Wonderware Operations Integration – Supervisory 3S CODESYS Server (G-1.2 Series)

Schneider Electric

Contents

WONDERWARE OPERATIONS INTEGRATION - SUPERVISORY 3S CODESYS SERVER (G-1.2 SERIES)... 3

Legal Information...4
Technical Support...5
Introduction to the CODESYS OI Server...6
Configuring the CODESYS OI Server...7
Configuring a Channel's Communication Settings...7
Setting a Device's Station ID...7
CODESYS OI Server Reference...11
Item Reference Syntax...11
Examples of Item References...12
CODESYS OI Server Error Codes...13
Appendix: Configure a CODESYS V3.x application to include symbols...16

Wonderware Operations Integration - Supervisory 3S CODESYS Server (G-1.2 Series)

This document describes the technical specifications and configuration options for the Wonderware® Operations Integration - Supervisory 3S CODESYS Server (or CODESYS OI Server, for short).

Last revision 2016-03-25

Legal Information

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

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

http://software.invensys.com/

For comments or suggestions about the product documentation, send an e-mail message to ProductDocumentationComments@schneider-electric.com.

Technical Support

Wonderware Technical Support offers a variety of support options to answer any questions on Wonderware products and their implementation.

Before you contact Technical Support, refer to the relevant section(s) in this documentation for a possible solution to the problem. If you need to contact technical support for help, have the following information ready:

- The type and version of the operating system you are using.
- Details of how to recreate the problem.
- The exact wording of the error messages you saw.
- Any relevant output listing from the Log Viewer or any other diagnostic applications.
- Details of what you did to try to solve the problem(s) and your results.
- If known, the Wonderware Technical Support case number assigned to your problem, if this is an ongoing problem.

Introduction to the CODESYS OI Server

These are the technical specifications for Wonderware Operations Integration - Supervisory 3S CODESYS Server.

Requirements

CODESYS OI Server requires Wonderware Operations Integration – Core G-1.2 or later.

Supported hardware and software

This OI Server enables TCP/IP Ethernet communication with any computer, controller, or device that hosts a CODESYS runtime, as long as the runtime is compatible with CODESYS PLCHandler 3.5 library.

For CODESYS application development, you can use the CODESYS Development System from 3S-Smart Software Solutions. For more information, go to: www.codesys.com

Conformance

The following hardware and software was used for conformance testing of this OI Server:

- Equipment:
  - Schneider Electric M241/M251
  - CODESYS SP WIN V3.5.1 Control
  - CODESYS SP PLCWinNT V2.4.4.0
  - Wago CPU 750-841
  - EATON XC-CPU202 CODESYS V3.5
  - Bosch Rexroth IndraLogic XLC L65 and IndraLogic L20

Configuring the CODESYS OI Server

Each server instance has its own hierarchy of objects, and each object has parameters that you need to configure in order to establish communication between the OI Server and individual devices on the network.

You can view a CODESYS server instance's configuration hierarchy under its **Configuration** node.

This section only describes how to configure object parameters for a CODESYS server instance. For more general information about adding and configuring objects, see "Configuring Your OI Server" in the *Operations Integration Server Manager Help*.

Configuring a Channel's Communication Settings

Configure the communication settings for a selected channel to ensure uninterrupted communication with the device network.

Advanced

> Click this button to open the *Advanced Settings* dialog box, which provides access to additional communication settings such as timeouts, retries, and buffer sizes. You might need to change these settings if the OI Server behaves unexpectedly during run time, but the default settings should work for most network configurations. For more information about these settings, see "Advanced Settings" in *Operations Integration Server Manager Help*.

Setting a Device's Station ID

Set the station ID for a selected device so that the OI Server can identify and communicate with it on the network.

Syntax

To connect to CODESYS V3.x device via CODESYS Gateway, use the following syntax:

```
[gateway IP address:]<device name or hexadecimal address>[:gateway port number]
```

To connect to a CODESYS V2.x device using TCP/IP communication via CODESYS Gateway, use the following syntax:

```
[gateway IP address:]TCP,<device IP address>[:device port number][,protocol (L4 or L2)]
```

To connect to a CODESYS V3.x device using ARTI3, use the following syntax:

```
ARTI3,<device name or hexadecimal address>
```

To connect to a CODESYS V3.x or CODESYS V2.x device in simulation mode using an exported symbol file, use the following syntax:

```
SIM,<symbol file>
```

The following syntax diagrams show all of the possible options:

For CODESYS V3.x device:

{ { | *gateway IP address*: }*device name or hexadecimal address*{ | :*gateway port number* } | ARTI3, *device name or hexadecimal address* | SIM, *symbol file* }

For CODESYS V2.x device:

{ { | *gateway IP address*: }{ TCP{ | ,{ | *device IP address*{ | :*device port number* } }{ | ,{ L4 | L2 } } } } | SIM, *symbol file* }

gateway IP address

The IP address of the CODESYS Gateway server that is managing communication with the CODESYS device(s). For CODESYS V3.x, if no address is specified, the default is 127.0.0.1 (i.e., localhost). For CODESYS V2.x, if no address is specified, it will connect directly using ARTI.

device name or hexadecimal address

The name (case sensitive) or hexadecimal address of the CODESYS device.

gateway port number

The port number of the CODESYS Gateway server that is managing communication with the CODESYS device(s). If no port is specified, the default is 1217.

device IP address

The IP address of the CODESYS device. If no address is specified, the default is 127.0.0.1 (i.e., localhost).

device port number

The port number of the CODESYS device. If no port is specified, the default is 1200.

protocol

Protocol to be used to communicate with the CODESYS device. The valid options are L4 (Level 4) and L2 (Level 2). If no protocol is specified, the default is L4.

symbol file

The file path to a symbol file (`*.sym` or `*.xml`) that has been exported from the CODESYS programming software. For more information, see the original CODESYS documentation, as well as the appendices at the end of this document.

Examples of valid station IDs for CODESYS devices

Example	Description
`192.168.1.10:0A56`	Connects to a CODESYS V3.x device at hexadecimal address 0A56, via a CODESYS Gateway server running at IP address 192.168.1.10 on default port 1217.
`127.0.0.1:0A56`	Connects to a CODESYS V3.x device at hexadecimal address 0A56, via a CODESYS Gateway server running at localhost on default port 1217.
`0A56:1480`	Connects to a CODESYS V3.x device at hexadecimal address 0A56, via a CODESYS Gateway server running at localhost on port 1480.
`MY-DEVICE-10:1480`	Connects to a CODESYS V3.x device named "MY-DEVICE-10", via a CODESYS Gateway server running at localhost on port 1480.
`192.168.1.10:0A56:1480`	Connects to a CODESYS V3.x device at hexadecimal address 0A56, via a CODESYS Gateway server running at IP address 192.168.1.10 on port 1480.
`192.168.1.10:TCP,192.168.1.50`	Connects to a CODESYS V2.x device at IP address 192.168.1.50 on default port 1200, using L4 protocol, via a CODESYS Gateway server running at IP address 192.168.1.10.
`192.168.1.10:TCP,192.168.1.50:1201,L2`	Connects to a CODESYS V2.x device at IP address 192.168.1.50 on port 1201, using L2 protocol, via a CODESYS Gateway server running at IP address 192.168.1.10.
`TCP, ,L2`	Connects to a CODESYS V2.x device at localhost on default port 1200, using L2 protocol over ARTI.
`TCP,192.168.0.50`	Connects to a CODESYS V2.x device at IP address 192.168.1.50 on default port 1200, using L4 protocol over ARTI.
`ARTI3,MY_PLC_NAME`	Connects to a CODESYS V3.x device named "MY_PLC_NAME", using ARTI3.

Example	Description
`SIM,C:\A\File.xml`	Connects to a CODESYS V3.x or CODESYS V2.x device in simulation mode, as specified in the CODESYS symbol file located at `C:\A\File.xml`.

Communicating with Schneider Electric SoMachine controllers

For Schneider Electric controllers based on CODESYS, such as the M241 and M251 from the SoMachine family of controllers, you can use the ARTI3 syntax described above. For example, to connect to an M251 PLC named M251_OEM:

```
ARTI3,M251_OEM
```

Communicating with Bosch controllers

Bosch Rexroth has two types of controllers: 1G and 2G. This section explains how to connect to them.

For Bosch 1G controllers, this OI Server was tested with the L40 1G PLC. For that controller, you can use the TCP syntax described above, where *<device IP Address>* is the IP address of the controller on the network. For example:

```
TCP,192.168.0.50:1200,L4
```

For Bosch 2G controllers, you need to install the Bosch 2G PLC Gateway. To do that, perform the following steps:

1. Run the IndraWorks setup. The installation wizard is displayed.
2. Click **Next** to see the list of products installed.
3. Select **Communication - IndraLogic XLC, IndraMotion MLC**.
4. Follow the installation wizard's instructions to complete the installation.

> **Note:** The gateway must be in the same subnet as the controller.

Once you have the gateway installed, you can use a modified version of the CODESYS Gateway syntax described above to connect to Bosch 2G controllers:

```
<gateway IP address>:<Bosch runtime address>:1317
```

gateway IP address

The IP address of the Bosch 2G PLC Gateway.

Bosch runtime address

This runtime address can be computed from the controller's IP address. It is composed of hexadecimal numbers representing the four fields of the IP address. For example, if the IP address is 192.168.23.32, the hexadecimal representations would be C0.A8.17.20 and the Bosch runtime address would be `0000.C0A8.1720`.

This OI Server was tested with the L65 2G PLC. For that controller, at IP address 10.168.24.51 and with the Bosch 2G PLC Gateway installed on the same computer (i.e., localhost) as this OI Server, the station ID would be:

```
127.0.0.1:0000.0AA8.1833:1317
```

Communicating with EATON controllers

For EATON controllers based on the XSoft v3.x software, you can use the ARTI3 syntax described above. For example, to connect to an EATON PLC named XC202_060c000:

```
ARTI3,XC202_060c00
```

For EATON controllers based on the XSoft v2.x software, you can use the TCP syntax described above. For example:

```
TCP,192.168.10.55:1200,L4
```

Communicating with WAGO controllers

For WAGO controllers based on the CODESYS V3.x software, you can use the ARTI3 syntax described above. For example, to connect to an WAGO PLC named PLC_758:

```
ARTI3,PLC_758
```

For WAGO controllers based on the CODESYS V2.x software, you can use the TCP syntax described above. For example:

```
TCP,192.168.10.25:2455,L2
```

CODESYS OI Server Reference

Use item references to access data stored in memory registers in connected devices, as well as to access standard system items in the OI Server itself.

This section only describes the item reference syntax and options for the CODESYS server. For more general information about item references, see "Managing Device Items" and "Item Reference Descriptions" in the *Operations Integration Server Manager Help*.

Item Reference Syntax

Item references in this OI Server use the following syntax.

CODESYS V3.x

For all variables in a CODESYS V3.x device, use the following syntax:

```
<application name>.<object name>.<variable name>
```

The following syntax diagram shows all of the possible options:
application name. *object name*. *variable name*

Where…
application name

The name of the CODESYS application.

object name

The name of the program organization unit (POU), global variable list, or other programming object that contains the variable. For example, **PLC_PRG**.

variable name

The name of the variable.

CODESYS V2.x

For local and global variables in a CODESYS V2.x device, use the following syntax:

```
[object name].<variable name>
```

The following syntax diagram shows all of the possible options:
{ *object name* }. *variable name*

Where…
object name

The name of the program organization unit (POU), global variable list, or other programming object that contains the variable. For example, **PLC_PRG**. This is required only if it is a local variable.

variable name

The name of the variable.

> 👉 **Note:** Even if it is a global variable without an object, you must include the period (.) before the variable name.

Notes

This OI Server supports all data types, but it cannot access specific parts of a variable. For example, it cannot read from or write to a single bit in an Integer variable. If you need that sort of functionality, modify your application to create more variables of the appropriate data types.

In order for this OI Server to communicate with the specified variables, the variables must be included as symbols in the CODESYS application. Use your application development software to make sure the symbols are included. The procedure to do this varies by device manufacturer and CODESYS version, but for an example of how to do it using the standard CODESYS V3.x software, see the appendix at the end of this document.

Examples of Item References

These are examples of valid item references for this OI Server. For more information about the referenced addresses, see the manufacturer's documentation for your device.

Item Reference		
CODESYS V3.x	CODESYS V2.x	
	Local	Global
Application.PLC_PRG.initPosition1	PLC_PRG.initPosition1	.initPosition1
Application.PLC_PRG.bHMIStart	PLC_PRG.bHMIStart	.bHMIStart
Application.PLC_PRG.Timer2[1,3,0].StartTime	PLC_PRG.Timer2.StartTime	.Timer2.StartTime

CODESYS OI Server Error Codes

The following tables describe the additional error codes that you might receive when poll/poke requests and operations fail.

Code	Description	Possible Causes	Solution
1	PLC not connected	• Lost connection to the PLC due to a hardware failure, such as PLC in error node, or cables issues. • Wrong Station field configuration.	• Check the **Station** field configuration, confirming that the IP Addresses for the Gateway (if it is used) and the PLC are correct, as well as the PLC ID number in hexadecimal format for CODESYS V3.x. • Check if the PLC is running and if you can ping it.
2	Login to PLC has failed	Some devices only allow a log-in of one application.	If there is another program connected to the PLC, such as CODESYS programming software, you need to disconnect it (i.e., log off). Then you should be able to communicate with the PLC.
3	No cyclic list has been found	Invalid list or no list variables to read.	Internal error related to the PLCHandler functions `CycDefineVarList` and `CycEnterVarAccess`.
4	PLCHandler is inactive	PLCHandler instance is not set active. This error happens when you use the INI file option and it is misconfigured.	Properly configure the INI file and the **Station** field.
5	Loading of the symbols has failed	There is no symbol configuration in the application.	Create the Symbol Configuration accordingly.
6	The defined communication interface is not valid or not supported	The interface is not supported (ARTI, Gateway). This error happens when trying to establish a connection with the PLC.	Check if your CODESYS configuration supports the desired interface (GATEWAY, ARTI, INI file).
7	Communication error ocurred during action	• Error while trying to start the communication with the PLC. • Exceeded number of retries to receive a response from the PLC before throwing a COMM_FATAL. Related to the PLCHanlder PlcConfig Struct.	Check if your PLC is properly configured and reachable.
8	Wrong or erroneous configuration of the PLCHandler	No configuration for this PLCHandler instance (Id unknown). This error happens when trying to establish a connection with the PLC and you are using a INI file that is not properly configured for that PLC instance.	Properly configure the INI file.
9	Invalid parameter	Invalid function parameters (for e.g. NULL). Usually happens when trying to retrieve the Variable Names from the PLC.	Internal error related to the PLCHandler functions `GetAllItems`, `GetItem`, and `CycEnterVarAccess`.
10	Communication interface not resp. Incorrectly installed (e.g., Gateway Dlls not available)	The interface can't start successfully (missing interface-dependent DLLs). This error happens when trying to establish a connection with the PLC.	If you are using the Gateway, check to see if it properly installed and running.
11	Method not yet supported resp. implemented	Spare error.	Not applicable.
12	Exception occurred during action	An exception occurred in the underlying interface. This error happens when transfering any application service to the PLC.	Internal driver error related to the PLCHandler function `SyncSendService`.
13	Timeout time exceeded	Time for the answer on a data package from the PLC exceeded. This could be caused by a wrong **Station** field configuration or the PLC is unreachable.	• Check the **Station** field. • Check if you can have access to the PLC using pinging and testing the TCP/IP ports.

Code	Description	Possible Causes	Solution
14	PLC already connected (at a further **`::Connect`** function call)	The driver tried to reconnect to a PLC that is already connected.	Internal error related to the PLCHandler function `Connect`.
15	Reconnect thread already active	Reconnect thread is still active. This error happens when trying to establish a connection with the PLC.	Internal error related to the PLCHandler function `Connect`.
16	Symbols available offline	Cannot open connection to the PLC but could load the symbol file offline. This error happens when trying to establish a connection with the PLC.	Internal error related to the PLCHandler function `Connect`.
17	Asynchronous operation	Asynchronous operation (e.g., cyclic read of variables) has not yet finished.	Internal PLCHandler error that should never happen on this driver. Contact technical support if this error occurs.
18	ActiveX error	Internal error.	The communication driver does not use this capability of PLCHandler, if you see this error it is probably a problem with the PLCHandler. Please contact technical support.
19	Target ID mismatch	PLC does not match to the passed target ID specified.	Use the programming software to scan the network and find the correct PLC ID.
20	Object not found	No object found for the required action (e.g., tried to get an element beyond the end of the list).	Contact technical support.
21	Components not loaded	No object found for the required action (e.g., tried to get an element beyond the end of the list).	Components required to establish communication are missing. Please contact your supplier to receive the additional files.
22	Busy	Last action still in progress, cannot start the required one.	• The driver tried to start a communication task before the previous one was completed. Contact the technical support. • If you are seeing intermittent communication problems because of this issue, please try increasing the number of retries.
23	Disabled	Driver tried to use the log feature but logging is disabled.	Contact technical support.
50	Invalid type	Results returned by the PLCHandler or specified by the driver are invalid.	Contact technical support.
51	Symbols not found	None of the variables specified match the symbols currently present in the PLC.	• Make sure that your symbols are properly added to the controller. • Verify if the name specified in the driver worksheet matches the variable name in the PLC.
52	Initialization error	The operating system does not have enough resources for the driver initialization.	Enable the protocol analyzer and run the driver again to retrieve further details.
53	Memory allocation error	• The driver could not allocate memory. • Internal programming error in the driver.	• Verify the memory available on your device. • If enough memory is available, contact technical support.
54	Driver is closing	Driver could not be initialized because it is in shutdown process.	Wait for until the driver close and then retry.
55	PLCHandler returned invalid code	PLCHandler function `GetLastError` returned zero after a read or write failure.	Contact technical support.

Code	Description	Possible Causes	Solution
0	OK	Communicating without error.	None required.
-15	Timeout waiting for message to start	• Disconnected cables. • PLC is turned off, in stop mode, or in error mode.	• Check cable wiring. • Check the PLC mode — it must be RUN.

Code	Description	Possible Causes	Solution
		• Wrong station number. • Wrong parity (for serial communication). • Wrong RTS/CTS configuration (for serial communication).	• Check the station number. • Increase the timeout in the driver's advanced settings. • Check the RTS/CTS configuration (for serial communication).

Appendix: Configure a CODESYS V3.x application to include symbols

This appendix describes how to configure a CODESYS V3.x application to include symbols, which other programs can use to communicate with the corresponding variables.

By default, a CODESYS V3.x application does not include symbols. You must add a Symbol Configuration object to your project, configure the object to include the variables that you want to communicate with, and then rebuild the application to create the symbol file.

To configure your CODESYS V3.x application to include symbols:

1. Open your project in the CODESYS V3.x application development software.
2. In the project explorer, right-click **Application**, and then on the shortcut menu, click **Add Object**.
   The *Add Object* dialog box is displayed.
3. From the list of objects, select **Symbol configuration**, and then click **Open**.
   A new Symbol Configuration object is added to your project, and it is opened for editing.
4. In the Symbol Configuration object, add the variables you want to communicate with — move them from the **Available variables** list on the left to the **Selected variables** list on the right.

   If you do not see your variables in the **Available variables** list, check the following:

   - For Local Variables (POU variables), the POU containing them must be called in a Task:
     1. Add a Task Configuration object to the application.
     2. Add a Task to the Task Configuration object.
     3. Add the POU to the Task.
   - For Global Variables, at least one of the variables from the Group must be used in at least one POU that is being called by one Task.
5. Close the Symbol Configuration object.
6. On the **Build** menu, click **Rebuild Application**.

Once your CODESYS application is configured to include the selected variables, it will automatically create the symbol file and then send it with the application to the runtime.

Note the location or file path of the symbol file. You might need to copy it to another location so that it can be accessed by other programs.