AVEVA™ Communication Drivers Pack

User Guide

Publication date: Friday, October 27, 2023

Publication ID: 868186

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

AVEVA

Chapter 1

About AVEVA™ Communication Drivers Pack Help

Using This Help

This document describes the user interface, features, and functions of the Communication Drivers Core. The OI Core and OI Server are now called as Communication Drivers Core and Communication Driver respectively. However, you may see “OI Core” or "OI Server" in some instances of documents, Operations Control Management Console (OCMC) tree hierarchy, upgrade path, and so on.

Technical Support

The Technical Support offers a variety of support options to answer any questions on products and their implementation. Before you contact the Technical Support, refer to the relevant section(s) in this documentation for a possible solution to the problem. If you need to contact Technical Support for help, keep the following information ready:

- version of the operating system you are using
- steps to recreate the problem
- exact wording of the error messages (if any)
- any relevant output listing from the Log Viewer or other diagnostic applications
- details of what you did to try to solve the problem(s) and your results
- In case of an ongoing issue, the case number assigned by the Technical Support.

Documentation Conventions

This documentation uses the following conventions.

- **Initial Capitals:** Paths and file names.

- **Bold:** Menus, commands, dialog box names, and dialog box options.
- `Monospace`: Code samples and display text.

AVEVA

Chapter 2

Getting Started

About the Communication Drivers Core

A Communication Driver is a component of a software system that connects your software application with sources of information on the factory floor. The OI Server Manager (also known as Communication Drivers Core) is a part of the Operations Control Management Console (OCMC) suite of utilities (previously known as System Management Console (SMC)). It enables the configuration, diagnosis, activation, or deactivation of a local Communication Driver located on a different node from the OI Server Manager.

Servers developed with the Toolkit can be stand-alone products or designed for use with System Platform based products. You can open multiple instances of the OI Server Manager at the same time; however, you can use only the first instance to create device hierarchies and configure a Communication Driver. In all other instances of the OI Server Manager, hierarchy and configuration settings are set to read-only.

Note: The OI Server Manager user interface may look different on different versions of Microsoft Windows. In addition, the version of Microsoft Management Console (MMC) installed on your computer may affect the content and behavior of the user interface.

Before You Begin

Before you begin configuring the Communication Driver, verify the following items:

- A PC is set up with the necessary network cards, and connected to the necessary networks.

- The Communication Driver and the OI Server Manager are installed with the proper licenses. For more information, see the AVEVA Enterprise Licensing documentation.
- The client software is installed.
- The device(s) is/are connected (networked) and, if necessary, programmed.
- Ensure you have the additional information necessary for configuration:
  - device network configuration and addresses
  - data items needed for the client application
  - device name/topic name/group name
  - desired update intervals

Note: The Secure Development Lifecycle (SDL) guidelines recommend against using automatically created users like aaUser and aaAdminUser with well-known or publicly documented passwords. Hence, the Communication Drivers Core supports the non-admin user to run the OCMC and configure software.

Supported Client Protocols

Client applications can connect to a Communication Driver using the following protocols:

- OPC
- SuiteLink
- DDE/FastDDE
- PCS

OPC

OPC (OLE for Process Control) is a non-proprietary set of standard interfaces based upon Microsoft's OLE/COM technology. This standard makes possible interoperability between automation/control applications, field systems/devices, and business/office applications.By providing a common and high performance interface, OPC avoids the traditional requirement of software/application developers to write custom drivers to exchange data with field devices. The work is done once, and can then be reused by HMI, SCADA, control and custom applications.

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.

SuiteLink is not a replacement for DDE, FastDDE, or NetDDE. 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 timestamp and quality indicator on all data values delivered to VTQ-aware clients.
- Extensive diagnostics of the data throughput, server loading, computer resource consumption, and network transport are made accessible through the operating system's performance monitor. This feature is critical for the operation and maintenance of distributed industrial networks.

- Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large node count.
- The network transport protocol is TCP/IP using Microsoft's standard WinSock interface.

Note: As of System Platform 2023, the SuiteLink connection is by default set to secure with TLS encryption. For more information refer to the Secure SuiteLink Connection section.

DDE/FastDDE

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

Important! Local DDE is supported only when the Communication Driver is configured as "Desktop Mode" and activated from its executable file or launched from InTouch. Local DDE is not supported when the Communication Driver is activated from the Operations Control Management Console (OCMC).

DDE

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

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

FastDDE

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

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

PCS

The Platform Common Services (PCS) framework enables communication between all AVEVA services that are listed on the framework.

PCS IData

A Communication Driver acts as a PCS IData provider. The PCS framework enables IData clients to directly access data from the Communication Driver at runtime.

With PCS connectivity, all activated Communication Drivers can communicate with IData clients such as Application Server. At present, IData versions V2 and V3 are supported.

OPC UA Methods

The PCS framework enables the invocation of methods in an OPC UA Server. Method calls are facilitated through the OPC UA Client in the Gateway Communication Driver.

Note: As part of the CDP installation, you can install the PCS plug-in for Communication Drivers, along with the PCS framework components (PCS Runtime and PCS Service Repository), only if the PCS framework is installed on the same machine. The PCS framework is installed with System Platform.
OPC UA Methods is supported on PCS framework 8.0. If an earlier version of the PCS framework is present on the machine, it is automatically upgraded to the 8.0 version during the CDP installation.
If no version of the PCS Framework is present on the machine, the PCS plug-in and PCS framework components cannot be installed during the CDP installation.

Windows Firewall Considerations

For the Communication Driver to function correctly on a firewall-enabled computer, specific applications or port numbers must be added to the firewall exception list. By default, the Communication Driver installation program adds the required entries in the firewall exception list. If you do not want the installation program to make the entries, you can add them manually. For information on how to add the applications and port numbers to the firewall exception list, see your firewall or Windows security documentation.

Applications in Firewall Exception List

Ensure the following applications are added to the firewall exception list (either automatically during installation or manually) on the computer where the OI Server Manager is installed:

- **DASWrapper.exe**
- **aaLogger.exe**
- **DASAgent.exe**
- **dllhost.exe**
- **mmc.exe**
- **OPCEnum.exe**
- **Slssvc.exe**
- **Asb.Watchdog.exe**
- **Asb.ServiceManager.exe**
- **Asb.Discovery.exe**
- **PCS.IdentityManager.Host.exe**
- **PCS.Portal.exe**

Port Numbers in the Firewall Exception List

Ensure the following port numbers are added to the firewall exception list (either automatically during installation or manually) on the computer where the OI Server Manager is installed.

- SuiteLink:
  - TCP 5413
- OPC:
  - TCP 135 (Microsoft DCOM)
  - TCP range 49152-65535 (OPC Server-specific ports)

- PCS:
  - TCP 808 for WCF communications between PCS Clients and Communication Drivers
  - TCP 7084 for node registration
  - TCP 7085 for node pairing
  - TCP 443 for PCS clients to discover and connect to Communication Drivers
  - UDP 1900 to discover SMS nodes on the network during SMS configuration
- TCP 445 for direct TCP/IP MS Networking access
- TCP and/or UDP ports configured in the device for use with this Communication Driver. For more information, see the documentation for the specific Communication Driver.

Uninstalling the Communication Driver does not remove the firewall exception list entries. You must delete the firewall exception list entries manually. For more information, see your firewall or Windows security documentation.

Installing a Communication Driver

Installation Checklist

The Communication Drivers Core must be installed prior to the installation of the Communication Driver.

Before installing the Communication Driver, please ensure the following:

- Exit all AVEVA programs, including executable (.exe) files and services
- Un-deploy the platforms where Communication Drivers Core will be installed
- Disconnect or close all third party OPC clients of existing DAServers/OI Servers/Communication Drivers
- Deactivate all DAServers/OI Servers/Communication Drivers

Installation Procedure

Download the Communications Driver from the **Connectivity Hub** section of the **Global Customer Support**.

To install the Communication Driver, double-click **Setup.msi**. When setup is complete, the message "Setup was successful" is displayed.

Silent Installation of Communication Drivers Core

The Communication Drivers Core supports silent (command line) installation. This features enables non-interactive installation of the Communication Drivers Core and its prerequisite products.

The prerequisite products that are installed during the silent installation includes:

- Activation-based Licensing components
  - AVEVA Enterprise Licensing Platform
  - AVEVA Enterprise License Manager
  - AVEVA Enterprise License Server

Silent_Install_Setup.bat is run from the command line and accepts install or uninstall as an argument to perform the respective installation/ uninstallation process.

Silent installation syntax

The basic syntax of the silent installation command consists of the full path to the Silent_Install_Setup.bat file (typically the DVD drive designation on your local computer), and the command line switch for silent installation/ un-installation. For example:

```
<DVD>: AVEVA Communication Drivers Pack 2023>Silent_Install_Setup.bat /<argument>
```

The usage instructions can be retrieved by the following command line:

```
<DVD>: AVEVA Communication Drivers Pack 2023>Silent_Install_Setup.bat /?
```

The above command line returns the supported operations - install and uninstall. The /install and /uninstall switches completely disable the graphical user interface of Silent_Install_Setup.bat. There is no input from or feedback to the end user.

To install the Communication Drivers Core and the prerequisite products

1. Open a command prompt using **Run as administrator**.
2. Run the batch file set up with the **/install** switch.

   ```
   C: AVEVA Communication Drivers Pack 2023.1>Silent_Install_Setup.bat /install
   ```

3. Wait for the system to display the confirmation message - **Silent installation completed successfully.**

Note: Before initiating the silent installation of Communication Drivers Core, it is strongly recommended that you stop all the DAServers, OI Servers, and Communication Drivers.

To uninstall the Communication Drivers Core

1. Open a command prompt using **Run as administrator**.
2. Run the batch file set up with the **/uninstall** switch.

   ```
   C: AVEVA Communication Drivers Pack 2023.1>Silent_Install_Setup.bat /uninstall
   ```

3. Wait for the system to display the confirmation message - **Silent uninstallation completed successfully.**

Note: The silent uninstall process uninstalls only the Communication Drivers Core and the dependent components. All the prerequisites and the independent components should be uninstalled manually.

Installing specific Communication Drivers silently

The silent install command by default installs all the Communication Drivers supported by the Communication Drivers Pack. To install specific Communication Drivers:

1. Open the **response.txt** file with a text editor. This file is present in the AVEVA Communication Drivers Pack 2023.1 installation media.
2. In the <install> tag, replace **ALL** in the line "FeatureForm.SFeatureList=ALL" with:

   ```
   AVEVA Communication Drivers Pack.<name of the server>_Server.
   ```

   For example, to install only the ABCIP Communication Driver, the <install> tag content is:

   ```
   FeatureForm.SFeatureList=AVEVA Communication Drivers Pack.ABCIP_Server.
   ```

   To install the ABCIP, GESRTP, and SIDIRECT Communication Drivers together, the <install> tag content is:

```
FeatureForm.SFeatureList=AVEVA Communication Drivers Pack.ABCIP_Server;AVEVA
Communication Drivers Pack.GESRTP_Server;AVEVA Communication Drivers
Pack.SIDIRECT_Server
```

3. Save the **response.txt** file in the same folder.
4. Run the silent install command.

Checklist for Setting Up a Communication Driver

If you are setting up a Communication Driver for the first time, perform these tasks in the order listed.

1. Review the items described in Before You Begin.
2. Learn how to navigate the OI Server Manager in the OCMC. See Navigating the OI Server Manager.
3. Add and configure objects in the Communication Driver configuration. See Configuring Your Communication Driver.
4. Configure the global parameters. See Configuring Global Parameters.
5. Add one or more device groups. See Managing Device Groups.
6. Add device items. See Managing Device Items.
7. Activate the Communication Driver. See Activating/Deactivating the Communication Driver.
8. Access data from the client, including specifying device item references. See Accessing the Data in Your Communication Driver.
9. Troubleshoot any problems. See Troubleshooting.

Licensing

The AVEVA Communication Drivers Pack uses the AVEVA Enterprising Licensing platform to effectively manage its product licenses.

With activation-based licensing, the Communication Drivers Core can acquire the license locally or remotely as follows:

- Locally, if the local node acts as a license server
- Remotely, if a designated node on the network acts as a license server

The License Manager and License Server are required for license activation, deactivation, and maintaining the license rights. These components can be installed separately, or as a selectable feature during the Communication Drivers Pack install. For a node to act as a license server, the "License Server" should be installed during the installation process.

For more information about AVEVA Enterprise Licensing, see the AVEVA Enterprise Licensing guide.

Configuring the License Server

The License Manager and License Server can be installed separately, on the local machine, or on remote machines. After installation, you can configure the machine name on which the server is installed and the local port used to access that server.

To configure the License Server

1. Start the Configurator.
2. On the left navigation pane, expand **AVEVA Enterprise Licensing Platform**, and select **AVEVA Enterprise License Server**.

   The **AVEVA Enterprise License Server** configuration screen appears.

   a. **Primary Server Name:** If the License Server is not installed on the local node, enter the License Server name, or select a server name from the drop down list of previously-configured License Servers (if any).

Note: This is the IP address/machine name of the server that hosts the relevant licenses.

   a. **Server Port:** Enter the server port number. The default port number is 55559.

   b. **Agent Port:** Enter the license server agent port number. The default port number is 59200.

   c. **Legacy Server Port:** Enter the legacy license server port number (license server installed before version 4.0). The default port number is 55555.

   d. Click **Test Connection** to verify the details are correct.

      The validation messages are displayed in the Configuration Messages section.

3. Click **Configure**.

   The license(s) are released from the host machine.

Reserving Licenses to Nodes

The license server grants license to nodes on a first-come first-serve basis. If you are utilizing a centralized license server, and the specific node that you are running requires a specific license functionality, you must reserve a Communication Driver license (Standard or Professional) to the node.

If a license is reserved to a node, any other node that requests a license first will retrieve it.

For example: Consider a license server that contains a mix of standard and professional licenses. If a particular node (say, node X) requires to run multi-instance functionality, which is a professional feature, ensure to reserve a professional license to that node. Otherwise, other nodes may allocate all the available professional licenses, preventing the node X from retrieving the professional license.

Activating a License

To activate a license

1. Start the License Manager.

   The License Manager is browser-based, and is located in the AVEVA folder (Start > AVEVA > AVEVA Enterprise License Manager).

   Alternatively, launch the URL in the browser: **https://<hostname>/AELicenseManager**; where <hostname> is the name of the computer hosting License Manager.

2. Activate the license for the machine.

   Once the license is activated, the logger shows a success message.

For detailed information about activating licenses, refer to the "Activating and Acquiring Licenses" section AVEVA™ Enterprise Licensing Help.

Deactivating a License

To deactivate a license:

1. Start the License Manager.

   The License Manager is browser-based, and is located in the AVEVA folder (Start > AVEVA > AVEVA Enterprise License Manager).

   Alternatively, launch the URL in the browser: **http://<hostname>/AELicenseManager**; where <hostname> is the name of the computer hosting License Manager.

2. Select the license in the license manager grid, and click **Deactivate.**

   Once deactivated successfully, the logger shows a success message.

For detailed information about deactivating licenses, refer to the "Activating and Acquiring Licenses" section in the AVEVA™ Enterprise Licensing Help.

Navigating the OI Server Manager

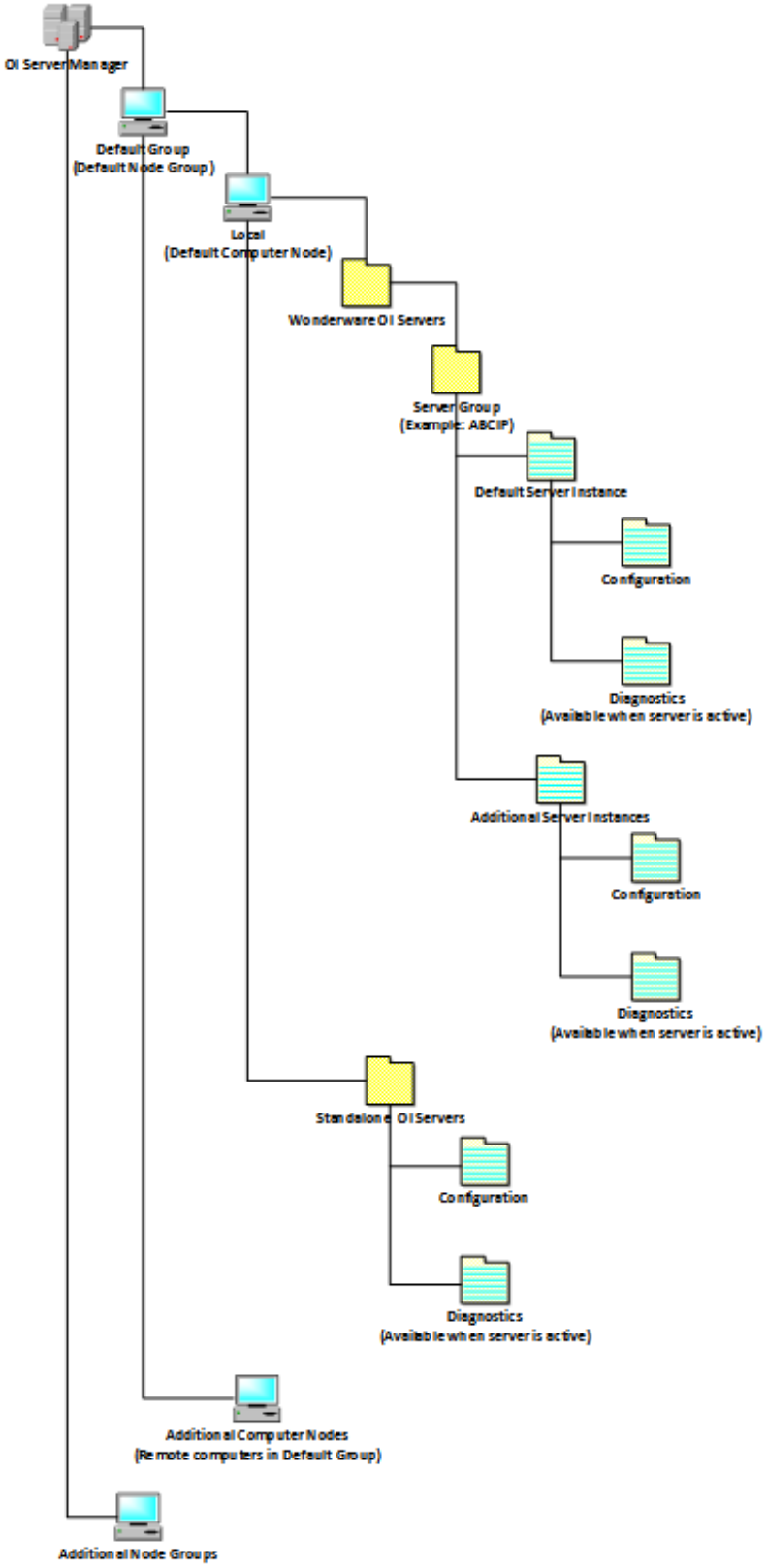
You can access the OI Server Manager from the OCMC. The OI Server Manager has a hierarchical tree of items, called the console tree, and a configuration pane, called the details pane. The configuration pane on the right, also called a faceplate, varies depending on the object being configured.

The OI Server Manager has one or more node groups in its hierarchy, and each node group comprises one or more nodes. A node represents a computer that hosts at least one Communication Driver. A server group comprises one or more server instances, and each server instance has its own hierarchy of configurable objects.

Communication Drivers that have been optimized for Operations Integration are contained in the **Operations Integration Supervisory Servers** folder. Only these Communication Drivers can have multiple server instances. Older, standalone Communication Drivers that have not been optimized are listed separately, and they cannot have multiple server instances.

Note: To enable multiple server instances, you must have a Professional (Tier 2) or Premier (Tier 3) license for Communication Drivers.

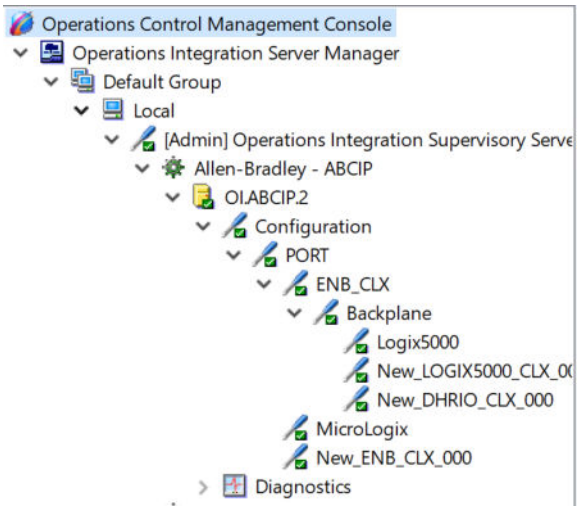
The following diagram depicts the OI Server Manager:



The **Operations Integration Supervisory Servers** folder, with its hierarchy of server groups and server instances, is shown only when you browse the **Local** node. All of the server instances on a remote computer are listed together, regardless of their server groups.

Example

If the ABCIP Communication Driver is installed on a local computer node, the OCMC shows the OI Server Manager with OI.ABCIP.1. When expanded, an active Communication Driver contains both **Configuration** and **Diagnostics** nodes. The green icon indicates that the Communication Driver is active. Inactive Communication Drivers, marked with a red icon, contain only the **Configuration** node.



Node Groups and Nodes

You can use the OI Server Manager to manage the Communication Drivers running on the local computer. Use node groups and nodes to organize the computers.

In the console tree, the **Default Group** node appears under the **Operations Integration Server Manager** node.

The **Default Group** lists all the computer nodes on the local domain, beginning with, by default, the local computer (**Local**), which is the computer on which the OI Server Manager is running.

Each computer node is itself a container if one or more Communication Drivers are running on that node. When you click different nodes in the console tree, the information in the details pane changes accordingly.

To add a node group (other than Default Group)

1. Right-click the **Operations Integration Server Manager** node, point to **New**, and then click **Node Group**.

   The **New Node Group** dialog box appears.

2. Type a name for the node group, and then click **OK**.

   The new node group is added under the **Operations Integration Server Manager** node.

To delete a node group (other than Default Group)

1. In the OI Server Manager, navigate to the node group.
2. Right-click the node group, and then click **Delete**.

   The **Delete** dialog box appears.

3. Click **Yes**.

   The node group is deleted.

To add a remote computer node

1. In the OI Server Manager, navigate to a node group.
2. Right-click on the node group, point to **New**, and then click **Node**.

   The **New Node** dialog box appears.
3. Click **Browse**.

   The **Browse Nodes** dialog box appears.
4. Select the domain name from the **Domain** list, and then select the remote computer name from the domain.

   The remote computer node is added.

To delete a remote computer node

1. In the OI Server Manager, navigate to the node.
2. Right-click the node, and then click **Delete**.

   The **Delete** dialog box appears.
3. Click **Yes**.

   The remote computer node is deleted.

Server Groups

A server group comprises one or more server instances that all use the same Communication Driver protocol.

When you install a specific Communication Driver on a computer, a server group and default server instance are automatically created. The server group has the same short name as the Communication Driver itself, along with the name of the manufacturer, for example, **Allen-Bradley - ABCIP**. There is a limit of one server group per Communication Driver per computer node.

Newer Communication Drivers that have been optimized for the Operations Integration are contained in the **Operations Integration Supervisory Servers** folder. Older, standalone Communication Drivers are listed separately below the **Operations Integration Supervisory Servers** folder.

Although a server group is automatically created when you install a Communication Driver, there are some cases in which you might need to create or remove a server group. For example, if you want to stop using a specific Communication Driver but not uninstall it, you can simply remove the server group from the computer node in the OI Server Manager. The Communication Driver will remain installed on the computer, and you can create the server group again if you need to.

Note: These procedures apply only to Communication Drivers that are contained in the **Operations Integration Supervisory Servers** folder in the OI Server Manager. Older, standalone OI Servers do not support creating or removing server groups. Instead, those OI Servers must be installed or uninstalled in their entirety.

Also, the **Operations Integration Supervisory Servers** folder, with its hierarchy of server groups and server instances, is shown only when you browse the **Local** node. All of the server instances on a remote computer are listed together, regardless of their server groups.

To create a server group

1. In the OCMC, navigate to the **Operations Integration Server Manager** folder.
2. Right-click **Operations Integration Supervisory Servers**, and then click **Create Server Group**.

   The **Create Server Group** dialog box is displayed.
3. In the **Server Group Name** list, select the Communication Driver that you want to use.
4. Click **OK**.

   A new server group for the selected Communication Driver is created in the OI Server Manager, and the server group contains a default server instance.

To remove a server group

1. In the OCMC, navigate to the **Operations Integration Server Manager** folder.
2. Expand the server group.
3. Make sure all of the server instances are deactivated. A server group cannot be removed while it contains active server instances. For more information, see Activating/Deactivating the Communication Driver.
4. Remove all of the server instances other than the default server instance. A server group cannot be removed while it contains multiple server instances. For more information, see Server Instances.
5. Right-click the server group, and then click **Remove Server Group**.
6. Click **Yes** to confirm.

   The server group and all of its instances are removed.

Server Instances

When you install a specific Communication Driver on a computer, a server group and default server instance are automatically created in the OI Server Manager. The **Operations Integration Supervisory Servers** folder, with its hierarchy of server groups and server instances, is shown only when you browse the **Local** node. Each server instance has its own configuration and diagnostics, can be activated and deactivated separately from all other server instances, and appears as a separate application to external clients. Newer Communication Drivers that have been optimized for Operations Integration support multiple server instances, which means that you can create additional server instances, and even clone existing server instances in those server groups. To enable multiple server instances, you must have a Professional (Tier 2) or Premier (Tier 3) license for Communication Driver.

The name of the default server instance is based on the short name of the Communication Driver itself. For example, for the Simulation Communication Driver, the default server instance is named **OI.SIM.1**. All server instance names follow this basic format. The middle part of a server instance name becomes the application name that external clients will use to access Communication Driver data. For example, if a server instance is named **OI.SIM_0001.1**, the corresponding application name will be **SIM_0001**.

Note: All of the server instances on a remote computer are listed together, regardless of their server groups.

For more information on creating, cloning, renaming, and deleting server instances, see Instantiating Data Sources.

License Status

You can view the icon of the active Communication Driver to determine its license status.

Icon	Description
	The Communication Driver in desktop mode and running with a valid license.
	The Communication Driver in desktop mode and running in the demo mode.
	The Communication Driver in desktop mode, and without a valid license.
	The Communication Driver is not licensed, and 32 free tag mode.
	The Communication Driver is running as a Windows service in active state, and with a valid license.
	The Communication Driver is running as a Windows service in active state, and in demo mode.
	The Communication Driver is running as a Windows service in active state, without a valid license, and not in a demo mode.
	The Communication Driver is not licensed, and 32 free tag in demo mode.

For more information about demo mode, see Demo Mode.

Accessing Operations Control Management Console (OCMC) using Different User Groups

You can access OCMC using a user login which is either in the Administrators, oiAdministrators user group, and/or other user groups. Only users who belong to the oiAdministrators or Administrators group can edit the configuration of the Communication Drivers. Users not belonging to these two groups can only view the Configuration and Diagnostics of the Communication Driver.

Administrators Users

When the OCMC is started with elevated privilege, the following operations can be performed:

- View the runtime diagnostics of Communication Driver
- Edit and view the configuration of Communication Driver
- Activate/Deactivate/Reset/Enable/Disable Communication Driver
- Create/Clone/Rename/Remove Communication Driver instances
- View the configuration of Communication Driver remotely on a different node

OI Administrator Users

OI Administrator users are users who are part of the oiAdministrators user group. These users can access the OCMC and have rights to perform all the actions, including:

- View the runtime diagnostics of Communication Driver
- Edit and View Communication Driver Configuration
- Activate/ Deactivate/Reset/Enable/Disable a Communication Driver
- Create Communication Driver/Clone/Remove Communication Driver instances

---

Note: The OI Administrator users cannot view the Configuration and Diagnostics of the remote Communication Drivers.

For the procedure on how to add a user to a user group, refer to the "Adding a member to a local group" section of the Microsoft documentation.

Standard Users

Standard users are the users who can access the OCMC and have only view access. Users part of user groups, other than oiAdministrators and Administrators, have permissions to:

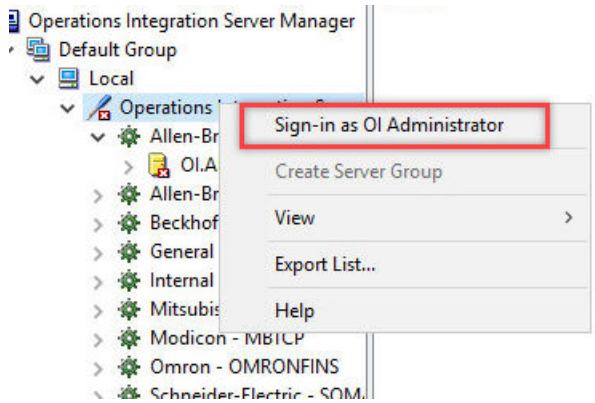
- View all installed Communication Drivers on a local node
- View Communication Driver diagnostics .

Note:
- Standard users cannot view the Communication Drivers installed on the remote node.
- Autobuild, MQTT Browser, and MQTT Publisher are only accessible to Administrator or OI Administrator users. To access them, you need to either start the OCMC with elevated privilege or sign-in as OI Administrator in the OCMC.

To sign in to OI Administrators Privileges in the OCMC:

If the OCMC is started without elevated privilege, or the user starting the OCMC does not belong to the oiAdministrators user group, you can still edit or update the configuration of Communication Drivers:

1. Under **Operations Integration Server Manager** > **Default Group** > **Local**, right-click the **Operations Integration Supervisory Servers**.
2. In the context menu that appears, select **Sign-in as OI Administrator**.



The **User Authentication** window appears.

3. In the **Domain** field, enter the user domain. Leave it blank if the user account is in the Windows work group.
4. In the **User Name** field, enter the user name of the user who is in the **oiAdministrators** user group.

   Note: If you start the OCMC with Windows elevated privilege, you are automatically in the OI Administrator mode.

5. In the **Password** field, enter the corresponding password of the user.
6. If you want to retain the privileges for a specific period, enter the required time in minutes in the **Keep sign-in** field. You can enter a value from 1 to 15 minutes.
7. Click **OK**.

Note: Once the specified time expires, the menu options will be disabled again.

When you access the OCMC as an unauthorized user, a  icon is displayed beside the **Operations Integration Supervisory Servers** node. When you enter the OI Administrator credentials, a  icon is displayed along with ["Admin"] beside the **Operations Integration Supervisory Servers** node.

Chapter 3

Configuring Your Communication Driver

Configuring Your Communication Driver on the Local Node

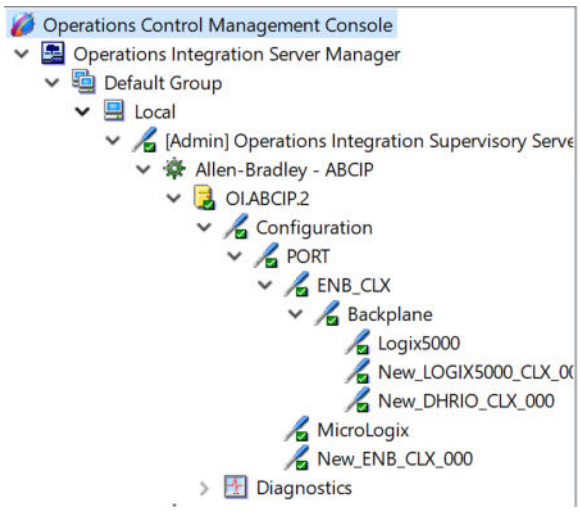
This section describes the different configuration procedure of the Communication Drivers that are installed on the local node.

About Configuring your Communication Driver

Each server instance in the OI Server Manager has both a **Configuration** node and a **Diagnostics** node (when the server is activated). Use the **Configuration** node to configure your Communication Driver for run-time.

The **Configuration** node has two functions:

1. The Global Parameters allow you to configure, and to adjust the Communication Driver's run-time performance. These parameters apply to all server instances in the server group. For more information, see Configuring Global Parameters.
2. The **Configuration** node has its own hierarchy of configurable objects. Each object represents a physical device, such as a channel, port, bridge, or PLC. Some Communication Drivers have two levels of objects in the hierarchy. For example, a parent object may represent a network channel, port, or bridge, while a child object may represent an individual device on that network.



Adding an Object

The first step in configuring the connection between the Communication Driver and a physical device is to add an object that represents the device.

To add an object

1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand **Default Group** (or the node group name), and then expand **Local**.
   b. Expand the Communication Driver, and then expand its **Configuration** node.
2. Right-click **Configuration**, and then click **Add Object**. The type of object varies by Communication Driver and node level. For more information, see the Communication Driver-specific documentation.

   The console tree shows a new object with its default name selected.
3. Type a new name, if desired.
4. Click **Save**.

   The object is added.

Renaming an Object

New objects in a Communication Driver configuration are given default names. You can rename an object to represent it clearly.

If you rename an object, the change takes effect immediately. However, you cannot rename an object if the Communication Driver is active and items are subscribed to the object. You can rename the object only after removing all subscribed items.

To rename an object

1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand **Default Group** (or the node group name), and then expand **Local**.

b. Expand the Communication Driver, and then expand its **Configuration** node.

2. Locate and right-click the object, and click **Rename**.
3. Type the new name, and then press **Enter**.

   The object is renamed.

Instantiating Data Sources

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.

Note: Creating multiple instances of Communication Driver or the Gateway Communication Driver requires a Professional level license.

To create new server instances in a server group (in addition to the group's default instance)

The creation of a new server instance uses the base instance as a template. All configuration parameters of the base template are copied to the new instance during the creation operation.

1. In the OI Server Manager, navigate to the server group:
   a. Expand the **Operations Integration Server Manager** node, expand **Default Group**, and then expand **Local**.
   b. Expand **Operations Integration Supervisory Servers**.
2. Right-click the server group, and then click **Create Server Instance**.

   The **Create OI Server Instance** dialog box is displayed.
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.

   Note: The server instance name — including the underscore and suffix — is limited to 16 characters. When you try to create or rename a server instance, names that exceed the limit will be rejected. These procedures apply only to Communication Driver that are contained in the **Operations Integration Supervisory Servers** folder in the OI Server Manager. Older, standalone OI Servers do not support multiple server instances.
5. Click **OK**.

   The new instances are created.

To clone an existing server instance

1. In the OI Server Manager, navigate to and expand the server group that contains the existing server instance.
2. Right-click the server instance, and then click **Clone Instance**.

   The Clone OI Server Instance dialog box is displayed.
3. In the **Custom Instance Name** box, type a custom name for the new instance. This is optional.

4. Click **OK**.

   The new instance is created.

To remove an existing server instance (other than a server group's default instance)

Make sure the server instance that you want to remove is deactivated. A server instance cannot be removed while it is active.

1. In the OI Server Manager, navigate to and expand the server group that contains the existing server instance.
2. Right-click the server instance, and then click **Remove Instance**.
3. Click **Yes** to confirm.

   The server instance is removed.

To rename an existing server instance (other than a server group's default instance)

Make sure the server instance that you want to rename is deactivated. A server instance cannot be renamed while it is active.

1. In the OI Server Manager, navigate to and expand the server group that contains the existing server instance.
2. Right-click the server instance, and then click **Rename Instance**.

   The Rename OI Server Instance dialog box is displayed.
3. In the **Custom Instance Name** box, type the new value that should be appended to the base name.
4. Click **OK**.

   The server instance is renamed.

Enabling or Disabling an Object

Similar to how the Communication Driver itself can be activated or deactivated, objects in a Communication Driver configuration can be enabled and disabled.

Rules for enabling and disabling objects:

- Objects in a Communication Driver configuration can be enabled and disabled when the Communication Driver is active or inactive. The default state of an object is enabled.
- You cannot disable the Communication Driver and **Configuration** nodes, or the nodes between them. However, objects that represent busses, ports, or bridges can be disabled.
- System items remain valid for disabled objects. All other items have "Out-of Service" quality updated to the client. No messages are sent to the device represented by a disabled object. Write transactions for a disabled object fail immediately with a bad HRESULT.
- Disabling an object also disables all of its child objects. The tree pane does not show child objects as disabled. When you enable an object, all of its child objects are enabled, except those objects that you explicitly disabled. Explicitly disabled child objects remain disabled until you enable them.

To enable or disable an object in Communication Driver configuration

1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.

b. Expand the Communication Driver, and then expand its **Configuration** node.

2. To enable/disable an object:
   - To enable: Right-click the object and then click **Enable**.
   - To disable: Right-click the object and then click **Disable**.

   The object's icon shows the change in the object's state: an enabled object shows a check mark (✓), and a disabled object shows a cross mark (X).

Configuring an Object's Parameters

Each object in a Communication Driver configuration has parameters that you can configure.

To configure an object's parameters

1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, and then expand its **Configuration** node.
2. Select the object in the Communication Driver configuration. The object's parameters are displayed in the details pane on the right.
3. Configure the object's parameters as needed.

   The parameters vary by Communication Driver, and if the Communication Driver has more than one level of objects, the parameters also vary by level or object type. For example, a channel object may have communication settings that you need to configure in order to establish communication between the Communication Driver and the network, while a device object may have address settings that you need to configure in order to identify a device on the network. For more information, see the documentation for your specific Communication Driver.
4. Click **Save**.

   The object's parameters are configured and saved.

Advanced Settings

Many of the Communication Driver included in the **Operations Integration Supervisory Servers** folder have a common set of advanced communication settings. These settings are in addition to the object parameters that are described in the documentation for your specific Communication Driver.

To determine if your Communication Driver has these settings, right-click **Configuration**. If your Communication Driver has these settings, you will see the **Add ChannelSelector Object** in the context menu options.

You might need to change these settings if the Communication Driver behaves unexpectedly during run time, but the default settings should work for most network configurations.

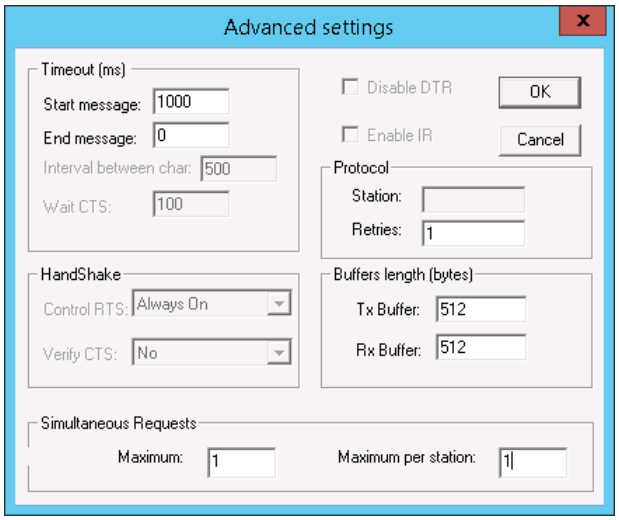
To change the advanced settings for a ChannelSelector object

1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.

b. Expand the Communication Driver.

2. Right-click the **Configuration** node and select the **Add ChannelSelector Connection** option. The object's parameters are displayed in the details pane on the right.
3. In the right pane, click **Advanced**.

   The **Advanced Settings** dialog appears.



4. Change the settings as needed.

Group	Setting	Description
Timeout	**Start message**	The timeout (in milliseconds) to receive the start of a message.
	End message	The timeout (in milliseconds) to receive the end of a message.
	Interval between char	The interval (in milliseconds) between characters in a message.
	Wait CTS	The timeout (in milliseconds) for the Clear to Send wait.
Handshake	**Control RTS**	Specifies whether to use the Request to Send control.
	Verify CTS	Specifies whether to use the Clear to Send verification type.

Simultaneous Requests	**Maximum**	The maximum number of requests (up to 32) that can be sent at the same time to all devices in the channel. **Note:** There is a limit of 100 total simultaneous requests across all channels in a server instance, which means you can have three channels at 32 each, four channels at 24 each, five channels at 20 each, and so on up to 100 channels at 1 each, or any combination thereof.
	Maximum per station	The maximum number of requests that can be sent at the same time to a single device in the channel. The limit varies, so for more information, see the documentation for the specific Communication Driver.
Protocol	**Station**	The station number or ID of the channel, according to the device protocol being used. Some master/slave protocols consider the Communication Driver to be another slave device and therefore require it to have its own station ID. For more information, see the documentation for the specific Communication Driver.
	Retries	The number of times that the Communication Driver will retry the same command before generating a communication error.
Buffers Length	**Tx Buffer**	The transmission buffer length (in bytes).
	Rx Buffer	The reception buffer length (in bytes).
Disable DTR		When disabled, no DTR signal is sent before starting a communication.
Enable IR		Available only on Windows Embedded target systems. Enables use of Infrared interface (COM2 port) rather than a standard serial port to communicate with devices.

5. Click **OK** to close the **Advanced Settings** dialog box.
6. Click **Save**.

   The advanced settings are saved.

Archiving Configuration Sets

A configuration set includes the global parameters; each channel and its parameters; and each device and its parameters, device groups, and device items. It lets you manage the settings of different Communication Driver configurations.

The Communication Driver contains a default configuration set. You cannot delete the default configuration set. However, you can create multiple configuration sets and switch between them. Archiving, clearing, and switching configuration sets can only be done when the Communication Driver is deactivated. Before you create a configuration set, verify that you have saved any changes you made to the global parameters. If you change a

parameter and then immediately create a configuration set, the original parameter value is saved as part of the configuration set, not the changed value.

You cannot use COM1 through COM9 as the name of a configuration set, as these names are reserved for system functions.

To archive a configuration set

1. In the OI Server Manager, navigate to the configuration node.
   a. Expand **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver.
2. Right-click **Configuration**, and then click **Archive Configuration Set**.
3. In the **Archive Configuration Set** dialog box, enter a name for **Configuration Set Name**, and then click **Archive**.

   All the current configuration values are saved to the set.

After you archive at least one configuration set, you can select it for use.

To select a configuration set

1. In the OI Server Manager, navigate to the configuration node.
2. Right-click **Configuration**, point to **Use Another Configuration Set**, and then click the configuration set that you want to use.

To change the parameter values saved in a configuration set, make sure the desired configuration set is shown, and then follow this procedure.

To change the parameter values in a configuration set

1. In the OI Server Manager, navigate to the configuration node.
2. Change the configuration parameters as needed.
3. Click the **Save** icon.

Clearing a configuration set returns the parameters to their default values.

To clear a configuration set

1. In the OI Server Manager, navigate to the configuration node.
2. Right-click **Configuration**, and then click **Clear Configuration Set**.
3. Read the warning message, and then click **Yes**.

   The parameters are set to the default values.

To delete a configuration set

1. In the OI Server Manager, navigate to the configuration node.
2. Right-click **Configuration**, point to **Delete Configuration Set**, and then click the configuration set that you want to delete.
3. Read the warning message, and then click **Yes**.

   The configuration set is deleted.

Resetting an Object

You can reset an object in a Communication Driver configuration in order to make modified configuration parameters take effect without deactivating and then reactivating the Server. Resetting an object also resets all of its child objects.

The **OI Server** and **Configuration** nodes, and the nodes between them, cannot be reset.

To reset an object

1. In the OI Server Manager, navigate to the **Configuration** node.
    a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
    b. Expand the Communication Driver, and then expand its **Configuration** node.
2. Right-click the object and then click **Reset**.

    The modified configurations of the object and all its child nodes take effect.

Deleting an Object

You can delete an object in a Communication Driver configuration when you no longer need it. You cannot delete an object if the Communication Driver is active and items are subscribed to the object. You can delete the object only after removing all subscribed items.

To delete an object in Communication Driver configuration

1. In the OI Server Manager, navigate to the **Configuration** node.
    a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
    b. Expand the Communication Driver, and then expand its **Configuration** node.
2. Right-click the object, and then click **Delete**.

    The object is deleted.

Configuring Your Communication Driver on the Remote Node

If you access the Operations Control Management Console (OCMC) as an Administrator, you can view the remote Communication Drivers and its Diagnostics node. However, you are not allowed to Activate, Deactivate, or make any configuration changes on the remote Communication Drivers for security reasons.

Chapter 4

Configuring Global Parameters

Each Communication Driver has a unique set of parameters that you can configure for your particular environment. For descriptions of those configuration parameters, see the Communication Driver-specific documentation.

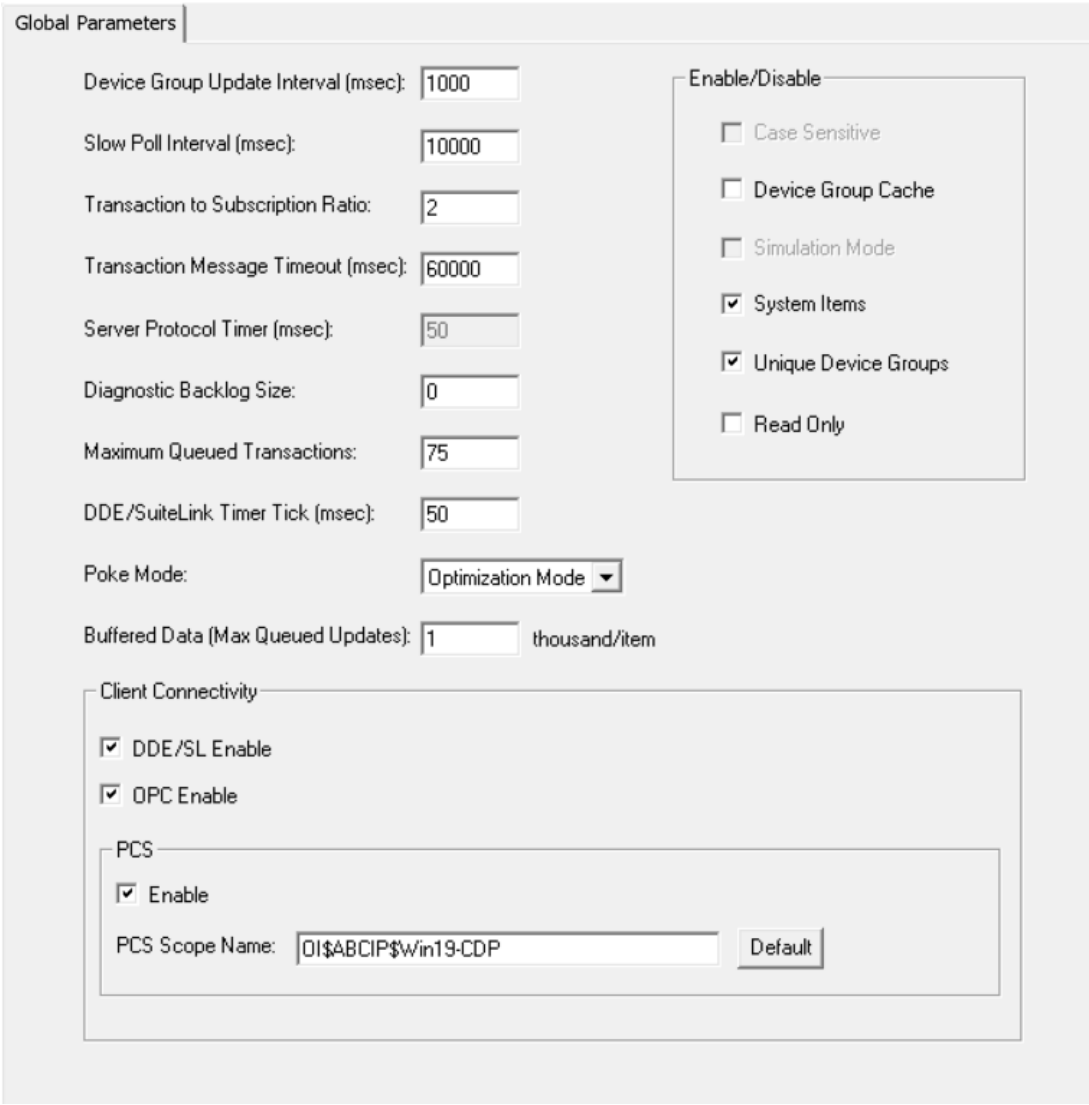
The OI Server Manager also allows you to configure a set of common, or global parameters for each Communication Driver.

Accessing the Global Parameters

To access the global parameters

1. In the OI Server Manager, navigate to the Configuration node.
   a. Expand the **Operations Integration Server Manager** node, expand the **Default Group** group, and then expand **Local**.
   b. Expand the Communication Driver, and then click **Configuration**.

   The **Global Parameters** tab appears in the details pane.



Note: The **Device Group Cache** option is for future use only and should not be selected. Some Communication Drivers, in fact, may disable this option.

2. Click **Save** to write the current data to the configuration set.

   If the Communication Driver is enabled and you have changed a global parameter, the new value takes effect when you save the configuration. For more information, see Hot Configuration.

   When you click **Save**, leading zeros are truncated from your entries in numerical fields. The behavior of the rest of the Communication Driver's configuration hierarchy depends on the refresh mechanism of the server-specific programming logic.

Configuring Intervals for Device Group Updates and Slow Polling

The device group update interval parameter specifies the update interval of the <Default> device group. If it is not specified, all unnamed device groups have an update interval of 1,000 milliseconds.

The slow poll interval parameter controls the interval that the Communication Driver polls the field device after it goes into slow poll mode. This occurs when certain connectivity problems occur, such as an unplugged PLC. When the Communication Driver again achieves connectivity with the field device, it returns to normal operation and is governed by the setting of the device group update interval.

To configure the update and poll intervals

1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, and then click **Configuration**.

   The **Global Parameters** tab appears in the details pane.
2. In the **Device Group Update Interval** box, type the value in milliseconds. The default value is 1,000.
3. In the **Slow Poll Interval** box, type an integer representing the interval, in milliseconds, between polls per transaction item. The default value is 10,000.
4. Click **Save**.

   The configurations of the update and poll intervals are saved.

Configuring Transaction and Subscription Settings

You can set the following transaction and subscription settings:

Transaction to Subscription Ratio

This parameter controls the ratio of transaction to subscription messages in a Communication Driver at the time when several transaction messages are pending. This value is the maximum number of transaction messages sent before sending a subscription message. The second half of the ratio is always 1. The default ratio is 2 transaction messages to 1 subscription message (2:1).

Transactions have a higher priority than subscriptions. If only one transaction message is pending, that message is sent first no matter how many subscription messages are due to be sent. If one or more transactions generate several transaction messages, this ratio applies to the number of generated transaction messages. This ratio ensures that a certain amount of subscription activity is guaranteed even in a transaction overloaded state. The value sets the number of allowed transaction messages to be arbitrated on the protocol before evaluation of due subscription messages is enforced. If subscription messages are due, one of them is sent before sending transaction messages again.

Transaction Message Timeout

This parameter sets the timeout for transactions (read/write/refresh/property) per message. The value is an integer representing the timeout in milliseconds per transaction message. Default is 60000 (corresponds to the default ValidDataTimeout in legacy I/O servers). A properly operating Communication Driver should never encounter this timeout because a transaction will always be completed by the protocol engine (successful or not). This timeout only prevents a client from hanging in a transaction forever if, for some reason, the transaction messages are never scheduled until completion. The timeout is specified on an individual message level. It is not the maximum amount of time a transaction can take. It is the maximum amount of time between message updates within a transaction. The timeout has to be set in such a way that under no circumstances can data acquisition on an individual message take longer than this timeout. The time it might take for a transaction containing several messages theoretically can be multiples of this timeout.

Diagnostic Backlog Size

Controls the maximum number of transactions displayed in the Transactions diagnostic node. The diagnostic backlog size parameter does not affect the number of transactions allowed by the Communication Driver. It strictly affects the number of transactions shown in the Transactions node.

Configuring Protocol Timers

You can use the server protocol timer to control the interval between protocol activities for timer-driven Communication Drivers. Your specific Communication Driver documentation will describe whether it uses this value.

The server protocol parameter corresponds to the default ProtocolTimerTick in legacy I/O Servers. This value is ignored by Communication Drivers that use an entirely event-driven protocol engine.

The DDE/SuiteLink timer controls the time between processing client requests though the DDE or SuiteLink protocols. If you set this value to a smaller number, it increases the responsiveness, but may reduce the throughput. The default value works well for most installations.

To configure timers

1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, and then click **Configuration**.

   The **Global Parameters** tab appears in the details pane.
2. In the **Server Protocol Timer** box, type the value, in milliseconds, between protocol activities if the Communication Driver is timer-driven. The default value is 50.
3. In the **DDE/SL Timer Tick** box, type the value in milliseconds. The minimum value of this parameter is 20, and the maximum value is 6,000. The default value is 50.
4. Click **Save**.

Configuring the Poke Mode

You can use the poke mode to control how the Communication Driver optimizes and folds pokes within a transaction. Select from the following three modes:

Control Mode

Preserves the poke order without folding. This mode is typically used by batch and control applications that depend on the order of the pokes, and on the processing of every item poked.

Transition Mode

Preserves the poke order with minimum folding by keeping the first, second, and last poke values of an item. This mode is used by batch and control applications that depend on the order of pokes but do not process every item poked.

Optimization Mode

Does not preserve the poke order and has maximum folding by only poking the last value of an item. This mode is typically used by HMI applications.

To configure the poke mode

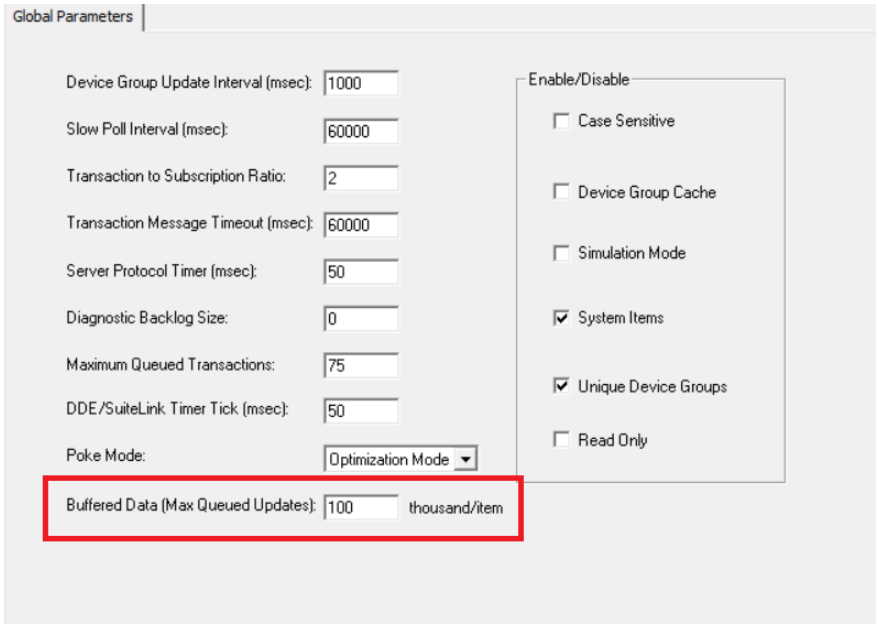
1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, and then click **Configuration**.

   The **Global Parameters** tab appears in the details pane.
2. In the **Poke Mode** list, select the mode. The default mode is Communication Driver dependent.
3. Click **Save**.

   The configuration of the poke mode is saved.

Configuring Buffered Data (Maximum Queued Updates)

Buffered data is data captured and stored locally on a remote or a local device for later transfer to a supervisory system for processing, analysis, and long-term storage. The buffered data feature enables efficient accumulation and propagation of VTQ (Value, Time, and Quality) data updates, without folding and data loss. Communication Drivers, such as MQTT, can acquire buffered data from a remote data source.



The **Buffered Data (Maximum Queued Updates)** sets the maximum number of buffered values that the Communication Driver can accumulate without folding. The minimum value of this parameter is 1, and the maximum value is 1,000. As the unit of Buffered Data (Maximum Queued Updates) is in thousands, if the value of the **Buffered Data (Maximum Queued Updates)** is 1, then up to 1000 data updates can be buffered per tag. If the value of the **Buffered Data (Maximum Queued Updates)** is 1000, then up to 1,000,000 data updates can be buffered per tag. When the queued data updates exceed this limit, the newest value replaces the oldest value.

For example:

- **Example1:** Consider you have set the value of **Buffered Data (Maximum Queued Updates)** to 1, which is 1000 data updates. If there is a buffered data of 1100 updates that are received by the Communication Driver, then the last 1000 data updates of buffered data will be sent to System Platform for processing and historization, the first 100 data updates of the buffered data will be lost.
- **Example2:** Consider you have set the value of **Buffered Data (Maximum Queued Updates)** to 2, which is 2000 data updates. If there is a buffered data of 1100 updates that are received by the Communication Driver, then all the 1100 updates of buffered data will be forwarded to System Platform for processing and historization. There will be no data loss because 1100 updates are below the maximum queued updates value.

For more information on buffered data, refer to the Working with Buffered Data section in the AVEVA™ Application Server User Guide.

Note: The default value of the **Buffered Data (Maximum Queued Updates)** parameter is 1 (that is, 1000 maximum buffered values) for most Communication Drivers. Some individual Communication Drivers impose a different value that has been documented in the respective section of the Communication Driver.

Configuring Client Connectivity

You can use the Client Connectivity section to select which protocol(s) a client needs to use to communicate with a Communication Driver on a per instance basis. The DDE/SL, OPC, and PCS protocols are selected by default when a Communication Driver is not running.

To configure Client Connectivity

1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, and then click **Configuration**.

   The **Global Parameters** tab appears in the details pane.
2. In the **Client Connectivity** section, uncheck the protocol(s) that you wish to disable. A client cannot connect to the Communication Driver instance using the disabled protocol(s).

   Note: The PCS section is disabled if the PCS Plug-In is not installed. For more information on installing the PCS Plug-In, refer to PCS in the 'Supported Client Protocols' section.
3. For PCS IData clients, there is an option to modify the **PCS Scope Name** field to any scope name of your choice.

   The scope name must be unique across all computer nodes in a Galaxy. The default scope name can be modified using the text field.

   A general syntax to configure the PCS Scope Name is:

   `OI$<OIServerInstanceName>$<ComputerName>`

   - **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 Communication Driver as viewed in the OCMC.
- **ComputerName** is the name of the machine on which the Communication Driver is running.

Example of PCS Scope Name:

Note: The scope name must not contain a colon :

It is recommended to follow this syntax to ensure that the scope name of a Communication Driver instance is unique within a Galaxy, though not compulsory.The PCS Scope Name can be customized to any preferred name. For example, if the computer running the Communication Drivers is the only computer in a Galaxy, and you want to shorten the PCS Scope Name used in item names, then PCS Scope Names such as PLC1, PLC2, ABCIP1, ABCIP2, or SIDIR1 could be configured for the Communication Driver instance.

Note: If a duplicate PCS Scope Name is defined in a Galaxy, then PCS clients may fail to connect to the Communication Driver with the duplicate name.

You can restore default PCS scope name by clicking **Default**. Refer to "Accessing Data Using PCS" for more details on the default PCS scope name syntax.

4. Click **Save**.

Configuring Case Sensitivity for Item IDs and Device Group Names

You can control how the Communication Driver scans items and device group names with respect to upper and lower case. The default is set to case insensitive, which is recommended if you are working with legacy applications.

Case sensitivity applies to item names (fully qualified item ID, including hierarchy names) and device group names.

To configure case sensitivity

1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, and then click **Configuration**.

   The **Global Parameters** tab appears in the details pane.
2. In the **Enable/Disable** area, select the **Case Sensitive** check box. By default, this check box is clear.
3. Click **Save**.

   Case sensitivity is configured for all item IDs and device group names.

Enforcing Uniqueness for Device Group Names

The option **Unique Device Groups** allows you to control the uniqueness of device group names across all device nodes.

- If you enable uniqueness checking, all device groups on a server node are considered one list across the namespace. And because device group names are not case sensitive, a specific name is allowed only one time, regardless of case.
- If you disable uniqueness checking, you can have the same device group name in different device nodes, but not in the same node (inside device groups, names are not case sensitive). This means that device group names are unique on one hierarchy level only.

By default, **Unique Device Groups** is selected. Checking for uniqueness is required when working with DDE/SuiteLink installations.

To enforce uniqueness for device group names

1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, and then click **Configuration**.

   The **Global Parameters** tab appears in the details pane.
2. In the **Enable/Disable** area, the **Unique Device Group** check box is selected by default as it is required for working with DDE/SuiteLink legacy installations. If you clear this check box, device group names must be unique on one hierarchy level only.
3. Click **Save**.

Enabling the Communication Driver to Run in Simulation Mode

Some Communication Drivers can simulate communication with a field device. Availability of a simulation mode is Communication Driver-specific. If the Communication Driver does not support this mode, the element is not available.

To enable simulation mode

1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, and then click **Configuration**.

   The **Global Parameters** tab appears in the details pane.
2. In the **Enable/Disable** area, select the **Simulation Mode** check box. You can clear this check box to set the Communication Driver in normal operation mode.
3. Click **Save**.

Showing or Hiding System Items

You can specify whether system items appear in the browser interface. This option also confirms recognition of system items as valid ItemIDs by the data acquisition interfaces.

To show or hide system items

1. In the OI Server Manager, navigate to the **Configuration** node.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, and then click **Configuration**.

   The **Global Parameters** tab appears in the details pane.
2. In the **Enable/Disable** area, select the **System Items** check box. By default, this option is selected for system items. If you clear this option, no system items appear in the Communication Driver namespace.
3. Click **Save**.

Configuring Read Only

You can set the Communication Driver to be read-only so that the Communication Driver will reject all poking (write) operations from its client.

Note: The Read Only option in the editor is disabled when the server is running. You cannot configure a server as Read-Only in runtime.

Chapter 5

Managing Device Groups

About Device Groups

Device groups are labels used by client applications when accessing the Communication Driver.

The Device Group Update Interval determines how often the Communication Driver polls the device and sends data to the client application. If you configure multiple device groups with different update intervals, the client application can receive data at various intervals.

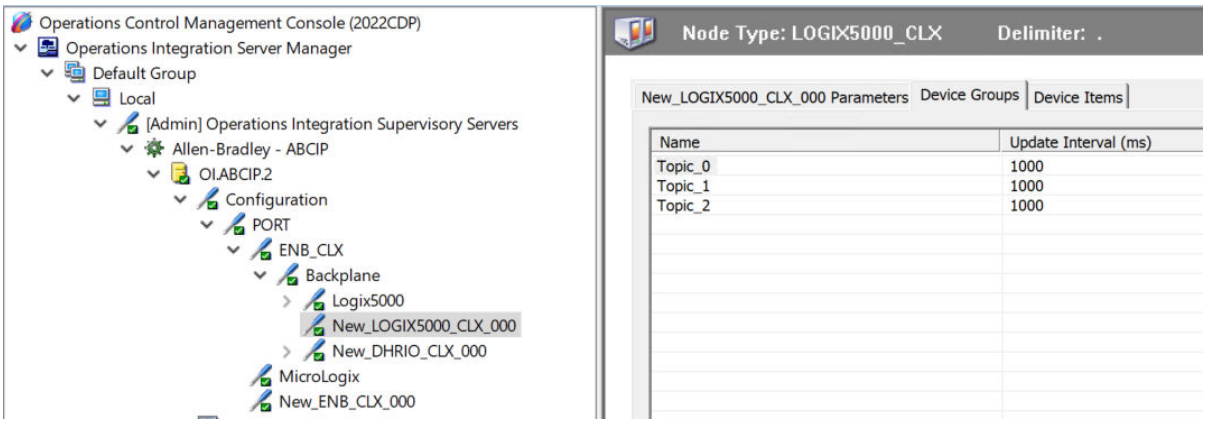
Small update intervals mean fast turnaround for data changes and a high overhead because a large amount of data is moving. Large update intervals mean slow turn around for data changes and a low overhead because not as much data is being passed to the client application.

For DDE/SuiteLink clients, the device group is the same as the DDE/SuiteLink topic. DDE/SuiteLink clients require that at least one device group be created for each device.

For OPC clients, the device group equals the OPC access path. The Communication Driver has a default device group for each device, and this device group cannot be deleted. If you are using OPC client applications, creating a device group is optional.

Viewing Common Device Groups

Under the **Configuration** node in the console tree, most Communication Driver specific items have the **Device Groups** tab as a common component in the details pane.



Note: Device group names are not case sensitive. Therefore, ensure that you do not use two device groups with the same name with different case. Each device group name must be unique, including case.

Adding a Device Group

Device groups allow you to specify an update interval for a set of device items. The device group does not contain any device items. The linkage is made when the client makes a request.

To add a device group

1. In the OI Server Manager, navigate to the device.
   a. Expand **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, expand **Configuration**, and then expand the channel, port, or bridge.
2. Select the device. The device's parameters are displayed in the details pane on the right.
3. In the details pane, click the **Device Groups** tab.
4. Right-click anywhere in the table, and then click **Add**. A device group is added with a default name and update interval.
5. Type a new name, if desired.
6. Click the **Save** icon.

   The new device group is added.

Renaming a Device Group

Changing the name of an existing device group requires that any client queries using the device group must be changed. Requests for data accepted by the Communication Driver before the change are not affected.

To change a device group name

1. In the OI Server Manager, navigate to the device.
   a. Expand **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, expand **Configuration**, and then expand the channel.
2. Select the device. The device's parameters are displayed in the details pane on the right.

3. In the details pane, click the **Device Groups** tab.
4. Right-click the name to be changed, and then click **Rename**. The current name is selected.
5. Type the new name.
6. Click **Save**.

   The device group name changes.

Modifying the Update Interval

The device group data consists of one item, the update interval. The update interval specifies the time period in milliseconds between Communication Driver reads of the device memory.

You can specify a value between 0 and 604800000 ms (that is, 168 hours). The default is 1000 (that is, 1 second).

To modify the update interval

1. In the OI Server Manager, navigate to the device.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, expand **Configuration**, and then expand the channel.
2. Select the device. The device's parameters are displayed in the details pane on the right.
3. In the details pane, click the **Device Groups** tab.
4. Right-click the interval, and then click **Modify Update Interval**. The current interval is selected.
5. Type the new interval.
6. Click **Save**.

   The update interval is modified.

Deleting a Device Group

When you delete a device group, the quality of items being accessed using the device group changes to BAD. The Communication Driver rejects new requests for data using the device group.

To delete a device group

1. In the OI Server Manager, navigate to the device.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, expand **Configuration**, and then expand the channel.
2. Select the device. The device's parameters are displayed in the details pane on the right.
3. In the details pane, click the **Device Groups** tab.
4. Right-click the device group to be deleted, and then click **Delete**.
5. Read the warning, and then click **Yes**.
6. Click **Save**.

The device group is deleted.

AVEVA

Chapter 6

Managing Device Items

About Device Items

Device Items are used to access data in the Communication Driver and the connected devices. The device item name is an alternative name for the item reference. It is an “alias” or a label for the data in the device. You can use this label instead of the item reference when you create the client application. Defining device items provides a more user-friendly way to name data in the device. Defining device items is optional.

Device items consist of two pieces: a name and an item reference. Once defined, you can access it in the client program either through item name or the item reference. The item reference identifies data in the device. The item reference is a PLC memory reference. Each device’s memory reference can have a different format. For more information, see your Communication Driver-specific documentation.

The actual item reference can be entered as the device item name. In this case, the item reference value can be left empty.

To provide diagnostic and operational information, there are several system items that do not access data in a device. They are grouped by function:

- Global system items
- Device-group-specific system items
- Device-specific system items

For more information about system items, see Standard System Items.

The defined device item names show up as OPC browsable items. While the Communication Driver is active, you can add and make changes to the device items. Changes take effect immediately, and the new items are immediately available to client applications. OPC clients that are already connected to the item are not affected until they release and re-acquire the item.

For detailed formats for specifying item references and how to subscribe to data items, see your Communication Driver-specific documentation.

Adding a Device Item

The device item name is an alias for the item reference. Device item names can be 256 characters long. Long names may be more explanatory, but your client application may have limited screen space.

To add a device item

1. In the OI Server Manager, navigate to the device.
   a. Expand **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, expand **Configuration**, and then expand the channel, port, or bridge.
2. Select the device. The device's parameters are displayed in the details pane on the right.
3. In the details pane, click the **Device Items** tab.
4. Right-click the column field, and then click **Add**. A device item is added with a default name.
5. Type a new name, if desired.
6. Click **Save**.

   The device item is added.

Renaming a Device Item

Changing a device item name affects new client requests for data. Requests for data already accepted by the Communication Driver are not affected.

To change a device item name

1. In the OI Server Manager, navigate to the device.
   a. Expand the **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, expand **Configuration**, and then expand the channel, port, or bridge.
2. Select the device. The device's parameters are displayed in the details pane on the right.
3. In the details pane, click the **Device Items** tab.
4. Right-click the name to be changed, and then click **Rename**. The current name is selected.
5. Type the new name.
6. Click **Save**.

   The device item is renamed.

Setting the Item Reference

You must know which memory locations you need and the memory location attributes before entering item references in the Communication Driver.

To set an item reference

1. In the OI Server Manager, navigate to the device.
   a. Expand **Operations Integration Server Manager** node**,** expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, expand **Configuration**, and then expand the channel, port, or bridge.
2. Select the device. The device's parameters are displayed in the details pane on the right.
3. In the details pane, click the **Device Items** tab.
4. Right-click the **Item Reference** field to be set, and then click **Rename**.
5. Type the item reference, and then press **Enter**.

   For a complete description of the item reference syntax and options, see the Communication Driver-specific documentation.
6. Click **Save**.

   The item reference saved.

Exporting and Importing CSV Files

To help you manage item references (tags) and device item names outside the OI Server Manager, the Communication Driver supports importing and exporting device item data in a comma-separated values (CSV) file. The CSV functions are only available when a device items tab is selected.

Each row in the CSV file represents an item, and each row contains two elements. The first element must be the item name; the second element must be the item reference string. The item name can be the same as the reference string or it can be an alias. If either the item name or reference string element contains a comma, use quotation marks around the element to avoid a collision with the comma delimiters.

Item names are not case-sensitive.

To export or import a device item list

1. In the OI Server Manager, navigate to the device.
   a. Expand **Operations Integration Server Manager**, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, expand **Configuration**, and then expand the channel, port, or bridge.
2. Select the device. The device's parameters are displayed in the details pane on the right.
3. In the details pane, click the **Device Items** tab.
4. To export a device item list:
   a. Right-click anywhere in the table, and then click **Export.**
   b. In the Save As dialog box, type a file name, select a directory, and then click **Save**.

   Items list in the **Device Items** tab are exported to the CSV file.
5. To import a device item list:
   a. Right-click anywhere in the table, and then click **Import**.
   b. In the Open dialog box, find the file containing the items to be imported, and then click **Open**.

Items contained in the file are now listed on the **Device Items** tab.

Deleting a Device Item

Deleting a device item name affects new client requests for data. Requests for data already accepted by the Communication Driver are not affected.

To delete a device item name

1. In the Communication Driver, navigate to the device.
   a. Expand **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, expand **Configuration**, and then expand the channel, port, or bridge.
2. Select the device. The device's parameters are displayed in the details pane on the right.
3. In the details pane, click the **Device Items** tab.
4. Right-click the item to be deleted, and then click **Delete**.
5. Read the warning, and then click **Yes**.
6. Click **Save**.

   The device item is deleted.

Clearing All Device Items

You can delete all device items for a device.

To clear all device items

1. In the OI Server Manager, navigate to the device.
   a. Expand **Operations Integration Server Manager** node, expand the node group, and then expand **Local**.
   b. Expand the Communication Driver, expand **Configuration**, and then expand the channel, port, or bridge.
2. Select the device.
3. In the right pane, click the **Device Items** tab.
4. Right-click the columns field, and then click **Clear All**.
5. Read the warning, and then click **Yes**.
6. Click **Save**.

   All device items are deleted.

Configuring Device Redundancy

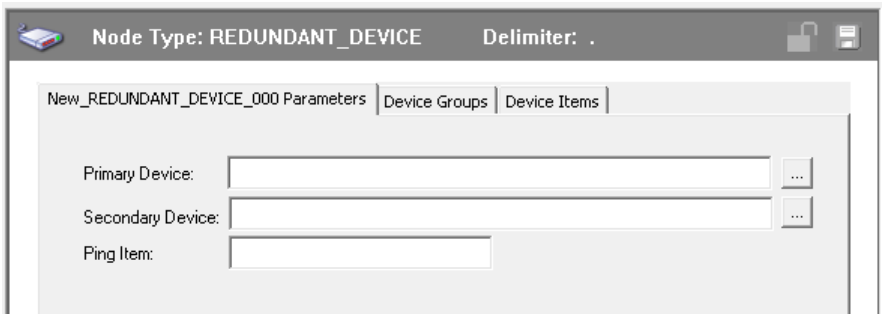
The OI Server Manager provides the ability to assign redundant device for fail-over protection in the event of device failure. Two devices must be configured in the same Communication Driver having identical item syntax.

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

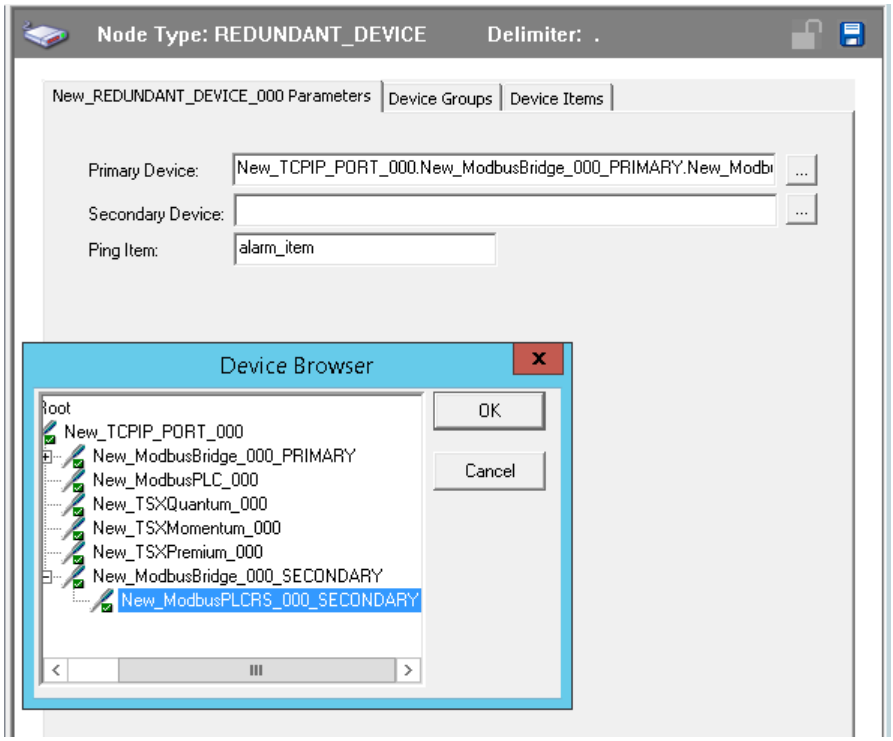
Note: At present, you cannot configure device redundancy using the PCS scope name. It is recommended to use the SuiteLink protocol while setting up a redundant connection between the Communication Driver and HMI/ SCADA client application.

To setup up a REDUNDANT_DEVICE from the configuration branch:

1. Set-up a primary device and hierarchy in the OI Server Manager in the OCMC.
2. Create at least one device item that can be shared between the primary and secondary devices to determine device status.
3. Set up a secondary device on the same Communication Driver. Once again, create an identical device item within the secondary device so that device status can be determined.
4. Right-click on **Configuration**, and select **Add REDUNDANT_DEVICE Connection**. An object called New_REDUNDANT_DEVICE_000 is created.



5. Rename the newly created object as appropriate. The New_REDUNDANT_DEVICE_000 configuration view is displayed in the Configuration branch of the hierarchy.
6. Enter or use the device browser to select the primary and secondary devices. Save the hierarchy node configuration by clicking on the save icon.



Note: The primary device and secondary device must be a PLC object, and not the Port or Bridge objects.

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

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

Run-time Behavior

The Communication Driver starts with the active device. The driver switches to the standby device when the active device fails to communicate. The value of the SYSStatus determines the communication failure.

Note: The value of the SYSStatus of the standby device must be TRUE in order to switch over to the standby device. Otherwise, there will not be any failover.

When SYSStatus shows a FALSE value at both active and standby devices, the driver considers a complete communication failure and mark all the items subscribed to the redundancy device hierarchy with the current time and the appropriate OPC quality. The driver activates the slow-poll mechanism to retry the communication to both devices until either one of the Ping Items returns to a good quality and update its SYSStatus item to TRUE.

When the driver switches to the standby device, the standby device becomes active and the originally active device becomes the standby.

When the active device becomes the standby device the Ping Item is not deleted from that the standby device. This ensures the standby is able to recover the communication again.

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

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

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

Chapter 7

Secure SuiteLink Connection

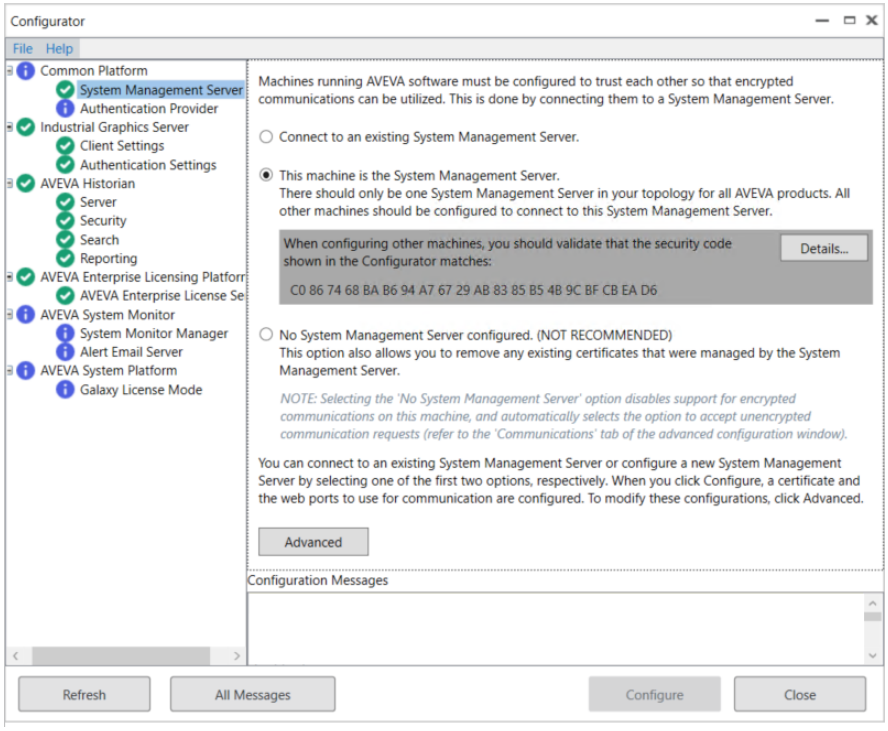
To ensure a high level of confidentiality and privacy, SuiteLink communication between a server and a client is encrypted by default.

Since the introduction of secure SuiteLink (V3), SuiteLink has been providing the fallback so that a SuiteLink server can accept incoming connections in a secure (V3, encrypted) or an unsecure (V2, unencrypted) mode together. From the System Platform 2023 release, by default, the fallback mode to accept V2 connection is disabled.

Note: Using a Secure SuiteLink connection is highly recommended. A warning message will be generated in the logger if the SuiteLink connection is unsecure. As of System Platform 2023, the SuiteLink connection is by default secure with TLS encryption.

If you want to enable the fallback mode to accept the both unsecure and secure connection, you have to make the configuration changes in the System Management Server as below:

1. In the Configurator, expand **Common Platform**, and select **System Management Server**.
2. Select the **This machine is the System Management Server** option.

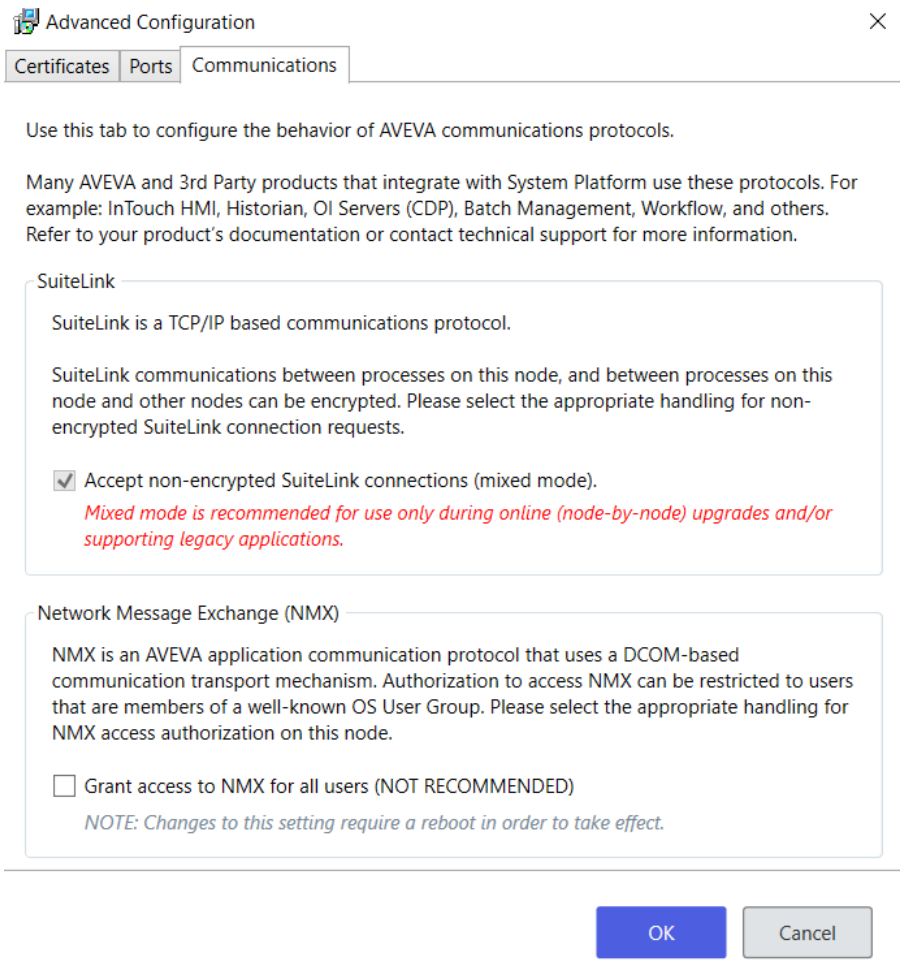


3. Click **Advanced**.
4. In the **Advanced Configuration** window that appears, click the **Communications** tab.
5. Select the **Accept non-encrypted SuiteLink connections (mixed mode)** check box and click **OK**. This enables the mixed mode. That is , both encrypted and non-encrypted connections are accepted. If you upgrade from the previous version, then this option is selected by default.
   - If **Accept non-encrypted SuiteLink connections (mixed mode)** selected: A client first attempts to establish a secure connection to the SuiteLink server. If a secure connection cannot be established, the client then establishes the connection to the server without security. This is how it was behaving earlier, prior to the System Platform 2023 release.
   - If **Accept non-encrypted SuiteLink connections (mixed mode)** unselected: Client connections to the SuiteLink server are only successful if the connection is secured, that is both nodes must be configured to use the System Management Server. This option ensures that the connection between the SuiteLink Server and SuiteLink clients is always secure (encrypted). If a secure connection is not available, the connection will not be allowed. A secure connection between client and server is only possible if both are configured to use the System Management Server.

     Note: If you want only the encrypted connection (V3), then unselect the **Accept non-encrypted SuiteLink connections (mixed mode)** check box, and click **OK**.

   Mixed mode is recommended for use under the following condition:

   - To support legacy applications that do not use encrypted SuiteLink communications. While upgrading an existing Communication Drivers Pack with unsecure SuiteLink to 2023 or later without configuring SMS, then fallback is disabled. SuiteLink no longer works after upgrade.You have to enable the mixed mode after the upgrade is complete.



6. Click **Configure**.

Note: Whenever the SuiteLink communication mode is changed, a system restart is required before the new mode will take effect.

For more information on System Management Server configuration, refer to the Common Platform Configuration section of the System Platform Installation Guide.

Configuring a Securing SuiteLink Connection

Follow the steps described below to configure a secure SuiteLink connection.

Step 1: Set up and Register with the System Management Server

a. **Setting up the System Management Server:** A computer with the application environment is designated as the System Management Server. The System Management Server node holds and distributes the security related information to the other nodes in the environment. The security information is in the form of server certificates.

b. **Registering with the System Management Server**: All the nodes which need to securely communicate with one another will have to register with the System Management Server node. All the nodes

registered with the System Management Server node are grouped together, and can communicate securely with one another.

Use the Configurator to configure the ASB System Management Server point to the System Management Server on the GR node.

Step 2: Server Initialization

It is highly recommended that you configure the SuiteLink connectivity in Communications Drivers in secure encrypted mode. However, if you have legacy installations of SuiteLink that have not been upgraded to the latest version, you will need to set the mixed mode(encrypted and unencrypted) in the System Management Server to allow the latest version of SuiteLink to communicate with the legacy SuiteLink servers.

User Credentials for Secure SuiteLink Server

A secure SuiteLink server needs to be run with a user credential that is either SYSTEM or is part of the ArchestrAWebHosting user group. Failure to do so may cause startup failure with errors showing up in the logger.

For more information about adding users to user-groups, refer to the Windows-specific documentation.

AVEVA

Chapter 8

Item Reference Descriptions

Use item references to access standard system items in the Communication Driver, and to access data stored in memory registers in connected devices. This section only describes the item reference and generic OPC syntax. For more information about item references, see Managing Device Items.

Device Registers

Item references are addresses of device registers, and device registers are specific locations in a device's memory.

In most cases, each item reference has the following attributes:

register type
What the memory location is used for. Register types include inputs, outputs, link relays, latch relays, timer bits, and so on. For limitations on register types, see the manufacturer's documentation for your device.

range
The range of valid addresses for a given register type. A complete address typically consists of a prefix indicating the register type and a numerical value indicating the specific memory location.

data type
The types of data the client can request when it accesses the specified memory location. Data types include Short, Word, BCD, and so on. The data type is often optional because most memory locations have a default data type.

access
What a client application can do with the memory location. Some locations are read only or write only, while others are fully read/write.

For a complete description of the item reference syntax and options, see the Communication Driver-specific documentation.

Standard System Items

System items provide easy access to the status and diagnostics information. Client applications can read data from them just like ordinary items.

However, in most cases the system item values are not directly acquired through the communications layer. System item values are usually generated through internal calculations, measurements, and tracking.

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 and the device the item is attached to.
- Device group (OPC access path/topic): A collection of items on the same physical location with the same protocol update rate.

For DDE/SuiteLink clients, SYSStatus always comes from the leaf level of a Communication Driver hierarchy branch, which is the destination PLC node. For OPC clients, SYSStatus can be accessed at all hierarchy levels. SYSStatus at the root level of the whole hierarchy tree is always good, as it represents the quality status of the local computer itself. For practical application, OPC clients should reference SYSStatus at any hierarchy levels other than the root.

All system items follow the same naming convention:

- All system items start with SYS.
- The Communication Drivers Core 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 server-wide.

Global System Item

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

System Item Name (Type)	Type/Access Rights	Description	Values
SYSLicensed	Boolean/Read	Binary status indication of the existence of a valid license. If FALSE, this item causes the Communication Driver to stop updating existing tags, to refuse activation of new tags, and to reject write requests in addition to setting quality for all items to BAD. If TRUE, the Communication Driver functions as configured. All instances have the same value.	RANGE: 0, 1 1: Valid license exists. 0: No valid license exists.

SYSReadOnly	Boolean/Read	Binary status indication of the Read Only state. 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 device(s) the Communication Driver is connected to.

System Item Name (Type)	Type/Access Rights	Description	Values
SYSStatus	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. For DDE/SuiteLink clients, SYSStatus always comes from the leaf level of a Communication Driver hierarchy branch, which is the destination PLC node. For OPC clients, SYSStatus can be accessed at all hierarchy levels. SYSStatus 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 SYSStatus at any hierarchy levels other than the root.	RANGE: 0, 1 1: Connection to the device is intact. 0: Error communicating with the device.
SYSErrorCode	Longint/Read	Detailed error code of the communications state to the device. The device group (OPC access path/ topic) does not affect the value.	>= 0: Good status (0 is the default state – connected. >0: is some device state like: connecting, initializing, etc. <0: Error status (value indicates the error).

SYSErrorText	String/Read	Detailed error string of the communications state of the device. The device group (OPC access path/ topic) does not affect the value.	Descriptive text for the communications state corresponding to the error code.
SYSStoreSettings	Integer/ ReadWrite	Makes the temporary update interval changes via the SYSUpdateInterval item permanent. If the client pokes a value of 1 into this system item, the currently set update interval is written to the server's configuration file. The value of this system item clears to 0 after being set, if the configuration file write is successful. If the write fails, then the value is set to -1. If the update interval changes via the SYSUpdateInterval item and this item is not poked to 1, the Communication Driver uses the original update interval for that topic the next time it is started. Reading the item always provides 0. Read/Write values are persisted only if you set this system item.	RANGE: -1, 0, 1 -1: Error occurred during saving the configuration file. 0: Read value always if status is OK. 1: Persist settings (cleared immediately).

Device Group-Specific System Items

The following system items refer to specific information regarding device groups that are configured in the Communication Driver.

System Item Name (Type)	Type/Access Rights	Description	Values
SYSUpdateInterval	DWord/ ReadWrite	Accesses the currently set update interval. It is the current update interval of the device group in milliseconds. A client can poke new values into this item. The value of zero indicates that non-system items on that topic are not updated (data for these items are not acquired from the device).	RANGE: 0…604800000 0: Topic inactive, no items are updated. Data acquisition is stopped. Greater than 0: Expected updated interval for the set of all items in the device group.
SYSMaxInterval	DWord/Read	It provides the rolling average in milliseconds of the last 10 actual	

		update durations for the Communication Driver to send and receive all the items in the same topic. This item is read-only. Not supported by some Communication Driver.	
SYSWriteComplete	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, indicating all write activities are complete – no pokes are pending. If values are poked into any items of the device group, the value of this item changes to 0, indicating write activity is currently in progress. If the server has completed all write activities, the value of this item changes to 1 if all pokes were successful or to -1 if at least one poke has failed. If the value of this item is not zero, the client can poke 1 or -1 to it (poke a 1 to clear errors, or a -1 to test a client reaction on write errors). If the value of this item is zero, it cannot be poked.	RANGE: -1, 0, 1 1: Write complete (no writes are pending – initial state). 0: Writes are pending. -1: Writes completed with errors.
SYSReadComplete	Integer/ ReadWrite	Accesses the state of initial reads on all items in the corresponding device group. The value is 1 if all active items in a device group have been read at least once. If at least one item in the device group is activated, this item changes to 0. It changes to 1 if all items have been read successfully, or to -1 if at least one item has a non-good quality.	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.

		Poking a 0 to this item resets the internal read states of all items in this device group. This resets this item to 0. If all items are read again after this poke, this item changes back to 1 or -1.	
SYSItemCount	DWord/Read	Accesses the number of items in the corresponding device group. This item is read-only.	RANGE: 0…2147483647 >=0: Number of items.
SYSActiveItemCount	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.
SYSErrorCount	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.
SYSPollNow	Boolean/ ReadWrite	Poking a 1 to this item forces all items in the corresponding device group to be read immediately (all messages in this device group become due). 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). Not supported by some Communication Driver.	

Redundant Device Specific System Items

These system items are specific to the Redundant Device.

System Item Name	Type/Access Rights	Description	Values
SYSForceFailover	Boolean/ ReadWrite	This is required to achieve the failover condition to be forced by client. **Note:** By poking a value of "1" (True) into the Force Failover item, a client can conveniently switch to the secondary device.	TRUE, FALSE
SYSActiveDevice	String/Read	This system item will show the current runtime active device.	Node Hierarchy Name
SYSFailoverTime	Time/Read	This system item will show the time at which the switch occurred.	Time at which the switch occurred
SYSStandbyDevice	String/Read	This system item will show the current runtime standby device.	Node Hierarchy Name
SYSSecondaryDevice Status	Boolean/Read	This system item will show the status of the secondary device. This is the status of the second device defined in the configuration and is not changed with any failover. RANGE: 0, 1	RANGE: 0, 1 (Contains the value of the system item SYSStatus)
SYSPrimaryDevice Status	Boolean/Read	This system item will show the status of the primary device. This is the status of the first device defined in the configuration and is not changed with any failover. RANGE: 0, 1	RANGE: 0, 1 (Contains the value of the system item SYSStatus)
SYSFailoverReason	String/Read	This system item will show the reason for the failover.	Descriptive text "ForceFailover" or the value of the system item SYSErrorText.

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

Generic OPC Syntax

The Communication Driver serves as a container for the OPC Groups, which provide the mechanism for containing and logically organizing OPC items. Within each OPC Group, an OPC-compliant client can register OPC

items, which represent connections to devices in the field device. In other words, all access to OPC items is maintained through the OPC Group.

The fully qualified name for an OPC item is called the Item ID (equivalent to Item Name). The syntax for specifying a unique Item ID is Communication Driver-dependent. In OPC Communication Driver, the syntax can be as follows:

```
AREA10.VESSEL1.TIC1.PLC.400001
```

where each component (delimited by a period) represents a branch or leaf of the field device's hierarchy.

In this example:

- `AREA10.VESSEL1.TIC1` is the link name
- PLC is the name of the target PLC
- 400001 is the specific data point (Item) desired.
    - An item is typically a single value such as an analog, digital, or string value.

The Item ID describes the syntax for defining the desired data point, and Access Path defines optional specifications for obtaining that data.

In Communication Drivers, the Access Paths are equivalent to Device Groups. It defines the update interval between the Communication Driver and the field device for accessing the values of data points in the PLC.

AVEVA

Chapter 9

Managing Your Communication Driver

About Managing Your Communication Driver

After you configure the Communication Driver, perform the following steps to access data with your client application:

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

Activating/Deactivating the Communication Driver

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

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

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

To activate the Communication Driver

1. In the OI Server Manager, expand the node group, and then expand **Local**.
2. Select and right-click the Communication Driver you want to activate, and click 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 Communication Driver

1. In the OI Server Manager, expand the node group, and then expand **Local**.
2. Select and Right-click the Communication Driver you want to activate, and 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: The Communication Driver with active OPC clients does not stop until the last OPC client shuts down.

OPC/COM Activation

The Communication Driver runs only out-of-process (that is, as a stand-alone process).

If a client uses the CLSCTX_ALL value in an activation call (e.g., CoCreateInstance) to activate the Communication Driver, out-of-process activation is triggered. Explicitly starting the Communication Driver in-process (that is, as a part of the client process) is not supported.

Using the CLSCTX_ACTIVATE _64_BIT_SERVER value in an activation call is also not supported.

Hot Configuration

Hot configuration allows you to configure some parameters in the Communication Driver while it is running.

If you change the value of a parameter and it immediately takes effect while the Communication Driver is running, the parameter is hot-configurable.

The following parameters and features are hot-configurable:

- Global parameters on the **Configuration** node.
- Adding, deleting, or modifying a device in a channel. (Modifying a device does not affect other devices except for the children of the modified device.)
- Adding, deleting, or modifying a device group, in the **Device Groups** tab on a device node.
- Adding, deleting, or modifying an item reference, in the **Device Items** tab on a device node.

There is no support for hot configuration of server-specific parameters. If you configure server-specific parameters while the Communication Driver is running, a warning message is displayed to restart the Communication Driver for the changes to take effect.

See the sections about the specific parameters for limitations or constraints.

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

Demo Mode

You can install the Communication Driver without a license. The Communication Driver runs without a license in Demo mode for 120 minutes.

While in demo mode the Communication Driver checks for a license every 30 seconds. At the end of 120 minutes:

- the Communication Driver stops updating items.
- all non-system items have a Bad quality status.
- new items are rejected.

After 120 minutes, if a license is not found, it logs a warning.

You can use the SYSLicensed system item to check the status of your license. This item returns true if the proper license is found or the Communication Driver is in demo mode (120 minutes), otherwise, it returns false.

When a valid license is found, it stops looking for a license, and begins running normally. For more information, see the AVEVA™ Enterprise Licensing Help.

Chapter 10

Accessing the Data in Your Communication Driver

About Accessing Data in your Communication Driver

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

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

For information on how to specify item references, see Item Reference Descriptions.

Accessing Data Using OPC

To connect to the Communication Driver with an OPC client application, be aware of the following parameters:

node name
The computer name identifying the node where the Communication Driver is located. Only required for remote access.

program name (ProgID)
The name (Programmatic IDentifier) of the specific server instance (e.g., `OI.ABCIP_0001.1`).

group name
An OPC group defined and created by the client. The device group is used as the OPC access path.

device group
A device group as defined on the Communication Driver. If omitted, the default device group is assumed.

link name
The hierarchy of nodes names, based on the node name configured in the OI Server Manager.

item name
The specific data element. This can be the device item name or the item reference.

The combination of the link name and item name form the OPC data path for any OPC client to access the data in the Communication Driver.

If the item specified is not valid for the device location, the Communication Driver does not accept the item, returns bad quality, and generates an error message in the logger.

Accessing Data Using DDE/SuiteLink

To connect to the Communication Driver using DDE/SuiteLink addresses, be aware of the following address fields:

node name
The computer name identifying the node where the Communication Driver is located. Only required for remote access.

application name
The application name for the specific server instance (for example, `OI.ABCIP_0001`).

topic name
A device group defined for the device.

item name
The specific data element. This can be the device item name or the item reference.

The DDE/SuiteLink topic is the equivalent to the device group.

Note: DDE is supported only by the default instance of the Communication Driver.

Running the Communication Drivers from the command line using DDE

DDE communication between Communication Driver and DDE clients can work only if both driver and client are running in the same user session. It means that the Communication Driver must not be automatically activated. It must run as an application started by the same interactive user that is running the DDE client.

For non DASWRAPPER-based Communication Driver

To run the Communication Drivers listed below, navigate to the corresponding installation folder, and double-click the .exe file.

- OI.ABCIP
- OI.ABTCP
- OI.Gateway
- OI.GESRTP
- OI.ITME
- OI.MBTCP
- OI.MQTT
- OI.SIDirect
- OI.WebSvc

The .exe file is usually located in the below file path:

C:\Program Files (x86)\Wonderware\OI-Server\<OI-Server Name>\bin\OIServer.exe

For example:

To run ABCIP Communication Driver directly from the command line, navigate to C:\Program Files (x86)\Wonderware\OI-Server\OI-ABCIP\bin and double-click ABCIP.exe

To run MBTCP Communication Driver directly from the command line, navigate to C:\Program Files (x86)\Wonderware\OI-Server\OI-MBTCP\bin and double-click MBTCP.exe

For DASWRAPPER-based Communication Drivers

To run the OI Servers not listed, follow the steps below:

1. Retrieve the ClassID and ProgID of the Communication Driver
   a. Ensure that the Communication Driver is configured to run in either Auto-Start or Manual-Start mode (desktop mode).

      Right-click the Communication Driver and select **Desktop mode (Must start from command line)**. If already active , deactivate and activate in desktop mode.

   b. Navigate to the services applet Communication Driver.

      Or, Open the Command prompt, and type: `services.msc`.

   c. In the **Name** column, locate the desired Communication Driver.

   d. Right-click the Communication Driver, and select **Properties**.

      The **OI Server Properties** dialog appears.

   e. In the **General** tab, the **Path to executable** displays the file path of the .exe file. Select to highlight and copy the path.

   Note: The value of path to executable is usually very long, and only a part of the string is shown. Ensure to copy the entire string into the copy clipboard or Notepad.

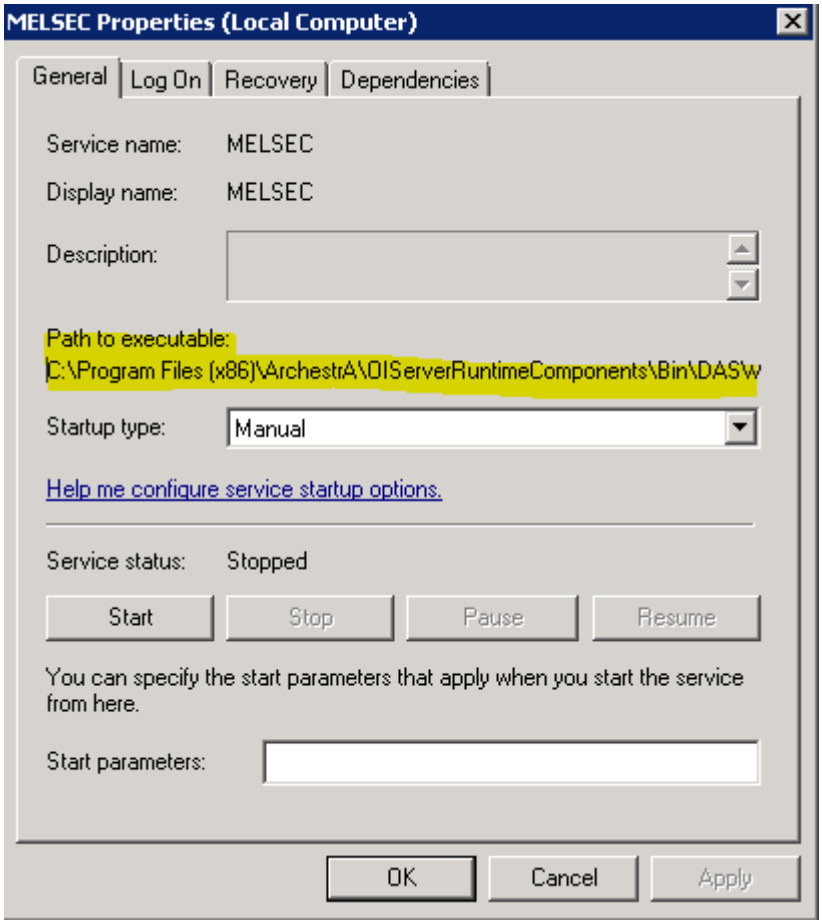
   A general executable path is given below:

   C:\Program Files (x86)\ArchestrA\OIServerRuntimeComponents\Bin\DASWRAPPER.exe -service /CLSID={<Value of Cass ID>} /ProgID=<Value of Prog ID>

   The value of the ClassID and ProgID are shown right after the parameter /CLSID= and /ProgID= respectively.

   For example: The general executable path of MELSEC Communication Driver is:

   C:\Program Files (x86)\ArchestrA\OIServerRuntimeComponents\Bin\DASWRAPPER.exe -service /CLSID={9EE7EE06-36F9-43D7-989B-4B96117F1A75} /ProgID=OI.MELSEC.1



2. Execute the command line to start the Communication Driver manually.
   a. Open the command prompt.
   b. Run the command. Replace the ClassID and ProgID with the values obtained from the above steps.

```
"C:\Program Files (x86)\ArchestrA\OIServerRuntimeComponents\Bin\DASWRAPPER.exe" /
CLSID={<OIServer Specific CLSID>} /ProgID=OI.Server.x
```

   It is highly recommended to put the above line in a batch file, to re-run the Communication Driver more readily.

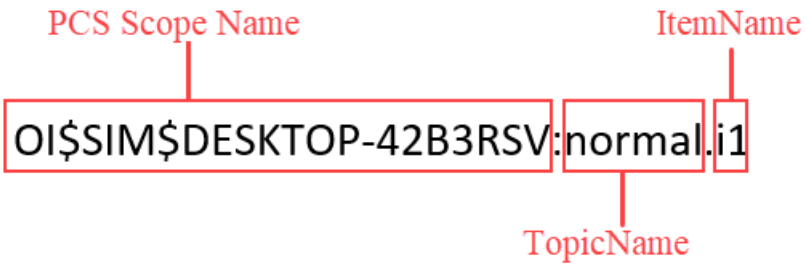
Accessing Data Using PCS

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

`<PCS Scope Name>:<TopicName>.<ItemName>`

- **PCS Scope Name**: The scope name configured for PCS connectivity for the Communication Driver.The syntax of the PCS Scope Name is `OI$<OIServerInstanceName>$<ComputerName>.`
- **TopicName:** A device group that is configured in the Communication Driver.
- **ItemName**: The full name of the item required by the specific protocol of the Communication Driver where the item is being subscribed.

Example of the PCS item syntax:



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 for more details.

Buffered Data and Late Data Support

As of Communication Drivers Pack 2023, data acquired by the Communication Drivers are buffered based on the received order until they are drained either by the OPC, SuiteLink, or PCS IData client of the Communication Driver. The size of the buffer is determined on a per item basis by the configuration parameter "Buffered Data" in the Global Parameters. Since the memory associated with the buffer is only allocated on a per-demand basis, any spike in memory consumption only occurs temporarily until the buffer is drained by either the OPC, SuiteLink, or PCS IData client of the Communication Driver. Refer to for configuration details.

If the acquired data in the Communication Driver are out of time sequence consecutively, the quality of subsequent data (that is, late data) will be indicated with the 0x0100 mask.

Consider two consecutive data values received by a Communication Driver:

The value of the first received value chronologically has VTQ:
value 1.0, timestamp 2022-02-28 11:00:00, quality 0x00C0

and the next received value has VTQ:
value 2.0, timestamp 2022-02-28 10:59:00, quality 0x00C0

The first received VTQ will be dispatched to SuiteLink/OPC/PCS IData clients as is but the second received value will be dispatched to SuiteLink/OPC/PCS IData clients with the data quality 0x01C0 since its timestamp (2022-02-28 10:59:00), is earlier than the timestamp of the VTQ immediately preceding it (2022-02-28 11:00:00). This will facilitate System Platform in recognizing and processing late data so that they can be reconstituted in the System Platform Historian in the proper time order in the VTQ.

Chapter 11

Using Auto-Build

Auto-Build is an engineering efficiency feature that allows you to read the templates and instances in the controllers capable of running the Auto-Build feature. For a one-to-one correspondence between the PLC and a Galaxy project, you can also replicate these structures to the Application Server.

Auto-Build browsing can be accomplished online if connected to the PLC, or offline if the PLC file (for example, L5X for ABCIP) is available.

Prerequisites for Auto-Build Operation

Ensure the following prerequisites are met before configuring the Auto-Build.

Browser Requirements

Auto-Build requires Microsoft Edge browser for functioning.

Installation Requirements

Auto-Build must be installed in the same node as an Application Server Galaxy Repository. You can only view the Auto-Build tab from the Application Server node. You cannot view the Auto-Build tab from a remote node.

Licensing Requirements

Auto-Build requires a professional Communication Driver license or higher. Refer the Licensing section for more details. If the license server is not configured, a warning message is displayed:

Unable to obtain license for Auto-Build.

Note: A standard license, or no license will allow the user to exercise Auto-Build configuration steps, but not building the objects in Application Server.

Required User Privileges

The user who launches the OCMC must be part of the oiAdministrators Windows Local Users and Groups to access the Auto-Build functionality, else an error message is displayed saying the user or the user group is not part of oiAdministrators group. If the user is not a part of the oiAdministrators group, you must add the user to the oiAdministrators group manually and re-login your computer. The user who installs the Communication Drivers Pack will be added to the oiAdministrators group automatically.

PLC Data Processing and Validation Rules

Rules to Configure Auto-Build

Following properties are configurable in rule file:

Sl. No.	Property Name	Description	Default Value
1	MaxAttributesPerArea	If the total number of attributes under Area reaches the max limit, the next object will be assigned to new Area.	25000
2	MaxObjectsPerArea	If the total number of objects under Area reaches the max limit, the next object will be assigned to new Area.	1000
3	ContainedObjectsWithGuidNames	If the value set to 1 then the name of the contained object will be GUID, which will uniquely identify the object. The original name will be moved to Description attribute of the object. The GUID name is given to the object in order to avoid long object name, as ArchestrA has limit of 32 chars for object name.	0
4	ServerName	Indicates the default server name listed in the DDE SuiteLink object server name configuration and depends on the PLC Type.	For example: ABCIP for ABCIP Communication Driver TI500 for TI500 Communication Driver
5	ServerNode	Indicates the default server node where the Communication Driver is running. The value depends on the PLC Type.	Localhost

6	ScanGroupPrefix	This value will specify the name of the Scan Group. The unique number will be appended at the end of the name, when multiple Scan Groups are created.	AutoBuildSG_
7	DIObjectPrefix	This value will specify the name of the DI Object. The unique number will be appended at the end of the name, when multiple DI Objects are created.	AutoBuildDI_
8	AreaPrefix	This value will specify the name of the Area Object. The unique number will be appended at the end of the name, when multiple Area objects are created.	AutoBuildArea_

Template Name Validation Rules

The template name validation follows the Object naming convention.

- The template name can be 32 characters long. However, if a prefix is added (Prefix_Template name), the maximum length of the template is 29 characters. See Prefix Syntax Validation Rules.
- The template name can contain alphanumeric characters (A-Z, a-z or 0-9)
- The template name can contain only the following special characters: $, #, _

For more information about the naming convention rules, refer to the section of the Application Server user guide.

Using Prefix for Identical PLC Programs

The two-character prefix is used to uniquely represent multiple PLC programs that are identical, or contain common data structures within the Application Server. Add the prefix during the Auto-Build operation to append the prefix to all the templates and instances built in the batch.

Note: Adding a prefix does not require any changes in the PLC. All references will still be mapped properly to the appropriate addresses in the controller.

Prefix Syntax

The syntax of the Prefix is given below.

```
<Prefix>_<Template Name>
<Prefix>_<Instance Name>
```

Examples:

AB_Template; AB_Instance

A1_Template; A1_Instance

Prefix Syntax Validation Rules

The validation rules for the prefix syntax are:

- The prefix can only be two characters long.
- The characters in the prefix are not case sensitive.
- The first character must be an alphabet (A-Z).
- The second character can be alphanumeric (A-Z, a-z, or 0-9).
- Special characters are not allowed.

Some of the incorrect prefix examples and the error messages are listed below.

Prefix Syntax	**Error Message**
Contains more than two characters • Aa1 • Z4Ab	Length should not be more than two characters
First character not an alphabet • $A • 1B	Please input valid input prefix
Prefix contains a special character • A$ • %B • %&	Please input valid input prefix

Note: The maximum length of the template is 32 character. However, if a prefix is added, the maximum length of the template is 29 characters (Prefix_template name : 2+1+o29).

Stages of Auto-Build Operation

Different stages of importing the templates and instances using Auto-Build feature are listed below.

1. Source & Destination
2. Pre-Check
3. Templates & Instances
4. Review & Submit

At any stage of the Auto-Build operation, to access the Help pages, click the help icon located on the upper-right section of the screen.If you make any changes to the configuration of the System Management Server (SMS) during run time, you must restart the **AVEVA Communications Backend Service** in the **Services** console of your machine.In the **Advanced Configuration** of the System Management Server (SMS), if you change the **HTTP Port** or the **HTTPS Port** then you must run the below command as an administrator in the command prompt:

If you change the HTTP Port field

```
netsh http add urlacl url=http://localhost:<http port configured in SMS>/oi/ user="NT Authority\Network Service"
```

If you change the HTTPS Port field

```
netsh http add urlacl url=https://localhost:<https port configured in SMS>/oi/ user="NT Authority\Network Service"
```

If you do not run the above commands after changing the ports and try to access the Auto-Build functionality, you will get a HTTP 503 error saying the service is unavailable.

Source & Destination

The **Source and Destination** section allows you to select a source to connect to the PLC, and also select a destination where the templates and instances should be built.

1. Select a source to run Auto-Build.
   - **Online Mode**: Online mode allows you to connect to the specific PLC,which is currently selected and from which the current session of Auto-Build has been activated. It requires the PLC to be connected and the OI Server to be activated.
   - **Offline Mode**: Offline mode allows you to use a file that has been exported from the PLC programming software.
2. Select a destination Galaxy to upload the new objects.
   a. **Galaxy Name**: Select the Galaxy Name from the list. As a best practice, select a newly created, unpopulated Galaxy. You must have a valid IDE_Runtime license for the galaxy list to be populated.
   b. **Device Group**: Select the Device Group from the list. This list is populated from the topics created in the **Device Groups** section. Select the device groups that adhere to the the following naming conventions:
      - The length of the device group name must be less than 255 characters.
      - The device group name can contain only alphanumeric or special characters like $,#,_.

      For more details about managing device group, see Managing Device Groups.
   c. **Prefix**: For multiple PLC programs that are identical or contain common data structures, enter the Prefix of the Object being uploaded.

      For more details about using prefix, and the prefix syntax validation rules, see Prefix Syntax Validation Rules.

      Note: Entering a prefix is optional, and not mandatory during the Auto-Build operation.
3. Click the forward arrow to proceed to the next page.

Note: Once the next page loads, the previous page number in the header frame is marked with a check mark, indicating that the step has been completed.

Information Icons

Hover the mouse pointer over the information icon to see the additional information about the respective field.

- **Online Mode:** Hover over the information icon next to **Online Mode** to display the limitations of online mode.

- Online mode excludes tag descriptions.
- Online mode exposes tags that have been configured in the PLC program as "External Access=None" or "Usage=Local".
- Online mode creates all the attributes as Read/Write.

- **Prefix:** Hover over the information icon next to the **Prefix** for more information about the usage of prefix.
  - Using a prefix is highly recommended, especially if the project connects to more than one PLC controller, where naming of templates and instances may overlap.
  - If updating an existing Auto-Build based project, ensure to use the same Prefix and Device Group used before.

Pre-check

The **Pre-check** screen displays the templates, instances, and attributes with syntax conflicts, and the corresponding error messages. These templates and instances will be skipped from the build operation.

- To export the error message to a CSV file for reference, click the download icon  next to the **Warning** message.
- Click the forward arrow to proceed to the next page.

Note: The warning icon  denotes the syntax conflicts for the templates and instances, respectively. For common syntax errors, See PLC Data Processing and Validation Rules.

Template & Instance

The **Template & Instance** screen allows you to select specific templates, attributes, and instances to be uploaded to the galaxy.

- Click **SELECT ALL Templates and instances** to select all the listed templates and instances. Click **CLEAR ALL Templates and instances** to clear the selection.
- Click **SELECT ALL Templates only** to select only the listed templates and not the instances. Click **CLEAR ALL Templates only** to clear the selection.
- Select **Display templates with Instances only** to filter templates that have associated instances.
- Browse through the templates list using the **Template Name Filter**.

To select a template and instance

1. Scroll through the list under **Templates**.
   - Select a template to display the list of associated attributes.
   - The associated instances are displayed in the **Instances** section.
2. Click the forward arrow to proceed to the next page.

Review & Submit

The **Review & Submit** page allows the review and submission of selected templates, their instances and attributes.

To review and submit a template

1. Click each template to view the related attributes and instances.
2. Verify the details are correct and click the forward arrow.

Monitoring Auto-Build Progress

To start the Auto-Build operation

- In the **You are about to start build operation** dialog, click **START BUILDING NOW** button to start the building process.

The progress bar displays the extent of completion of the building operation.

To stop the Auto-Build operation

1. Click the **Stop** button during the progress to terminate the Auto-Build operation.

   The progress displays the terminating process. Upon completion, the dialog displays the information that the Build was stopped by the User.

2. Click **OK.**

   The galaxy templates and instances built prior to the stop request will reflect in the Galaxy.

Note: Click anywhere on the screen during the operation to minimize the progress to the title bar. Click the title bar again to view the progress.

To view the Logs

- Open the Log Viewer within the OCMC to view the logs recorded during the Auto-Build operation.

Validations during Build Operation

- Only one Auto-Build operation is allowed at a time. If another object is uploaded to the same galaxy when the first upload operation is in progress, the following warning message is displayed:

  Build operation is in progress. Please allow the current operation to complete before starting the new operation.

- The Galaxy list populates only if you have a valid IDE_Runtime License. Otherwise, an error message is displayed:

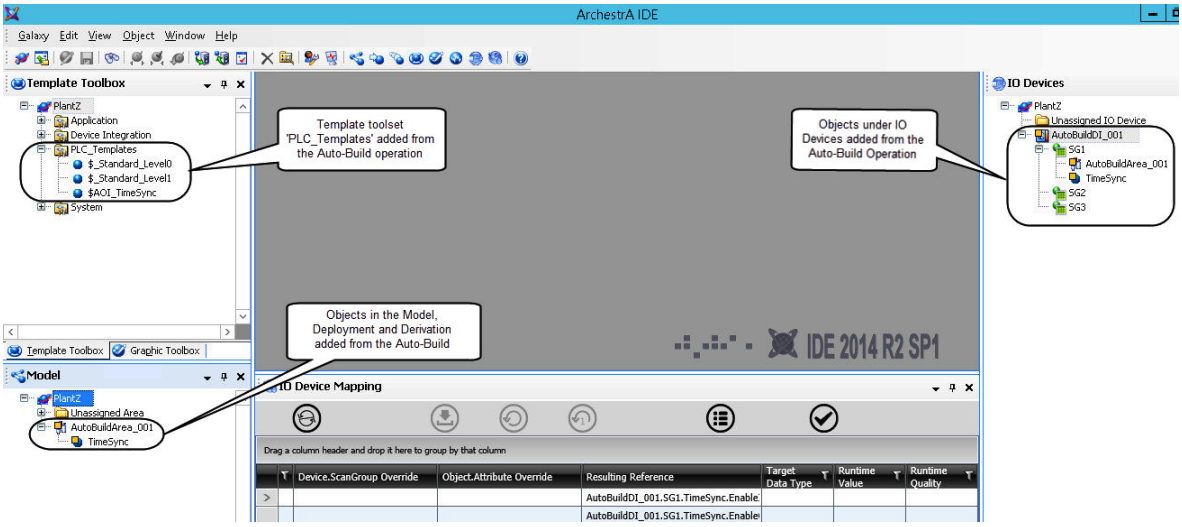
  Failed to retrieve information from the Industrial Application Server License. Either the Industrial Application Server License is not available or it has expired.

- If the selected galaxy is security enabled, the build operation fails with the following warning message:

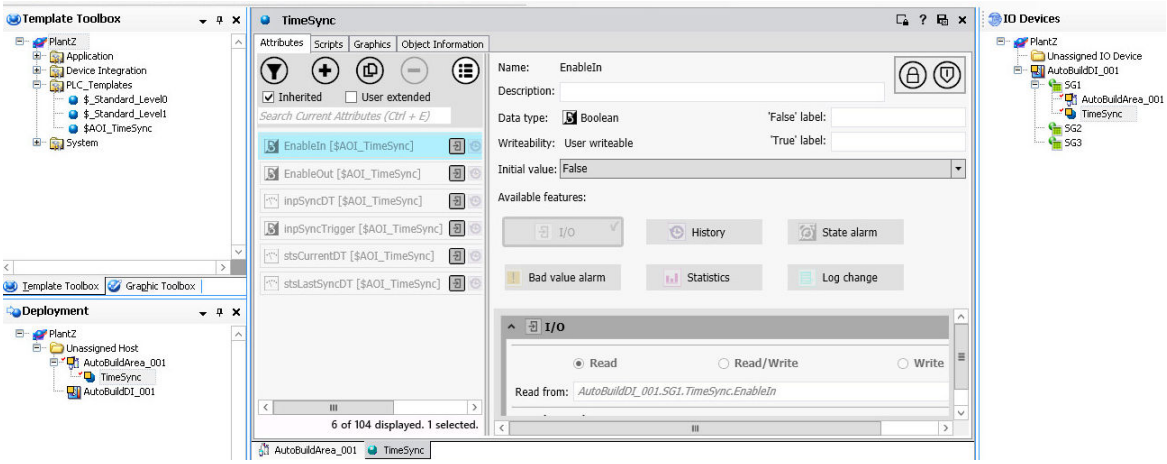
  Build operation failed for Galaxy 'XXX'. Please check logger for more details.

Template Generation in the Application Server

The instances submitted from Auto-Build are generated in the selected Galaxy within the IDE.



The attributes of the selected instance are displayed in the center of the IDE screen as shown below.



AVEVA

Chapter 12

Troubleshooting

Basic Tools for Troubleshooting

You can troubleshoot problems with the Communication Driver using the:

- Windows Task Manager
- Windows Performance and Alerts (PerfMon) application, also called Performance Monitor
- Communication Driver Diagnostics
- Log Flag Editor
- Log Viewer

Your client application may let you view error messages, monitor the status of requests, and allow you to request data on the status of the Communication Driver and connected devices. For more information, see your client application documentation.

Using the Diagnostics Node

The OI Server Manager has information that may be useful in troubleshooting problems. When the Communication Driver is active, a **Diagnostics** node is present below the **Configuration** node in the console tree of the OCMC.

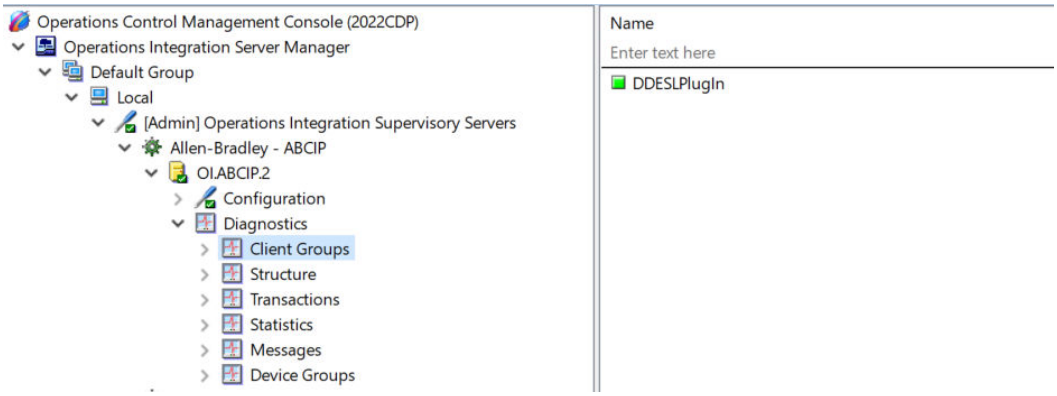
All Communication Driver include at least the following six diagnostic nodes:

- Client Groups
- Structure
- Transactions
- Statistics
- Messages
- Device Groups

Viewing Status Information

When you select individual diagnostic nodes or items contained in a diagnostic node, the Details pane shows data for that node or item.

The following example of a diagnostic hierarchy shows a DDE/SuiteLink plug-in for the **Client Groups** node.



What is shown on your computer may vary depending on the plug-in(s) you have installed.

A particular Communication Driver may have additional diagnostic nodes and additional data shown under the common diagnostic nodes. For more information and description about additional features, see your Communication Driver documentation.

If the string value obtained by the Communication Driver contains a tab character or newline character, they are filtered out and replaced with the sequence "\t" and "\n" respectively.

Descriptive icons are present with each data element that indicates the quality of the presented information. These icons are described in the following table:

Icon	Description
	The clock indicates that the data shown on the line was not updated during the last update interval and the data item has just been scrolled into view. The clock is shown until the next update interval, when it changes to other icons described in this table.
	White - Initializing (no VTQ available yet, no time-out yet)
	Red - Error Active
	Yellow - Warning Active
	Green - Normal Active
	Red Marked - Bad Inactive

	Yellow Marked - Warning Inactive
	Green - Normal Inactive

Sorting and Filtering

Sorting the Columns

Sorting feature helps you to easily sort the data elements of different quality. You can sort the **Name** column in all the Diagnostics nodes. You can sort the **Quality/Result** column in **Device Groups** and **Messages** nodes, **Client Quality** column in **Client Groups** node, and **Quality** column in **Structure** and **Transaction** nodes.

- To sort a column, click on the column header.

Filtering the Columns

Filtering feature helps you to filter the data elements in different ways. You can apply multiple filters at the same time by applying filters in different columns. After applying the filter you can also sort the filtered items.

To filter a column:

- Enter the filter text in the required column.

For example, to filter the items by name, type the required name of the item in the filter text box of the **Name** column. It displays the items with the given name. It also supports regular expressions. In the filter box enter the regular expressions like `.*, \[.?\], \.` and so on.

Example

To apply a filter to the below items, you can use different characters, as explained in the following tables:

- FillLevel.V
- FillLevel.PV
- FillLevel.SP
- FillLevel.SL
- bFeeder01_Running
- bFeeder02_Running
- bFeeder11_Running
- bFeeder111_Running
- fHumidity.PV
- FHumidity.PV
- fHumidity.SP
- fLine_Speed.PV
- fLine_Speed.SP
- fTempurature.PV
- fTemparature.SP

- Tank.Line_1.Value
- Tank Line2
- Tank_Line21
- TankLine2$

Special Pattern characters

Character	Description	Example
.	Any character except line terminators (LF, CR, LS, PS).	“FillLevel.S.” results the items “FillLevel.SP” and “FillLevel.SL”
\d	A decimal digit character	“bFeeder\d\d_Running” results “bFeeder01_Running”, “bFeeder02_Running” and “bFeeder11_Running”
\s	a whitespace character (same as [[:space:]]).	See character classes in the below tables
\S	any character that is not a whitespace character (same as [^[:space:]]).	See character classes in the below tables
\w	an alphanumeric or underscore character (same as [_[:alnum:]]).	See character classes in the below tables
\W	any character that is not an alphanumeric or underscore character (same as [^_[:alnum:]]).	See character classes in the below tables
\character	the character as it is, without interpreting its special meaning within a regex expression. Any character can be escaped except those which form any of the special character sequences above. Needed for: ^ $ \ . * + ? ( ) [ ] { } |	“TankLine2\$” results “tankLine2$”
[class]	the target character is part of the class (see character classes below)	See character classes in the below tables
[^class]	the target character is not part of the class (see character classes below)	See character classes in the below tables

Qualifiers

Character	Description	Example

*	Any atom (character/digit) is matched 0 or more times	• “FillLevel*” results “FillLevel.V”, “FillLevel.PV”, “FillLevel.SP” and “FillLevel.SL” • “*PV” results “FillLevel.PV”, “fHumidity.PV”, “fHumidity.PV”,”fLine_Speed.PV” and “fTempurature.PV” • “*Speed*” results “fLine_Speed.PV” and ”fLine_Speed.SP”
?	Any atom (character/digit) is optional (matched either 0 times or once).	• “FillLevel.?V” results “FillLevel.V”, “FillLevel.PV” and “FillLevel.SV” • “bFeeder??_Running” results “bFeeder01_Running”, “bFeeder02_Running” and “bFeeder11_Running”
+	The preceding atom is matched 1 or more times.	“bFeeding1+_Running” results “bFeeder11_Running” and “bFeeder111_Running”
{n}	The preceding atom is matched exactly n times	“bFeeding1{3}_Running” results “bFeeder111_Running”
{n,}	The preceding atom is matched n or more times.	“bFeeding1{2}_Running” results “bFeeder111_Running”
{n,m}	The preceding atom is matched minimum n times and not more than m times	“bFeeding1{1, 2}_Running” results “bFeeder11_Running” and “bFeeder111_Running”

Alternatives

Character	Description	Example
|	Separates two alternative filter patterns or sub patterns.	“*Level*|bFeeder??_Running” results “FillLevel.V”, “FillLevel.PV”, “FillLevel.SP”, “FillLevel.SL”, “bFeeder01_Running”, “bFeeder02_Running” and “bFeeder11_Running”

Assertions

Character	Description	Example
^	Either it is the beginning of the target sequence or follows a line terminator.	"^bFeeder01_Running" results "bFeeder01_Running"

$	Either it is the end of the target sequence or precedes a line terminator.	“FillLevel.V$” results “FillLevel.V”

Character classes

Character	Description	Example
[:classname:]	Either it is the beginning of the target sequence or follows a line terminator.	“[a-z]Feeder01_Running” results “bFeeder01_Running”
[[:alnum:]]	alpha-numerical character	“Tank.Line_[[:alnum:]].Value” results “Tank.Line_1.Value”
[[:alpha:]]	alphabetic character	“[[:alpha:]]Feeding11_Running” results “bFeeder11_Running”
[[:digit:]]	decimal digit character	Tank.Line_[[:digit:]].Value” results “Tank.Line_1.Value”
[[:lower:]]	lowercase letter	“[[:lower:]]Humidity.PV” results “fHumidity.PV”
[[:print:]]	printable character	-
[[:space:]]	whitespace character	“Tank[[:space:]]Line2” results “Tank Line2”
[[:upper:]]	uppercase letter	“FillLevel.[[:upper]]L” results “fHumidity.PV”
[[:d:]]	decimal digit character	“Tank.Line_[[:d:]].Value” results “Tank.Line_1.Value”
[[:s:]]	whitespace character	“Tank[[:space:]]Line2” results “Tank Line2”
[^[:space:]]	a character class that matches any character except a whitespace	“Tank[^[:space:]]Line21” results “Tank_Line21”

Client Group Diagnostics

If you click the **Client Groups** node, all individual client groups are shown in the Details pane.

You can see the following data for each individual client group shown:

- Data quality icon
- Name of the client
- Number of items
- Number of active items
- Number of items with errors

- Update interval

The data quality icons are marked as active/inactive, depending on their corresponding OPC group state. The following table describes the icon states:

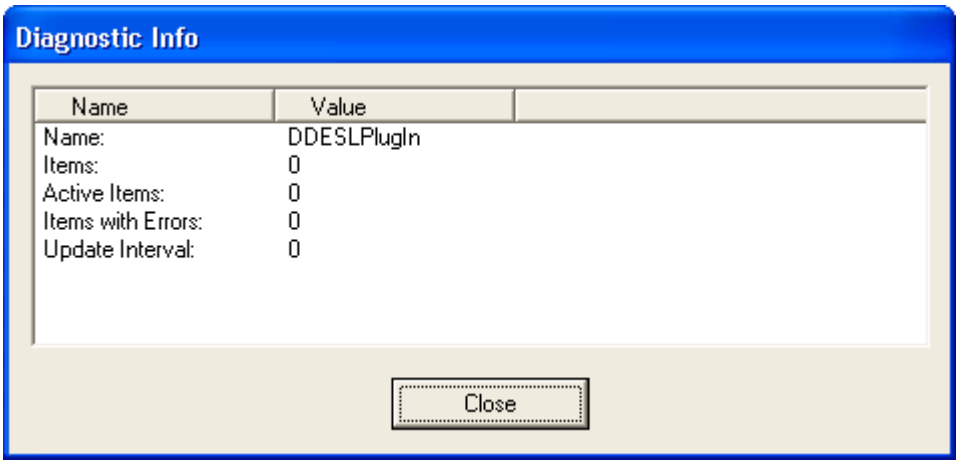
Icon State	Icon Color	Client Group
Normal		The OPC quality is good for all items in this group.
Warning		The OPC quality is uncertain or bad for at least one item in this group and the OPC quality is good for at least one item.
Error		The OPC quality is uncertain or bad for all items.

If you select an individual client group in the console tree, the following data is shown in the Details pane:

- Data quality icon
- Name of the client
- Client value and time
- Client quality
- Subscription message
- State
- Location
- Device group

If you select an individual client group in the console tree, the same normal/warning/error icon colors apply.

When you double-click any item name in the **Details** pane, the **Diagnostic Info** dialog box appears.



The **Diagnostic Info** dialog box provides detailed information about the client group's diagnostics. This information varies between client groups and the type of data described. For example, the **Diagnostic Info** dialog box shows quality for transactions except for write transactions. It shows write status for write transactions.

Structure Diagnostics

The Communication Driver namespace may include many levels of hierarchy, some of which have not been added and are not visible in the console tree. In the **Structure** diagnostic node, you can view data for as many levels of hierarchy as possible based on the structure of the Communication Driver.

Data for each structural hierarchy includes:

- Data quality icon
- Name of the item
- Number of items
- Number of items with errors
- Read/Write status
- Value
- Time
- Quality
- Number of messages
- Device groups

The **R/W Status** column always shows `R/W`, which indicates that the structure supports both read/refresh/property transactions and write transactions. For more information, see R/W Items in Diagnostics.

The structure icons are marked as active or inactive. The hierarchical elements (structural branches) are always marked as active, and the normal/warning/error color of the icons is set as follows:

Icon State	Icon Color	Hierarchical Element
Normal		The status error code is greater than or equal to zero (not negative), and the OPC quality is good for all items on this level.
Warning		The status error code is greater than or equal to zero (not negative), and the OPC quality is uncertain or bad for at least one item on this level.
Error		The status error code is less than zero (negative).

Hierarchical items (leaves) are marked as active if they are referencing at least one active group item. They are marked as inactive if they are referencing all inactive group items. The normal/warning/error color of the icons is set as follows:

Icon State	Icon Color	Item
Normal		The OPC quality is good.
Warning		The OPC quality is uncertain.

Error		The OPC quality is bad.

If the **Structure** node shows an error indicator, it does not necessarily mean that an error condition exists at the hierarchical level in focus. It means that an error condition exists at some point down the hierarchical tree.

Transaction Diagnostics

The **Transaction** diagnostic node contains all individual transactions.

Data for the **Transaction** node includes:

- Data quality icon
- Type of transaction
- Item ID
- Number of items
- Status
- Start time
- End time

Note: Items shown below the **Transaction** node in the hierarchy depend on how a Communication Driver developer has customized the code and configuration of an individual Communication Driver. Items below the root level in this hierarchy are subject to customization and configuration of individual Communication Driver.

The **Status** column indicates whether the transaction status is `COMPLETE`, `ERROR`, or `INCOMPLETE`. The following table shows icon colors for the normal/warning/error conditions for the **Transaction** node:

Icon State	Icon Color	Transaction
Initializing		Transaction is in progress.
Normal		Transaction has completed successfully, and the OPC quality is good for all items or write completed with success.
Warning		Transaction has completed successfully, and the OPC quality is uncertain or bad for at least one item or write completed with error in at least one item.
Error		Transaction has completed with error, the OPC quality is uncertain or bad for all items or write completed with error.

Pending individual transactions are marked with a white icon (the state is undefined/pending). A transaction item element is always marked as active. Viewing completed transactions refers to viewing a snapshot. Data for an individual transaction item includes:

- Data quality icon
- Name of the item

- Read/write status
- Value
- Time
- Quality/result
- Message ID (MsgID)
- Location

The **Quality/Results** column shows the write complete code for demand write transactions and data quality for all other transactions. The **R/W Status** column displays `R` for read/refresh/property transactions or `W` for write transactions. The location is the path of the hierarchical item name.

For a demand write transaction, the data shown in the **Quality/Results** column indicates the success or failure of the item in the transaction.

- All zeros (00000000) specify that the item is successful.
- A positive number specifies that the item is successful but indicates a condition that you must record. You can double-click the item to open the Diagnostic Info dialog box, and view the write status data for additional information about the item.
- A negative number in the Quality/Results column indicates that the item in the transaction is not successful.

The following table describes the status icons for the normal, warning, and error conditions of the transaction items:

Icon State	Icon Color	Transaction Item
Normal		The OPC quality is good; write completed with success.
Warning		The OPC quality is uncertain.
Error		The OPC quality is bad; write completed with error.

Statistic Diagnostics

The **Statistics** diagnostic node has no sub-nodes because it shows general Communication Driver data.

The types of statistics available include:

- Server state
- DAS Engine version
- Start time
- Current time
- Client groups
- Client group items
- Client group errors

- Device items and device errors
- Messages

For each type of statistic, the data shown includes the following:

- Data quality icon
- Type of statistic
- Value
- Unit/info

The following table describes the status icons for the normal, warning, and error conditions of the client group and device items:

Icon State	Icon Color	Statistic
Normal		The number of items with errors is 0.
Warning		The number of items with errors is not zero and not equal to the corresponding total number of items.
Error		The number of items with errors is equal to the corresponding total number of items.

Message Diagnostics

The **Messages** diagnostic node shows data for the node and for the individual messages.

Data shown for the node includes:

- Data quality icon
- Message ID
- Number of items
- Number of items with errors
- Status

Note: Items shown below the **Messages** diagnostic node in the hierarchy depend on how an Communication Driver developer has customized the code and configuration of an individual Communication Driver.

The following table describes the status icons for the normal, warning, and error conditions of messages:

Icon State	Icon Color	Message
Normal		The OPC quality is good for all items in this message.
Warning		The OPC quality is uncertain or bad for at least one item in this message, and the OPC quality is good for at least one item.

Error		The OPC quality is uncertain or bad for all items in this message.

Both messages and message items are always marked as active. The data for an individual message item includes:

- Data quality icon
- Name of item
- Read/write status
- Value
- Time
- Quality
- Message ID (MsgID)
- Location

The **R/W Status** column displays:

- `R` for read/refresh/property messages
- `W` for write messages
- `R/W` for both read/refresh/property and write messages

For more information, see R/W Items in Diagnostics.

The **Location** column shows the path of the hierarchical item name.

The following table describes the status icons for the normal, warning, and error conditions of message items:

Icon State	Icon Color	Message Item
Normal		The OPC quality is good.
Warning		The OPC quality is uncertain.
Error		The OPC quality is bad.

Note: Items below the root level in this hierarchy are subject to customization and configuration of individual Communication Driver.

Device Group Diagnostics

The **Device Groups** diagnostic node shows data for the node and for the individual device groups.

The data that is shown for the node includes:

- Data quality icon
- Device group

- Update interval
- Update actual time
- Messages
- Time per message
- Number of items
- Number of active items
- Number of items with errors
- Location

Only the **Client Groups** and **Device Groups** nodes show system items.

Note: Items shown below the **Device Groups** node in the hierarchy depend on how a Communication Driver developer has customized the code and configuration of an individual Communication Driver. Items below the root level in this hierarchy are subject to customization and configuration of individual Communication Driver.

The following table describes the status icons for the normal, warning, and error conditions for the device groups:

Icon State	Icon Color	Device Group
Normal		The OPC quality is good for all items in this device group.
Warning		The OPC quality is uncertain or bad for at least one item in this device group, and the OPC quality is good for at least one item.
Error		The OPC quality is uncertain or bad for all items in this device group.

Device group items are marked as active if at least one active group item from any active group is referencing the device group items. They are marked as inactive if no active item in any active group is referencing the device group items. Referencing indicates the same fully qualified ItemID/hierarchical name and the same OPC access path/device group.

Data for an individual device group item includes:

- Data quality icon
- Name of the item
- Update actual time
- Number of Messages
- Message
- Time per message
- Read/write status
- Value
- Time

- Quality
- Message ID (MsgID)
- Location

The **R/W Status** column displays:

- `R` for read/refresh/property device group items
- `W` for write device group items
- `R/W` for both read/refresh/property and write device group items

For more information, see R/W Items in Diagnostics.

The **Location** column shows the path of the hierarchical item name.

The following table describes the status icons for the normal, warning, and error conditions for the device group items:

Icon State	Icon Color	Message Item
Normal		The OPC quality is good.
Warning		The OPC quality is uncertain.
Error		The OPC quality is bad.

R/W Items in Diagnostics

The **R/W Status** column appears in the details pane for the **Structure**, **Transactions**, **Messages**, and **Device Groups** nodes.

It shows any pending read/write status as follows:

- Any item that has been updated with VTQ (good or bad) is marked `R`. Before being updated with VTQ, it is marked with a hyphen (-) for read-uninitialized.
- Any item that has no pending write operations, because it either contains all completed write activities or has never been poked, is marked `W`. If write operations are pending on an item, it is marked with a `-W`.

The normal state of an item is `R/W`. All others reflect temporary or special states, such as an item being in an initialization state, an item is processing a write, or an error accessing item data occurred.

1. In the **Structure** dialog box, an `R` indicates:
   - The item is updated from any subscription or transaction with a corresponding fully qualified itemID at least once.
   - A VTQ for any client group item pointing to this item is available if the item is reading from cache and the item is active.

   The `R` is missing only for a short period of time during initialization of an active item. Reasons for a missing an `R` state include:

- Item is not active long enough to acquire data.
- Item cannot access data.
- System item SYSPollNow is poked and the device item is not updated yet.

In the **Structure** dialog box, a `W` indicates:

- No pokes are pending on this item.
- All pokes are complete.
- No write transaction items are connected to this item.

2. In the **Transaction** dialog box, demand reading or demand writing creates a new transaction item. Each transaction item represents a unique demand operation on this item. Transaction items display `R` in the **R/W Status** column during read transactions and `W` during write transactions. The `R` indicates that the item's VTQ was acquired in this transaction for this item, and `W` indicates that the item poke is complete. `R` also can represent a refresh or property read. A refresh is typically a device group-wide read (either from the field device or from cache) that is triggered by the client. A property read is a read of any property of an item (usually a property other than VTQ) by the client.
3. In the **Message** dialog box, an `R` indicates that item VTQ was acquired for this item and a `W` indicates that the item poke is complete.
4. In the **Device Groups** dialog box, an `R` indicates that the item VTQ was acquired for this item. `W` is always set because there are no transaction items and a poke is always complete.

Time Zone Format for Diagnostic Data

All diagnostic data have a date/time component. The system sends diagnostic time information embedded in diagnostic strings or info. This allows the client to show the time in the selected format.

You can view the current time zone component of the date/time stamp or set it by selecting one of the three available formats from the **Time Zone** menu, located on the menu bar. The **Time Zone** menu is available when you select the OI Server Manager or any branch under it in the hierarchy.

The time zone formats are as follows:

- UTC (Coordinated Universal Time): The date/time is shown as per the UTC time of the Communication Driver.
- Client Time Zone: The date/time shown reflects the client's time zone (that is, Communication Driver UTC + client time zone).
- Server Time Zone: The date/time shown reflects the Communication Driver's time zone (that is, Communication Driver UTC + Communication Driver time zone).

Another characteristic of time data associated with diagnostic data is related to the duration of the existence of the data. These characteristics are shown in three formats as follows:

- Short-lived data: Used for time stamping data points of end devices (for example, PLCs, RTUs, and control processors). This format shows hours, minutes, and seconds.
- Medium-lived data: Used for transaction start and end times. This format shows month, day, year, hours, minutes, and seconds.
- Long-lived data: Used for the **Statistics** diagnostic node for the Communication Driver start time and current time. This format shows the day of the week, month, day, year, hours, minutes, and seconds.

OPC Quality Flags

OPC clients show a variety of possible quality flags for an item's data value. These clients typically show quality flags in one of the two forms: textual description or hexadecimal value.

If your OPC client shows quality flags in textual form, see the "Description" table below for descriptions of these flags. If your OPC client shows quality flags in hexadecimal form, see the "Hexadecimal Quality Code Cross-Reference" table below.

For descriptions of hexadecimal quality flags, first check the hex value in the "Hexadecimal Quality Code Cross-Reference" table. Then, use the three numbers in associated Description Cross-Reference column to find the Textual Quality Words and Description in the "Description" table.

The three descriptions provide a complete explanation for the quality code shown in your OPC client. For example, hex quality code 0x001B (**1**, **10**, **20** cross-reference numbers) is described as follows:

- **1** Bad quality - The Value is not useful.
- **10** Comm failure - Communication has failed. There is no last known value available.
- **20** Constant - The value is a constant and cannot move.

Hexadecimal Quality Flag	Description Cross-Reference
0x0000	1, 4, 17
0x0001	1, 4, 18
0x0002	1, 4, 19
0x0003	1, 4, 20
0x0004	1, 5, 17
0x0005	1, 5, 18
0x0006	1, 5, 19
0x0007	1, 5, 20
0x0008	1, 6, 17
0x0009	1, 6, 18
0x000A	1, 6, 19
0x000B	1, 6, 20
0x000C	1, 7, 17
0x000D	1, 7, 18
0x000E	1, 7, 19
0x000F	1, 7, 20
0x0010	1, 8, 17

0x0011	1, 8, 18
0x0012	1, 8, 19
0x0013	1, 8, 20
0x0014	1, 9, 17
0x0015	1, 9, 18
0x0016	1, 9, 19
0x0017	1, 9, 20
0x0018	1, 10, 17
0x0019	1, 10, 18
0x001A	1, 10, 19
0x001B	1, 10, 20
0x001C	1, 11, 17
0x001D	1, 11, 18
0x001E	1, 11, 19
0x001F	1, 11, 20
0x0040	2, 4, 17
0x0041	2, 4, 18
0x0042	2, 4, 19
0x0043	2, 4, 20
0x0044	2, 12, 17
0x0045	2, 12, 18
0x0046	2, 12, 19
0x0047	2, 12, 20
0x0050	2, 13, 17
0x0051	2, 13, 18
0x0052	2, 13, 19
0x0053	2, 13, 20
0x0054	2, 14, 17
0x0055	2, 14, 18
0x0056	2, 14, 19

0x0057	2, 14, 20
0x0058	2, 15, 17
0x0059	2, 15, 18
0x005A	2, 15, 19
0x005B	2, 15, 20
0x00C0	3, 4, 17
0x00C1	3, 4, 18
0x00C2	3, 4, 19
0x00C3	3, 4, 20
0x00D8	3, 16, 17
0x00D9	3, 16, 18
0x00DA	3, 16, 19
0x00DB	3, 16, 20

Description Table for Cross References

Cross Reference	Textual Quality Words	Description
1	Bad	The Value is not useful.
2	Uncertain	The quality of the value is uncertain.
3	Good	The quality of the value is good.
4	Non-specific	There is no specific reason for the quality state.
5	Configuration error	There is some server specific problem with the configuration. For example, the item in question has been deleted from the configuration.
6	Not connected	The input must be logically connected to some entity but it is not. This quality may reflect that no value is available at this time. For example, the value may not have been provided by the data source.
7	Device failure	A device failure has been detected.
8	Sensor failure	A sensor failure has been detected. (Numbers 17 through 20 in this table may provide additional diagnostic information in some situations.)

9	Last known value	The communication has failed. However, the last known value is available. The "age" of the value may be determined from the time stamp in the OPCITEMSTATE.
10	Comm failure	The communication has failed. There is no last known value available.
11	Out of service	The block is offscan or otherwise locked. This quality is also used when the active state of the item or the group containing the item is inactive.
12	Last usable value	Whatever device was writing this value has stopped doing so. The returned value should be regarded as "stale." This differs from Bad quality (No. 1) with Last Known Value (No. 9). The later status is associated specifically with a detectable communications error on a "fetched" value. The former status is associated with the failure of some external source to "put" something into the value within an acceptable period of time. The "age" of the value can be determined from the TIME STAMP in OPCITEMSTATE.
13	Sensor not accurate	Either the value has "clamped" at one of the sensor limits (related to No. 18 or 19) or the sensor is out of calibration through some form of internal diagnostics (related to No. 17).
14	Engineering units exceeded	The returned value is outside the limits defined for this parameter. In this case (per the Fieldbus Specification) the Limits descriptions (No. 17 - 20) indicate which limit has exceeded but they do not necessarily imply that the value cannot move farther out of range.
15	Sub-normal	The value is derived from multiple sources and has less than the required number of good sources.
16	Local override	The value has been overridden. Typically, this means that the input has been disconnected and a manually entered value has been "forced. "
17	Not limited	The value is free to move up or down.
18	Low limited	The value has "clamped" at some lower limit.
19	High limited	The value has "clamped" at some high limit.
20	Constant	The value is a constant and cannot move.

Using the Log Viewer

Error messages are created by the Communication Driver and logged by the Logger. You can view these messages with the Log Viewer. The Log Viewer help files explain how to view messages and how to filter which messages are shown.

Log Flags are categories of messages. The Log Flag Editor User Guide contains an explanation of the categories. Using the Log Flag Editor, you can specify which log flags the Communication Driver creates.

Note: Generating large numbers of diagnostic messages can impact Communication Driver performance. You should not run in production with any more flags than those set when the Communication Driver is installed. To troubleshoot you can turn on more flags, but there is a performance impact. For more information, see the *Log Flag Editor User Guide*.

To open the Log Flag Editor:

1. In the OCMC, expand **Log Viewer**, and then expand the log viewer group.
2. Select **Local**.
3. On the Action menu, click **Log Flags**.

In general, look at error and warning messages to determine if a problem exists. To determine whether the Communication Driver is communicating with a device, you can enable the DASSend and DASReceive log flags. From these you can determine whether or not the device is responding.

Basic Log Flags

These are the basic log flags for all AVEVA components.

Error
A fatal error, the program cannot continue. By default set on by logger.

Warning
The error is recoverable. A client called with a bad parameter, or the result of some operation was incorrect, but the program can continue. By default set on by logger.

Start-Stop
Each main component logs a message to this category as it starts and stops.

Info
General diagnostic messages.

Ctor-Dtor
C++ classes of interest log messages to this category as they are constructed and destructed.

Entry-Exit
Functions of interest log messages to this category as they are called and return.

Thread Start-Stop
All threads should log messages to this category as they start and stop.

Communication Driver Log Flags

Messages created for these log flags are for Communication Driver common components and contain information about internal Communication Driver activities.

DACmnProtFail
Some failure occurred in the common components while sending a message, updating an item, or otherwise moving data. Typically, this represents some unexpected behavior in the server-specific DLL.

DACmnProtWarn

Some problem occurred that interfered with sending messages, updating items, or otherwise moving data. Common examples are slow poll, value limiting during type conversion, and transaction timeout messages.

DACmnTrace
Normal processing of client program requests and data movement to and from the server-specific DLL are traced on this log flag. Use this in conjunction with DACmnVerbose to get the most information.

DACmnVerbose
Many log flags used by the DAS common components are modified occasionally by DACmnVerbose. When DACmnVerbose is set, the logging of messages on other log flags includes more information.

DACmnSend
Operations that revolve around sending messages to the server-specific DLL.

DACmnReceive
Events surrounding messages that include blocking and unblocking of hierarchies, that are returned by the server-specific DLL.

Communication Driver-Device Interface Log Flags

Messages created for the following log flags are specific to an individual Communication Driver and contain information about communications between the Communication Driver and device.

DASProtFail
An error in the protocol occurred, for example, device disconnected. The program can continue, and, in fact, this category is expected during normal operation of the program. Must be set on by the generic DAS code when the Communication Driver starts.

DASProtWarn
Something unexpected occurred in the protocol, for example, a requested item with an otherwise valid item name is not supported by this device. Must be set on by the generic DAS code when the Communication Driver starts.

DASTrace
General diagnostic messages of a protocol-specific nature. For example, you can provide the number of items in a message for a specific protocol, then optimize based on the number.

DASVerbose
Modifies all other DAS logging flags. When on, provides detailed messages.

DASSend
Protocol messages sent to the device are logged to this category.

DASReceive
Protocol messages received from the device are logged to this category.

DASStateCat1, DASStateCat2, DASStateCat3, DASStateCat4
These are general categories for use by the server developer. As DeviceEngine-generated state machines are created by the Communication Driver, they can be told to log state machine messages to one of the following: DASStateCat1, DASStateCat2, DASStateCat3, or DASStateCat4. These messages indicate when a state is made the active state, when a state handler is run, when a state handler completes, and when a timeout occurs for a state machine.

DASStateMachine

By default, DeviceEngine-generated state machines created by the Communication Driver log to this category unless specifically told to log to one of the DASStateCatN categories. In addition, general state machine messages are logged to this category. These messages indicate when a state machine is created and deleted.

Using the Windows Tools

Windows has two tools that may be useful in troubleshooting performance problems.

You can find quick verification that the Communication Driver process is running by looking at the Windows Task Manager. It also provides information on the user, CPU, and memory usage of the processes.

If you need more information, or need to gather data while not logged in, you can use the Performance and Alerts application. The Performance application is one of the administrative tools found in the Windows Control Panel. For more information, see the Microsoft Management Console (MMC) help files on the Performance application.

OPC Connectivity, DCOM, Windows Firewall, and Anonymous Access

The OPC connectivity between two computers relies 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.

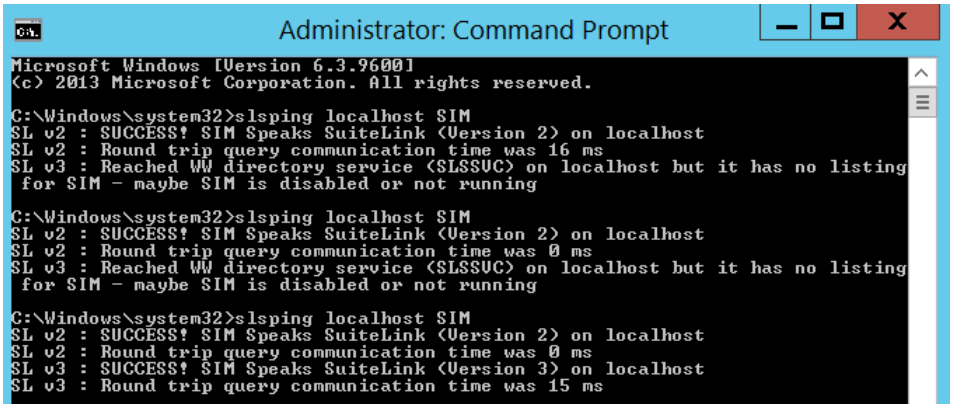
Solution: As an initial test, disable the firewall on either or both the OPC client and OPC server computers. If the problem is resolved by disabling the firewall, revert to enabling the setting and confirm that the DCOM port 135 inbound and outbound rules are configured in the firewall and are set to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

SuiteLink Troubleshooting

To know which communication is enabled in the machine:

Open the Command Prompt as an Administrator and type the below command and press Enter.

```
slsping <machine name> <Communication Driver name>
```



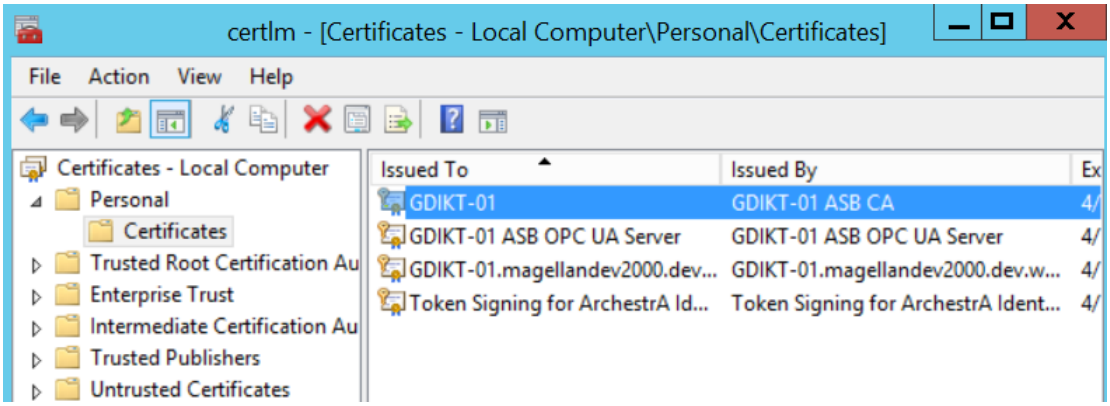
SL V2 SUCCESS indicates unsecure communication is enabled.

SL V3 SUCCESS indicates secure communication is enabled.

If you want the communication secure, then you need to change the SMS configuration in the Configurator. Once you change the configuration in the configurator, you need to deactivate and reactivate the communication driver and then run the command in the Command Prompt to check if the communication has changed to secure.

To know which certificate is configured:

SuiteLink uses the same SMS certificate which is configured through the Configurator. You can find the certificate by searching certlm.msc to open the certlm window.



AVEVA

Chapter 13

Managing Security for Communication Drivers

General Considerations for Security

Before you review the information in this section, it is recommended that you go through the following checklist to ensure you plan to cover the security areas that apply to your ICS and organization.

Security Area	**Reference Section**
Physical and virtual access to the host	General Guidelines for Securing the Host
Latest Windows patches applied	Windows Updates
Protecting the host from viruses and malware	Scanning the Host
Access to content on the host	Protecting the Applications and Content on the Host
Securing your network	Securing the Network
Configuring services and ports	Managing Network Services and Ports
Securing client/server communication	Securing Communication between the Client and Server
User and group management	Securing Systems through Authentication and Authorization

Security Area	Reference Section
Planning for emergencies	Contingency Planning

For a list of security feature help topics refer to the table at the end of this section.

Introduction

This appendix provides a general overview on how to securely deploy your AVEVA software product as an Industrial Control Systems (ICS) application.

This appendix is not meant to be comprehensive, and it does not provide any detailed instructions. It is only a collection of basic concepts and recommendations that you can use as a checklist to secure your own systems. If you need help with a specific item in this guide, see the official documentation for that item -- for example, if you need help with your anti-virus software, see the documentation for that software.

AVEVA's approach to securing site networks and ICS software is driven by the following principles:

- View security from both Management and Technical perspectives
- Ensure that security is addressed from both IT and ICS perspectives.
- Design and develop multiple network, system and software security layers.
- Ensure industry, regulatory and international standards are taken into account.
- Aim to prevent security breaches, supported by detection and mitigation.

These principles are realized by implementing the following security recommendations:

- Prevent security breaches using the following components:
  - Firewalls
  - Network-based intrusion prevention/detection
  - Host-based intrusion prevention/detection
- Segregate IT and Plant networks
- Include a clearly defined and clearly communicated change management policy. For example, firewall configuration changes.

Note: AVEVA strongly recommends following the guidelines prescribed by the U.S. Department of Commerce for securing ICS software. The document "Guide to Industrial Control Systems (ICS) Security" [NIST Special Publication 800-82 Revision 2] (https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf) provides detailed information about ICS, typical system topologies, security threats and vulnerabilities, and recommendations for implementing security measures.

Securing the Host

Given the sensitive nature of industrial control, it is important to secure not only the ICS software, but also:

- the host on which it runs
- the network to which it is connected
- the hardware used for the ICS software.

Note: The "host" is the Windows computer or Windows Embedded device on which your ICS software is installed and running.

There are several factors to consider for securing the host including:

- Access to the host
- Keeping track of and applying the latest Windows updates
- Keeping the host computer free of viruses and malware
- Protecting the applications and content on the host

Each of these factors is covered in the sections below.

General Guidelines for Securing the Host

Here are a few guidelines to secure the host:

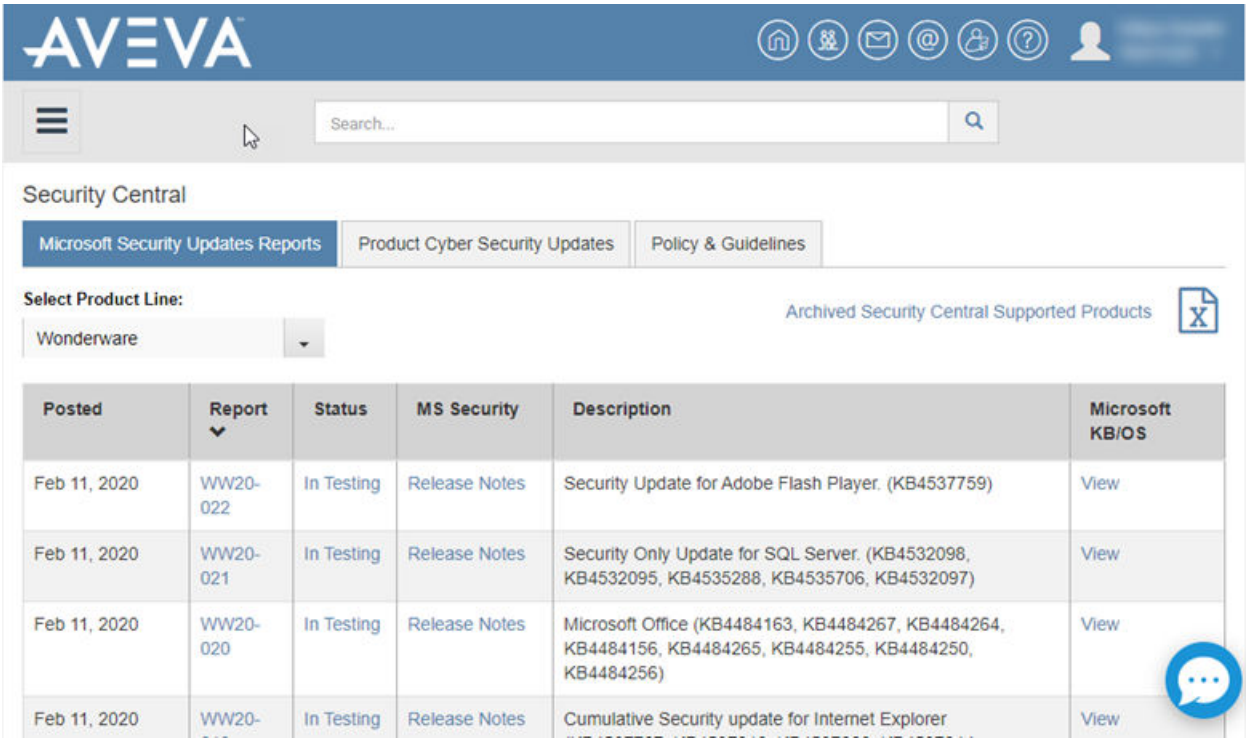
- Use an account with administrative privileges to install the ICS software, and one without administrative privileges to run the ICS software.
- Restrict configuration of ICS to a limited set of users.
- Consider running the ICS software as a Windows service, if that option is available. If the ICS software is run as a service, run it as a low privileged virtual service account.
- Once the host is fully configured and placed in its permanent location, restrict physical access and remote access to it so that only authorized personnel (for example, system administrators, application engineers, run-time operators) can use it.
- Consider disabling or removing physical ports (for example, USB, memory card) that might be used to connect external storage devices and then transfer data.

Windows Updates

Check that the Windows operating system on the host is a version that is under what Microsoft calls "mainstream support", which means Microsoft actively maintains and releases updates for it. Older versions of Windows are under Microsoft "extended support", which means they are not actively maintained and therefore might become vulnerable without notice. For more information about the different versions of Windows and the different levels of support, see [Windows lifecycle fact sheet](https://support.microsoft.com/en-us/help/13853/windows-lifecycle-fact-sheet).

Automate Microsoft product updates using Microsoft Windows Server Update Services (WSUS), which enables you to manage and distribute updates to computers on your network. For more information about WSUS, see [Windows Server Update Services](https://docs.microsoft.com/en-us/windows-server/administration/windows-server-update-services/get-started/windows-server-update-services-wsus). If the host does not or will not have a reliable connection to the WSUS server, perhaps because it is located on a private network, you can either develop a procedure to manually apply updates or consider changing the operating system to a Long-Term Servicing Channel (LTSC) version of Windows, which is updated less frequently.

In addition, AVEVA ICS software is tested for compatibility with Microsoft updates the results of which are published on the Security Central site (https://softwaresupportsp.aveva.com/#/securitycentral). Security advisories and bulletins are also published on this site.



ICS Software Updates

Check that the ICS software on the host has all the recommended patches and hot fixes installed.

Some AVEVA applications release regular updates, which should be applied as soon as possible as they may contain security-related fixes.

Note: AVEVA's Global Customer Support (GCS) group publishes a Technology Matrix (https://gcsresource.aveva.com/TechnologyMatrix/Home/Index) for AVEVA software products. This matrix lists the Windows operating system versions against which a software product has been tested for compatibility. In addition, it lists compatible runtime, browser, and virtualization environments for the software. It also includes a list of other products that can be installed on the same computer and lists other products with which this software can communicate.

Scanning the Host

Use both anti-virus and anti-malware software and file integrity checking software to regularly scan the host.

Windows includes Windows Defender by default, but you may choose to install and use additional software that scans for more types of malware or performs other functions. If you do that, make sure the software is provided by a reputable company. And, as with the operating system, if the host does not or will not have reliable access to the software's update service, develop a procedure to manually apply updates. If you develop a manual update procedure, it should account for all devices on a network or at a site, because a single outdated device can leave the entire network or site vulnerable.

Protecting the Applications and Content on the Host

To protect the applications and content on the host:

- Enable Windows Firewall, and configure it to close all ports that are not used by the ICS software. For more information about port usage, see Managing Network Services and Ports.
- Disable Windows features like remote desktop and file sharing, and remove unnecessary programs like games and social media.
- Restrict access to the files, databases, registry and other resources on the host.
- Use Windows BitLocker to encrypt the hard drive of computers that are either mobile or not located in a secure facility. However, BitLocker may impact the performance of computers.
- Consider using server-class storage (SANs) infrastructure to avoid storing sensitive data on mobile devices.
- If your application stores data in SQL Server, Windows authentication can provide better application security than SQL Authentication. If you switch from Windows Authentication to SQL Authentication, a pop up dialog will appear recommending that you use Windows Authentication for this reason. If you choose to ignore this warning and proceed with SQL Authentication, click **OK**. A similar message will be logged in the OCMC (SMC) Log Viewer.

AVEVA leverages the security built into the Windows operating system to store and manage encryption keys. The encryption keys are stored in a local storage location called the encryption store. For more information about the Windows encryption store, refer to the Microsoft documentation, located at Certificate Stores (https://docs.microsoft.com/en-us/windows-hardware/drivers/install/certificate-stores).

Phases of Data Protection

Data exists in three different phases, and protection must be provided for each phase:

- At rest
- In transit
- In use

Data at rest

Data at rest is data that is not currently being used or accessed, such as data stored on a hard drive, laptop, flash drive, RAID array, network attached storage (NAS), storage area network (SAN), or archived/stored in some other way. Data protection at rest aims to secure inactive data stored on any device or network. For protecting data at rest, you can simply encrypt sensitive files prior to storing them and/or choose to encrypt the storage drive itself. BitLocker Drive Encryption, which you can invoke via the Windows Control Panel, can be used to invoke whole-drive encryption.

In the context of SCADA and ICS systems, data at rest includes stored configuration data, historical data, backups, and other static data. The duration of storage, that is, long term or short term, does not impact this classification of data at rest. Protection for data at rest is applicable for as long as the data exists in this condition; it is not a fixed condition.

Proper authorization rights need to be set in place to ensure the data is not being viewed by unauthorised users. Other steps can also help, such as storing individual data elements in separate locations, such as a corporate-approved offline backup to decrease the likelihood of attackers gaining enough information to commit fraud or other crimes. Offline backups are the best mitigation against the threat of ransomware.

Data in transit

Data in transit, or data in motion, is data that is actively moving from one location to another.

In the context of SCADA and ICS systems, this encompasses deploying a project to a run-time node, transmitting process variables, VTQ data, and other data that is sent between nodes in a running, production system. This includes alerts and alarms.

Data protection in transit is the protection of this data while the data traveling, including the following examples:

- From node to node within a network
- From network to network
- Accessed via internet
- Transferred from a local storage device to a cloud storage device

Wherever data is moving, effective data protection measures for in-transit data are critical as data is often considered less secure while in motion. Best security practice is to ensure TLS 1.2 encryption is used for all communications using the HTTPS protocol.

Data in use

Data in use refers to data that is being processed or accessed either locally or remotely. This generally involves placing data into memory (RAM) for access and processing by applications and users, potentially multiple users across different computers, mobile devices, remote terminals or other device. Data in use is particularly vulnerable to attack. To protect data in use, encryption, user authentication, and identity management is highly recommended.

In the context of SCADA and ICS systems, data in use can apply to databases, such as those used actively by a historian or deployed to a run-time node. This needs to be safeguarded by a secure transfer channel.

Configure Encryption in SQL Server

We recommend you enable encrypted connections for SQL Server. You enable encrypted connections for an instance of the SQL Server Database Engine and use SQL Server Configuration Manager to specify a certificate. The server computer must have a certificate provisioned. To provision the certificate on the server computer, you import it into Windows. The client machine must be set up to trust the certificate's root authority.

SQL Server can use Transport Layer Security (TLS) to encrypt data that is transmitted across a network between an instance of SQL Server and a client application. The TLS encryption is performed within the protocol layer and is available to all supported SQL Server clients.

Enabling TLS encryption increases the security of data transmitted across networks between instances of SQL Server and applications. However, when all traffic between SQL Server and a client application is encrypted using TLS, the following additional processing is required:

- An extra network round trip is required at connect time.
- Packets sent from the application to the instance of SQL Server must be encrypted by the client TLS stack and decrypted by the server TLS stack.
- Packets sent from the instance of SQL Server to the application must be encrypted by the server TLS stack and decrypted by the client TLS stack.

Certificate Requirements

For SQL Server to load a TLS certificate, the certificate must meet the following conditions:

- The certificate must be in either the local computer certificate store or the current user certificate store.

- The SQL Server Service Account must have the necessary permission to access the TLS certificate.
- The current system time must be after the **Valid from** property of the certificate and before the **Valid to** property of the certificate.

Install on a Server

With SQL Server 2019 (15.x), certificate management is integrated into the SQL Server Configuration Manager. SQL Server Configuration Manager for SQL Server 2019 (15.x) can be used with earlier versions of SQL Server.

If using SQL Server 2012 (11.x) through SQL Server 2017 (14.x), and SQL Server Configuration Manager for SQL Server 2019 (15.x) is not available, follow these steps:

1. On the **Start** menu, click **Run**, and in the **Open** box, type **MMC** and click **OK**.
2. In the MMC console, on the **File** menu, click **Add/Remove Snap-in**.
3. In the **Add/Remove Snap-in** dialog box, click **Add**.
4. In the **Add Standalone Snap-in** dialog box, click **Certificates**, click **Add**.
5. In the **Certificates snap-in** dialog box, click **Computer account**, and then click **Finish**.
6. In the **Add Standalone Snap-in** dialog box, click **Close**.
7. In the **Add/Remove Snap-in** dialog box, click **OK**.
8. In the **Certificates** snap-in, expand **Certificates**, and then expand **Personal**.
9. Right-click **Certificates**, point to **All Tasks**, and then click **Import**.
10. Right-click the imported certificate, point to **All Tasks** and then click **Manage Private Keys**. In the **Security** dialog box, add read permission for the user account used by the SQL Server service account.
11. Complete the **Certificate Import Wizard**, to add a certificate to the computer, and close the MMC console. For more information about adding a certificate to a computer, see your Windows documentation.

Export Server Certificate

To export the server certificate:

1. From the **Certificates** snap-in, locate the certificate in the **Certificates / Personal** folder.
2. Right-click the **Certificate**, point to **All Tasks**, and then click **Export**.
3. Complete the **Certificate Export Wizard**, storing the certificate file in a convenient location.

Configure Server

Configure the server to force encrypted connections. The SQL Server service account must have read permissions on the certificate used to force encryption on the SQL Server. For a non-privileged service account, read permissions will need to be added to the certificate. Failure to do so can cause the SQL Server service restart to fail.

To configure the server:

1. In **SQL Server Configuration Manager**, expand **SQL Server Network Configuration**, right-click **Protocols for <server instance>**, and then select **Properties**.
2. In the **Protocols for <instance name> Properties** dialog box, on the **Certificate** tab, select the desired certificate from the drop-down for the **Certificate** box, and then click **OK**.
3. On the **Flags** tab, in the **ForceEncryption** box, select **Yes**, and then click **OK** to close the dialog box.

4. Restart the SQL Server service.

Configure Client

Configure the client to request encrypted connections.

1. Copy either the original certificate or the exported certificate file to the client computer.
2. On the client computer, use the **Certificates** snap-in to install either the root certificate or the exported certificate file.
3. Using SQL Server Configuration Manager, right-click **SQL Server Native Client Configuration**, and then click **Properties**.
4. On the Flags page, in the **Force protocol encryption** box, click **Yes**.

Note: The sections in this topic have been derived from the Microsoft documentation. For more information, refer to the topic 'Enable encrypted connections to the Database Engine' in Microsoft Documentation.

Securing the Network

Usually, the host computer will have some sort of network access; it is increasingly rare for an ICS device to run as an entirely standalone device. The host may use the network to communicate with other ICS components such as controllers, sensors, databases, remote clients, and even other hosts in peer-to-peer relationships. You may also use the network to manage several ICS devices from a development or supervisory computer.

Once you determine that the host will have network access, decide how it will connect to the network. In recent years there has been a shift from wired networks (that is, "Ethernet") to wireless networks ("Wi-Fi"), even for business and industrial uses. We recommend against using Wi-Fi for your ICS network because you do not have physical control over who or what might access the network. Any computer or device within range of the Wireless Access Point (WAP) can try to access the network, and even if the network is ostensibly secure, an intruder can intercept and analyze network traffic and potentially discover a vulnerability.

Nevertheless, if you decide to use Wi-Fi for your ICS network, enable all access control features on the WAP including encryption (for example, WPA/WPA2), a strong password and a list of authorized MAC addresses. Do not try to "hide" the Wi-Fi network by disabling broadcast of the Service Set Identifier (SSID), because doing so actually generates more network traffic that can be intercepted and analysed.

Segmenting the ICS Network

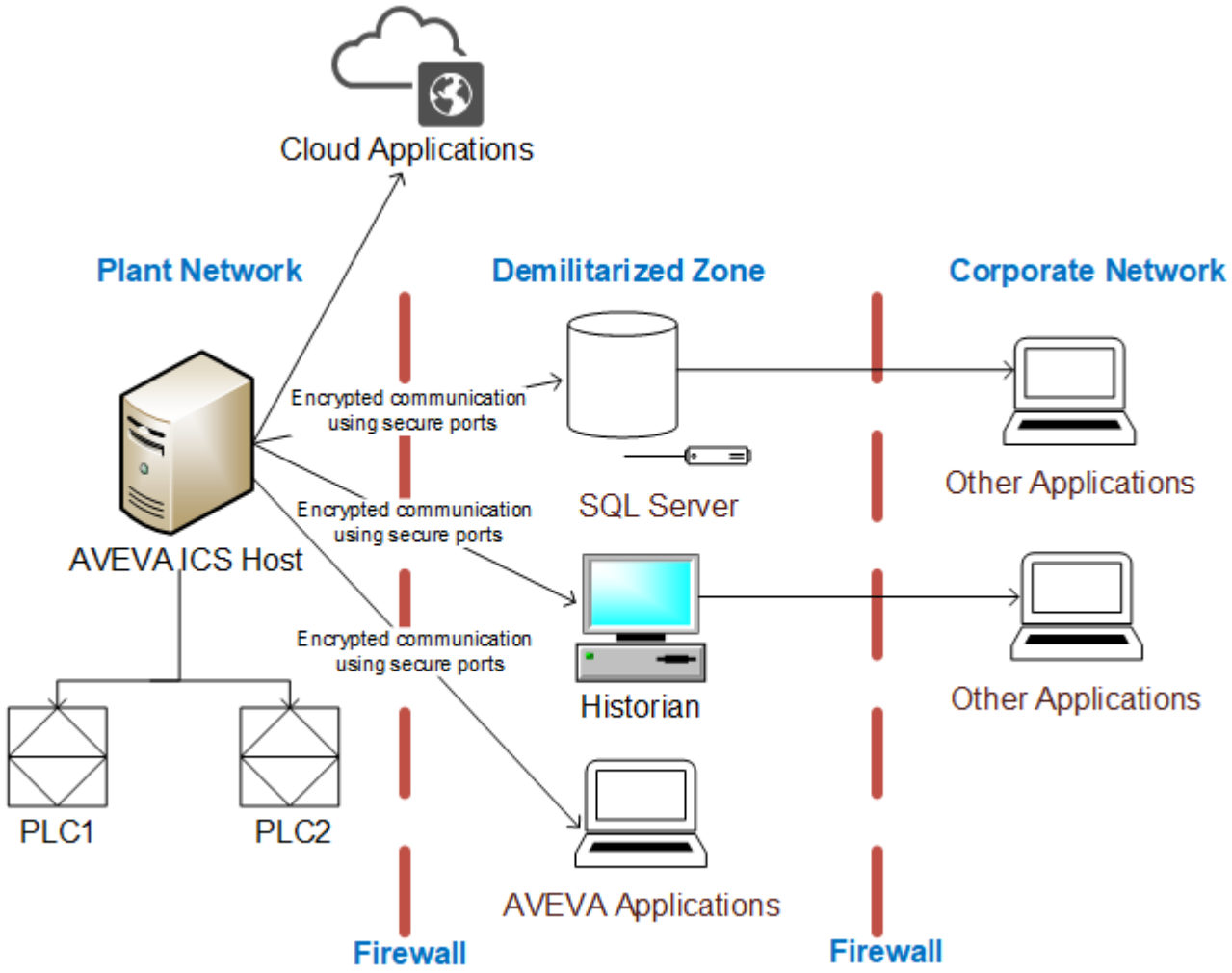
The ICS network itself can be either physically or logically segmented from your other corporate networks. A physically segmented network is by definition the most secure. The network hardware and all computers and devices connected to it form a single closed network with no physical connection to any other network, so an intruder cannot access the network unless they also have access to the physical location.

In contrast, a logically segmented network is physically connected to your other corporate networks and/or the public internet, but it uses various methods to segregate ICS network traffic from other network traffic. This may include:

- Using a unidirectional gateway
- Implementing a Demilitarized Zone (DMZ) network architecture with firewalls to prevent network traffic from passing directly between the corporate and ICS networks
- Having different authentication mechanisms and credentials for users of the corporate and ICS networks.

- The ICS should also use a network topology that has multiple layers, with the most critical communications occurring in the most secure and reliable layer.

Given below is a sample deployment topology.



In no case should your ICS network and devices be directly accessible from the public internet. If there is some part of your ICS that you want to be accessible, (for example, if you want be able to view web-enabled HMI screens from a browser or smart phone), your ICS software should include features that securely pass the necessary traffic between your ICS network and a public-facing server.

Managing Network Services and Ports

A network port is an endpoint of communication in an operating system. While the term is also used for hardware devices, in software it is a logical construct that identifies a specific process or a type of service. In other words, a network port is conceptually different from hardware ports like USB, memory card, and even the wired network connection.

Computers and devices can access many different network services at the same time by communicating on different network ports. Each network service or communication protocol has an associated port number. Some port numbers are specified by international standards, and therefore they are universally recognized. Other port

numbers are claimed by proprietary software, and in most cases they can be changed in the software settings if there is a conflict with other software or services.

Firewalls control network traffic by either accepting or refusing communication on these network ports. If a port is open, it accepts communication, and if a port is closed, it refuses communication. Almost every layer of a network -- from the operating system on an individual computer or device, to the router that manages traffic within a network, to the gateway that manages traffic between networks -- has its own firewall.

The documentation for your ICS software should include a list of network ports that are commonly used by the software. Given the nature of ICS, the list typically includes services like web, email, file transfer, external databases, device drivers, and the ICS software itself for server-client communications. Configure the firewalls to open only those network ports that are actually used by your ICS. Disable all unused services and close all unused ports.

Securing Communication between the Client and Server

Like most server-client applications, your ICS software should support secure communication between the server and client in order to prevent the messages sent between those two stations from being read by any other stations on the same network. Note that this is different from securing the network itself in order to prevent unauthorized access to the network.

This sort of communication is also sometimes known as "Encrypted Channel" because it uses the Transport Layer Security (TLS) standard to encrypt the server-client messages. The latest version of the standard is TLS 1.3 (released August 2018), but it is not yet in common use. The latest version of the standard in common use is TLS 1.2 (released August 2008). TLS supersedes the earlier Secure Sockets Layer (SSL) standard, although SSL is still used in older applications.

Certificates

TLS and SSL use a system of certificates and keys to digitally "sign" the messages sent between the server and client. When the server establishes communication with the client (and vice versa), it presents its certificate which identifies its name, network address, organization, physical location, and so on. The client can then choose to either accept or refuse the certificate as presented. If it accepts the certificate, it agrees to accept messages encrypted with the same certificate, and it uses the associated key to decrypt those messages.

When you configure this sort of communication, you need to choose one of the following:

- Using self-signed certificate
- Using certificates signed by a Public Certificate Authority (CA)
- Using Domain-issued certificates or certificates signed by a Private Certificate Authority using systems like Microsoft Active Directory Certificate Service (AD CS)

A self-signed certificate is issued and signed by the same application that presents it. Self-signed certificates are easy to create and manage, but they are secure only if you control both the server and the client and therefore control which certificates are installed on each.

In contrast, CA-signed certificates are slightly difficult and expensive to acquire, but they are more flexible than self-signed certificates because you do not need to control both the server and the client. If you configure the server to present a CA-signed certificate, the client will accept the certificate because it recognizes the Certificate Authority.

Domain-issued certificates are internal certificates typically managed by your IT department. They are issued and validated by an Active Directory Certificate Authority. Domain-issued certificates are free and can be issued instantly.

You need to renew CA-signed and Domain-issued certificates at regular intervals.

For more information about how to enable Encrypted Channel features and manage self-signed certificates in your ICS software, see the documentation for that software. However, acquiring a CA-signed certificate and then using it to sign other certificates is typically beyond the scope of ICS software documentation.

Note: Encrypted and unencrypted communications typically use different network ports.

Cloud-based Systems

It is possible that your ICS software might access cloud-based solutions, or might itself be hosted on the Cloud. It is important to mitigate the risks associated with cloud-based access and hosting.

Accessing Cloud-Based Solutions

Several AVEVA applications are now being made available through the Cloud, and ICS software may need to connect to these applications. One of the main risks associated with accessing cloud-based applications is unauthorized access. Connecting ICS software to Cloud solutions must be done in a secure manner, and needs to use secure protocols such as Transport Layer Security (TLS).

It is important that data integrity is maintained at all times. Use data classification to identify data that is sensitive and data that can be made public. Secure computers, storage and networking in order to secure the data that is stored and transmitted. Work with your Cloud Service Provider (CSP) to configure users, assign access levels and monitor and control access. Ensure that the CSP's buildings are physically secure and protected from unauthorized access.

Cloud-based ICS Software

While hosting ICS software on the Cloud provides several benefits such as flexibility, scalability and availability, it is also fraught with security risks such as susceptibility to hacking resulting in damage to the organization's reputation . Therefore, it is important to implement a security strategy before you make your ICS software accessible on the Cloud. For securing ICS software on the Cloud, you need to consider the following:

- Securing access points by putting in place authentication, monitoring and support mechanisms.
- Implementing cloud-based, centralized security measures including encrypting communications using TLS.

Note: It is recommended that you review the NIST Cybersecurity Framework (https://www.nist.gov/cyberframework) for additional information.

Securing Systems through Authentication and Authorization

Typically, ICS software is comprised of a large number of systems, each accessed by a variety of users including engineers, operators and managers. The level of access that each type of user requires is different. So, it is necessary to manage user authentication and authorization to secure the system.

Authentication

Authentication is the process of verifying a user's/system's identity. Authentication can be managed in the following ways:

- Within the ICS software through application accounts
- Through Windows accounts, which can be local to a single computer
- Through Authentication systems (see the next section for details)

While ICS software allows for user and role management, it can become cumbersome and complicated to manage a large number of user accounts as employees and roles change. Because of this, use of Windows accounts is generally preferred.

Authentication Systems

Authentication systems such as Active Directory and Lightweight Directory Access Protocol (LDAP), referred to as authentication servers, are a repository of and provide centralized management for all system accounts and individual user accounts. An authentication protocol is used for all communication between authentication servers and the user or server requesting authentication.

Even though use of authentication systems provides improved scalability, the following factors must be considered depending upon the size and complexity of your operations:

- It is important that the authentication servers are highly secured.
- The authentication server system creates a single system for managing all system accounts. Therefore, it requires to be available at all times. To ensure minimal disruption during an emergency, redundancy must be considered.
- Permit caching of user credentials only for users who have authenticated their identity recently.
- Networks that support the authentication protocol must be reliable and secure to assist in trouble-free authentication.

It may also be worthwhile implementing two-factor authentication using additional applications such as PingID.

Authorization

Authorization is the process of providing the correct level of privileges to users by applying access rules to authenticated users, systems (HMIs, field devices and SCADA servers) and networks (remote sites' LANs).

Managing Users and Groups through Windows

When you configure security, you may choose to do one of the following:

- Keep the configuration local to a single application.
- Share the configuration between multiple applications.
- Manage the configuration as part of the network domain (for example, using Active Directory). This option typically allows users to have the same user account for the network, the host, and the ICS software. Using Active Directory gives you the following advantages:
  - A centralized repository for user and group data, enabling effective implementation of security policies and procedures.
  - Provides a single point of access to all network resources after the user is identified and authenticated.

To manage users and groups:

- First define a specific role for each group, and then configure the group privileges to fit that role.

- Groups may overlap, but it is often better to have clearly separate groups and then assign individual users to multiple groups, if necessary.
- Set or change the password for the ICS software's default user (e.g., "guest").
- Define stringent password policies to force users to create strong passwords. Enforce mandatory password updates on a regular basis.

Managing Users and Groups through ICS Software

Your ICS software should have a built-in security system that controls who may use the software and what privileges they have.

Users should be assigned permissions that determine what each user is authorized to do within the ICS system. Permissions can be managed either on a per-account basis or on a group basis by making use of roles. Group or role-based access control is preferred as it greatly simplifies management. Users can be moved from one role to another as the organization's needs change, and can also be members of multiple roles if required.

Each user should have their own user account with a unique user name and a strong password. The user account can then be assigned to one or more groups.

Accounts should always be assigned the least privileges necessary to perform their functions. Accounts with Windows Administrator permissions should be reduced to the minimum, and typically only used to install and configure the software. Likewise, accounts with SQL Server SysAdmin privileges should be reduced to the minimum, and typically only used to install and configure the software.

In most cases, the ICS software will allow associating Windows Groups with roles within the product. While defining and assigning roles, consider the following:

- Roles should be defined to have the least privileges necessary for their functionality.
- Roles should be limited to a single purpose in order to simplify the permissions assigned to them.
- Users can be members of multiple roles if necessary.

Contingency Planning

Incidents are inevitable. It is, therefore, important to develop a strategy to detect an incident quickly and respond to it in a timely manner in order to minimize loss and protect your system. An organization must consider contingencies arising from incidents such as fire, flood and so on, and those arising from failure of hardware or software components. Cyber attacks such as ransomware are becoming more common and must also be considered.

An organization should have contingency plans in place to cover the entire range of failures and eventualities. Employees should be trained and be familiar with the contents of the contingency plans.

As part of planning for contingencies, it is important to establish a site, physically separated from the central one, that has replication capability. Doing so will ensure the integrity of an operational system where the central site is at risk from fire, floods or other disasters. The replication capability includes having duplicated hardware, and requires software configuration and key state information to be periodically propagated from the central site to the recovery site. Each recovery scenario is unique, so it is important to consult with system integration experts regarding the design of communications equipment, hardware and the configuration of the software.

Protecting the data stored in your system is also of paramount importance. Full and incremental backups must be scheduled on a regular basis. Backups should be verified by running tests to restore from backed up data. Backups should be stored offline so that they are safe from cyber attacks such as ransomware.

Organizations should also have business continuity and disaster recovery plans that are similar to contingency plans. These plans are covered briefly in the sections to follow.

Auditing and Logging

As part of implementing security for ICS software, it is important to incorporate auditing and logging activities on various systems and networks.

Auditing and logging provide information on the current state of your ICS, and help to ensure that the system is functioning as expected. If an incident occurs, you can use the activity logs to trace the origin of the incident to a computer, user or network. Auditing and logging can also help with troubleshooting issues.

If you are connecting to cloud-based solutions, audit all virtual machines (VMs) to ensure data integrity.

Business Continuity Planning

Business continuity planning addresses strategies to maintain or re-establish production in the event of any disruption. These disruptions may be caused by a natural disaster (flood, earthquake, etc), by an intentional or unintentional man-made event (arson, operator error, power outage, etc), or by system failure.

Depending upon the duration of the potential ICS application outage caused by a disruption, operational recovery plans for short-term outages and disaster recovery plans for long-term outages must be formulated. It is also important to employ physical security for areas of a production site that house data acquisition and control systems that might have higher-level risks. Your business continuity plan should specify system and data recovery procedures for your systems. Once the recovery procedures are documented, a schedule should be developed to test the recovery procedures. Particular attention must be paid to the verification of backups of system configuration data and product or production data. The procedures should be reviewed periodically.

If you are accessing cloud-based solutions, ensure that systems are available at all times. In case of a disaster, services should switch to a new physical location to provide continued service.

Disaster Recovery Planning

A disaster recovery plan (DRP) is a set of procedures to protect and recover an IT infrastructure in case of a disaster. It contains the procedures to follow before, during and after a disaster. Disasters can be natural, environmental or man-made (intentional or unintentional).

A DRP is essential for continued availability of the ICS, and should cover the following:

- When the DRP should be activated depending upon an event, its duration and its severity.
- Detailed course of action for operating the ICS manually until external connections are secured.
- Personnel responsible for each procedure.
- Processes for securely backing up data and restoring it. This should cover:
  - Requirements for building redundancy.
  - File backup procedures.
  - Frequency of backups.

  - Storage mechanism for full and incremental backups.
  - Safe storage of installation media, license keys and configuration information.
  - List of individuals responsible for performing, testing, maintaining and restoring backups.
- List of personnel with physical and virtual access to the ICS.
- Detailed configuration information about the components of the ICS.
- Schedule for testing the DRP.

Conclusion

Security lapses present a serious threat to ICS software and infrastructure. Therefore, it is important for every organization to:

- Be proactive about preventing security lapses
- Identify potential lapses
- Detect them in a timely manner when they occur
- Address lapses to ensure minimum disruption and maximum availability

To this end:

- Computers and networks must be secured
- Users and groups must be authenticated and authorized
- Contingency plans must be in place to recover from untoward or intentional events

Refer to the document "Guide to Industrial Control Systems (ICS) Security" [NIST Special Publication 800-82 Revision 2](https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf) for additional details and recommendations.

Security Configuration for Communication Drivers

The table below lists the security areas that you need to configure for Communication Drivers and the details of the section(s) in this guide that provide the corresponding instructions.

Security Area(s)	**Topic(s) in this guide**	**Summary**
Encrypting SuiteLink communication for secure connection.	Secure SuiteLink Connection	You can encrypt SuiteLink communication between a SuiteLink server and a SuiteLink client.
Enabling security in the communication.	Configuring the System Management Server	Incorporates security measures, including support for the TLS 1.2 protocol for secure encrypted communications between nodes, single sign on (SSO), and certificate management.

Security Area(s)	Topic(s) in this guide	Summary
Establishing secure connection with MQTT Broker	Configuring an MQTT Data Source Connection	Incorporates security measures, including support for the TLS 1.3 protocol for secure encrypted communications between MQTT Broker and MQTT Communication Driver.
Authenticating the Connection using SNMP V3.	Authenticating the Connection	Allows you to configure the security settings for every station.
Setting the OPC UA connection security parameters:	Configuring an OPC UA Data Source Object	Allows you to update Security Policy, Security Message Mode, and User Credentials.