EcoStruxure™ Control Expert

Modicon Communication Server

User Guide

EIO0000004083.03
06/2021

www.se.com

Legal Information

Table of Contents

Safety Information

Important Information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

Before You Begin

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠WARNING

UNGUARDED EQUIPMENT

- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

Start-up and Test

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check are made and that enough time is allowed to perform complete and satisfactory testing.

⚠WARNING

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

Operation and Adjustments

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Book

Document Scope

This manual describes the features and use of the Modicon Communication Server software.

NOTE: The specific configuration settings contained in this guide are intended to be used for instructional purposes only. The settings required for your specific configuration may differ from the examples presented in this guide.

Validity Note

This document is valid for Modicon Communication Server V2.01 software.

The technical characteristics of the devices described in the present document also appear online. To access the information online, go to the Schneider Electric home page www.se.com/ww/en/download/.

The characteristics that are described in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the document and online information, use the online information as your reference.

Overview

What’s in This Part

Introduction

What's in This Chapter

General

Overview

The Modicon Communication Server (MCS) allows you to configure and communicate with Modicon PAC Controllers through Wonderware's System Platform Management Console (SMC).

Functions of the Modicon Communication Server

The functions of the Modicon Communication Server include:

- Support tags of all data types (as supported by OPC UA Server Expert)
- Integration of Wonderware's System Platform Management Console
- Provides diagnostic information

Modicon Communication Server Features

What's in This Chapter

Overview

This chapter describes Modicon Communication Server product features.

Modicon Communication Server Components

Software Components

Modicon Communication Server is a software package that can be installed on a Windows™ PC, and consists of the following software components:

- Modicon Communication Server: an OPC UA server that provides OPC UA clients with access to the data of:
  - Modicon PAC family, page 63 – including standalone, safety, and Hot Standby PACs – programmed with EcoStruxure™ Control Expert, and
- Wonderware's System Management Console: used to configure the OPC UA server.

Supported Features

Modicon Communication Server supports:

- OPC United Architecture version 1.03, including:
  - Core Server Facet profile, page 11
  - Data Access Server Facet profile, page 11
  - Transport protocol with UA-Binary encoding
- OPC UA security, including:
  - Message security modes: None, Sign, Sign&Encrypt
  - Security policies: None and Basic256Sha256.
- Authentication , including:
  - Identity Token types: Anonymous IdentityToken, UserName IdentityToken and X509 IdentityToken.

Supported Profiles

Introduction

The primary purpose of the Modicon Communication Server is to provide an OPC UA communication channel between a Modicon PAC and OPC UA clients. The data of the Modicon PAC devices is mapped to variables in the OPC UA server and made available to OPC UA clients.

NOTE: The terms of each connection between an OPC UA client and the OPC UA server are determined by the client, which sets the attributes of the connection between the client and server.

Supported Profile

The Modicon Communication Server supports the **Standard 2017 UA Server Profile**, with restrictions, page 12. For more information, refer tot he OPC Foundation website at: http://opcfoundation.org/UA-Profile/Server/StandardUA2017.

Supported Facets

The Modicon Communication Server supports the following Facets:

- **Server Category > Facets > Core Characteristics**:
  - **Core Server Facet** (http://opcfoundation.org/UA-Profile/Server/CoreFacet), with restrictions, page 12.
- **Server Category > Facets > Data Access**:
  - **Data Access Server Facet** (http://opcfoundation.org/UA-Profile/Server/DataAccess)
- **Security Category > Facets > Security Policy**:
  - **None** (http://opcfoundation.org/UA/SecurityPolicy#None)

    NOTE: The security policy **None** can be enabled, page 38 using the Modicon Communication Server.
  - **Basic256Sha256** (http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256)
- **Security Category > Facets > User Token > General**:
  - **Anonymous** (http://opcfoundation.org/UA-Profile/Security/UserToken/Anonymous)

    NOTE: The user token **Anonymous** can be enabled by using the Modicon Communication Server.
- **Security Category > Facets > User Token > Server**:
  - **User Name Password** (http://opcfoundation.org/UA-Profile/Security/UserToken/Server/UserNamePassword)
  - **X509 Certificate** (http://opcfoundation.org/UA-Profile/Security/UserToken/Server/X509Certificate)

    NOTE: The user token **X509 Certificate** can be disabled by using the Modicon Communication Server.
- **Transport Category > Facets > Client-Server**:
  - **UA-TCP- UA-SC UA-Binary** (http://opcfoundation.org/UA-Profile/Transport/uatcp-uasc-uabinary)

Modicon Communication Server Stack Services

Supported OPC UA Services

The OPC UA server supports the following service sets and services:

Service Set	Services
Attribute	• Read • Write
Method	• Call
MonitoredItem	• CreateMonitoredItems • ModifyMonitoredItems • DeleteMonitoredItems • SetMonitoringMode
SecureChannel	• OpenSecureChannel • CloseSecurechannel
Session	• CreateSession • ActivateSession • CloseSession
Subscription	• CreateSubscription • ModifySubscription • DeleteSubscription • SetPublishingMode • Publish • Republish
View	• Browse • BrowseNext • TranslateBrowsePathToNodeIds

NOTE: For a description of these service sets and services, refer to the document *OPC Unified Architecture Specification Part 4: Services (Release 1.04)*.

Prerequisites, Installing, and Licensing the Modicon Communication Server

What's in This Part

Overview

This part describes prerequisites, installing, and licensing the Modicon Communication Server.

Prerequisites, Installing, and Licensing the Modicon Communication Server

What's in This Chapter

Prerequisites

Accessing the Modicon Communication Server Software

The latest version of the Modicon Communication Server software is available from the website www.se.com/en/download.

Selecting a Host PC for Installation

You have to install the Modicon Communication Server software on the same PC used to host the Wonderware System Platform.

Host PC Software and Hardware Requirements

The software and hardware requirements to install the Modicon Communication Server software are the same as for a Wonderware System Platform.

For more details, refer to the *ReadMe* and *Release* notes.

Installing the Modicon Communication Server

Using the Installation Wizard

After you download the Modicon Communication Server software package from Schneider Electric, follow these steps to install the software using the installation wizard:

Step	Action
1	Verify that the hardware and software requirements are fulfilled, page 15. Navigate to the location on your host PC where you saved the installation files, and double-click **Setup.exe** to run the wizard. **NOTE:** For installations on PCs running Windows version 7 or earlier, you may see a system message indicating "Publisher: Unknown", asking if you want to allow the setup.exe program to make changes to your computer. This message arises because older versions of the Windows operating system will not recognize the certificate provided by Schneider Electric. Click **Yes** to continue with the installation.
2	Click **Next >** to start the installation process.
3	Click **Next >** to view the **ReadMe** and **Release Notes**.
4	After reading the **ReadMe** and **Release Notes**, click **Next >**.
5	In the **License Agreement** screen, read the license agreement carefully and select **I accept the terms in the license agreement** option, and then click **Next >**.

Step	Action
6	In the **Customer Information** screen, enter the required details and click **Next >**.
7	In the **Destination Folder** screen, click **Next >** to install in the selected path or click **Change...** to select a different path for the installation.
8	In the **Ready to Install the Program** screen, click **Install**. The wizard shows how installation is progressing.
9	When installation is complete, click **Finish**.

NOTE: If OPC UA Server Expert is installed in your PC, ensure that the controller IP address is unique between Modicon Communication Server and OPC UA server.

After initial installation, you can use the wizard to repair or uninstall the Modicon Communication Server software installation.

Installed Components

The installation wizard loads the following software components onto the host PC:

- Modicon Communication Server: an OPC Unified Architecture v1.03 compliant server.
- Modicon Communication Server Configuration Service: a Windows™ service for configuring the OPC UA server.
- Modicon Communication Server UI Service: a configurable Windows™ service for System Management Console with restricted features.
- The following administrative tools for certificate management (which require Security Administrator privileges to use):
  - *trustentity*: used to add certificates, or a certificate authority (CA) that issues certificates, to the certificate trust list, page 56.
  - *setcertificate*: used to replace the auto-generated self-signed certificate with a certificate issued by a certificate authority (CA).

Re-Installing or Repairing the Modicon Communication Server

You can use the installation wizard to re-install or repair the Modicon Communication Server software. If you re-install or repair the Modicon Communication Server software:

- If your configuration is using the self-signed certificate, page 55 generated as part of the previous installation, note that a new self-signed certificate will be generated. You will want to add the new self-signed certificate to the certificate trust list.
- If your configuration is using a certificate issued by a certificate authority (CA), page 55, it will need to be re-issued by the CA and re-trusted using the *setcertificate* tool.

Upgrading the Modicon Communication Server

If previous version of Modicon Communication Server exists in your computer, you can use the installation wizard to upgrade to version 2.01. If you upgrade the Modicon Communication Server software:

- The **Device** configuration and **Security** settings will be preserved.
- If your configuration is using the self-signed certificate, page 55 generated as part of the previous installation, note that a new self-signed certificate will be generated. You will want to add the new self-signed certificate to the certificate trust list.

- If your configuration is using a certificate issued by a certificate authority (CA), page 55, it will need to be re-issued by the CA and re-trusted using the *setcertificate* tool.

Product Licensing

Licensing

The current release of Modicon Communication Server does not require a product license to operate.

Getting Started

What's in This Part

Getting Started

What's in This Chapter

Overview

This chapter describes the quick steps to get started using EcoStruxure™ Control Expert - Modicon Communication Server V2.01 after installation to establish communication between Modicon controllers and Aveva System Platform.

For installation procedure and prerequisites, refer to installation, page 15 chapter.

Configure Modicon Communication Server

General

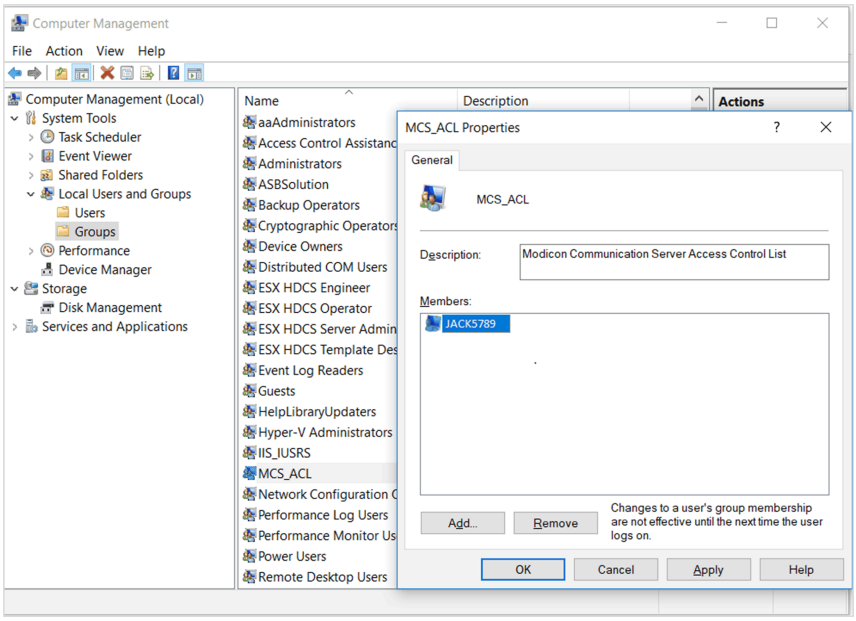
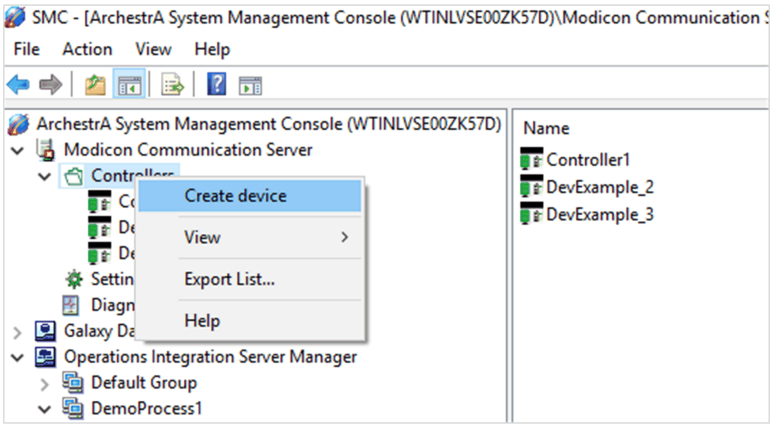
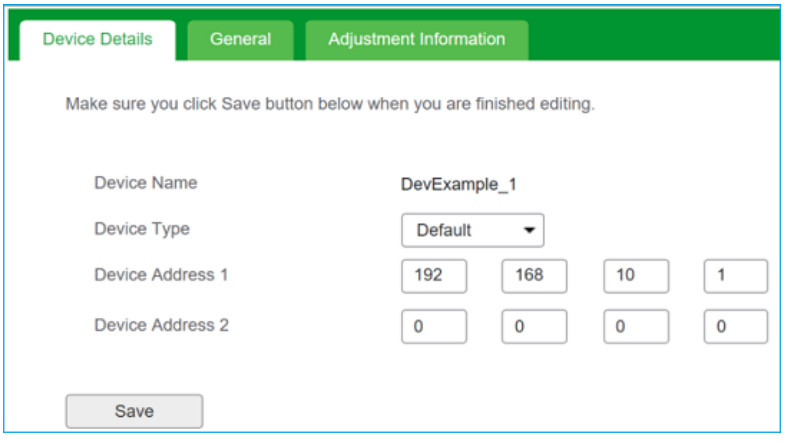
You can configure the Modicon Communication Server with security enabled or disabled:

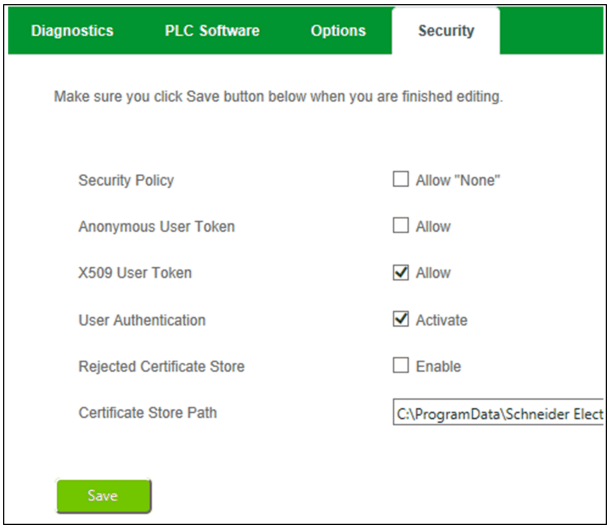
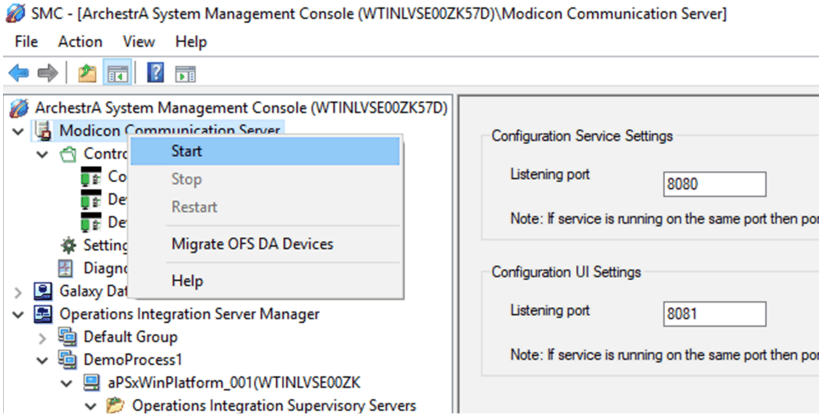
- Communication with Security Enabled, page 19
- Communication with Security Disabled, page 23

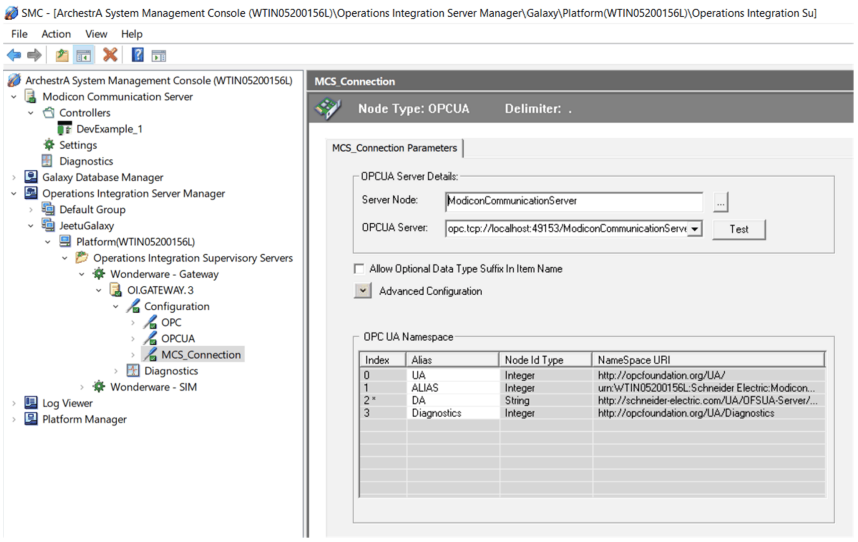
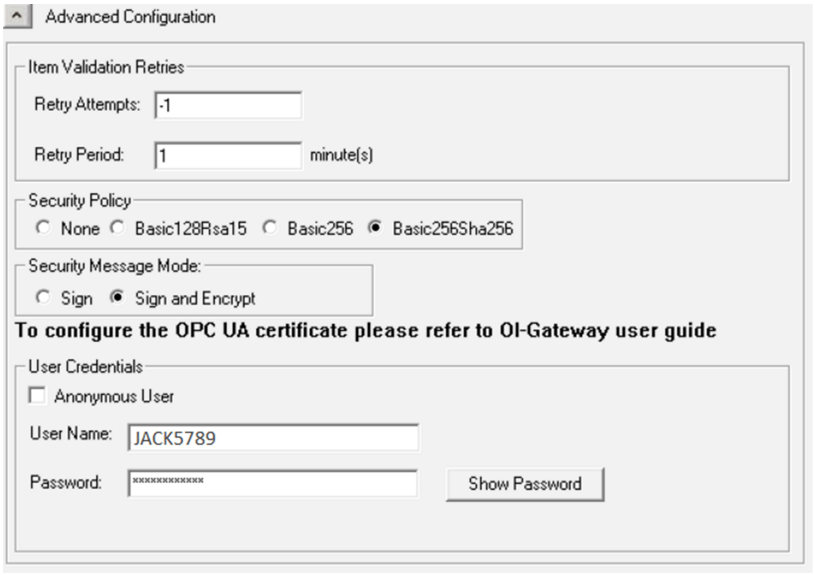
Communication with Security Enabled

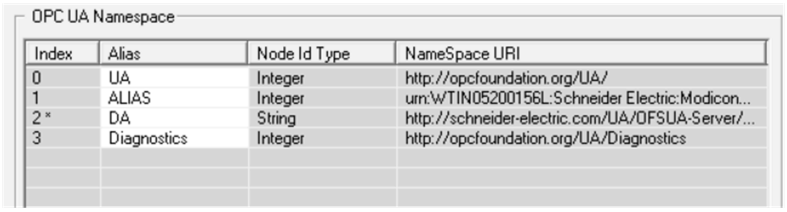
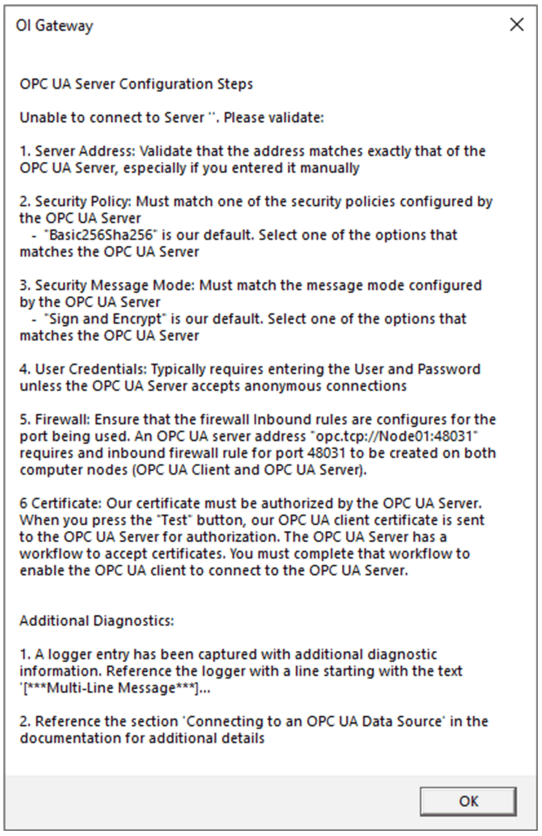
NOTE: This is the default configured mode and is recommended for the Wonderware System Platform.

Step	Action
1	Open **Computer Management** from **Start** menu.
2	Navigate to **System Tools** > **Local Users and Groups** > **Groups**.
3	Find MCS_ACL group and add current user or any user. This user will be used as an authorized user in the Modicon Communication Server for Wonderware System Platform required for communication.

Step	Action
	
4	Launch **System Platform Management Console** from windows **Start** menu or run `smc.msc` command.
5	In the Modicon Communication Server tree, right click on **Controllers** and click **Create device**.  **Result**: A new device alias is created with default name under **Controllers** tree. To rename, right click on the default name and click **Rename**.
6	Click on a specific device under **Controllers** tree to display its configured properties in the details pane.
7	In the **Device Details** tab, select the required type of controller and configure the device IP address. 

Step	Action
	NOTE: Device Address 1 is mandatory and unique, while **Device Address 2** is optional and required only for establishing network level redundancy. Click **Save** to apply the changes.
8	In Modicon Communication Server tree, click **Settings** to display its properties in the details pane.
9	In **Security** tab, configure as shown below:  Ensure that **Security Policy** and **Anonymous User Token** check boxes are cleared. Click **Save** to apply the changes.
10	Right click on the Modicon Communication Server node and click **Start** to start the server.  **NOTE:** If any changes are made in the Modicon Communication Server, you have to **Restart** the server.
11	In the SMC hierarchy, click **MCS_Connection** as shown below:

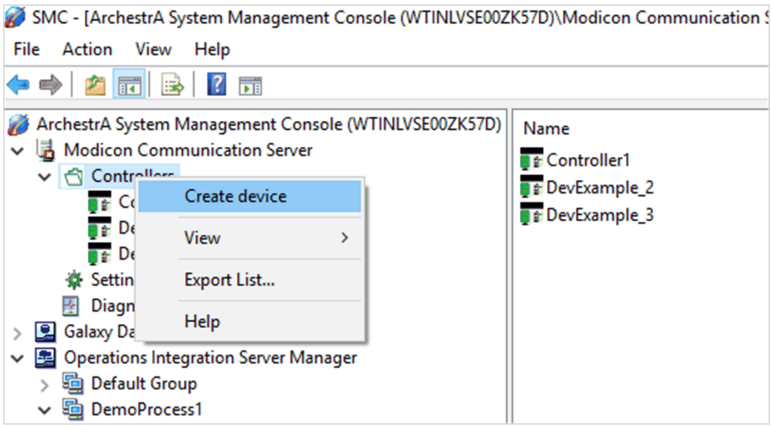
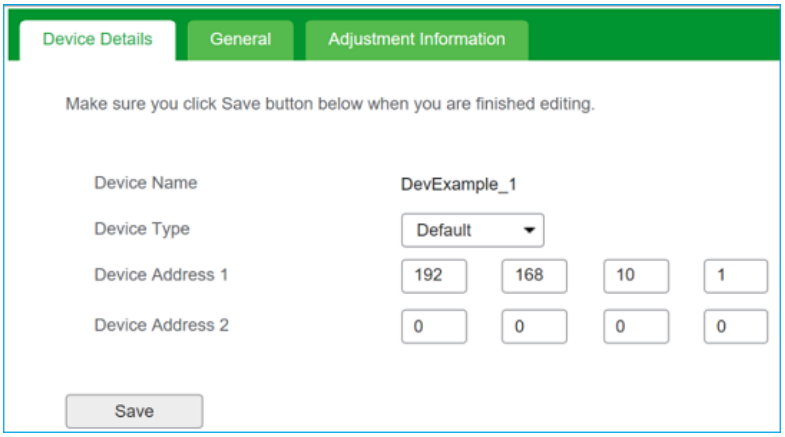
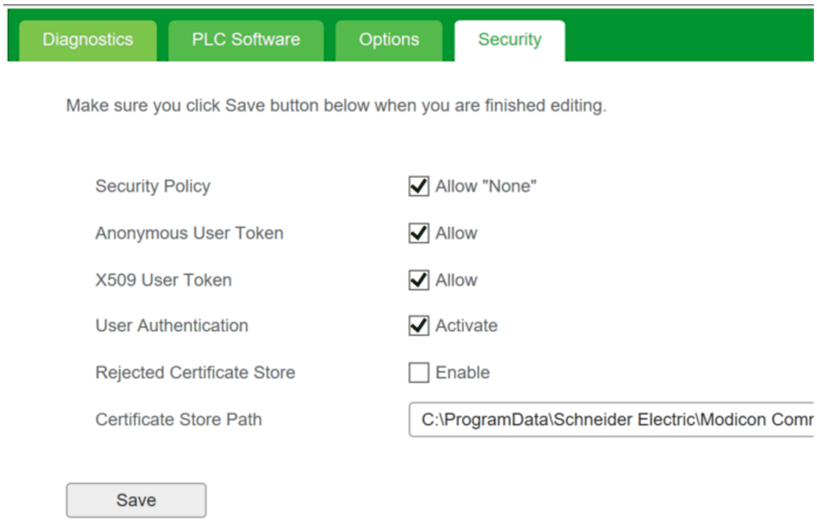
Step	Action
	
12	In the **Advanced Configuration**, page 40 section, enter **User Name** and **Password** which you have provided for the user in step 3:  Ensure that **Security Policy** is selected as **Basic256Sha256**, **Security Message Mode** as **Sign and Encrypt**, and **Anonymous User** check box is cleared. **NOTE:** If TCP listening port of Modicon Communication Server has been changed then **OPCUA Server** field has to be updated with latest port.
13	Click **Save** to apply the changes.

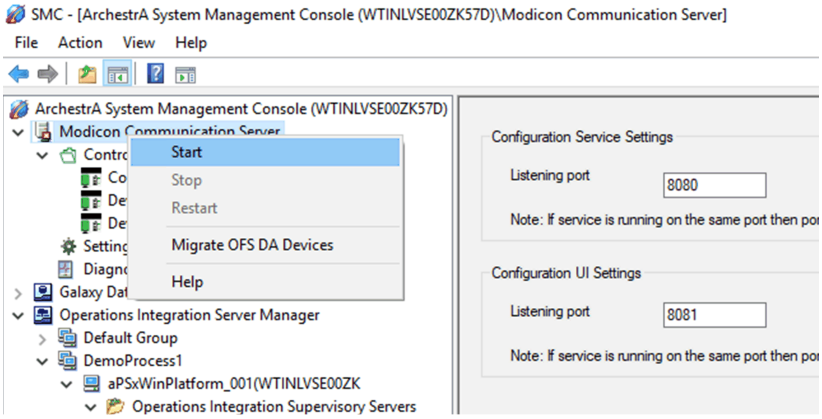
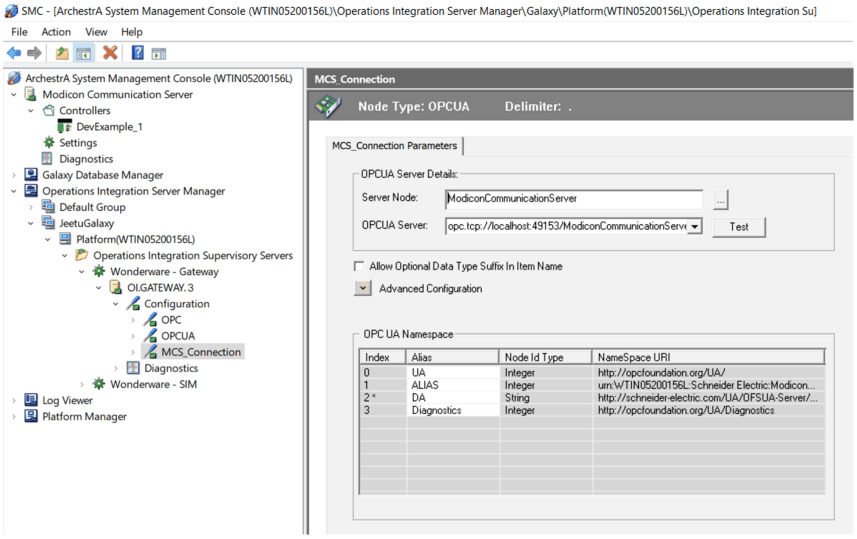
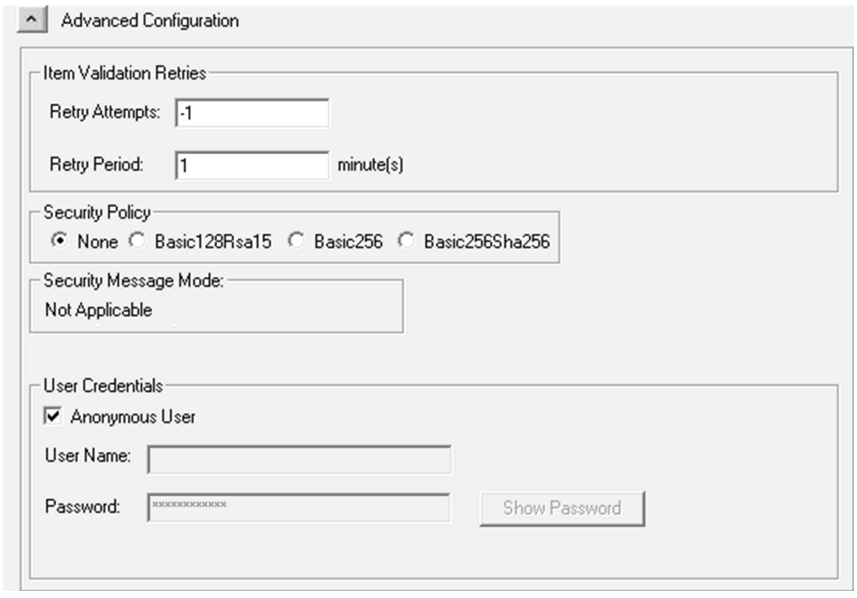
Step	Action
14	Click **Test** to establish the connection. If the connection is successful, then **OPC UA Namespace** will be displayed as shown below:  If the connection is not successful, then the following similar message will be displayed:  Click **OK** and repeat the steps to ensure that no steps has been missed. For troubleshooting, refer to the chapter describing about Troubleshooting, page 82.
15	Right click on **OI.GATEWAY.3** and select **Activate** to activate the gateway.

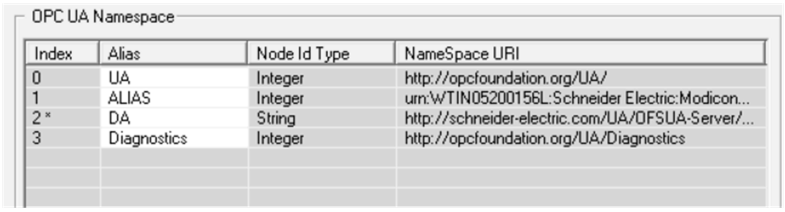
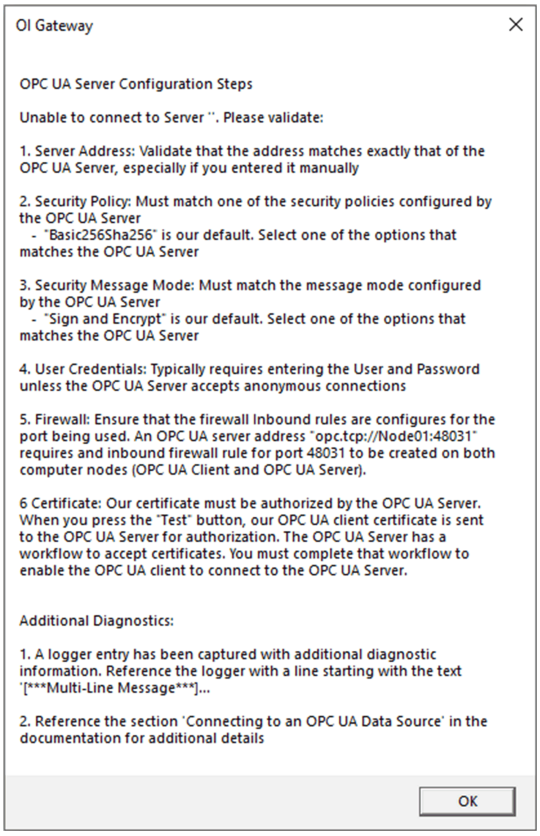
Communication with Security Disabled

NOTE: We do not recommended this mode due to cyber security.

Step	Action
1	Launch **System Platform Management Console** from windows **Start** menu or run `smc.msc` command.
2	In the Modicon Communication Server tree, right click on **Controllers** and click **Create device**.

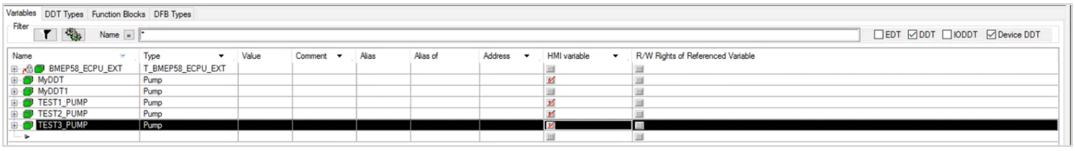
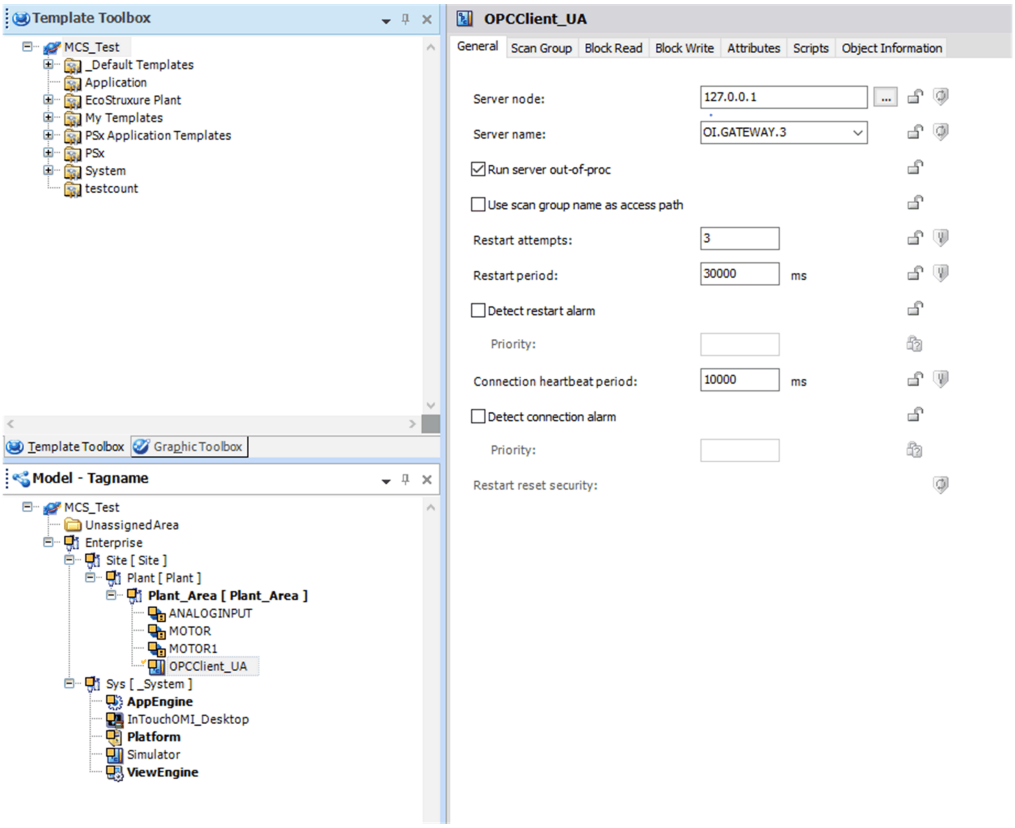
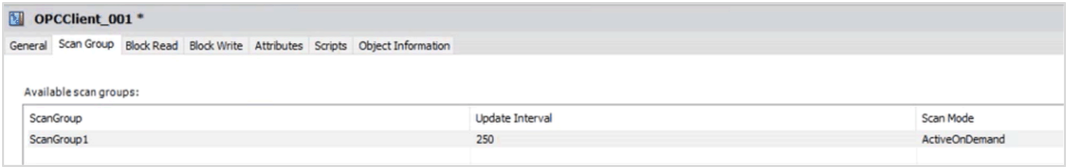
Step	Action
	 **Result**: A new device alias is created with default name under **Controllers** tree. To rename, right click on the default name and click **Rename**.
3	Click on a specific device under **Controllers** tree to display its configured properties in the details pane.
4	In the **Device Details** tab, select the required type of controller and configure the device IP address.  **NOTE: Device Address 1** is mandatory and unique, while **Device Address 2** is optional and required only for establishing network level redundancy. Click **Save** to apply the changes.
5	In Modicon Communication Server tree, click **Settings** to display its properties in the details pane.
6	In **Security** tab, configure as shown below:  Ensure that **Security Policy** and **Anonymous User Token** check boxes are selected.

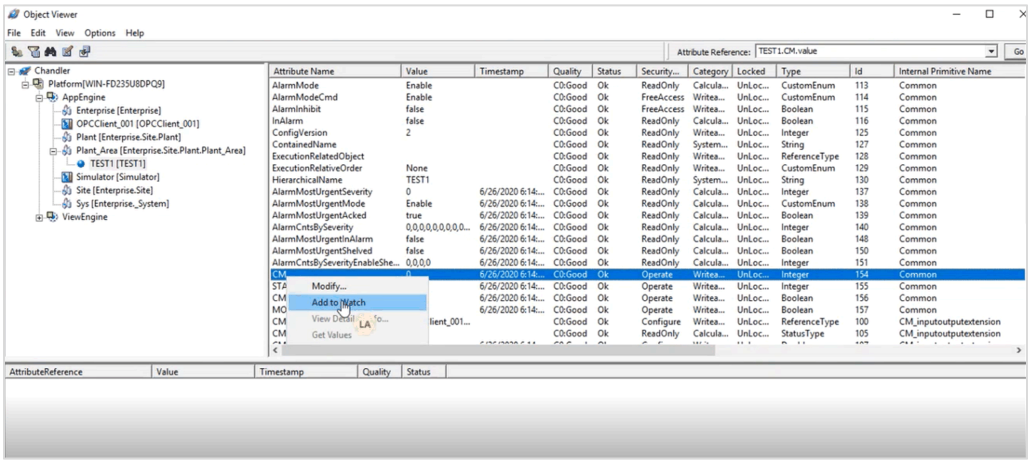
Step	Action
	Click **Save** to apply the changes.
7	Right click on the Modicon Communication Server node and click **Start** to start the server.  **NOTE:** If any changes are made in the Modicon Communication Server, you have to **Restart** the server.
8	In the SMC hierarchy, click **MCS_Connection** as shown below: 
9	In the **Advanced Configuration**, page 40 section, select **Security Policy** as **None** and **Anonymous User** check box is selected. 
10	Click **Save** to apply the changes.

Step	Action
11	Click **Test** to establish the connection. If the connection is successful, then **OPC UA Namespace** will be displayed as shown below:  If the connection is not successful, then the following similar message will be displayed:  Click **OK** and repeat all the steps to ensure that no steps has been missed. For troubleshooting, refer to the chapter describing about Troubleshooting, page 82.
12	Right click on **OI.GATEWAY.3** and select **Activate** to activate the gateway.

Communication in AVEVA System Platform with Modicon Controllers

Test Communication in AVEVA System Platform with Modicon Controllers

Step	Action
1	In Control Expert create a variable and ensure that the HMI variable is enabled and the variable is deployed to the controller or simulator. 
2	In AVEVA System Platform, create an instance of `$OPCClient` from **Template Toolbox** system and configure: • **Server node**: IP Address of the server machine. • **Server name**: `OI.GATEWAY.3`  • Create **Scan Group**: 
3	From **Template Toolbox** create a new instance of `$Userdefined` and add an attribute for the type matching as defined in the Control Expert with `IO` Features.
4	Configure attribute address with below syntax: `<OPCClient Instance name>.<ScanGroup Name>.MCS_Connection.DeviceGroup./DA/0:<Controller Name>!<Variable name in Controller>`

Step	Action
5	**Deploy** the **AppEngine** and the created instance.
6	Launch **Object Viewer**, select the attribute and **Add to Watch** list and verify that the values are updated based on the value modified in the Control Expert.  **NOTE:** If the **Quality** column displays **Bad** status (due to repair activity such as IP conflict or update in the Modicon Communication Server), **Deactivate** and then again **Activate** the gateway.

User Interface

What's in This Part

User Interface

What's in This Chapter

Introducing the Modicon Communication Server

Using the Modicon Communication Server

You can use the Modicon Communication Server to:

- Create an alias, which represents a PAC or other OPC UA client device.
- Configure properties for an alias.
- Configure global properties for the Modicon Communication Server.

NOTE:

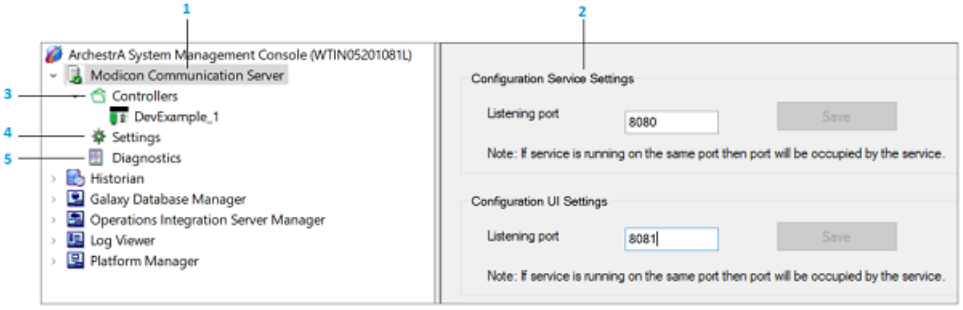
- Only a user with Administrator privileges can run the Modicon Communication Server.
- All configuration settings made, either for an alias or for the server, are made statically and take effect only after the server is stopped, then restarted. The making of dynamic changes, while the OPC UA server is running is not supported.

Launching the Modicon Communication Server

To launch the Modicon Communication Server, go to **Start > Wonderware Utilities** and click **Wonderware System Platform Management**.

The Modicon Communication Server is located in the SMC node as shown below.

The following illustration is an example of the Modicon Communication Server user interface in the SMC:



1 Modicon Communication Server, page 31

2 Details pane, page 31

3 Controllers, page 32

4 Settings, page 32

5 Diagnostics, page 32

Modicon Communication Server

This is the top branch of the Modicon Communication Server which allows you to set the port of **Configuration Service Settings** and **Configuration UI Settings** services.

To **Start/Stop/Restart** the server, refer to operating the Modicon Communication Server section, page 50.

Details Pane

The details pane displays the configuring settings associated with the item selected in the navigation tree.

Configuration Settings

What's in This Chapter

Overview

This chapter describes the settings of the Modicon Communication Server.

Controllers

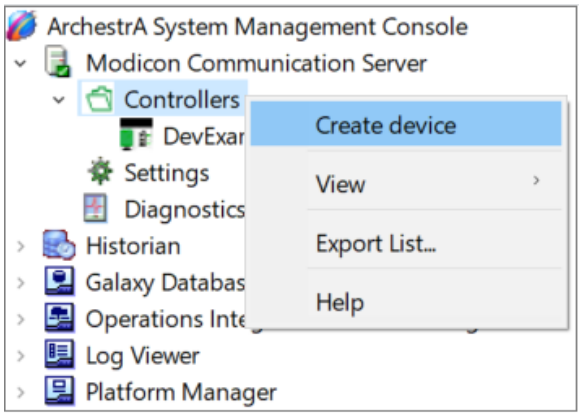
Overview

This section describes how to use the controllers node to create and configure a device alias.

Managing Device Aliases

Creating a Device Alias

The following illustration is an example of creating a device alias:



In the Modicon Communication Server tree, right click on **Controllers** and click **Create device**. A new device alias is created with default name under **Controllers** tree.

To rename, right click on the default name and click **Rename**.

NOTE: No two device aliases can be assigned the same name.

To configure the settings of device alias, refer to Configuring the Device Alias Properties, page 33.

Configuring the Device Alias Properties

Editing Properties for a Selected Device Alias

Select a specific device alias under **Controllers** tree to display its configured properties in the details pane.

The following properties can be configured for a selected device alias:

- Device details, page 33
- General, page 33
- Adjustment information, page 34

Device Details

Property	Description
Device Name	Name of the device.
Device Type	Select the type of controller from the drop-down list. Based on the type of controller you have selected, the default values are updated in the **Adjustment information**, page 34 tab.
Device Address 1	The required main address of the device, and the only address if **Device Address 2** is not configured.
Device Address 2	(Optional) An alternate address of the device. It is used to create a second communication path between the Modicon Communication Server and the OPC UA client device (typically a PAC) when the PAC link redundancy, page 65 feature is used.

NOTE: The name and address properties are unique for each device alias, and have no default settings.

General

Property	Description
Topological Objects	**Populate**: Enable to allow the population of State RAM topological objects for a device configured in EcoStruxure™ Control Expert.
PLC Embedded Data	**No Communication Break**: If enabled, in the event of an application build change, it supports uninterrupted operations by permitting the OPC UA server to preload and then switch to a new data dictionary. This is effective only if the **Data dictionary > Preload on build changes** option in **Tools > Project Settings... > General > PLC embedded data** is selected in EcoStruxure Control Expert.
Preload Settings	Select one of the following symbol loading and device connection options to be performed at OPC UA server startup: • **No Preload**: No device connection nor symbol access is performed. • **Device**: Symbol loading and device connection are performed.
Dynamic Consistency	**New Symbol Detection**: If enabled, the OPC UA server detects and populates new symbols created in EcoStruxure™ Control Expert application. This is effective only if **PLC Embedded Data > Using Data Dictionary > Preload on build changes** is enabled in EcoStruxure™ Control Expert.
Option	**Read Only**: Determines the read/write access to the property: • If selected, all variables relating to the device present read-only access rights. • If disabled, all variables relating to the device present read/write access rights.

Adjustment Information

NOTE: The default values are displayed based on the type of device you have selected in the **Device Details** tab.

For the default values of the selected device, refer to the default values, page 34 table.

Section	Property	Description
Adjustment Information	**Max Channels**	The maximum number of channels allocated to the device. Range: 1 to 256
	Max Pending	The maximum number of requests that can be sent in parallel. Range: 0 to 256
	Device Timeout (ms)	A delay used to manage the status transition of the device. Range: 3000 to 32767 ms **NOTE:** The recommended value is at least 3 times the **Frame Timeout**. The value can be monitored using the *#PLCQualStatua*, page 79 *Specific DataItem*.
	Frame Timeout (ms)	The permissible delay (in milliseconds) between request and response. The maximum length of time the OPC UA server will wait for an response from a device after sending a request. Range: 1000 to 10900 ms to a maximum of one-third the configured **Device Timeout**.
Enhanced Adjustment Information	**Reconnection Retry Number**	The number of times the OPC UA server attempts to reconnect to the OPC UA client represented by the device alias, after the server and client have been disconnected. The number includes any communication operations, such as item monitoring, and read/write attributes service set operations. If the device remains disconnected after the specified number of retries, the OPC UA server ignores requests for any additional communication operations for the **Disconnection Timeout** period. Range: 0 to 100
	Disconnection Timeout (min)	The duration (in minutes) during which requests for communication operations are ignored, after the number of unsuccessful reconnection attempts reaches the **Reconnection Retry Number**. Range: 0 to 1440 min

NOTICE

DEVICE COMMUNICATION IS MAINTAINED DISABLED

To keep device communication enabled, configure the value of the `Disconnection timeout` parameter carefully adapted to your system, or keep the default value.

Failure to follow these instructions can result in equipment damage.

Default Values

The table describes the default values of the selected **Device Type**:

Parameter	Device Type			
	Default	**M340/M350**	**M580-20xx**	**M580-30xx, M580-40xx, M580-50xx, M580-60xx**
Max Channels	4	8	12	16
Max Pending	0	8	12	16
Device Timeout (ms)	5000	5000	5000	5000
Frame Timeout (ms)	1000	1000	1000	1000

Parameter	Device Type			
	Default	M340/M350	M580-20xx	M580-30xx, M580-40xx, M580-50xx, M580-60xx
Reconnection Retry Number	0	0	0	0
Disconnection Timeout (min)	0	0	0	0

Settings

Overview

This section describes the configuration settings of the Modicon Communication Server.

Diagnostics Settings

Configuring Log and Snapshot Files

Use the **Settings > Diagnostics** tab to configure diagnostic log and snapshot features. Both log and snapshot files contain the information displayed in the:

- **Diagnostics** node, page 38 when the Modicon Communication Server is running as a standalone application.
- **Notifications** window, when the server is running as a Windows™ service.

Settings

Property	Description
OverWrite Logs	Enable to overwrite the log files (application diagnostics, communication diagnostics, and snapshot) each time the OPC UA server is started.
Application Diagnostics	Enable to activate the **Application Diagnostics** log trace file and then complete: • File path field: Enter the log file name (.txt or .log) and path. • **Max Size (Mb):** Set the maximum log file size (Range: 500 to 4000 Mb). **NOTE:** When the log file exceeds this size, the log file is discarded.
Communication Diagnostics	Enable to activate the **Communication Diagnostics** log trace file and then complete: • File path field: Enter the log file name (.txt or .log) and path. • **Max Size (Mb):** Set the maximum log file size (Range: 500 to 4000 Mb). **NOTE:** When the log file exceeds this size, the log file is discarded.
Snapshot	Enable to activate the **Snapshot** log trace file and then complete: • File path field: Enter the log file name (.txt) and path. • **Max Size (Mb):** Set the maximum log file size (Range: 500 to 4000 Mb). **NOTE:** When the log file exceeds this size, the log file is discarded. **Snapshot Period (s)**: Rate (in seconds) at which the **Snapshot** file will be periodically updated (in the range 10 to 120 s in 1 s increments). For more details, refer to the **Snapshot** file, page 51.
Verbose Mode	Enable to retrieve the detailed information in **Application Diagnostics** and **Communication Diagnostics** log files.

PLC Software Settings

Configuring Symbol File Checking and PAC Datatype Mapping

Use the **Settings > PLC Software** tab to set the time for cyclic redundancy check (CRC) of the symbol file, and data type mapping of the data types used by the EcoStruxure™ Control Expert configuration software.

Settings

Dynamic Consistency

Property	Description
Cyclic Consistency Check Rate (s)	Sets the period for symbol database (data dictionary) consistency checking. An inconsistency triggers a database reload and an update of all variables definition. Reload also is triggered by a communication detected error on inconsistent items (Range: 1 to 32767).

Project Files Options

<table>
<tr><th>Property</th><th colspan="2">Description</th></tr>
<tr><td rowspan="8">Unit/Control Expert Symbols</td><td colspan="2">Use Native Types: If enabled, variable instances linked to EcoStruxure™ Control Expert String, DATE, TOD, DT and TIME data types are converted to the OPC UA built-in String data type, in accordance with the IEC1131-3 representation.</td></tr>
<tr><td colspan="2">If Use Native Types is disabled, the following data type conversion occurs:</td></tr>
<tr><td>EcoStruxure™ Control Expert Type</td><td>OPC UA Datatype</td></tr>
<tr><td>String</td><td>Byte array</td></tr>
<tr><td>DATE</td><td>UInt32</td></tr>
<tr><td>TOD</td><td>UInt32</td></tr>
<tr><td>DT</td><td>Double</td></tr>
<tr><td>TIME</td><td>UInt32</td></tr>
<tr><td colspan="2">For more details, refer to the examples, page 37.</td></tr>
<tr><td>Unity/Control Expert DATE, TOD and DT Instances</td><td colspan="2">Use Regional Settings: If Use Native Types is enabled, this option enables formatting of the String datatype attribute of variable instances linked to EcoStruxure™ Control Expert DATE, TOD and DT datatypes according to configured Windows regional settings.
For example:
DATE can be formatted:
• M/d/yyyy
• MM/dd/yyyy
• yyyy-MM-dd
Where:
• MM = month with leading 0
• M = month with no leading 0
• dd = day with leading zero
• d = day with no leading zero
• yyyy = 4-digit year
Time can be formatted:
• H:mm:ss
• HH:mm:ss
Where:
• HH = 24 hour display with leading 0
• H = 24 hour display with no leading 0
• mm = minute with leading zero
• m = minute with no leading zero</td></tr>
</table>

Property	Description
Unity/Control Expert TIME Instances	**Display Underscore**: If **Use Native Types** is enabled, this option enables formatting of EcoStruxure™ Control Expert variable instances of the *String* datatype by displaying the underscore.
Byte Array Management	**Manage as ByteString**: If enabled, the *DataItem* datatype *Byte array* is converted to the *ByteString* datatype.

Examples: Use Native Types

Example 1: If **Use Native Types** is enabled, the following conversion occurs:

EcoStruxure™ Control Expert Type	Value Sample	OPC UA Built-in Type
DATE	D#2017-05-17	String
TOD	TOD#07:44:01	String
DT	DT#2017-05-17-07:44:01	String
TIME	T#07h44m01s100ms	String

Example 2: If **Use Native Types** is disabled, the following conversion occurs:

EcoStruxure™ Control Expert Type	Value Sample	OPC UA Datatype
DATE	0x20170517	UInt32
TOD	0x07440100	UInt32
DT	4.2922532071416873e-154	Double
TIME	27841100	UInt32

Options Settings

OPC UA Server Options

Use the settings in the **Settings > Options** tab to set a variety of optional settings for the Modicon Communication Server.

Settings

Property	Description
Server URI	Displays the URI to connect the Modicon Communication Server from OPC UA, page 39 client.
DNS Scanning TCP/IP	If enabled, causes the OPC UA server to use DNS to resolve device names for OPC UA clients (for example, PAC_Pump instead of 192.168.2.10).
TCP Listening Port	Configures the TCP listening port of the OPC UA server in the range 49152 to 65535. The listening port is part of the OPC UA server URL.
Processor Affinity	This option defines a 32-bit vector in which each bit set to 1 represents a logical processor on which the tasks of the OPC UA server process can run. Bit 0 represents processor 0, bit 1 represents processor 1, and so on. The bit vector is logically ANDed with the system bit vector (that is 2^n-1, where n is the number of logical processors). A null bit vector (that is value is 0) is set to system bit vector. The value is displayed in hexadecimal. Range: 0 to 99999999

Security Settings

OPC UA Server Security

Use the **Settings > Security** tab to set OPC UA security settings for the Modicon Communication Server.

Settings

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

If you enable **Security Policy > Allow "None"** and **Anonymous user token > Allow**, it can result in unauthorized equipment operation as it is accessible and modifiable by any clients.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Property	Description
Security Policy	**Allow "None"**: If enabled, configures the OPC UA server so that an OPC UA client can create a session or open a channel using an OPC UA client application instance certificate that is not validated, page 58.
Anonymous User Token	**Allow**: If enabled, configures the OPC UA server so that an OPC UA client can create an anonymous session (a session where the user identity token cannot be validated, page 58).
X509 User Token	**Allow**: If enabled, configures the OPC UA server so that an OPC UA client can create a session using an X509 identity token.
User Authentication	**Activate**: If enabled, configures the OPC UA server to apply user authentication and authorization, page 60.
Rejected Certificate Store	**Enable**: If enabled, a read/write file is created that stores a list of certificates rejected by the validation policy, page 57. In the **Certificate Store Path**, enter the path to store the rejected certificate.

Diagnostics

Overview

This section describes the diagnostics tab of the Modicon Communication Server.

Diagnostics

Overview

The **Diagnostics** node displays informational and detected error messages.

Detected error messages include:

- Modicon Communication Server service detected errors.
- PAC and device communication detected errors.

The diagnostic messages are logged in to the folder: *C:\Users\Public\Documents\Schneider Electric\Modicon Communication Server\Log*.

File name: Diagnostic_<Date>.txt

OPC UA Operations Integration Gateway Configuration

What's in This Chapter

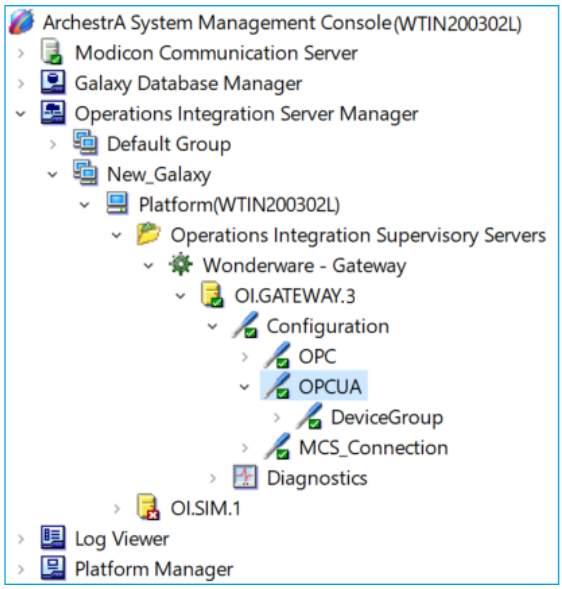
OPC UA Operations Integration Gateway Configuration

Overview

The Operations Integration (OI) Gateway when configured connects to the OPC UA server.

Configuring an OPC UA OI Gateway

Follow the below steps to configure the OI Gateway for OPC UA server:

Step	Action
1	In the SMC hierarchy, click **OPCUA** as shown below:  **Result**: The **OPCUA Server Details** section is displayed in the details pane.
2	In the **Server Node** field, click ... to browse the server node, or enter the server node name or the IP address.
3	In the **OPCUA Server** field, select the OPC UA Server URI, page 37 from the drop-down list. **NOTE:** The OPC UA server list is automatically populated if the OPC UA server is running, and is discoverable. Otherwise, enter the OPC UA server address manually in the **OPC UA Server** field. You can click **Allow Optional Data Type Suffix in Item Name** check box to add data type as a suffix to an item name.
4	Click **Test** to establish the connection, and update the OPC UA namespace.

Advanced Configuration

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

Property	Description
Item Validation Retries	The **Item Validation Retries** allows to configure the retry details for item validation. • **Retry Attempts**: Indicates the number of retry attempts for the validation. The valid range is between -1 to 10000000. The default value is -1. • **Retry Period**: Indicates the retry period (in minutes) for each retry of the item validation. The valid range is 1 to 10000000. The default value is 1 minute.
Security Policy	The **Security Policy** indicates the authentication mode used for the connection. • **None**: No security is applied (default). • **Basic128Rsa15**: Indicates that Basic128Rsa15 security is applied. • **Basic256**: Indicates that Basic256 security is applied. • **Basic256Sha256**: Indicates that Basic256Sha256 security is applied. **NOTE:** The Modicon Communication Server does not support **Basic128Rsa15** and **Basic256**.
Security Message Mode	The **Security Message Mode** indicates the message mode of the connection. If the **Security Policy** is set to **None**, the **Security Message Mode** is **Not Applicable**. If the **Security Policy** is set to any options other than **None** (that is, Basic128Rsa15, Basic256, or Basic256Sha256), the **Security Message Mode** options (**Sign** and **Sign and Encrypt**) populates for selection. • **Sign**: All messages are signed but not encrypted. • **Sign and Encrypt**: All message are signed and encrypted.
User Credentials	This section allows you to configure the user credentials for the OPC UA connection. Select the **Anonymous User** checkbox to allow the OPC UA client to connect to the OPC UA server without credentials. If your OPC UA client wants to connect to an OPC UA server that does not support anonymous connections, the OPC UA client has to provide a valid user name and password. Clear the checkbox to provide a **User Name** and **Password** in their respective fields. **NOTE:** The **User Name** has to be member of MCS_ACL group.

MCS Operations Integration Gateway Configuration

What's in This Chapter

MCS Operations Integration Gateway Configuration

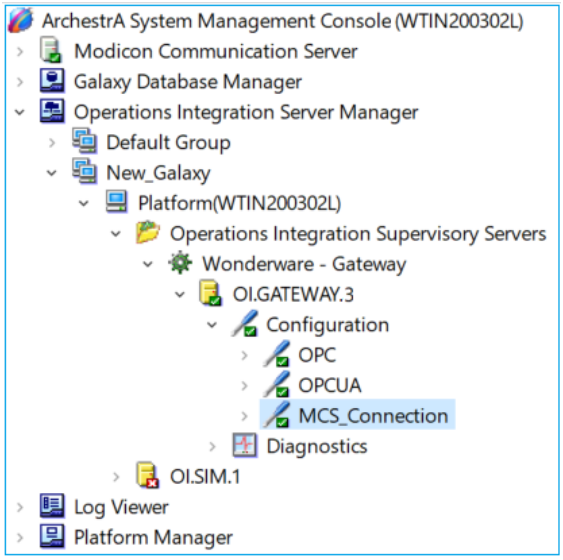
Overview

The Operations Integration (OI) Gateway when configured connects to the MCS server.

The Modicon Communication Server installation creates the new node **MCS_ Connection** for OI Gateway.

MCS OI Gateway

In the SMC hierarchy, click **MCS_Connection** as shown below:



Result: The configuration settings of the **MCS_Connection** is displayed in the details pane.

All the settings are preconfigured, you only need to enter the **Password** in the **Advanced Configuration** section. Click **Test** to establish the connection and to update the OPC UA namespace.

For more details about the configuration settings, refer to the advanced configuration, page 40 section.

Device Group

Overview

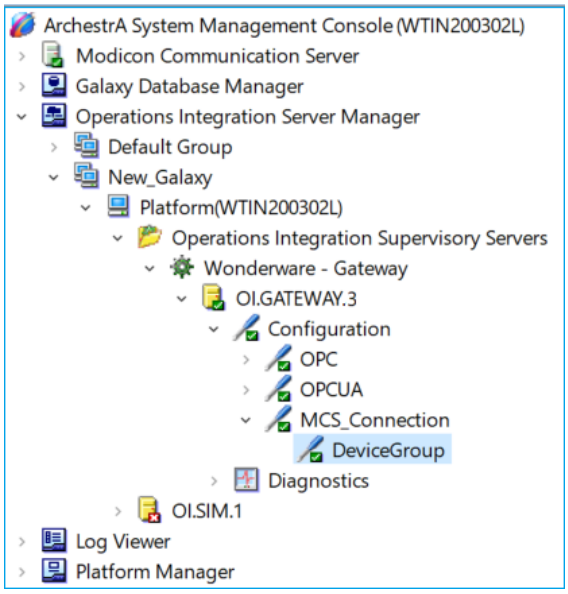
Device group allows you to specify an update interval for a set of device items and also allows you to view the tags associated with each device. The device group

update interval determines how often the OI Server polls the device and sends data to the client application.

The Modicon Communication Server creates the default device group (**DeviceGroup**) under **MCS_Connection** node.

Device Group Parameters

In the SMC hierarchy, click **DeviceGroup** as shown below:

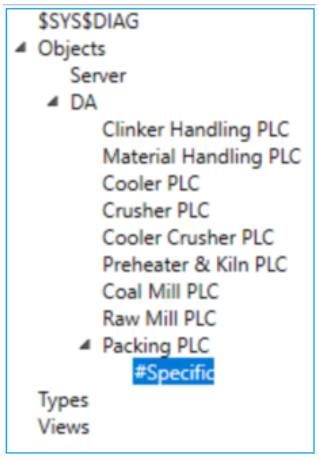


In **DeviceGroup Parameters** tab, click **Browse OPCUA Server** button to view the tags associated with each device in **OPCUA Tag Browser**, page 42 window.

OPCUA Tag Browser

The **OPCUA Tag Browser** window displays the list of devices configured in MCS and the associated tags.

To view the tags, go to the **Objects** hierarchy and click **#Specific** under respective device as shown below:



The following tag details are displayed in the browser table:

Name	Description
Tag name	Displays the tag name along with its path.
Display Name	Displays only the name of the tag.

Name	Description
Data type	Displays the data type of the tag.
Access Level	Displays the access level (read/write) of the tag.

Migrating from OFS DA to OPC UA Device

What's in This Chapter

Migrating OFS DA to OPC UA Device

Overview

You can use the Modicon Communication Server to migrate the configuration of OFS DA to OPC UA devices using the data defined in the configuration file (DeviceConfig.xml) of EcoStruxure™ Process Expert for AVEVA System Platform.

Configuration

Follow these steps to migrate from OFS DA to OPC UA:

Step	Action
1	In MCS hierarchy, click **Modicon Communication Server**. **Result**: The configuration section is displayed in the details pane.
2	In **OFS DA Configuration XML file** field, click ... to browse the path of the configuration file. For example: *C:\ProgramData\Schneider Electric*
3	Click **Save Settings**. **NOTE:** • Select the path only where the configuration file is located. if the selected path is wrong then **Select a valid OFS DA XML file** message is displayed in the dialog box. Click **OK** and select the appropriate path. • If an .xml file is not selected, then **Format of the selected file is invalid. You must select an XML file.** message is displayed in the dialog box. Click **OK** and select the XML configuration file exported from OFS DA.
4	Right click on **Modicon Communication Server** and click **Migrate OFS DA Devices**. **Result**: The Modicon Communication Server will import the devices which are in the configuration file. **NOTE:** • If **OFS DA Configuration XML file** field is empty then **Configure OFS DA XML file in Modicon Communication Server Setting** is displayed. • If a device existing in the Modicon Communication Server and also in the configuration file has same device name and IP address, then a conflict, page 44 occurs. • If you have selected the configuration file which is already migrated and contains same data, then **Configuration up-to date** message is displayed. • If the selected configuration file is not as per the schema then **Input XML file is invalid.** message is displayed.

Conflict

A conflict occurs when a device existing in the Modicon Communication Server and also in the configuration file has same device name and IP address.

The Modicon Communication Server will update the device configuration based on the **Device Name**, **Device Type**, and **Device Address** (IP address) as described in the below table:

Modicon Communication Server - OFS DA Mapping Scenarios			**External Factors**	**Action**
Device Name	**Device Type**	**Device Address**		
Same	Same	Same	–	Ignore
Same	Same	Different	–	Update
Same	Different	Same	IP address already exist in the MCS.	Conflict (no action)
Same	Different	Same	IP address does not exist in the MCS.	Update
Same	Different	Different	–	Conflict (no action)
Different	Same	Same	Device name already exist in the MCS.	Conflict (no action)
Different	Same	Same	Device name does not exist in the MCS.	Update
Different	Same	Different	–	Conflict (no action)
Different	Different	Same	–	Create
Different	Different	Different	–	Create

Modicon Communication Server

What's in This Part

Overview

This part describes the Modicon Communication Server.

Modicon Communication Server Operating Characteristics

What's in This Chapter

Overview

This chapter discusses the operating limitations, performance considerations, and recommended design practices for operating the Modicon Communication Server.

Modicon Communication Server Operating Characteristics

Limitations

The maximum memory amount that can be allocated to Modicon Communication Server is 4 GB (3 GB for x86 platforms).

NOTE: If either limit is exceeded, the server Address Space state enters in a limits exceeded state.

Other limitations, the context in which they occur, and their consequences if exceeded are set forth below:

Limit	Value	OPCUA Service	Service Parameter	Effects
Maximum session count	100	*CreateSession*	(Not Applicable)	*Bad_TooManySessions* service result code
Minimum session timeout	10 s	*CreateSession*	Requested SessionTimeout	revisedSession Timeout
Maximum session timeout	3600 s	*CreateSubscription*	Requested SessionTimeout	revisedSession Timeout
Maximum subscription count	300	*CreateSubscription*	(Not Applicable)	*Bad_TooManySubscriptions* service result code
Minimum publishing interval	100 ms	*CreateSubscription*	Requested Publishing Interval	revisedPublishingInterval
Maximum publishing interval	3600 s	*CreateSubscription*	Requested Publishing Interval	revisedPublishingInterval
Maximum subscription lifetime	3600 s	*CreateSubscription*	Min(Requested Publishing Interval, 3600000) * Requested LifetimeCount	revisedLifetimeCount
Maximum Notifications Per Publish	10000	*CreateSubscription*	maxNotificationsPerPublish	Notifications maximum capacity is thus (1000/ revisedPublishingInterval) * 1000 notifications per second.
Minimum sampling interval	100 ms	*CreateMonitorledtems*	MonitoringParameters. SamplingInterval	revisedSampling Interval

Limit	Value	OPCUA Service	Service Parameter	Effects
Maximum Message Queue Size	100	*CreateMonitorledtems*	MonitoringParameters. QueueSize	revisedQueueSize
Maximum Monitored Items Count	100000	*CreateMonitorledtems*	N.A.	*Bad_TooManyMonitoredItems* service result code

Modicon Communication Server Performance

Application Performance

If execution of an Modicon Communication Server task cannot be completed within the configured time, an application overload notification is displayed in the **Diagnostics** node, page 38.

Notifications that occur:

- Intermittently, indicate a transient host system overload.
- Persistently, indicate the host system (in term of memory quota, processor performances, other applications running) lacks capacity to run Modicon Communication Server.

Communication Performance

When the communication flow sent to a device exceeds the communication capacity of that device:

- The notification *Communication overload* is displayed in the **Diagnostics** node, page 38 along with the device configured address.
- The Status code of *DataChangeNotification.monitoredItems[].Value* DataValue is set to *Bad*.

Notifications that occur:

- Intermittently, indicate a transient host system overload.
- Persistently, indicate the number of polling requests sent to the device to fulfill the Monitored Items sampling rate exceeds the device capacity.

  NOTE: You can check the number of requests sent to a device per timer in the Timer list which is configured in the Snapshot, page 35.

For example, if the Timer list includes the following:

- Timer Number : 3

- Timer 100 ms : 1 polling req and 0 custom req

- Timer 1000 ms : 87 polling req and 0 custom req

- Timer 2000 ms : 0 polling and 3 custom req

The following formula yields the maximum number of requests that can be sent per second to the device communication adapter:

Max flow = Sum (1000*Polling req/Timer (in ms))

In this example:

Max flow = (1000*1/100) + (1000*87/1000) + 0 = 97 requests /s

Coordinating Server, Device and PAC Communications

Each PAC can handle a different number of requests in each scan of the MAST task. The following table gives the CPU Request per PAC scan time according to the CPU reference:

Platform	CPU	Communication adapter	Request per PAC scan time	Max Channel/Max Pending, page 34 Possible Configuration
M340	BMXP3410••	Embedded port	8	8/0, 8/8, 4/8
	BMXP3420••	Embedded port	8	8/0, 8/8, 4/8
	BMXP3420••2	Embedded port	8	8/0, 8/8, 4/8
M580	BME•581020	Embedded port	8	8/0, 8/8, 4/8
	BME•5820•0	Embedded port	12	12/0, 12/12, 4/12
	BME•5830•0	Embedded port	16	16/0, 16/16, 4/16
	BME•5840•0	Embedded port	16	16/0, 16/16, 4/16
	BME•585040	Embedded port	16	16/0, 16/16, 4/16
	BME•586040	Embedded port	16	16/0, 16/16, 4/16

A general rule for computing the communication capacity in request per second is:

Max capacity = 1000*Request per PAC scan time/PAC scan time (in ms)

For example, with Request per PLC scan time = 16 and PLC scan time = 50ms:

Max capacity = 1000*16/50 = 320 requests /s

If **Max Flow** computed above is greater than **Max Capacity**, increase the Monitored Items sampling rate, or decrease the number of Monitored Items.

NOTE: With PAC requests length =1024, each Read request can contain up to 500 values of OPC Variables with datatype INT16 or UINT16.

Recommended Design Practices

When designing EcoStruxure™ Control Expert application, take note of the number of OPC UA nodes your design will create. For example:

- For a one-dimensional ARRAY instance with <array elements count> elements, <array elements count> nodes are created in Modicon Communication Server data access address space.
- For a multi-dimensional ARRAY instance of <n> dimensions, with <array elements count1>, <array elements count2> … <array elements countn> elements per respective dimension, <array elements count1>*<array elements count2>*…*<array elements countn> nodes are created.
- For a DDT instance, one node for each DDT member is created.

In the above examples, if an ARRAY element or a DDT member is itself an ARRAY or a DDT, then the same node computation rules apply on the element.

When configuring the OPC UA client session timeout and subscriptions lifetime settings (computed as LifetimeCount * Publishing rate), set reasonable values. Reasonable values are values that will reduce the likelihood the server will pointlessly prolong inactive sessions and subscriptions. Setting unreasonably large values can cause the server to significantly increase its workload, and reach the monitored items limitation.

Because the Modicon Communication Server does not support the TransferSubscription OPC UA service, a subscriptions lifetime setting that is longer than the session timeout is overridden by the session timeout. This is because, as soon as the session is deleted on timeout, all subscriptions are deleted without waiting the expiration of the subscription lifetime.

Modicon Communication Server Operations

What's in This Chapter

Overview

This chapter describes the Modicon Communication Server running mode.

Operating the Modicon Communication Server

Operating the Modicon Communication Server from System Platform Management Console (SMC)

You can use the Modicon Communication Server to monitor, diagnose, and log the server events for the following data types:

- Application diagnostic data: Includes detected errors related to the application execution. When the optional verbose mode is configured, it also logs OPC UA events.
- Communication diagnostic data: Includes detected errors related to communications between PACs and devices. When the optional verbose mode is configured, it also logs detailed communication information.
- Snapshot file data: Contains runtime and statistics information displayed in the snapshot log file.

  NOTE:

  - Detection of application errors is enabled in the **Settings > Diagnostics** tab, page 35 in the Modicon Communication Server by selecting and configuring the **Application Diagnostics** setting.
  - Detection of communication errors is enabled in the **Settings > Diagnostics** tab, page 35 in the Modicon Communication Server tool by selecting and configuring the **Communication Diagnostics** setting.
  - Logging of dynamically updated runtime information and statistics is enabled in the **Settings > Diagnostics** tab, page 35 by selecting and configuring the **Snapshot** setting.

Configuration is performed using the Modicon Communication Server. Refer to the user interface, page 29 chapter for information on how to use it to configure the OPC UA server functions.

NOTE: A restart of Modicon Communication Server is required to take new configuration in to effect.

Starting the Modicon Communication Server

To start the server, right click on **Modicon Communication Server** and click **Start**. The icon  indicates that the server is started.

Stopping the Modicon Communication Server

To stop the server, right click on **Modicon Communication Server** and click **Stop**. The icon  indicates that the server is stopped.

Stopping the service (service mode) does not immediately shut down the Modicon Communication Server. A short waiting time is required for the shut down process to finish. For this reason, wait a short period before re-starting the server so that the previous session can completely shut down. If you decide to proceed, a detected error message is displayed.

Restarting the Modicon Communication Server

To restart the server, right click on **Modicon Communication Server** and click **Restart**. The icon  indicates that the server is restarted.

Snapshot File

The **Snapshot** file, page 35 contains the field device specific data, active OPC UA client session and subscription information, and monitored service messages.

Device-specific data includes the following for each listed device:

Field	Description
<Device address>=<alias name> [Primary Address Standby Address]	Combination of: • device address 1 • configured alias name • actual primary address and communication status, page 64. • actual standby address and communication status. **NOTE:** Undefined, if not configured.
UNITY device	Device configured type.
Device Identity	• Actual device identity, or • *Unknown* if not configured.
Device Version	• Actual device version, or • *Unknown* if not configured.
Device Status, page 64	• *Good* • *Uncertain* • *Bad*
Frame TO	Configured **Frame Timeout**.
Device TO	Configured **Device Timeout**.
Max Channel	Actual Max Channel used; not the **Max Channels** configured setting.
Max Pending Req	Actual Max Pending used; not the **Max Pending** configured setting.
Max Waiting Req	Device-specific requests waiting list size.
Topologic objects is	The setting of the **Topological object** property: • *Set* • *Not set*
Symbol Table from PLC or Symbol table used: <filepath>	Type of configured symbol link: • *Data dictionary* • <path> of the XVM or CSV file.

Field	Description
Is using PLC DataDictionary or NOT Using PLC DataDictionary	The type of configured symbol link.
Application (Symbol table)	Application name taken from XVM symbol link.
Application (Device)	Application name taken from the PAC.
Consistency Status	*Consistent*: if the PAC application and PAC symbol link are consistent. For other cases, refer to symbol link consistency policy, page 73.
Device	The read and write properties of the device, set in the **Option > Read Only** parameter, page 33: • *READ_ONLY*. • *READ_WRITE*
Device access	The data access, page 64 status: • *ENABLED*. • *DISABLED*
Request Length	Length of PAC communication requests.
Req Sent+Rcv	Sum of number of sent and received requests.
Bytes Sent	Number of bytes sent to the PAC.
Bytes Rcv	Number of bytes received from the PAC.
Ref Count	Reference counter of the device.
Nb of Error	Number of communication detected errors.
Last Error Code	Last detected error code.
State Cnt Good =	Number of devices with the respective device status, page 64.
State Cnt Uncertain =	
State Cnt Bad =	
Nb of Waiting Req (max reached =)	Current number of waiting requests and the maximum number of waiting requests reached.
Best Access Time	Shortest time between communication request and communication access.
Worst Access Time	Longest time between communication request and communication access.
Last Access Time	Most recent time between communication request and communication access.
Average Access Time	Average time between communication request and communication access.
Nb Var Desc	Number of internal objects for *DataItem*, page 75 management.
Nb Specific Var Desc	Number of internal objects for *Specific DataItems*, page 79 management.
Timer number	Number of active timers.
Timer list	List of active timers, with the number of polling and custom requests sent to the device.
Application Version (Symbol table)	Application signatures list.
Application Version (Device)	Device signatures list.
DataDictionnary option	The server will use the PAC embedded data dictionary as a source for variable symbols: • Checked: detection is enabled • Unchecked: detection is disabled (Refer to **PLC Embedded data > Using Data Dictionary** in the General Alias properties window, page 33.

Field	Description
No communication break option	The server is configured, on an application build change, to switch to a pre-loaded new data dictionary and avoid a communication break: • Checked: enabled • Unchecked: disabled (Refer to **PLC Embedded data > No Communication Break** in the General Alias properties window, page 33.
New symbol detection option	Automatic detection and population of new symbols is: • Checked: enabled • Unchecked: disabled Refer to **Dynamic consistency > New symbol detection** in the General Alias properties window, page 33.

The **Sessions** section in the **Snapshot** file describes the current OPC UA server session which includes:

Field	Description
Session Name	Readable string that identifies the session, page 62, set in *CreateSession()* parameter.
Identity	The data access status: • *Anonymous*: for an anonymous session. • *UserName*: provided by a UserName identity token, page 60. • *Certificate Subject Name*: provided by an X509 identity token, page 60.
SessionID	The *NodeId* assigned by the server to the session.
Connection	Local date and time where and when CreateSession() was invoked.
Last Contact	Local date and time where and when the last OPC UA service was invoked.
Subscriptions Count	Number of subscriptions in the session.
Monitored Items	Total number of monitored items in the session.
Session Timeout	Actual session timeout in ms. In brackets, the requested session timeout (set in *CreateSession()* service.

The **Subscriptions** section in the **Snapshot** file describes a list of subscriptions requested by an OPC UA client and created by the OPC UA server which includes:

Field	Description
Subscription ID	The server-assigned identifier for the subscription.
Publishing Enabled	• TRUE: publishing of NotificationMessages is enabled for the subscription. • FALSE: publishing of NotificationMessages is disabled for the subscription.
Publishing interval	The actual publishing interval that the server will use (in ms).
Keep Alive Count	The number of keep-alive messages sent to the client during the current subscription. When the number of consecutive publishing cycles – in which there have been no Notifications to report to the client – reaches this counter, a publish request is de-queued and used to return a keep-alive message. This keep-alive message informs the client that the subscription is still active.
Life Time Count	When the number of consecutive publishing cycles – in which there have been no publish requests available to send a publish response – reaches this counter, the subscription is closed.
Item Count	the number of monitored items attached to the subscription
Notifications count	Actual number of NotificationMessages published.

Field	Description
Publish Count	Number of de-queued publish requests.
Seq. Number	Sequence number of last notification message.

Certificate Management

What's in This Chapter

Overview

This chapter describes how the OPC UA server accepts, trusts, and validates application instance certificates.

Application Instance Certificates

Two Types of Application Instance Certificates

The Modicon Communication Server supports the use of two types of certificates:

- Self-signed certificates.
- Certificates issued by a certificate authority (CA).

Self-Signed Certificates

A self-signed certificate is an identity certificate that is signed by the same entity whose identity it certifies. When you install the Modicon Communication Server application, you also install the following self-signed instance application certificate:

ModiconCommunicationServer.der

This self-signed certificate is installed on the host PC in the following location:

`C:\Program Files (x86)\Schneider Electric\Modicon Communication Server\PKI`

An administrator can add this self-signed certificate to the OPC UA client *Certificate Trust List*, page 56.

The self-signed certificate has a lifetime of 5-years, which is renewed upon re-installation or repair, page 16 of the Modicon Communication Server software.

Certificates Issued by a Certificate Authority

Instead of using the auto-generated self-signed certificates in your project, you can simplify the deployment and administration of certificates by using a CA (or a CA chain). The CA can be used to issue both OPC UA server and OPC UA client certificates.

Follow these steps to obtain a certificate from a CA and replace the self-signed certificate:

Step	Action
1	Locate the Certificate Signing Request file (*OPCUAServerExpertCsr.pem*) located on the host PC at: `C:\Program Files (x86)\Schneider Electric\Modicon Communication Server\PKI`
2	Send the CSR to the intended CA, which processes the CSR and issues a new certificate (<IssuedCertificate>).
3	An administrator replaces the self-signed application certificate with the CA issued certificate by running: `"C:\Program Files (x86)\Schneider Electric\Modicon Communication Server\PKI\setcertificate"` <IssuedCertificateFullPath> **NOTE:** • The *setcertificate*, page 16 tool is installed as part of the OPC UA server installation. • Use only certificates that have the .der extension.
4	An administrator adds the CA certificate or CA chain to the *Certificate Trust List*, page 56.

NOTE: When used with the /restore argument, the setcertificate tool restores the original Modicon Communication Server self-signed certificate.

Managing the Modicon Communication Server Certificate Trust List

Introducing the Certificate Trust List

The Modicon Communication Server certificate trust list (CTL) contains the application certificates of OPC UA clients, with which the OPC UA server can communicate. The CTL is stored on the local machine *Windows Certificate Store* (WCS), in the *MCS_CTL* store

An administrator is required to trust a self-signed OPC UA client certificate (or a CA chain root certificate) by adding the certificate to the CTL.

NOTE: Similarly, only an administrator can untrust an OPC UA client certificate by removing it from the CTL.

The administrator can add an application certificate, in either of two ways:

- Running the *trustentity* tool from a command line interface (CLI). Using this tool, the administrator transparently adds a certificate to the CTL, without requiring of the administrator any knowledge of the WCS.

  NOTE: The *trustentity* tool is installed on the host PC as part of the Modicon Communication Server installation, page 15.

- Running the *WCS Explorer* CTL management application. Using this tool, an administrator can access a graphical interface and execute commands for adding, removing, and editing certificates.

Adding a Certificate by Running the *trustentity* Tool from a CLI

To use the *trustentity* tool to add a CTL from the command line interface, follow these steps:

Step	Action
1	On the host PC, open a **Command Prompt** window.
2	Execute the following command: `"C:\Program Files (x86)\Schneider Electric\Modicon Communication Server\PKI\trustentity" <CertificateToTrustFullPath>` Where: `<CertificateToTrustFullPath>` represents the full path to the certificate residing on the local host machine. For example: C:\certs\mycertificate.der

Managing Certificates Using the *WCS Explorer* CTL Management Software

You can use the *WCS Explorer* software to manage certificates in the CTL as follows:

Step	Action
1	Open the *WCS Explorer* management software in one of the following ways: • Select the **Modicon Communication Server Windows Certificate Store** shortcut icon that was created on the host PC during the OPC UA server installation. • Run the following command from a CLI: `C:\Program Files (x86)\Schneider Electric\Modicon Communication Server\PKI\ModiconCommunicationServer.msc`
2	In the *WCS Explorer* software, select **MCS_CTL**.
3	Using the *WCS Explorer* software, you can perform the following actions: • Add a certificate: ◦ Select **Action > All Tasks > Import**. ◦ Follow the *Certificate Import Wizard* instructions. • Remove a certificate: ◦ Select the certificate to be removed. ◦ Select **Action > Delete**. • Edit a certificate: ◦ Double-click on a certificate, then perform the desired edits.

Certificate Validation Policy in Modicon Communication Server

Strict Validation Policy

The OPC UA server applies a strict validation policy to each certificate path at start-up or when a client initiates an OPC UA connection. Rejected certificates are added to the **Rejected Certificate Store**, if this selection is enabled in the **Settings > Security** tab, page 38 of the Modicon Communication Server.

Potential Validation Process Detected Errors

The validation policy applied by the OPC UA server can detect the following validation errors. Each potential detected error is described below, including if the detected error can be ignored, and is ignored by the applied validation policy:

Validation Step	Detected Error	Cause	Seriousness	
			Can be ignored?	Ignored by Policy?
Host Name	Bad_CertificateHostNameInvalid	The HostName in the URL used to connect to the server does not match one of the HostNames specified in the server certificate.	Yes	No
Certificate Structure	Bad_CertificateInvalid	The certificate structure is invalid or one of the issuer certificate structures is not valid.	No	No
Signature	Bad_SecurityChecksFailed	The certificate signature is invalid or one of the CA certificates in the chain has an invalid signature or is not found. An issuer certificate may be not found because it exists but is not a CA.	No	No
Trust List Check	Bad_SecurityChecksFailed	The certificate is not trusted and at least one of the issuers in the chain is not trusted.	No	No
Validity Period	Bad_CertificateTimeInvalid Bad_CertificateIssuerTimeInvalid	The current time does not fit the certificate validity period or one of the issuer certificate validity periods.	Yes	No
Certificate Usage	Bad_CertificateUseNotAllowed Bad_CertificateIssuerUseNotAllowed	The set used for the Certificate does not match use requested for the Certificate (i.e. Application, Software or CA)[1].	Yes	No
1. The application certificate needs to include: *digitalSignature*, *nonRepudiation*, *keyEncipherment* and *dataEncipherment*.				

Causes of Certificate Type Validation Detected Errors

The certificate type validation policy applies to certificates of the following types:

- OPC UA server application instance certificates
- OPC UA client application instance certificates provided by the *OpenSecureChannel()* and *CreateSession()* services, provided that the client establishes a security policy other than *None*.
- X509 user token certificate

The probable cause and effect of a certificate validation detected error are set forth below:

Certificate Type	Case	Behavior	Probable Cause
OPC UA server application instance certificate	Starting the Modicon Communication Server	The application start is aborted.	Validity Period
OPC UA client application instance certificate	*OpenSecureChannel* or *CreateSession*	Service detected error. Session activation detected error.	Trust List Check or Validity Period
	At least one session is created	The secure channel is closed by the OPC UA server at secure channel renewal (each hour)	Validity Period[1]
X509 certificate user token	ActivateSession	Service detected error. Session activation detected error.	Trust List Check or Validity Period

Causes of User Token Validation Detected Errors

The probable cause and effect of a user token validation detected error are set forth below, for user tokens types other than *Anonymous*:

Type	Case	Behavior	Probable cause
UserNamePass-word	ActivateSession	Service detected error. Session activation detected error.	The token does not match any Windows account. or The Windows account does not belong to MCS_ACL group.
X509Certificate	Refer to the topic *Causes of Certificate Type Validation Detected Errors* regarding for the certificate type *X509 certificate user token*.		

Certificate Renewal

It is possible to renew self-signed certificates by performing an installation repair. The 5 year lifetime of the certificate takes effect as of the date the repair is performed.

Note that:

- An installation repair can be performed only when the Modicon Communication Server is stopped.
- If Modicon Communication Server certificate is to be issued by a certification authority (CA), page 55, the CA needs to process the updated Certificate Signing Request. The previous issued certificate is lost.

Security Management

What's in This Chapter

Overview

This chapter describes default security settings, and user authentication and authorization methods.

Default Security Policies

Security Policy Settings at Installation

When you install the Modicon Communication Server software, page 15, the Modicon Communication Server is installed and default settings are applied.

Default Security Configuration Settings

The OPC UA server is initially installed with the following default security settings, page 38:

- The **Security Policy > Allow None** option is de-selected. Thus, when attempting to connect with the OPC UA server in its default configuration, an OPC UA client needs to use the **Basic256Sha256** security policy.
- The **Anonymous user token > Allow** option is de-selected. Thus, when attempting to activate a session with the OPC UA server in its default configuration, an OPC UA client needs to use one of the following user token options:
  - UserName identity token
  - X509 identity token
- The **User authentication > Activate** option is selected. Thus, if the user authentication or authorization checks has not succeed, the OPC UA client cannot establish a session with the Modicon Communication Server.

Refer to the Security settings topic, page 38 for more information on these settings, and how to use the SMC to edit them.

User Authentication and Authorization

Authenticating and Authorizing User Tokens

When user authorization is activated and anonymous user tokens are not allowed (i.e., the default configuration settings, page 38) for the Modicon Communication Server, an OPC UA client needs to provide a user identity token when executing the *ActivateSession()* service. The user identity token can be a UserName identity token, or an X509 identity token – if X509 user tokens are allowed, page 38. Only an authenticated and authorized user can connect to the Modicon Communication Server.

The following discussion of identity tokens assumes user authorization has been activated, and X509 user tokens are allowed.

UserName Identity Token

When the OPC UA client provides a UserName identity token, the Modicon Communication Server performs authentication and authorization checks, as follows:

- Authentication checking is based on Windows™ user accounts. A UserName identity token is authenticated if its content matches a Windows user account (local or in domain) with the same user name and password combination.
- Authorization checking is based on an access control list (ACL). A Windows user group (MCS_ACL) is created by the setup and initially includes an administrator with permission to add authorized Windows user accounts to the ACL.

Only an authenticated user, with an associated Windows user account that is included in an MCS_ACL group, can connect to the Modicon Communication Server.

X509 Identity Token

When the OPC UA client provides an X509 identity token, the Modicon Communication Server performs user authentication and authorization checks, as follows:

- Authentication relies on the use of public and private keys, as follows:
  - The certificate signature set in the *userTokenSignature* parameter of the *ActivateSession()* service (generated with the private key associated with the X509 certificate) is decrypted using the public key provided in the certificate.
  - The decrypted certificate signature is compared to a certificate signature that is computed using the signature algorithm provided in the *userTokenSignature* parameter. If both signatures are equal, the user token is authenticated.
- Authorization is based on the Modicon Communication Server Certificate Trust List (CTL), page 56. If the CA, and all its intermediate CA (if applicable), for the user appears as trusted in the CTL, the user token is authorized.

Only an authenticated user, for which the certificate issuer is trusted in the CTL, can connect to the Modicon Communication Server.

For more details, refer to the Security Deployment Guide (see EcoStruxure™ Control Expert, Modicon Communication Server, Security Deployment Guide).

Communication Links

What's in This Chapter

Overview

This chapter describes how an OPC UA client can establish a connection with the Modicon Communication Server, how PAC data can be accessed, and how to establish redundant connections between the Modicon Communication Server and the PAC.

Connecting an OPC UA Client to the Modicon Communication Server

OPC UA Client Connecting to the Modicon Communication Server

You can programmatically obtain the Modicon Communication Server supported EndPoint URLs using either the *GetEndPoints()* or the *CreateSession()* service.

NOTE: The Modicon Communication Server does not support registration with either a Local Discovery Server or a Global Discovery Server.

The following EndPoint URLs are supported:

- Opc.tcp://<server_computer_name>:<listening_port>/ModiconCommunicationServer

  and

- Opc.tcp://<server_IPAddress>:<listening_port>/ModiconCommunicationServer

Where:

- <server_computer_name> is the name of the computer running Modicon Communication Server.
- <server_IPAddress> is the IPv4 or IPv6 address of the computer running the Modicon Communication Server (excluding IPv4 local host and IPv6 link local address).
- <listening_port> is set to 49152 by default, but can be configured in the range 49152...65535.

OPC UA Client Monitoring Modicon Communication Server Status

An OPC UA client can query the Modicon Communication Server ServiceLevel variable (as formally defined in the OPC Unified Architecture part 5) to monitor the ability of the Modicon Communication Server to provide data. The following features are combined in ServiceLevel computation:

- The workload of the Modicon Communication Server: for V1.0, only the workload *Low* is presented.
- The address space state: *Empty*, *Build*, or *Updating*.

The ServiceLevel variable can have the following values

ServiceLevel Value	Work Load	Address Space State
3	Any	Empty
255	Low	Build
199	Low	Updating

For example, the ServiceLevel value 199 indicates that services in the View Service Set may generate a detected error. In this case, it is recommended that an OPC UA client wait until the ServiceLevel value returns to value 255 before attempting to perform any of the View Service Set services.

Communicating with PACs and Devices

Supported Networks and Devices

Modicon Communication Server operates on the following platforms and communication networks:

Platform	Communication Protocol	
	Ethernet TCP/IP	USB
M340	Built-in channel BMXNOE0100 BMXNOE0110 BMXNOC0401	Built-in channel
M580	Built-in channel BMENOC03••	Built-in channel

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

Check that you are operating a platform supported by Modicon Communication Server V1.X.X (Please refer to the preceding table.)

Failure to follow these instructions can result in death, serious injury, or equipment damage.

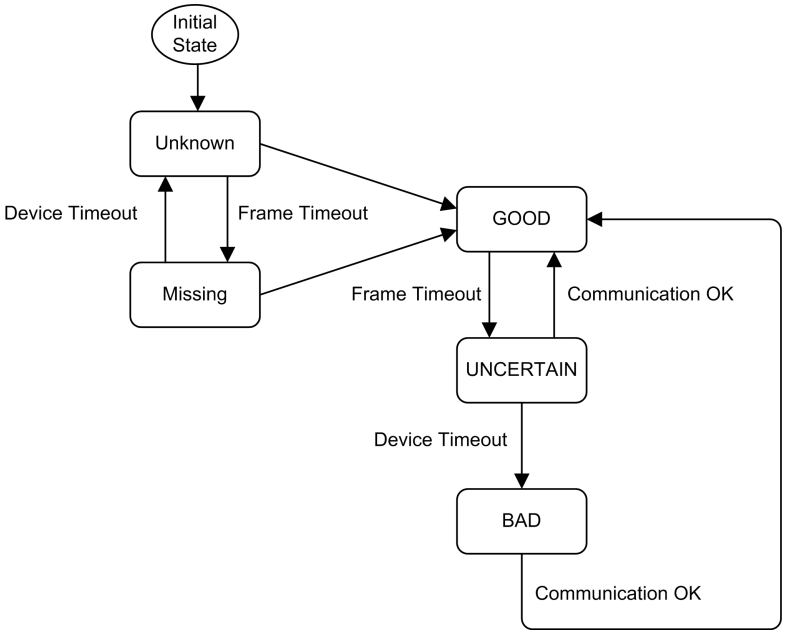
Requirements for Access to PAC or Device Variables Via OPC UA DataItems

The ability of the Modicon Communication Server to read or write PAC or device variables using DataItems depends on these preconditions:

- A communications path is established between the Modicon Communication Server and the PAC or device.
- Communication status, page 64 is ONLINE.
- Data access status, page 64 is ENABLED.

Device Status

A PAC or device can transition through several states, as follows:



NOTE:

- **Device Timeout** and **Frame Timeout** can be configured in the **Controllers** node, page 34.
- **Device Status** is displayed in the Modicon Communication Server **Snapshot** file, when the server is running as a standalone application.

Communication Status

The communication status of a device is related to the device status, as follows:

Status	Description
ONLINE	Device Status is GOOD or UNCERTAIN.
OFFLINE	The cause for OFFLINE communication status can be: • Device Status is BAD. • Server communication is stopped because DEMO mode has expired and licensing information is missing. • Device communication is disabled because PAC link redundancy, page 65 is configured, and the Standby communication path did not pass the communication path consistency check. • Device communication is disabled because of the configuration of the Disconnection timeout property, page 34.

NOTE: Communication status is included as part of both the Primary address and the Standby address as displayed in the Modicon Communication Server **Snapshot** file, when the server is running as a standalone application.

Data Access Status

The data access status of a device is related to the device status, as follows:

Status	Description
ENABLED	Data access status is ENABLED if: • Communication status is ONLINE. and • The symbol link (i.e. the collection of symbols derived from PAC variables) is consistent with the source PAC variables.
DISABLED	Data access status is DISABLED if: • Communication status is OFFLINE. or • The symbol link is not consistent with the source XVM symbol link file.

NOTE: Data access status is displayed as **Device access** in the Modicon Communication Server **Snapshot** file, when the server is running as a standalone application.

PAC Link Redundancy

Introduction

You can configure the Modicon Communication Server to establish redundant UMAS links to an EcoStruxure™ Control Expert PAC. Communication redundancy relies on the use of:

- Two separate physical communication paths: a primary path and a standby path.
- Two different IP addresses attributed to the same PAC with a unique alias.

The two physical communication paths are achieved by using:

- Two separate Ethernet ports in the PAC local main rack. For example, the two separate connections could be made to:
  - Two ports, each on a different Ethernet communications module in the local main rack.

    Or
  - A CPU port and a port on an Ethernet communications module installed in the local main rack.
- Two Ethernet ports, each on a separate network interface card (NIC), installed on the host PC.

The switchover from primary communication path to standby communication path can be performed automatically (upon the loss of the primary path) or in response to a command. In each case, the switchover is transparent to an OPC UA client application, for example a SCADA. The switchover does not impact to the OPC UA client.

Using the PAC link redundancy feature, you can design a redundant network architecture that provides for a communication path switchover – and continued communication – if communication to part of the network is lost.

Establishing Primary and Standby Communication Paths

PAC link redundancy is enabled using the Modicon Communication Server tool to enter two different IP addresses for the PAC in the **Device address 1** and **Device address 2** fields.

Primary Communication Path:

Neither communication path, with its associated IP address, is configured as the primary (or the standby) path. Instead, the first communication path to establish a connection and successfully transmit application information is designated the

primary communication path. At this moment, the primary communication path status is ONLINE and communication is supported by the OPC UA server.

Standby Communication Path:

After the primary communication path is established, the remaining path becomes the standby communication path. A connection check and a consistency check are performed on the standby communication path every 10 seconds. the results of these communication path checks can be:

- Communication check: **OK** or **Not OK**.
- Consistency check: **OK** if all the following conditions exist (**Not OK** otherwise):
  - The PAC firmware versions (in the primary and standby paths) are the same.
  - The application signatures (in the primary and standby paths) are compatible.

    NOTE: Application GUID and variable layout need to be the same.
  - The tested network frame lengths are the same.

Configuring PAC Link Redundancy to Operate with Two PACs

If the application GUID is the same for both the primary communication path and the standby communication path, it is possible to operate the PAC link redundancy feature by connecting each path to a different PAC.

When building the PAC application for a two-PAC design, take the steps set forth below so that EcoStruxure™ Control Expert will apply the same application signature to the applications in the primary PAC and the standby PAC:

Step	Action
1	Build the application in EcoStruxure™ Control Expert for the primary PAC. Use the communication settings, including IP address, that apply to the primary communication path.
2	Download the application from EcoStruxure™ Control Expert to the primary PAC.
3	Use the **Build Changes** command in EcoStruxure™ Control Expert to re-build the application for the standby PAC. Use the communication settings, including IP address, that apply to the standby communication path.
4	Download the application from EcoStruxure™ Control Expert to the standby PAC.

Impact of Communication Path Status on PAC Link Redundancy

After the primary communication path is established and is operating ONLINE, the status of both communication paths - primary and standby - is determined by the communication check and consistency check of the standby communication path, as follows:

- If the PAC link redundancy feature is configured, but the standby communication path is not identified (for example, not physically connected), then:
  - The standby communication path is OFFLINE and no switchover can be operated.
  - The primary communication path remains ONLINE.

- If the standby communication path is identified, then the consistency check is performed.
  - If the standby communication path consistency check is successful:

    - The standby communication path is ONLINE.

    - A switchover can be performed from primary to standby communication paths.
  - If the standby communication path consistency check is unsuccessful:

    - The system is not operational. In this case, both the primary and standby communication paths are OFFLINE.

    - The communication becomes unusable and no switchover can be performed.

⚠CAUTION

INCONSISTENT CONFIGURATION NOT DETECTED

Keep the Primary communication path and Standby communication path connected to the same PAC.

Failure to follow these instructions can result in injury or equipment damage.

Triggering a Switchover

A switchover from the primary communication path to the standby communication path can be triggered in two ways:

- Automatically, in response to loss of the primary communication path (for example, a send or receive detected error).

  NOTE: A send or receive detected error can arise from a request timeout due to a device communication latency, which cannot be differentiated from a communication interrupt. To avoid loss of data, the last request is re-sent to the new primary communication path after a switchover. As a result, switching from primary to standby communication paths has no effect on client application variables.
- By command, by writing to the *#SwitchPrimaryAddress* *Specific DataItem*, page 80.
  - 0 sets the communication path associated with **Device address 1**.
  - 1 sets the communication path associated with **Device address 2**

Managing a Switchover

The switchover from one communication path to the other is managed as follows:

- If the standby communication path was ONLINE at the last connection check, the switchover can be performed. The standby communication path becomes the new primary communication path.
- If the standby communication path was OFFLINE due to disconnection at the last check, a switchover can be attempted. A switchover will be performed when access to the standby communication path is achieved, unless a the original primary communication path is re-established and ONLINE.
- If the standby communication path was OFFLINE as a result of not passing the most recent consistency check, no switchover can be performed.

Modicon Communication Server Redundancy

Introduction

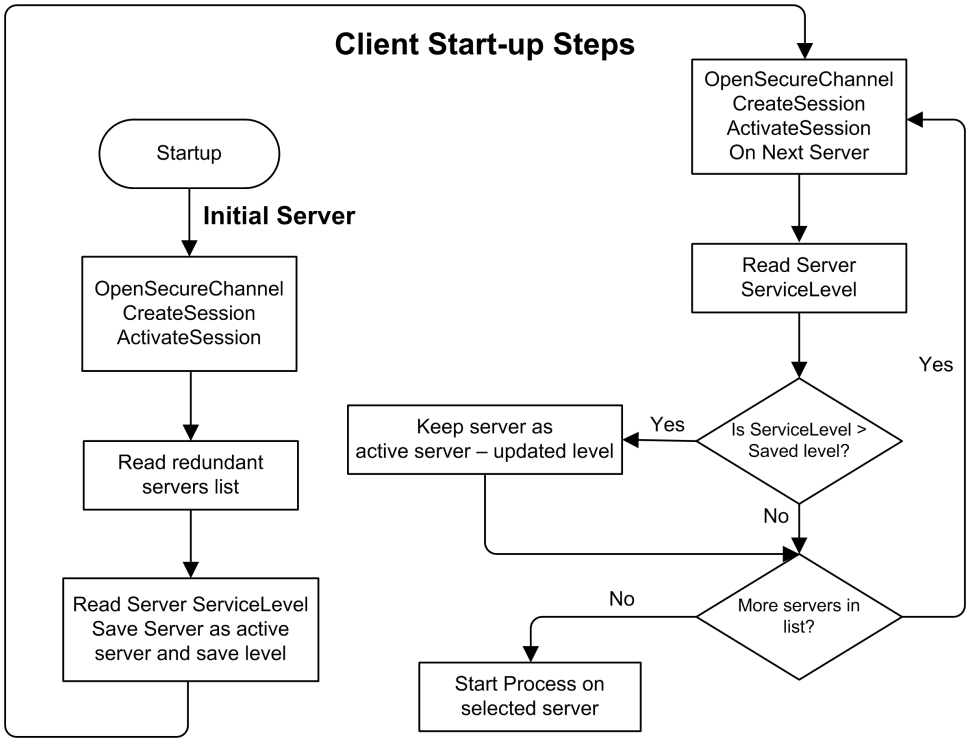
The Modicon Communication Server supports non-transparent server redundancy with warm failover mode.

In warm failover mode the backup, or alternate, server (or servers) can be active, but cannot connect to actual data points. This is typically the case in systems where the underlying devices are limited to a single connection. For example, the underlying devices, such as PACs, may have limited resources that permit only a single server connection. In this scenario, only one server at a time can consume data.

Initial Client Connection to a Redundant Server Set

The following figure provides an overview of the steps a client typically performs when first connecting to a redundant server set.

NOTE: The figure does not cover all possible detected error scenarios.



NOTE: As implemented in the Modicon Communication Server, the redundant servers list needs to be configured in the OPC UA client.

Supported Client Actions

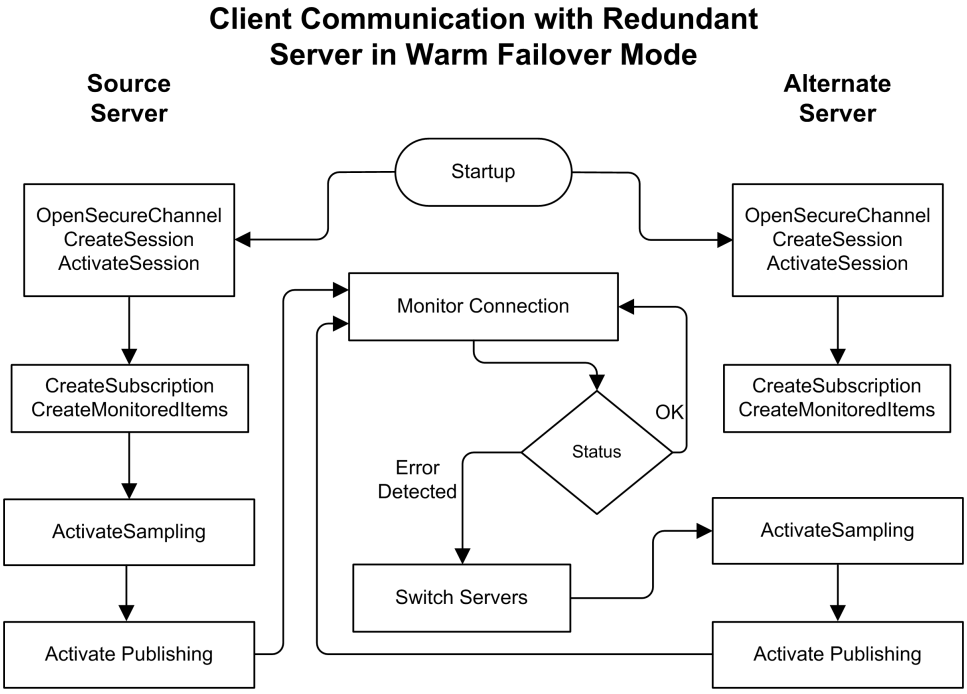
The list of available and unavailable client actions in non-transparent redundancy with warm failover mode include the following:

Client Options in Warm Failover Mode	Supported?
On initial connection in addition to actions of Active Server:	
Connect to more than one Modicon Communication Server	No
Create subscriptions and add monitored items	No
Activate sampling on the subscriptions	Yes

Client Options in Warm Failover Mode	Supported?
Activate publishing	Yes
At Failover:	
OpenSecureChannel to backup Modicon Communication Server	No
CreateSession on backup Modicon Communication Server	No
ActivateSession on backup Modicon Communication Server	No
Create subscriptions and monitored items	No
Activate sampling on the subscriptions	Yes
Activate publishing	Yes

Server Communications

A client would perform the following steps when communicating with a server using warm failover mode:



OPC UA Information Model

What's in This Part

Overview

This part describes the OPC UA information model, with emphasis on the Modicon Communication Server address space.

OPC UA Data Model

What's in This Chapter

Information Model

Namespace Index References

The following namespace URLs (published in the *NamespaceArray* variable as defined in the *OPC Unified Architecture Specification Part 5: Information Model, Release 1.04 (November 22, 2017)* , for the associated *NamespaceIndex* values as defined in *OPC Unified Architecture Specification Part 3: Address Space Model, Release 1.04 (November 22, 2017)* are supported by the Modicon Communication Server:

NamespaceIndex	**URL**	**Usage**
0	http://opcfoundation.org/UA/	OPC UA namespace
1	urn:localhost:Schneider Electric: Modicon Communication Server	Local server
2	https://schneider-electric.com	Modicon Communication Server data access
3	http://opcfoundation.org/UA/Diagnostics	OPC UA diagnostics generated by Modicon Communication Server

Modicon Communication Server Data Access Address Space

What's in This Chapter

Modicon Communication Server Modeling Elements

Modicon Communication Server DataItem, DA Root, and Alias Models

OPC UA Node Modeling

A *DataItem* designates an instance of an OPC UA variable node, with has*hasTypeDefinition ReferenceType* targeting *DataitemType* as defined in the *OPC UA Specifications part 8: Data Access*.

A *Folder* designates an instance of an OPC UA Object node, with *hasTypeDefinition* targeting *FolderType* as defined in the *OPC UA Specifications part 5: Information Model*.

The following notation is used to qualify an OPC UA node:

<template>=<UAType>,<NodeId>,<DisplayName><TargetReference>

Where:

- <template> designates a template name for the node.
- <UAType> is the target of a *hasTypeDefinition ReferenceType*. Possible values include:
  - *"Folder"*
  - *"DataItem"*
- <TargetReference> designates a node template which is the (0..*) target of *Organizes ReferenceType*, where <template> is the *SourceNode*.

Other patterns used to set the OPC UA node attributes include:

Node attribute	Value
Node Type	String
NodeId Identifier	0:<NodeId>
BrowseName	2:<DisplayName>
DisplayName	<DisplayName>

DA Root and Alias Modeling

A root *Folder* node is published as <ROOT_DA>="Folder", "","DA",<Alias>

An Alias configured in the Modicon Communication Server tool with the name <AliasName> is structured as follows:

<Alias>="Folder",<AliasName>,<AliasName>,<EntityName>

Examples of <EntityName> are described in the following topics and include *DataItem*, *Mapping*, and *Specific*.

Modicon Communication Server Address Space

Overview

This section describes the Modicon Communication Server address space published in the OPC UA server.

Linking the Modicon Communication Server to EcoStruxure™ Control Expert Symbols

Accessing EcoStruxure™ Control Expert Address Space

EcoStruxure™ Control Expert address space is published by translating the collection of variables (or symbols) associated with each PAC and associating them with an Modicon Communication Server alias created in the Modicon Communication Server. You can associate PAC variable data with the OPC UA server. To associate, embed the PAC data dictionary in the OPC UA server.

NOTE: The *NamespaceIndex* for all *NodeIds* published in the Modicon Communication Server address space and related to data access is **2**.

Refer to the part of this document that describes the information about how to create and configure an alias, page 32.

How the OPC UA Server Checks Symbol Link Consistency

On symbol link loading and modification, the OPC UA server checks for consistency between linked symbols and variables in the PAC application, as follows:

- On symbol link modification, the address space is checked for consistency:
  - For modifications that has not change the PAC application symbols layout (modification of the application, or new symbol creation) the symbol link is not reloaded.

    NOTE: New symbol creation is checked only if the **Dynamic consistency > New Symbol Detection** option is set as a General Property of the device alias, page 33, and applies only to an embedded data dictionary symbol link.
  - For modifications that modify the PAC application symbols layout (for example, symbol removal, symbol property edits, or symbol internal memory allocation modification) the symbol link is reloaded and the address space is updated.

Until the address space update is completed:

- the *OPC UA View Service Set* is not available.
- PAC communication required to fulfill *OPC UA Attribute Service Set* and *OPC UA MonitoredItem Service Set* is lost.

  NOTE: Loss of communication can be avoided where the symbol link is based on an embedded data dictionary, and preloading of the edited data dictionary is configured.

Configuring the No Communication Break of OPC UA Server During Data Dictionary Modification

A change in the data dictionary symbols layout (for example, symbol removal, symbol property edits, or symbol internal memory allocation modification) can cause loss of communication between the OPC UA server and OPC UA client

while the server undergoes a symbol link resynchronization (i.e. during which the variable definitions are updated).

It is possible to avoid this loss of communication by allowing the OPC UA server to preload the edited data dictionary before it is applied to the PAC application. To configure a data dictionary preload, follow these steps:

Step	Action
1	In EcoStruxure™ Control Expert, select **Tools > Project Settings...**. The **Project Settings** dialog opens.
2	Select **Project Settings > General > PLC embedded data**, then with **Data dictionary** previously selected, select **Preload on build changes**.
3	In the **Effective Build changes time-out (sec)** field, enter a value that exceeds the time required by the OPC UA server to update the variables definition.
4	Select **OK** to close the dialog.

Supported Data Types

EcoStruxure™ Control Expert Supported Variable Types

Modicon Communication Server provides access to symbolized EcoStruxure™ Control Expert variable instances of the following data type categories:

- Elementary Data Type (EDT)
- Derived Data Type (DDT)
- Arrays (single or multiple dimension)
- Device Derived Data Type (DDDT)
- Elementary Function Block (EFB) and Derived Function Block (DFB) inputs, outputs, inputs/outputs, and public parameters typed with any of the preceding types.

  NOTE: Generic Data Types (ANY_ARRAY x) are not supported.

Discoverable Variable Instances

The OPC UA server supports instances of variables declared in the following Control Expert locations:

- Global namespace
- Process namespace
- Safety namespace (read-only)
- Program Units, including public and external variables, inputs and outputs EDT parameters

Discoverable Objects

The following objects are supported:

- State RAM topological objects, page 78 (%MW, %M, %S, %SW, %KW, %I and %IW) can be populated by alias configuration.
- Local I/O objects configured with topological (versus DDDT) address: only symbolized I/O objects are populated.
- Local I/O objects configured with IODDT instances: only EcoStruxure™ Control Expert referencing IODDT field are populated.

Variable Data Type Conversion

The IPC UA server can discover and convert to OPC UA data types the following basic variable types supported by the EcoStruxure™ Control Expert data logic model:

EcoStruxure™ Control Expert Elementary Data Type	OPC UA Built-In Data Type
BOOL	Boolean
EBOOL	Boolean
INT	Int16
DINT	Int32
UINT	UInt16
UDINT	UInt32
REAL	Float
BYTE	Byte
WORD	UInt16
DWORD	UInt32
DATE	The display of time and string data types is determined by configuration settings entered for *Use native types* and *Use regional settings* in the PLC Software, page 36 window for the device alias.
TIME	
TOD	
DT	
STRING	

EcoStruxure™ Control Expert Variable Modeling

EcoStruxure™ Control Expert General Variable Structure

All EcoStruxure™ Control Expert variable instances, except elementary function blocks (EFBs) and derived function blocks (DFBs), are structured as follows:

<EntityName>=*"DataItem"*,<*VariableNameBrowsePath*>,<FinalName> , <VariableChild>

Where:

- <VariableChild >=*"DataItem"*,<*VariableNameBrowsePath*>,<FinalName>, <VariableChild>
- <VariableNameBrowsePath>=<AliasName>!<FullVariableBrowsePath>
- <FullVariableBrowsePath> describes a EDT instance with name <FinalName> or the full browse path used to access a final element of a DDT or Array instance for which the final name is <FinalName>.
  - The browse path to access a DDT element <DDTElement> of a DDT instance <DDTInstance> is formed with <DDTInstance>.<DDTElement> syntax.
  - The browse path to access an array element at index <Index> of an array instance <ArrayInstance> is formed with <ArrayInstance>[<Index>]

EcoStruxure™ Control Expert EFB, DFB and POU Variable Structure

EcoStruxure™ Control Expert EFBs, DFBs, and program organization units (POUs) with name <ContainerName> are structured as follows:

<EntityName>=*"FOLDER"*,<*ContainerNameBrowsePath*>,<ContainerName>, <VariableChild>

Where:

- <*ContainerNameBrowsePath*>=<AliasName>!< ContainerName>

Example

Consider the following example of a variable declaration made in EcoStruxure™ Control Expert:

```
DDT1 as Struct

aBool BOOL

anArray ARRAY[0...1, 0...1] of BOOL
```

For the variable *aDDT1*, which is an instance of DDT1 and is accessed with the alias *MyAlias*, the exhaustive list of populated nodes and the target node of the *Organizes ReferenceType* are presented below

Node	Target Node
NodeAlias	Node1
Node1	Node11
Node1	Node12
Node12	Node121
Node12	Node122
Node121	Node1211
Node121	Node1212
Node122	Node1221
Node122	Node1222

The associated Node attributes are described below:

Node	NodeId	BrowseName	DisplayName
NodeAlias	ns=2&&s=0:MyAlias	MyAlias	MyAlias
Node1	ns=2&&s=0:MyAlias!aDDT1	2:aDDT1	aDDT1
Node11	ns=2&&s=0:MyAlias!aDDT1.aBool	2:aBool	aBool
Node12	ns=2&&s=0:MyAlias!aDDT1.anArray	2:anArray	anArray
Node121	ns=2&&s=0:MyAlias!aDDT1.anArray[0]	2:anArray[0]	anArray[0]
Node12 2	ns=2&&s=0:MyAlias!aDDT1.anArray[1]	2:anArray[1]	anArray[1]
Node1211	ns=2&&s=0:MyAlias!aDDT1.anArray[0][0]	2:anArray[0][0]	anArray[0][0]
Node1212	ns=2&&s=0:MyAlias! aDDT1.anArray[0][1]	2:anArray[0][1]	anArray[0][1]
Node1221	ns=2&&s=0:MyAlias! aDDT1.anArray[1][0]	2:anArray[1][0]	anArray[1][0]
Node1222	ns=2&&s=0:MyAlias! aDDT1.anArray[1][1]	2:anArray[1][1]	anArray[1][1]

DataItem Node Attributes

Description Node Attribute

The *Description* node attribute for a *DataItem* is set to the property “Comment” of the matching variable configured in EcoStruxure™ Control Expert.

AccessLevel Node Attribute

The *AccessLevel* node attribute of a *DataItem* includes the following bits:

- Bit 0 (*Current Read*) is set to 1.
- Bit 1 (*Current Write*) is set to 0 if either of the following is true:
  - The matching EcoStruxure™ Control Expert variable *Constant* property is set to 1.
  - The **Option > Read Only** General property, page 33 of the device alias, to which the variable belongs, is selected.

  Otherwise Bit 1 is set to 1.

Effect of Byte Array Management On *Datatype*, *ValueRank* and *Value* Attributes

The following settings apply to the *Datatype*, *ValueRank*, and *Value* attributes for the following Control Expert categories, when **Byte array management > Manage as ByteString is not** set in the PLC Software, page 36 settings configuration:

Category	*Datatype*	*ValueRank*[1]	*Value*
EDT	Refer to Variable Data Type Conversion, page 75	*Scalar*	Scalar
DDT	Byte	*OneDimension*	Array
Array of EDT	Refer to Variable Data Type Conversion, page 75	*OneDimension*	Array
Other Array	Byte	*OneDimension*	Array
1. Scalar values can be negative, zero, or positive. OneDimension values can be 0 or positive.			

The following settings apply to the *Datatype*, *ValueRank*, and *Value* attributes for the following Control Expert categories, when **Byte array management > Manage as ByteString is** set in the PLC Software, page 36 settings configuration:

Category	*Datatype*	*ValueRank*[1]	*Value*
EDT	Refer to Variable Data Type Conversion, page 75	*Scalar*	Scalar
DDT	ByteString	*Scalar*	Scalar
Array of EDT	Refer to Variable Data Type Conversion, page 75	*OneDimension*	Array
Other Array	ByteString	*Scalar*	Scalar
1. Scalar values can be negative, zero, or positive. OneDimension values can be 0 or positive.			

DataItem Properties

DataItem Property Model

Every EcoStruxure™ Control Expert variable modeled as a *DataItem* has (0..*) *hasProperty ReferenceType* targeting a *Variable* node instance, which is defined as a Property (that is, the instantiated variable has a *hasTypeDefinition ReferenceType* targeting a *PropertyType* as *VariableType*).

The following notation is used to qualify a *DataItem* property of a node, for which NodeId String Identifier is 0:<*VariableNameBrowsePath*>

- <PropertyId>=<Description>,<DataType>

DataItem Property Patterns

The following patterns are used to set the OPC UA node attributes:

Node attribute	Value
NodeId Type	String
NodeId Identifier	2:<*VariableNameBrowsePath*>?<PropertyId>
BrowseName	2:<*Description*>
DisplayName	<*Description*>
DataType	<*DataType*>

DataItem Property List

DataItem properties can include the following:

<Proper-tyId>	<Description>	<Data-Type>
5000	The initial value of the variable.	Int32
5001	The kind of a variable: may be elementary, structured, function-block, section, and so forth.	String
5005	The size, useful for not elementary data types.	Int32
5007	The type name, as known by the programming tool.	String
5010	The free string of the variable.	String
5011	The topological address of the variable.	String
5012	Time stamp event support.	Bool

State RAM Topological Objects

Populating State RAM Topological Objects

You can populate EcoStruxure™ Control Expert device State RAM topological objects in the address space. To do this, select **Topological object > Populate** in the General properties, page 33 configuration for the alias of a EcoStruxure™ Control Expert device.

State RAM Topological Object Model Structure

The following structure is applied to State RAM topological objects:

<EntityName>= <Mapping>

Where:

- <Mapping>=“Folder”,<MappingBrowsePath>,”#Mapping”,<TopoObject>
- <MappingBrowsePath>=<AliasName>!#Mapping
- <TopoObject>=“DataItem”,<TopoObjectBrowsePath>,<TopoAddress>
- <TopoObjectBrowsePath>=<AliasName>!<TopoAddress>
- <TopoAddress> is the topological address of the variable (for instance: %MW10)

Specific DataItems

Diagnostic Supporting DataItems

Specific DataItems are *DataItems* not related to PAC variables, but which support diagnostics operations. *Specific DataItems* are populated only for a device alias that represents a Control Expert PAC.

Modeling Structure

Specific DataItems are structured as follows:

<EntityName>=<Specific>

Where:

- <Specific>=“Folder”,<SpecificBrowsePath>,”#Specific”,<SpecificItem>
- <SpecificBrowsePath>=<AliasName>!#Specific
- <SpecificItem>=“DataItem”,<SpecificItemBrowsePath>,<SpecificItemName>
- <SpecificItemBrowsePath>=<AliasName>!<SpecificItemName>
- <SpecificItemName> is the name of the Specific DataItem

Specific DataItems Collection

The collection of *Specific DataItems* and their relationship to the *DataType* attribute and to the *CurrentWrite* value of the *AccessLevel* node attribute are as follows:

<SpecificItemName>	Datatype	Current Write	Description
#AppliName	String	0	Application name of the PLC application
#AppliVersion	String	0	Application version of the PLC application
#DeviceIdentity	String	0	PLC product reference
#PLCQualStatus	Int16	0	See below, page 79
#SwitchPrimaryAddress	UInt16	1	See below, page 80

#PLCQualStatus

#PLCQualStatus supports monitoring the communication status, page 64 with a PAC device. Possible read-only values are:

Value (hex)	Description
0x00C0	Communication with the device is correct (Device state is GOOD).
0x0040	No communication with the device since less than the configured Device Timeout (Device state is UNCERTAIN.)
0x000C	No communication with the device since the configured Device Timeout (Device state is BAD).
0x0	Device is not identified ((Device state is Unknown or Missing).
0x0018	Device is inconsistent, page 73.

#SwitchPrimaryAddress

#SwitchPrimaryAddress supports monitoring the status of the PAC Link redundancy function, page 65. It can be used to trigger a switchover from the primary communication path to the standby communication path, provided the standby communication path is both configured and ONLINE. Possible values are:

Value	Description
0	Primary communication path is **Device address 1**.
1	Primary communication path is **Device address 2**.
2	Standby communication path is not configured.

NOTE: When the current value of *#SwitchPrimaryAddress* is:

- 0: You can change the primary communication path to **Device address 2** by using a write command to change the value to 1.
- 1: You can change the primary communication path to **Device address 1** by using a write command to change the value to 0.

#SwitchPrimaryAddress and DataValue

As defined in *OPC UA Part 4: Services* the *DataValue* type is used to transport a *Variable* value and a timestamp in both the *Read* service result and the *DataChangeNotification* data that is used for monitored items. In Modicon Communication Server, the *sourceTimestamp* reflects the timestamp applied when a value is received from the PAC or device (real time communication channel).

Possible *DataValue StatusCode* states include the following:

State	Description
GOOD	Communication with the device is correct and the primary communication path is ONLINE.
BAD	Communication with the device is interrupted, the primary and standby communication paths are OFFLINE.
UNCERTAIN	Communication with the device is correct, but the standby communication path is OFFLINE.

Troubleshooting

What’s in This Part

Troubleshooting

What's in This Chapter

Troubleshooting

Detected Issues and Possible Solutions

This table lists issues that may be detected in Modicon Communication Server and provides possible solutions:

Issue	Possible Cause	Solution
IP Timeout occurred.	IP is not reachable or slow connection.	Check IP is reachable. Diagnostics node displays the information about device communication. Separate log file also will have entry for device communication error.
Modicon Communication Server cannot read the configuration.	Modicon Communication Server Configuration Service and Modicon Communication Server UI Service has stopped.	Restart the Modicon Communication Server Configuration Service and Modicon Communication Server UI Service from Windows services.
No diagnostic messages are displayed under diagnostic node.	Modicon Communication Server is not running.	• Start the Modicon Communication Server if it is not running. • Start the Modicon Communication Server UI Service if it is not running. • Refresh the diagnostic message page to reconnect and load the content.
Unable to connect to server.	• Modicon Communication Server is not running. • Configured port is different or blocked. • Security settings are not configured properly.	• Start the Modicon Communication Server if it is not running. • Ensure the port of Modicon Communication Server is configured in OI Gateway. • Check the Security settings of Modicon Communication Server are matching with the OI gateway Security settings.
Modicon Communication Server is not communicating to the server.	The new configuration settings have not taken into effect.	Follow these steps: 1. Stop the Modicon Communication Server and then Deactivate the OI Gateway service. 2. Start the Modicon Communication Server and then Activate the OI Gateway service.
Modicon Communication Server immediately stops after starting the server without displaying any detected error message.	Port assigned to the Modicon Communication Server conflicts with other application and hence is not available. A message is displayed to port conflict.	Change the listening port in SMC under **Modicon Communication Server > Settings > Options > TCP Listening Port** with an available port.
SMC does not display settings or individual controller configuration and displays the page as: **"This page can't be displayed. Turn on TLS 1.0, TLS 1.1, and TLS 1.2 in Advanced settings ...**.	TLS settings are disabled on the system.	Launch **Internet Options** either from search bar in start menu or from **Internet Explorer > Tools > Internet options**. Navigate to **Advanced** tab and check below options are enabled: • Use TLS 1.0 • Use TLS 1.1 • Use TLS 1.2
The user interface of settings and devices under **Controllers** is not updated to the latest after migrating Modicon Communication	The user interface of Modicon Communication Server V2.01 is not updating due to temporary files of Modicon Communication Server V1.0 stored in a web cache.	Clear the temporary files and cache from the web browser settings.

Issue	Possible Cause	Solution
Server V2.01 from Modicon Communication Server V1.0.		
Modicon Communication Server service is not working on restart of the machine.	IP address of the machine on which the Modicon Communication Server is installed has changed, resulting in invalid security certificates.	• Repair Modicon Communication Server using installer to regenerate security certificates. • Deactivate and activate `OI.GATEWAY.3`. • Possible solution to avoid such scenario is to disable `IPv6` and assign static `IPv4` address to the machine where Modicon Communication Server is installed.
Unable to get data in AVEVA System Platform object viewer.	Control Expert variables are not set with HMI variable attributes.	Enable “HMI Variable” attribute for Control Expert variables intended to communicate with AVEVA System Platform using the Modicon Communication Server.

Cybersecurity

What’s in This Part

Cybersecurity

What's in This Chapter

What is Cybersecurity?

Introduction

Cyber threats are deliberate actions or accidents that can disrupt the normal operations of computer systems and networks. These actions can be initiated from within the physical facility or from an external location. Security challenges for the control environment include:

- diverse physical and logical boundaries
- multiple sites and large geographic spans
- adverse effects of security implementation on process availability
- increased exposure to worms and viruses migrating from business systems to control systems as business-control communications become more open
- increased exposure to malicious software from USB devices, vendor and service technician laptops, and the enterprise network
- direct impact of control systems on physical and mechanical systems

Guidelines for Modicon Communication Server

- Modicon Communication Server Configuration Service and Modicon Communication Server UI Service is running under your local system account. You have to configure and protect the system with recommendations.
- Windows Certification Store is managed by the user and the user have to protect it.
- By default, the self-signed certificates are used for securing the communication of Modicon Communication Server. However, you can configure the Modicon Communication Server and Wonderware System Platform communication channel to use the CA signed certificate.

For more details, refer to the Security Deployment Guide (see EcoStruxure™ Control Expert, Modicon Communication Server, Security Deployment Guide).

Index

V

W

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

EIO0000004083.03