to 0 after value reaches Int32_Max.
MonitoredItemCount	Int32	R	Number of monitored items in all subscriptions/connections
MonitoredItemErrCount	Int32	R	Number of items with bad quality in all subscriptions/connections
NodeName	String	R	Name of the node the Gateway Communication Driver is running on
ResetTotals	Bool	R/W	Reset the values of the following system items: ConnectionEvents, TotalReads, TotalWrites, TotalItemReads, TotalItemWrites, TotalReadErrors, TotalItemReadErrors, TotalWriteErrors and TotalItemWriteErrors
Run	Bool	R/W	Setting this item to True (non-zero) enables the system items calculation and reporting; Setting this item to False (0) disables the system items calculation and reporting. The quality of system items is set to 0x14 (LAST KNOWN VALUE) if the system item reporting is not enabled.

			Default value: True
StartDateTime	DateTime	R	Date and Time (in UTC) when the Gateway Communication Driver was started
SubscriptionCount	Int32	R	Number of subscriptions in all connections
TotalItemReadErrors	Int32	R	Number of read item errors in all connections since the Gateway Communication Driver starts or last reset. (Does not include subscription updates or items in the diagnostic namespace)
TotalItemReads	Int32	R	Number of read items in all connections since the Gateway Communication Driver starts or last reset. (Does not include subscription updates or items in the diagnostic namespace)
TotalItemWriteErrors	Int32	R	Number of write item errors in all connections since the Gateway Communication Driver starts or last reset. (Does not include items from the diagnostic namespace)
TotalItemWrites	Int32	R	Number of write items in all connections since the Gateway Communication Driver starts or last reset. (Does not include items from the diagnostic namespace)
TotalReadErrors	Int32	R	Number of read operation errors in all connections since the Gateway Communication Driver starts or last reset. (An error is counted if one or more item reads failed, or the entire read is not executed)
TotalReads	Int32	R	Number of reads operations in all connections since the Gateway Communication Driver starts or last reset.
TotalWriteErrors	Int32	R	Number of write operation errors in all connections since the Gateway Communication Driver starts or last reset. (An error is counted if one or more item writes failed, or the entire write is not executed)

TotalWrites	Int32	R	Number of writes operations in all connections since the Gateway Communication Driver starts or last reset.
UpTime	Int32	R	Elapsed time in seconds since the Gateway Communication Driver was started.

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=<DeviceGroup>/<Item>
```

where,

<DeviceGroup>: device group hierarchy name

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

For an OPC UA connection with multiple subscription device groups, the system items hierarchy ($SYS$DIAG) displays the system items associated with all the device groups in that OPC UA connection.

<Item>	**Type**	**R/W**	**Description**
ActiveItemCount	Int32	R	Number of active items in this subscription
IsEnabled	Bool	R	Subscription group is active when True
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
PublishInterval	Int32	R	The frequency of data updates for this subscription in milliseconds (ms)
SampleInterval	Int32	R	The fastest rate at which data updates are getting sampled for this subscription in milliseconds (ms)
ResetTotals	Bool	R/W	Reset TotalWrites, and TotalUpdates
TotalUpdates	Int32	R	Total number of received values since subscription creation
TotalWrites	Int32	R	Total number of write operations since subscription creation

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>: one of the diagnostic item names from the following table.

<Item>	Type	R/W	Description
CertificateExpiry	DateTime	R	Date & Time of application certificate expiry (Time in UTC)
CertificateSubject	String	R	Certificate subject name
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
IsConnected	BOOL	R	At least one UA Server session established=True, No sessions established=False
KeepAliveInterval	Int32	R	Keep Alive interval in milliseconds
MessagePolicy	Int32	R	0=none, 1=Sign, 2=Sign&Encrypt
NamespaceAliases	String[]	R	Array of namespace aliases.
NamespaceUris	String[]	R	Array of namespace URIs
ResetTotals	BOOL	R/W	Setting this to True (1) resets the value of TotalConnectionRetries system item
SecurityPolicy	Int32	R	0=none, 1=Basic256, 2=Basic128Rsa15
ServerUri	String	R	End point value of the OPC UA Server connected from Gateway Communication Driver
TotalConnectionRetries	Int32	R	Number of connection retries since the Gateway Communication Driver starts. Note: Item in intended to indicate the number of connection retries between the UA Client and the UA Server.
UserName	String	R	User Name which is configured for secured session with UA Server (empty if IsAnonymousUser=True)

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 the protocols that support it, the data quality is consistent with the OPC Quality.

- If supported by both source and client, data quality is passed through the Gateway Communication Driver unmodified.
- If not supported by the client, data quality is dropped (irrespective it is supported by the data source is not).
- If supported by client, and not supported by the source, the data quality is fabricated and is always good.

An exception is when Gateway Communication Driver cannot communicate with the target data source.

Chapter 9

Reference

Gateway Communication Driver Architecture

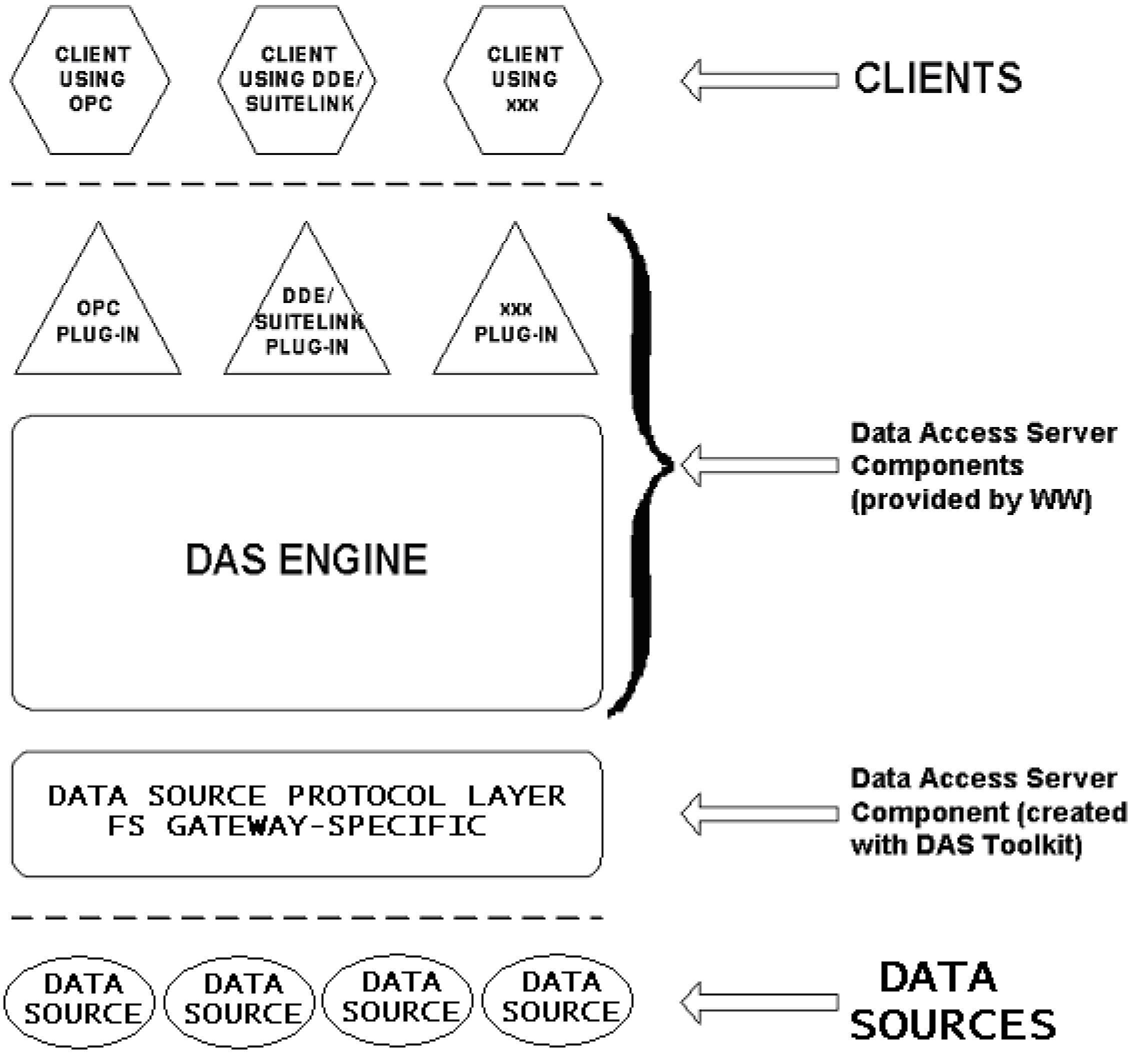
Gateway Communication Driver is a collection of components that work in concert to provide communications access with a variety of data sources and clients. These components include:

- **OI Server Manager**: This is the MMC snap-in, that is part of the OCMC suite of utilities, supplied with Gateway Communication Driver . It provides the necessary user-interface for diagnostics, configuration, and activation.
- **Client Plug-ins**: These are the components that are added to Gateway Communication Driver 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 Gateway Communication Driver to define the communications between particular data sources and clients.

Gateway Communication Driver is comprised of three physical parts:

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

The following diagram describes the Gateway Communication Driver system architecture.


CLIENT USING OPC
CLIENT USING DDE/ SUITELINK
CLIENT USING XXX
CLIENTS
OPC PLUG-IN
DDE/ SUITELINK PLUG-IN
XXX PLUG-IN
DAS ENGINE
Data Access Server Components (provided by WW)
DATA SOURCE PROTOCOL LAYER FS GATEWAY-SPECIFIC
Data Access Server Component (created with DAS Toolkit)
DATA SOURCE
DATA SOURCE
DATA SOURCE
DATA SOURCE
DATA SOURCES

Each physical part of Gateway Communication Driver is comprised of three sets of modules, each required for a fully functioning Gateway Communication Driver.

- a set of .exe and/or .dll modules
- the plug-ins and the OI Engine
- he Data Source Protocol Layer (Gateway Communication Driver-specific) created by the Toolkit user

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.

Communication Driver Engine

The Communication Driver 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

Gateway Communication Driver has the following characteristics:

- The OI Engine is dynamically linked to the other Gateway Communication Driver components. In other words, a new OI 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 Engine would attach to all existing Gateway Communication Driver 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 Gateway Communication Driver without any involvement of the code of the other components.
- Gateway Communication Driver 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 Gateway Communication Driver as well as other Communication Drivers. 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 Gateway Communication Driver as well as all Communication Driver, in addition to Gateway Communication Driver-specific/developer-defined diagnostic data.

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

Using Self-Signed Digital Certificates

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 connectivity to exchange data with any OPC UA compliant server. You can use a variety of third-party utilities to generate and manage digital certificates.

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.

Location of OPC UA Server Public Certificate

Gateway Communication Driver automatically trusts the external OPC UA servers if their public certificates are stored in the following folder in DER format.

Public certificate (.der file): **C:\ProgramData\Wonderware\OI-Server\$Operations Integration Supervisory Servers$\OI.GATEWAY\CertificateStores\trusted\certs folder**

If you start the Gateway Communication Driver by connecting it to any OPC UA server, Gateway Communication Driver will also automatically deposit the OPC UA server public certificate to the above folder.

Location of Gateway Communication Driver Public Certificate

To establish trust of Gateway Communication Driver to the external OPC UA server, see the documentation of the external OPC UA server. You may need to import the public certificate file of Gateway Communication Driver to the certificate configuration tool of the external OPC UA server provider. The public certificate of Gateway Communication Driver can be found at:

Public certificate (.der file): **C:\ProgramData\Wonderware\OI-Server\$Operations Integration Supervisory Servers$\OI.GATEWAY\CertificateStores\certs\OIGateway OPCUA.der**

Chapter 10

Troubleshooting

Using Troubleshooting Tools

This group of help topics describes troubleshooting tools that can be used to deal with Communication Driver 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 Communication Driver. 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 Communication Driver.

You can troubleshoot problems with Gateway Communication Driver using the Log Viewer, a snap-in to the OCMC. 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 InTouch HMI software. The client application can use system device items to determine the status of nodes and the values of some parameters.

To determine the version of your Gateway Communication Driver

1. Locate the Gateway DLL (Gateway.dll).
2. Right-click and select **Properties.**

   The **Properties** dialog appears.
3. Select the **Version** tab.

You can find the version of your Gateway Communication Driver listed under **File Version**.

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.

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

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

where,

Gateway: name of Gateway Communication Driver application.

<Device Group>: exact device group defined in Gateway Communication Driver for the data source.

$SYS$Status: 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>: assembly of hierarchy node names leading to a specific data source.

$SYS$Status: 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, Gateway Communication Driver 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.

Connectivity with the OPC UA Server

The connectivity with the OPC UA Server can fail for one or more of the following reasons:

- The OPC UA Server is not running, and is is not discoverable.
- The configuration on the OPC UA faceplate does not match that on the OPC UA Server.
- The firewall settings on a specified port blocks the OPC UA Server connectivity.

Diagnosing the Firewall Constraints

You can diagnose the firewall constraints leading to the OPC UA server connectivity failure by one of the following methods.

Method 1: Using Telnet to diagnose the connection

1. Enable the **Telnet Client** Windows feature in W**indows Features** configuration panel.

2. Open the Command Prompt window, and type `Telnet <IP Address> <port>`

   Example: `Telnet 10.228.106.139 48010`

3. If the connection is successful, the Command Prompt window goes blank. Else, the command will timeout.

Method 2: Use PsPing utility from Windows SysInternals

1. Open the Command Prompt window, and type `Psping -t <IP Address>:<Port Number>`

   Example: `psping –t 10.228.106.139 48010`

2. If the target node/port is reachable and not blocked by the firewall, the command will return the duration in milliseconds.

Resolving the OPC UA Server connectivity failure

Ensure that the firewall configuration in the target node (where OPC UA server is running) allows this connection. You can manage the firewall settings using one of the following solutions.

Solution 1: Register the program through an Inbound Rule (Program)

1. Open the **Windows Defender Firewall** control panel item, and click **Advanced Settings**.
2. Click **Inbound Rules**, and click **New Rule**.
3. In the **New Inbound Rule Wizard**, select **Program**, and click **Next**.
4. Select **This program path:**, and use the Browse button to select the program (.exe file), and click **Next**.
5. Select **Allow the connection** and click **Next**.
6. For **When does this rule apply?**, select **Domain**, and click **Next**.
7. Give a name and description for the setting, and click **Finish**.

Solution 2: Register the program through an Inbound Rule (Port)

1. Open the **Windows Defender Firewall** control panel item, and click **Advanced Settings**.
2. Click **Inbound Rules**, and click **New Rule**.
3. In the **New Inbound Rule Wizard**, select **Port**, and click **Next**.
4. Select **TCP**, and enter the port number in the **Specific local ports** field. Click **Next**.
5. Select **Allow the connection** and click **Next**.
6. For **When does this rule apply?**, select **Domain**, and click **Next**.
7. Give a name and description for the setting, and click **Finish**.

Solution 3: Modify the Incoming Connection rule

1. Open the **Windows Defender Firewall** control panel item, and click **Turn Windows Defender Firewall on or off.**
2. Under **Domain network settings**, clear the **Block all incoming connections…** check box.

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 Communication Driver conversations.

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

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

Note: These items represent the status of communication between the client and Gateway Communication Driver.

Using DDEStatus and IOStatus in Excel

The status of communications between Gateway Communication Driver 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: name of the InTouch application.

[DDE][IO] Status: built-in topic name used to monitor the status of communications between Gateway Communication Driver and InTouch.

ArchestrA: exact access name defined in Gateway Communication Driver for the data source.

Reading Values from the Gateway Communication Driver into Excel

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

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

Example formula:

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

where,

Gateway: name of the Gateway Communication Driver application.

ArchestrA: exact device group name defined in Gateway Communication Driver for the data source.

<tagname>: 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,Gateway Communication Driver 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 Gateway Communication Driver 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.

Gateway Communication Driver 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 Communication Driver to the data source succeeds or fails.

In the case of write success, the Gateway Communication Driver 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 Communication Driver 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 Gateway Communication Driver 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 GetErrorString.

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

- If ArchestrA responds with Nak, the Gateway Communication Driver sends an E_FAIL error code to the OPC Client.

- If Gateway Communication Driver 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 Gateway Communication Driver or ArchestrA, the OPC_E_INVALIDHANDLE error code is sent.
- If the item name is not valid in Gateway Communication Driver or ArchestrA, 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, Gateway Communication Driver sends an E_FAIL error code to the OPC Client.
- If Gateway Communication Driver 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 Gateway Communication Driver or the data source, the OPC_E_INVALIDHANDLE error code is sent.
- If the item name is not valid in Gateway Communication Driver or the data source, the OPC_E_UNKNOWNITEMID error code is sent.

Runtime Diagnostics and Error Reporting

For each data source connection, Gateway Communication Driver 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 Communication Driver 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.

- In the case of a DDE/SuiteLink data source, the connection is to an application and a topic.
- In the case of an ArchestrA data source, the connection is to a Platform through Message Exchange.
- In the case of an OPC data source, the connection is to an OPC Server through COM/DCOM object creation.

Communication Failures

This section describes the behavior of the Gateway Communication Driver in case of failed communications with the data source, client, or a remote OPC server.

Failed Communication with Data Source

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

- The Gateway Communication Driver 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 Communication Driver is not responsible for starting the data source server, unless the source protocol supports it. OPC has this capability.

- The Gateway Communication Driver 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.

The behavior of the Gateway Communication Driver during the communications failures with an ArchestrA data source is described below:

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

Failed Communication with Client

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

- The Gateway Communication Driver un-subscribes (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, Gateway Communication Driver maintains subscriptions to all items on the data source previously subscribed to by the failed client.)
- The Gateway Communication Driver accepts future attempts to reconnect from the client. Reconnection is the responsibility of the client.

Failed Communication with a remote OPC Server

The Gateway Communication Driver behaves in the following manner in case of failed communication with a remote OPC Server:

- The OPC Client application fails to create an OPC Group
- The OPC Client application does not display data updates. Consequently, data values remain unchanged or display ‘Bad Quality’
- The logger reports a COM error 0x80040202

This cause of this issue can be either an Invalid Username/Password, or Guest-only access. For the troubleshooting steps of the above issues, refer to the *Communication Failure: OPC Callback and Issues* section in the Communication Drivers Pack Help.