

# CODESYS ARTI v.4.18

# Driver Help

2025-08-15



# Table of Contents

<b>1. Introduction</b>	<b>5</b>
<b>2. Release Notes</b>	<b>6</b>
<b>3. Disclaimer</b>	<b>8</b>
<b>4. Limitations</b>	<b>9</b>
<b>5. Recommendations</b>	<b>10</b>
<b>6. Connecting to the Controller</b>	<b>11</b>
6.1. Ethernet	11
6.1.1. Ethernet Connection	11
<b>7. Settings</b>	<b>12</b>
7.1. General	12
7.2. Advanced	13
7.2.1. Commands	13
7.3. Stations	13
7.4. Additional Functionality	14
7.4.1. CODESYS Settings	14
<b>8. Addressing</b>	<b>16</b>
8.1. Using CODESYS v2	16
8.2. Using CODESYS v3	16
8.3. Name List Import	16
8.3.1. Using CODESYS ARTI v3 or Internal Controller	16
8.3.2. Using CODESYS v2	17
8.3.3. Project Options	17
8.3.4. Project Attributes	18
8.4. Data Types	18
8.4.1. Standard Data Types	18
8.4.2. User Defined Data Types	19
8.5. Bit Accessing of Integer Data Types	19
8.6. Station Addressing	19
8.6.1. Station Tag	20
<b>9. Routing</b>	<b>21</b>
<b>10. Import Module</b>	<b>22</b>
10.1. Driver-Specific Import Details	22
<b>11. Efficient Communication</b>	<b>24</b>
11.1. Packing of Signals	24
11.2. Waste	24
<b>12. Troubleshooting</b>	<b>26</b>



# 1. Introduction

This manual describes how to use the HMI driver together with a CODESYS V2/V3 runtime system that uses the ARTI protocol.

For information about the controller we refer to the manual for the current system.

## 2. Release Notes

Ver- sion	Release	Description
4.18	July 2025	Added support for new platform. Removed support for serial communication.
4.17	April 2025	Bug fix for controller tagnames with maximum length of 200 characters. Tagname added to Datatype mismatch error. Updated information for unsupported 8-bit data types. Security for the login credentials has been updated. Updating a driver from 4.15 to 4.16 or newer requires new credentials to be entered.
4.16	July 2024	Updated secure communication. Added node ID to IP address settings field. Added Datatype mismatch error message.
4.15	December 2023	Added support for secure communication with the PLC for the V3 TCP/IP protocol. Corrected problem with runtime controller settings. Updated driver help file. Removed support for End of Life HMI models.
4.14	May 2023	Updated driver interfaces. Methods for handling strings are replaced with more secure ones. Corrected problem on import of STRING<1> data type. Tag comments can now be imported from xml file. Corrected problem when using bit addressing in int, word, dint, dword etc. Corrected runtime crash problem due to iX addressing missing tags in (ARTI) controller. ELAU controller login failure fixed. Corrected the issue when the driver stops responding.
4.13	June 2020	Corrected value update issue after PLC online change. Corrected problem when writing to byte tags. Improved error message when invalid symbols are found.
4.12	July 2019	Corrected issue when loading recipes with more than 100 items.
4.11	June 2019	Corrected problem when writing string data. Corrected issue when using array tags.
4.10	March 2019	Corrected performance issues. Made it possible to disable symbol load during connect.
4.09	June 2018	Increased maximum address length to 200 characters.

Ver- sion	Release	Description
4.08	February 2018	Corrected problem when using momentary tags.
4.07	October 2016	Corrected problem when loading recipes. Corrected problem when using station addressing.
4.06	June 2016	Added support for new HMI platform.
4.05	May 2016	Added support for station indexing. Added support for enabling/disabling stations. Corrected problem when importing files from certain CODESYS versions.
4.04	June 2013	Corrected connection problem towards certain controllers.
4.03	May 2013	Updated CODESYS runtime version to 3.5. Added station handling.
4.02	December 2010	Support for Elau controllers.
4.01	September 2010	Updated import module to support array brackets []. Support for new HMI models.
4.00	November 2009	Initial version.

### 3. Disclaimer

Please note that changes in the controller protocol or hardware, which may interfere with the functionality of this driver, may have occurred since this documentation was created. Therefore, always test and verify the functionality of the application. To accommodate developments in the controller protocol and hardware, drivers are continuously updated. Accordingly, always ensure that the latest driver is used in the application.



## 4. Limitations

- Maximum device name length including separators, node and object names is 200 characters.
- Device name must not start with a digit.
- Tag address indexing is not supported.
- 64-bits data types are not supported.
- Only 1-dimensional ARRAY variables are supported.

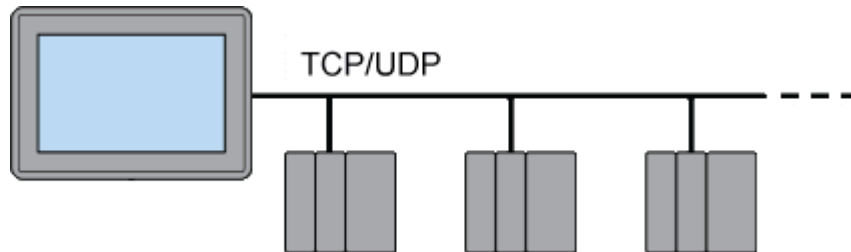
## 5. Recommendations

- Do not import more CODESYS tags to the HMI project than necessary since a large tag list can affect the communication performance towards the controller.
- When using data exchange from another controller to the ARTI driver it is recommended to use a timer and not value change event to trigger the data exchange. This is to reduce the number of write commands to the controller since these normally takes longer time than a read.

## 6. Connecting to the Controller

### 6.1. Ethernet

#### 6.1.1. Ethernet Connection



Connection in a network is made according to Ethernet standards.

To extend the network a switch may be used.



#### NOTE


When connecting to the controller, all included symbols are uploaded. Depending on the number of symbols, there could be a delay before the values are shown in the HMI.

For further information about settings in the controller, cable specifications and information about connecting the controller to the HMI we refer to the manual for the current controller.

## 7. Settings

The settings should correspond to the Communication Parameters in CODESYS.

### 7.1. General

Parameter	Default value	Description
Communication mode	Ethernet	Only Ethernet communication available.
Default station	1	Default value when no station is specified.
Update rate	200	The cyclic update rate of the driver in ms. Note that a low update rate time increases the CPU load in the HMI.
Byte order	Intel	Select if Intel or Motorola byte order is used. When using Ethernet this is set in the Stations tab.
ELAU controller	True	Select if connecting to an ELAU controller to set the special ELAU settings. When using Ethernet this is set in the Stations tab.
Load symbols when connecting	True	<p>If enabled the symbols from the controller will be loaded to the HMI during startup, this means that the connect time will be dependant of the number of variables in the controller.</p> <p>If disabled the variables will not be loaded meaning that the time it takes to connect will be shorter, however polling the variables will be slightly slower.</p> <div>  <b>NOTE</b>  Only used for V3 TCP/IP connections. </div>
Synchronize write data	True	<p>When enabled the data written in the HMI will be both sent to the PLC and synchronized with the drivers internal buffer. Can be disabled to force the driver to only write to the PLC and then await the reply before updating the buffer, for example when having rapid changes of the value in the PLC directly after a write operation.</p> <p>Note that some flicker can occur after a value is written if this setting is disabled.</p>

## 7.2. Advanced



Parameter	De-fault value	Description
Timeout	2000	The number of milliseconds of silence on the port before the next retry is sent.  Note that functions that uses the HMI as a gateway for passing on communication may require a higher Timeout value. Transparent mode, Routing, Passthrough mode, Modem and Tunneling are such functions.
Retries	3	Number of retries before a communication error is detected.
Offline station retry time	5	How long to wait after a communication error before trying to re-store communication.
Enable stations during startup	True	Sets if the configured stations shall be default active during project start. If set to true the driver will initiate communication to all stations with active controller tags and all ACTIVE-tags will be set to 1 (see chapter <a href="#">Addressing</a> ). If set to false the stations will be default deactivated, to initiate communication towards a certain station the ACTIVE tag needs to be set to 1 for the specific station.
Command line options		Special commands that can be passed to the driver. Available commands are described in the chapter <a href="#">Commands</a> below.

### 7.2.1. Commands

No commands are available for this driver.

## 7.3. Stations

Column	Default value	Description
Station	1	The controller station number.  Maximum number of stations that can be configured: 20 Value range: [0-255]
IP Address / Node ID	192.168.98.120	The address of the controller. When running CODESYS version 2 or version 3 using TCP connection this must be set to a valid IP address. When using CODESYS version 3 UDP connection this must be set to the controllers node ID or name. This is the same name/ID as in the CODESYS tool (see chapter <a href="#">Settings</a> ).
Port	11740	The port number to connect to in the controller. <b>Not used when connecting via V3 UDP/IP.</b>  <b>Commonly used port numbers:</b> <ul style="list-style-type: none"> <li>• Wago: 2455</li> <li>• CODESYS V2 SoftPLC: 1200</li> <li>• CODESYS V3 SoftPLC: 11xxx</li> </ul> Value range: [0-65535]

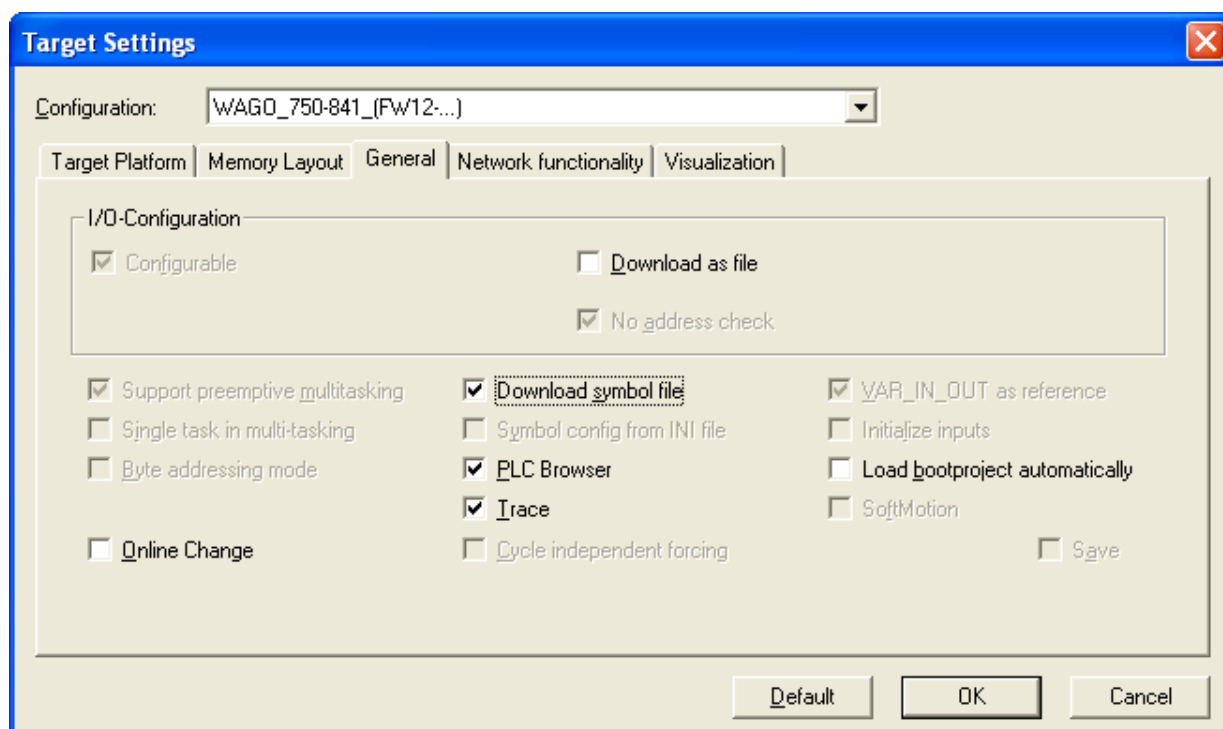
Column	Default value	Description
Protocol	V2 TCP/IP Level 2 Route	The protocol used to communicate with the controller.
Byte order (only for V2 TCP/IP protocol)	Intel	Select if Intel or Motorola byte order is used and which interface the controller uses for communication (Gateway or ARTI).
ELAU controller	False	Select if connecting to an ELAU controller to set the special ELAU settings.
User (only for V3 TCP/IP protocol)		<p>Used for secure communication with the PLC.</p> <p>The maximum length is 59 characters.</p> <p>Setting the station to use anonymous login can be done by deleting the login credentials or just deleting the username (using empty username).</p> <div>  <p><b>NOTE</b> When upgrading from version 4.15 to a later version, you must manually update the <b>User</b> and <b>Password</b> in the driver settings.</p> </div>
Password (only for V3 TCP/IP protocol)	*****	<p>Used for secure communication with the PLC.</p> <p>The maximum length is 59 characters.</p> <p>The password is hidden and "*****" string is shown instead. The password is encrypted after the input is finished.</p> <p>New password will be accepted only if the user press enter after entering the password.</p> <div>  <p><b>NOTE</b> When upgrading from version 4.15 to a later version, you must manually update the <b>User</b> and <b>Password</b> in the driver settings.</p> </div>

## 7.4. Additional Functionality

### 7.4.1. CODESYS Settings

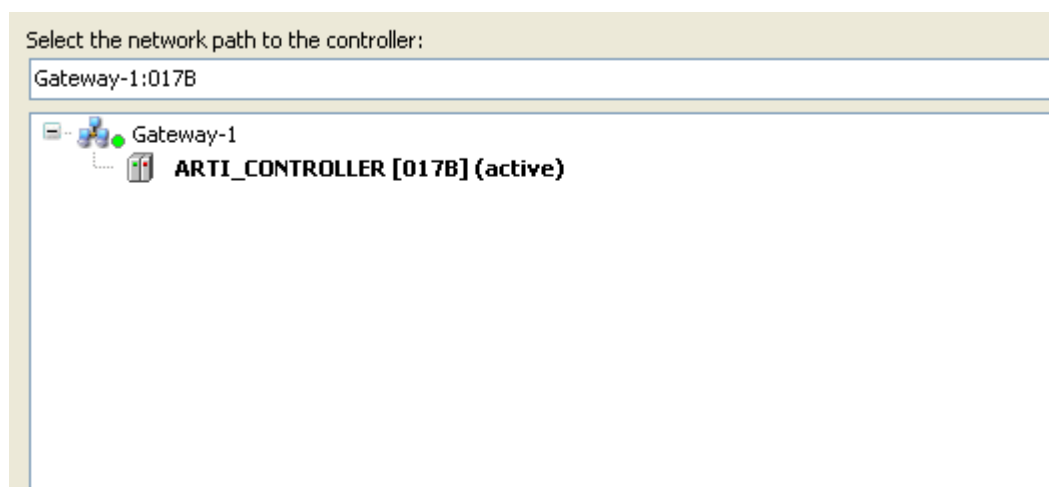
#### 7.4.1.1. Using CODESYS v2

When creating the project in CODESYS, check "Download Symbol File" (in Target Settings > General).



#### 7.4.1.2. Using CODESYS v3

The node name/ID can be seen under Communication settings in the device properties. Either the name or the ID inside the square brackets can be used.



## 8. Addressing

The available device names are created in the programming tool CODESYS.

The name of a variable created by the symbol configuration is composed according to the following syntax:

### 8.1. Using CODESYS v2

*<POU name>.<variable name>*

E.g. "PLC\_PRG.LocalVar".

Global variables are composed according to the following syntax:

*.<variable name>*

E.g. ".GlobVar"

### 8.2. Using CODESYS v3

*<application name>.<POU name>.<variable name>*

E.g. "MyApplication.PLC\_PRG.a" or "MyApplication.GVL.a"

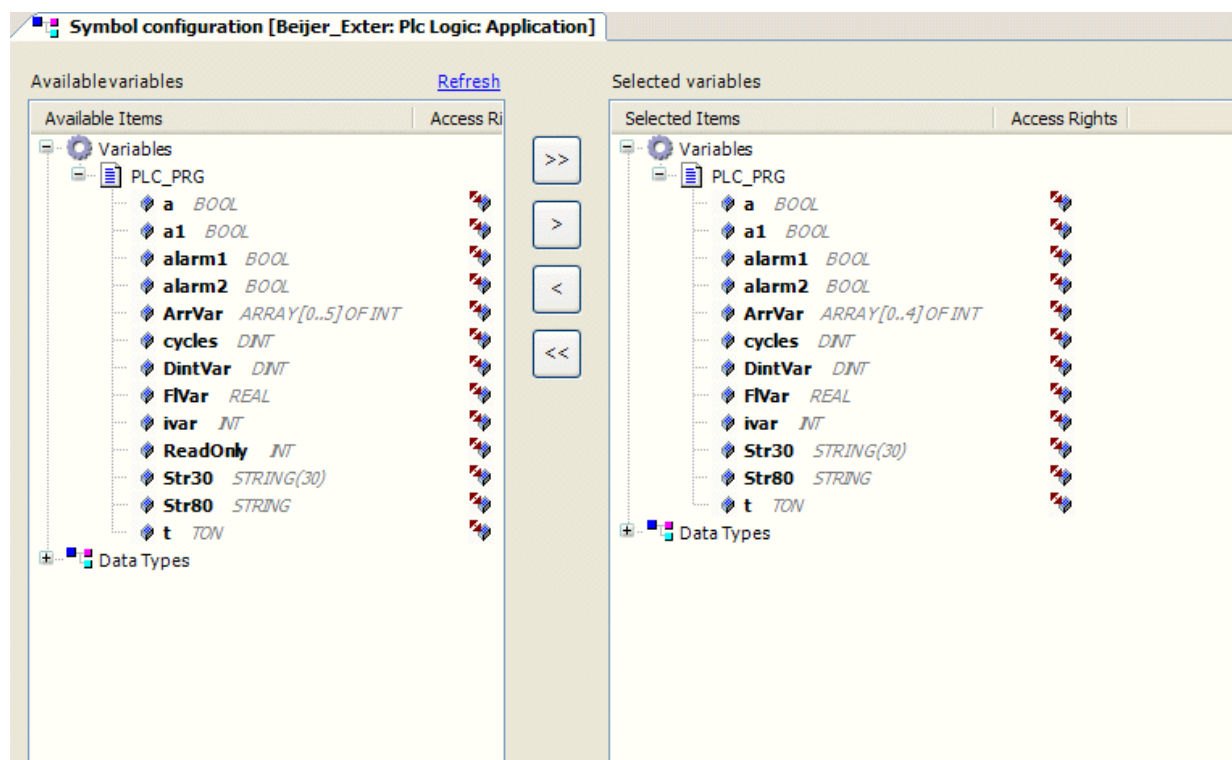
For the purpose of accessing the variable, the symbol name must always be specified completely.

The variables must then be imported to the HMI programming tool, see chapter [Name List Import](#).

### 8.3. Name List Import

#### 8.3.1. Using CODESYS ARTI v3 or Internal Controller

All variables needed in the HMI must be selected in the Symbol configuration in CODESYS.





The symbols defined for an application get exported to a xml-file in the CODESYS project directory. This file is named according to the following syntax:

*<project name>.<device name>.<application name>.xml*

E.g. proj\_xy.Device.application.xml.

The XML-file is used for import in the HMI programming tool.

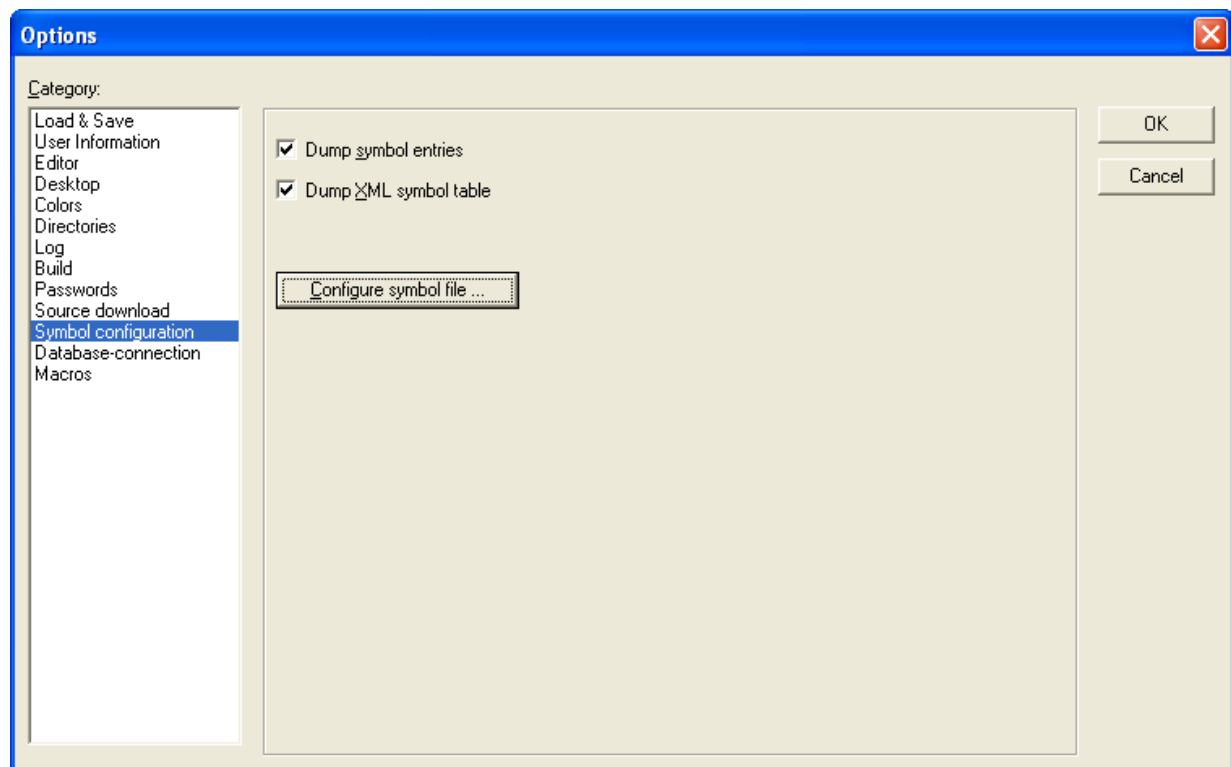
**NOTE**

Do not include more variables than necessary in the Symbol configuration.

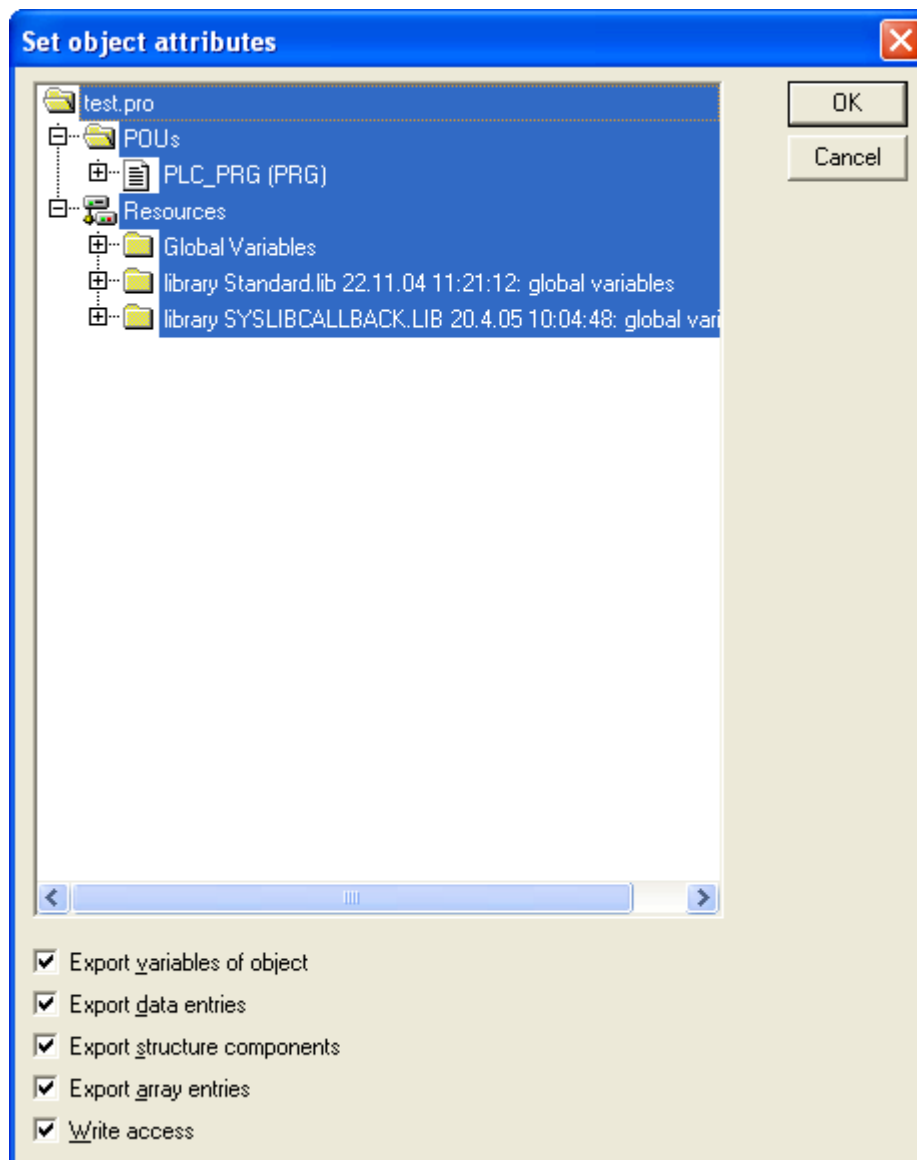
### 8.3.2. Using CODESYS v2

The symbols defined for an application get exported to a SDB-file (Symbol Data Base) in the CODESYS project directory. Make sure all variables are marked for export in CODESYS. See settings below.

### 8.3.3. Project Options



### 8.3.4. Project Attributes



The SDB-file is used for import in the HMI programming tool.

## 8.4. Data Types

The import module supports variables of standard data types and user defined data types.

### 8.4.1. Standard Data Types

The following data types in the CODESYS programming tool are considered standard data types by the import module:

- BOOL
- WORD
- DWORD
- INT
- UINT

- UDINT
- DINT
- STRING \*
- REAL
- TIME (will be imported as unsigned 32-bit)
- ...and 1-dimensional ARRAY of the types above

\* String length for a STRING variable in the PLC should be maximum 80 characters. Declare a STRING variable either with a specified size (str:STRING(35)) or default size (str:STRING) which is 80 characters. The size for a STRING variable in the HMI must be the same as the corresponding STRING size in the PLC.

The BYTE, SINT and USINT data types are not supported independent of the setting of the **Load symbols when connecting** parameter in the **Settings** menu. When it is set to **True**, using these data types will cause a **Type mismatch** error. A write will cause a **Write failed** error. When set to **False**, only the low byte is valid on a read and only for USINT and BYTE. For the signed data type SINT the data shown is invalid. A write will cause a **Write failed** error.

The 64-bit data types LWORD, LINT and LREAL are not supported.

For all functions in the HMI where the device addresses are calculated from a base address, only ARRAY variables can be used.

#### 8.4.2. User Defined Data Types

The import module also handles user defined types if the data type contains standard data types above. If variables of structured data types are used, one device per structure member is created.

### 8.5. Bit Accessing of Integer Data Types

Bit accessing of integer data types is supported in the HMI for digital devices. The syntax is:

*<POU name>.<variable name>.<BitNo>*

E.g. "PLC\_PRG.LocalVar.5".

The bit expansion of integers are not included in the file used for import, and therefore also not in the Name List.



#### NOTE

Writing bit accessed integer variables are less efficient, since read before write has to be done i.e. reading the WORD/DWORD, masking the bit and finally writing the WORD/DWORD. Therefore bit accessed integers should be used with caution.

### 8.6. Station Addressing

For communication with other Ethernet stations than the default one, the station number is given as a prefix to the device. The station number is referring to a specific Ethernet station ID. Both static and indexed values can be used for station addressing.

#### Example:

05:PLC\_PRG.VariableName addresses variable PLC\_PRG.VariableName in Ethernet station 5.

I2:PLC\_PRG.VariableName addresses variable PLC\_PRG.VariableName in the Ethernet station defined by the value in index register 2. For more information about indexing please see the manual for the designer.

PLC\_PRG.VariableName addresses variable PLC\_PRG.VariableName in the default station.

### 8.6.1. Station Tag

The x:ACTIVE-tag is a bit tag that makes it possible to enable/disable certain stations. The x corresponds to the station number for the defined station.

The station is enabled if the value for the station tag is set to 1 and the driver will then communicate with the addressed node. If the value is set to 0 the station is deactivated meaning that no requests will be sent out to the specific node, the values shown in the HMI will then be the last read values from the node.

During start up of the project all station tags is set to default 1 (active).

#### Example:

1:ACTIVE = 1

2:ACTIVE = 0

The tag values will be updated for station 1 but not for station 2.

1:ACTIVE = 1

I1:ACTIVE = 0

Station 1 is active, station defined by the value in index register 1 is deactivated.

## 9. Routing

The driver does not support any routing mode.

## 10. Import Module

The import module is a component of the driver that enables automatic import and parsing of tag definitions from a PLC project file or directly via communication with the PLC. Within iX Developer, the driver uses the import module to retrieve tag names, data types, memory addresses, and structured data types from the PLC. This facilitates seamless integration and mapping between the PLC and the control or SCADA system, reducing manual configuration efforts and ensuring consistency between engineering and runtime environments.

### 10.1. Driver-Specific Import Details

The import module makes it possible to import symbols and structs from XML/SDB export files generated by CODESYS. To use this feature, select **CoDeSys V2/V3 import format** in the **Import Tags** dialog in iX Developer.

It is possible to define more than one station to which the variables shall be imported to. This can be used if more than one controller has the same defined variables.



#### NOTE

The maximum device name length in CODESYS including separators, node and object names is 200 characters, however the tag name length in the designer is calculated as [path to the project files] + tag name. This means that if the length of the path is long the device name length might need to be shorter than 200 characters.

When an export file is selected the following dialog appears:

Import to station(s)

☒ Add variables for default station

Enter the stations and/or station ranges the variables shall be imported to separated by commas.  
For example 1,4,10-16  
Variables will be added with the station number in front of the name and address.

If Add variables for default station is checked variables will be imported and added without an initial station number.

For more information please see the driver manual.

OK Cancel

In the "Import to station(s)" field, stations where variables are to be imported can be entered. If more than one station is needed, separate them with a comma. Stations can be defined individually or as a range.

The variables will be added with the station number as a prefix to the name and address, e.g. 2\_VariableName and 2:VariableAddress.

If the "Add variables for default station" is checked, variables from the file will be added without an initial station number before name and address.

**Example 1:****Station(s): 1-3,6**

- Add variables for default station : yes

**Added tags:**

- VariableName, VariableAddress
- 1\_VariableName, 1:VariableAddress
- 2\_VariableName, 2:VariableAddress
- 3\_VariableName, 3:VariableAddress
- 6\_VariableName, 6:VariableAddress

If only the default station variables are to be added the Station(s) field can be left blank.

**Example 2:****Station(s):**

- Add variables for default station : yes

**Added tags:**

- VariableName, VariableAddress

**NOTE**

Importing tags is provided for your convenience. You can always manually enter tags from within iX Developer. If you run into any issues with importing tags, simply manually enter the tags within iX Developer.

It is generally recommended to import only the tags, that will be used from your HMI to iX Developer. Importing all available tags from the import file may lead to problems with iX Developer.

## 11. Efficient Communication

### 11.1. Packing of Signals

When tags are transferred between the driver and the controller, all tags are not transferred simultaneously. Instead they are divided into messages with several tags in each message. By decreasing the number of messages that have to be transferred, the communication speed can improve. The number of tags in each message depends on the used driver.

**NOTE**

ASCII Strings and arrays are packed into one message for each object.

**NOTE**

Having different pollgroups will affect how requests are generated.

### 11.2. Waste

To make the message as efficient as possible, the waste between two tag addresses must be considered. The waste is the maximum distance between two tag addresses that you can have and keep them in the same message. The waste limit depend on the used driver.

**NOTE**

Waste is only valid for number-based addressing, not for Name-based addressing.

**NOTE**

Waste can only be calculated between two similar datatype tags, not between different datatypes tags.

#### Scenario 1

When integer tags with address 4, 17, 45, 52 are used with waste limit of 20, this will end up creating two messages.

First message with address 4 and 17 (tag address difference is  $13 \leq 20$ ).

Second message with address 45 and 52 (tag address difference is  $7 \leq 20$ ).

**Reason:** Difference between 17 and 45 is more than the waste limit of 20, hence creating 2nd message.



**Scenario 2**

When integer tags with address 4, 17, 37, 52 are used with waste limit of 20, this will end up creating one message.

**Reason:** Difference between the consecutive tags is less than or equal to the waste limit of 20, hence creating one message.

**Conclusion**

**Scenario 2** is more efficient than scenario 1.

## 12. Troubleshooting

The meaning of error messages from the controller shown by the driver.

Error message	Explanation
Invalid variable	A variable not existing in the controller is accessed or the symbols have not been loaded yet by the controller.
PLC not connected	PLC not connected or no PLC program exists.
Loading Symbols failed	The PLC is connected but the symbols has not yet been loaded.
No symbol file	The symbol configuration has not been downloaded to the controller.
Trying to reconnect	The connection to the PLC has been lost or no PLC program exists.
Read failed	Reading of the variable(s) failed. The variable(s) may not exist in the PLC program.
Write failed	Writing of the variable(s) failed.
Variable is read only	Write failed, variable is read only.
Login Failed	Failed to connect to the PLC (wrong credentials).
User access denied	Failed to connect to the PLC (wrong credentials).
x: Invalid station ID	The driver is trying to access a device in a station that is not defined in the Stations configuration.