# Start Up

Beijer ELECTRONICS

## WARP Engineering Studio, Getting started

KI00373 2016-04

## 1 Function and area of use

*WARP Engineering Studio* is the common software platform for Beijer Electronics hardware providing seamless integration of our products. The graphical configuration tool creates single or multiple projects with our HMI and PLC programming tools. Failsafe and efficient with auto-configuration and smart objects, WARP Engineering Studio helps the user to speed-up engineering!




## 2 About this start up document

This Start-Up document is not a complete manual. It's a getting started document explaining the basics of *WARP Engineering Studio* and how to use this in combination with *iX Developer* and *CODESYS 3.5* based applications from our product range.

Note that screen copies may differ from the version of software you are using, the examples are only intended to show the general functionality.

*WARP Engineering Studio*, may hereafter be named *WARP* in this document!

This and other start up documents can be provided from our homepage.
For feedback, please use the address: manual@beijer.se

**Copyright © Beijer Electronics, 2016**
*This documentation (below referred to as 'the material') is the property of Beijer Electronics. The holder or user has a non-exclusive right to use the material. The holder is not allowed to distribute the material to anyone outside his/her organization except in cases where the material is part of a system that is supplied by the holder to his/her customer. The material may only be used with products or software supplied by Beijer Electronics. Beijer Electronics assumes no responsibility for any defects in the material, or for any consequences that might arise from the use of the material. It is the responsibility of the holder to ensure that any systems, for whatever applications, which is based on or includes the material (whether in its entirety or in parts), meets the expected properties or functional requirements. Beijer Electronics has no obligation to supply the holder with updated versions.*



**Head office**
Beijer Electronics AB
Box 426, Stora Varvsgatan 13a
SE-201 24 Malmö, Sweden
www.beijerelectronics.com | +46 40 35 86 00

KI_eng_130531.dot

# 3 Supported hardware and software

*WARP Engineering Studio* supports the following hardware:

- iX TxX panels, as well as rugged and marine panels
- iX TxX SoftControl and SoftMotion panels
- BFI frequency inverters
- BSD servo
- Crevis remote I/O (EtherCAT and Modbus)
- Crevis programmable I/O
- NEXTO modular PLC
- Industrial data communication (Korenix), Ethernet Switches

The main communication interface supported by WARP in above hardware are:

- MODBUS RTU
- MODBUS TCP
- EtherCAT

Software's that can be installed and launched by WARP:

- CODESYS 3.5 SP8 (support IEC 61131-3 PLC programming languages)
- iX Developer 2.20.2296 (HMI programming tool) and the update 2.20 SP1
- BCS Tools 2.05 (support IEC 61131-3 PLC programming languages)
- IOGuidePro (Crevis distributed I/O configuration tool)
- BSD Tool (separate tool to configure, tune and diagnose the servo applications)
- BFI Tools (separate frequency inverter configuration tool)
  http://www.drive-support-studio.com/OTS/Beijer/downloads/deploy/BFI_Tools.htm

# 4 Table of contents

**1 Function and area of use ..... 1**

**2 About this start up document ..... 1**

**3 Supported hardware and software ..... 2**

**4 Table of contents ..... 3**

**5 Basic functionality and specification ..... 5**

*5.1 First step! ..... 5*
*5.2 System requirements ..... 5*
*5.3 Installation of WARP Engineering Studio ..... 5*
*5.4 Device files, smart objects and other software tools ..... 7*
*5.5 User interface ..... 8*
*5.5.1 Launch ..... 8*
*5.5.2 Ribbon menu ..... 9*
*File ..... 9*
*Home ..... 10*
*View ..... 12*
*Tools ..... 13*

**6 Functionality ..... 14**

*6.1 Bill of materials ..... 14*
*6.2 Documents ..... 15*
*6.3 Product catalog ..... 16*
*Search and filter the product catalog ..... 16*
*6.3.1 Change version ..... 17*
*6.4 Property grid ..... 18*
*6.4.1 Device identification name ..... 18*
*6.4.2 Distributed IO properties ..... 19*
*Power ..... 19*
*Port settings ..... 20*
*Misc. - slots ..... 20*
*6.4.3 Distributed IO modules initial settings ..... 21*
*6.4.4 BFI Inverter select Expansion Module ..... 21*
*6.4.5 BFI Inverter initial parameters ..... 22*
*6.4.6 Serial communication settings ..... 22*
*6.5 Scan network ..... 23*
*6.6 Show messages ..... 23*
*6.7 Show variables ..... 24*
*6.8 Restore points ..... 25*
*6.9 Auto resolve ..... 26*
*6.9.1 Correcting connections between devices ..... 26*
*6.9.2 Correcting the order of I/O modules ..... 27*

**7 Project example ..... 28**

*7.1 New project ..... 28*
*7.2 Add devices to Workspace ..... 28*
*7.3 Build up the network modules ..... 29*
*7.4 Connect the device in the workspace ..... 30*
*7.5 Add a smart object to the SoftControl ..... 31*
*7.6 Generate project ..... 31*
*7.6.1 Saving the new project - Save As ..... 32*
*7.7 Messages during generate ..... 32*
*7.7.1 Generate message box ..... 32*
*7.8 Variable declarations and WARP objects in CODESYS ..... 33*
*7.9 Generate the HMI project ..... 34*
*7.10 Download to device ..... 34*
*7.11 Zoom in workspace ..... 34*
*7.12 Device identification name ..... 35*
*7.13 Making name change before generate ..... 35*
*7.14 Use the variables in PLC code ..... 36*
*7.15 Tip! SmartCoding in CODESYS ..... 36*

**8 Project backup ..... 37**
*8.1 Save project ..... 37*
*8.2 Project archive ..... 37*

**9 Appendix ..... 38**
*9.1 Notes regarding Generate, Regenerate, Download and Upload ..... 38*
*9.2 How to check compiler and library versions in CODESYS 3.5 ..... 39*
*Check CODESYS compiler version ..... 39*
*Check a device version ..... 39*
*Check CODESYS library version ..... 39*
*9.3 CODESYS project objects ..... 40*
*How to add a POU ..... 40*
*How to add a Task ..... 40*
*How to add a POU to a Task ..... 41*
*9.4 Trouble shooting SW use logfiles ..... 41*
*9.5 Trouble shooting in the HW communication ..... 41*

# 5 Basic functionality and specification

## 5.1 First step!

For better understanding and use of *WARP Engineering Studio* we recommend:

- Basic knowledge of *CODESYS* and *iX Developer*. There is a quick start in the help files of the software's that can be of good help.
- Prepare the computer with the programming tools needed for the device and configuration you are about to use. Software and device packages can be downloaded and installed from the "Smart Store" selection in WARP main menu.
- Download product manuals, brochures, drawings etc. from our web page. Or download from inside the Documents window in WARP by activating the ribbon View - Documents.
- Pressing [F1] in WARP will open the Getting started document.

## 5.2 System requirements

Recommended requirements running WARP, iX Developer, CODESYS 3.5 and BCS Tools:

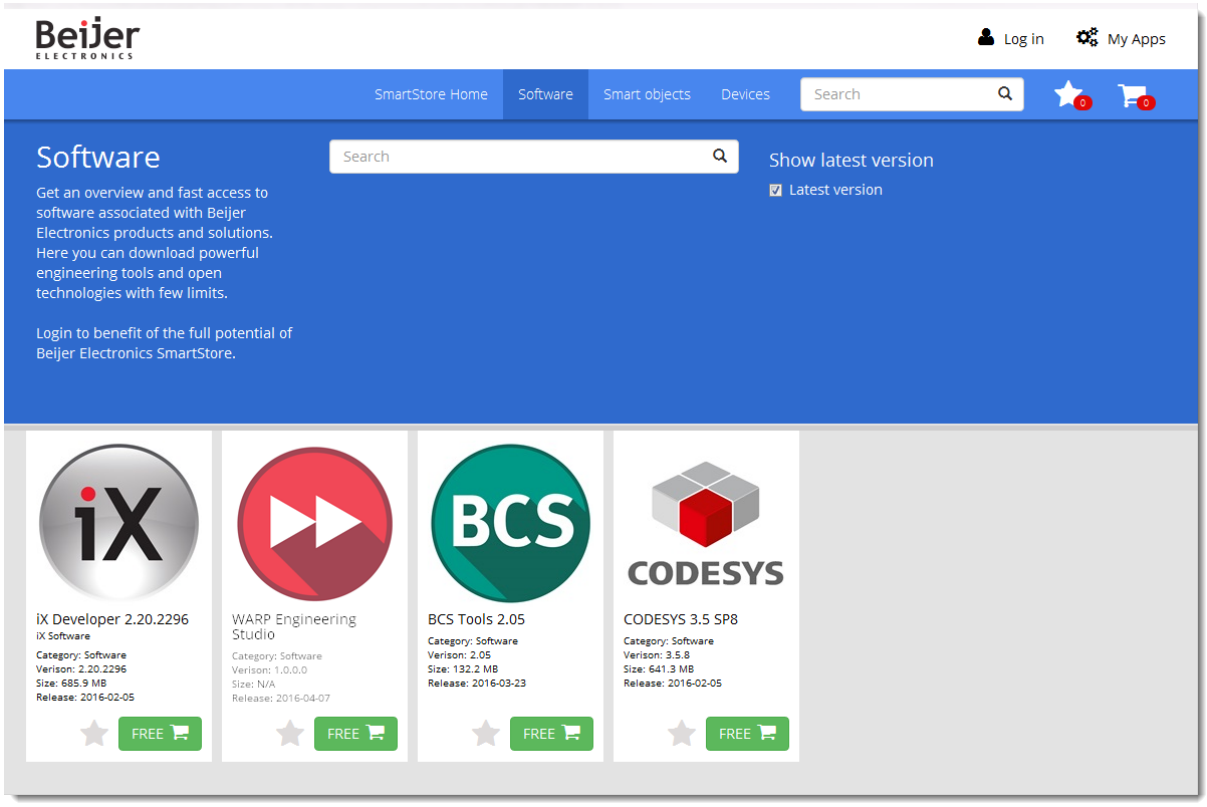
- Operating systems: Windows 7, Windows 8 or Windows 10 (32/64-bit)
- Computer: 8GB RAM, Intel i5 or equivalent

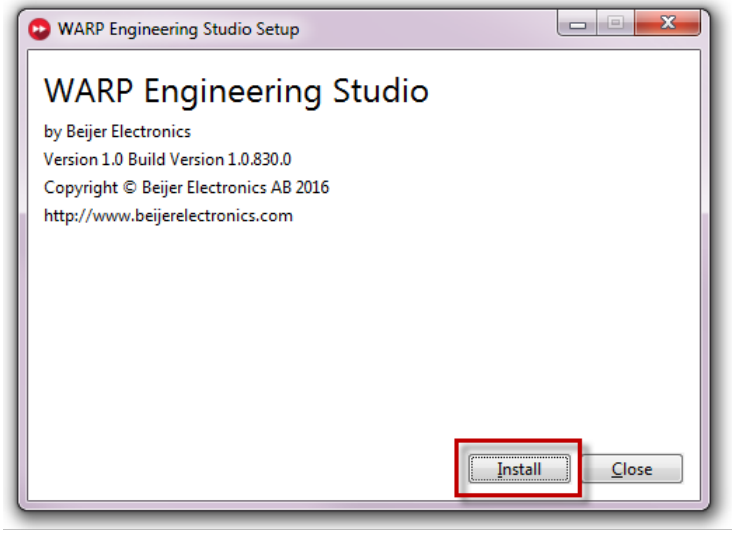
Minimum requirements:

- Operating systems: Windows 7 32-bit
- Computer: 4 GB RAM, Intel Core2Duo

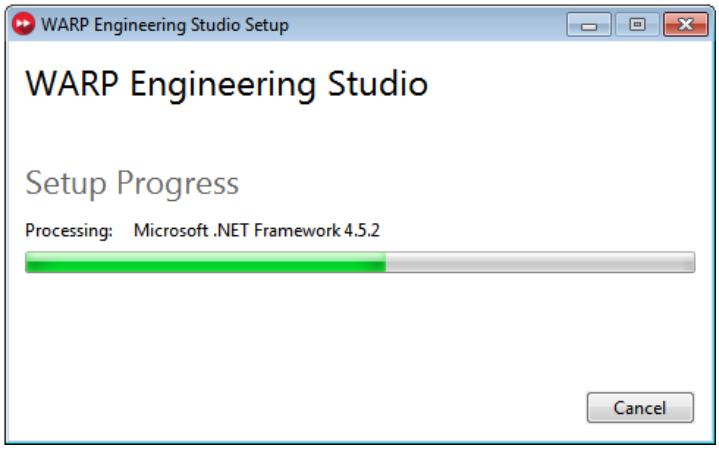
## 5.3 Installation of WARP Engineering Studio

Download the WARP installation from *Smart Store* and run the installation file *WARPSetup*.

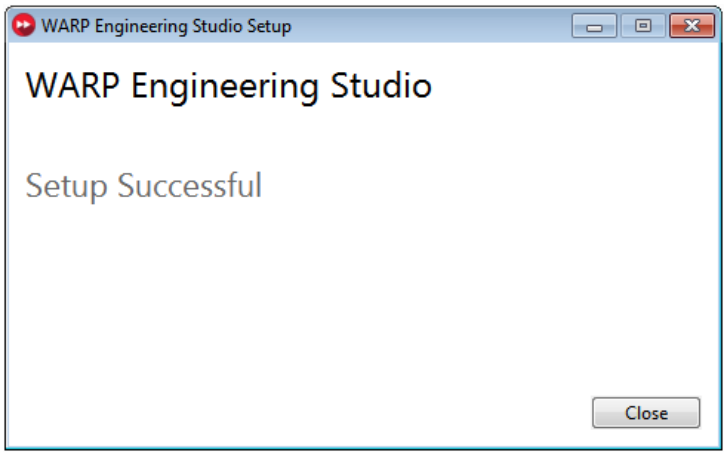





Please note, that during installation the Microsoft .NET Framework** may be installed as well. Be patient as this can take some time if not installed before, it's downloaded automatically when the computer is connected to Internet.




When the installation of WARP has finished this massage will appear.

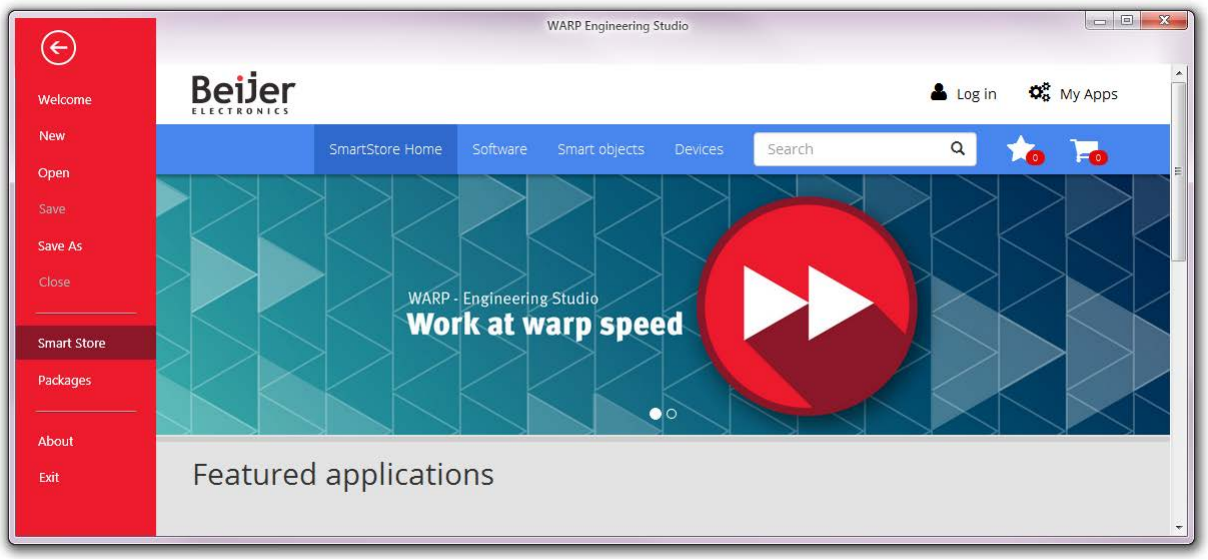



** The .NET Framework is an application development platform from Microsoft that provides many services and class library for application development. Because your computer is using Windows operating system the .NET Framework is most likely already installed. Though, when installing an application that requires the .NET Framework, the application's setup program may install a specific version of the .NET Framework. You do not need specific knowledge about the .NET Framework or its operation, for the most part .NET Framework is completely transparent to the user.

## 5.4 Device files, smart objects and other software tools

The device and software tools are very easy to install after WARP has finished installation. From the Main menu of WARP is a direct link to the download area of Smart Store.

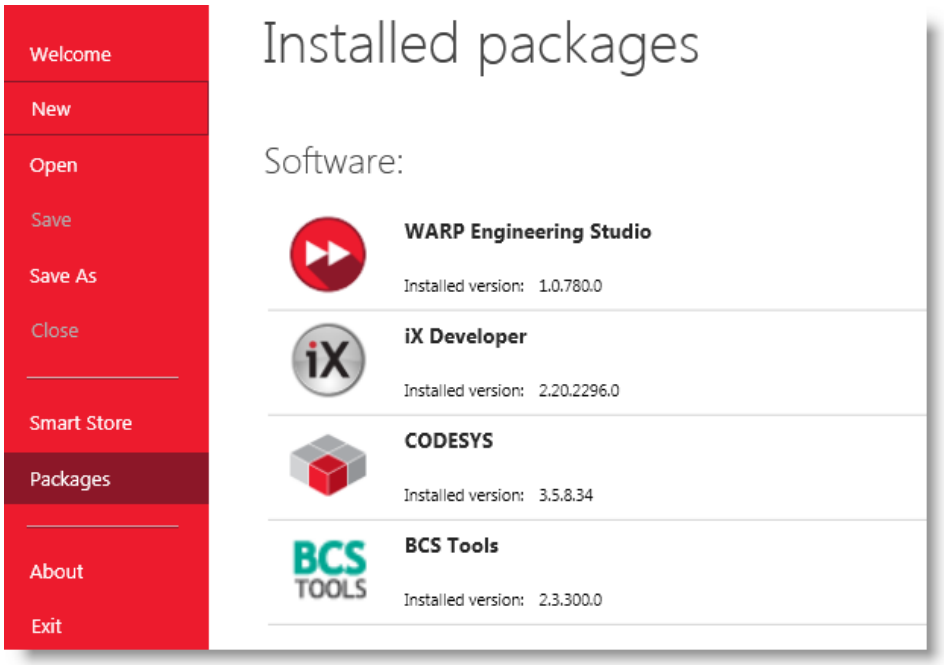
During download of a WARP package from Smart Store, the user will be asked to install the component or software. From Smart Store the user can download package installation files (*.warppackage) of hardware and software devices for the Product Catalog in WARP.




In the Package view of WARP, user may manually install / uninstall the packages.

Installed versions of software tools supported by WARP are displayed in this view, for example iX-Developer, CODESYS and BCS Tools.

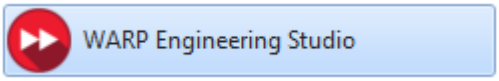
All downloaded device packages of the product catalog are displayed here too.



## 5.5 User interface

### 5.5.1 Launch

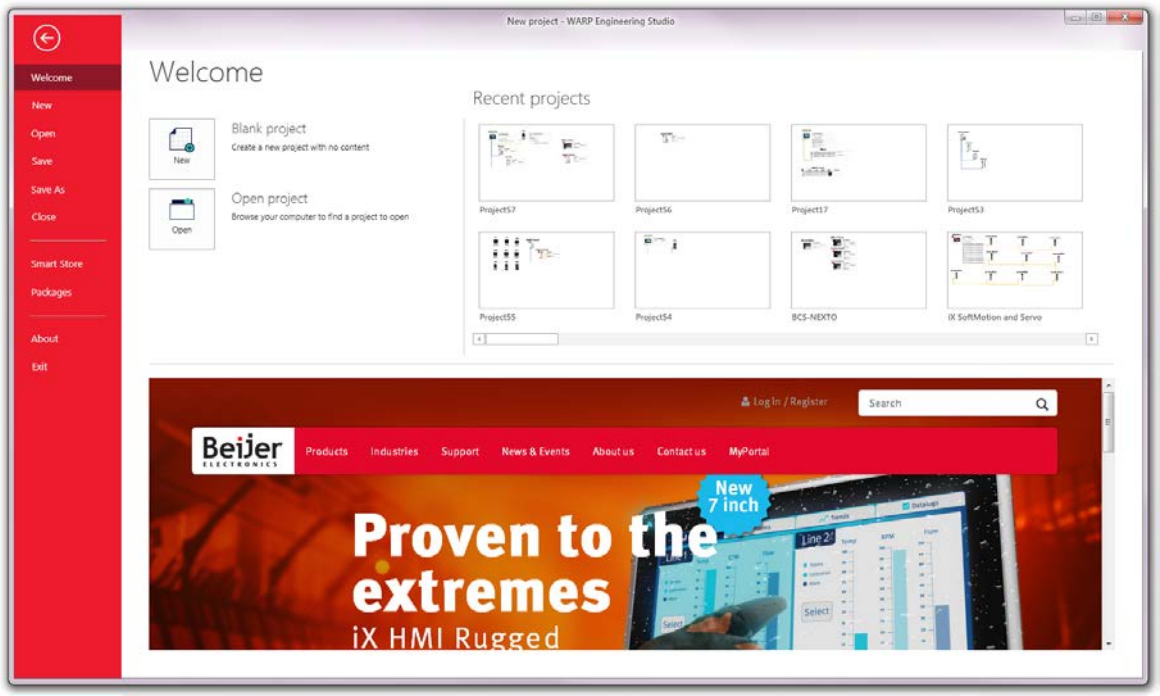
After finish, the installation the icon of WARP Engineering Tool can be found on the Start Menu.




When launching WARP, the splash screen is initially shown.




WARP will start up with a Welcome screen. It gives a fast access to create Blank project or Open project from a quick selection. In the right pane is the graphical selection of recent projects.



## 5.5.2 Ribbon menu

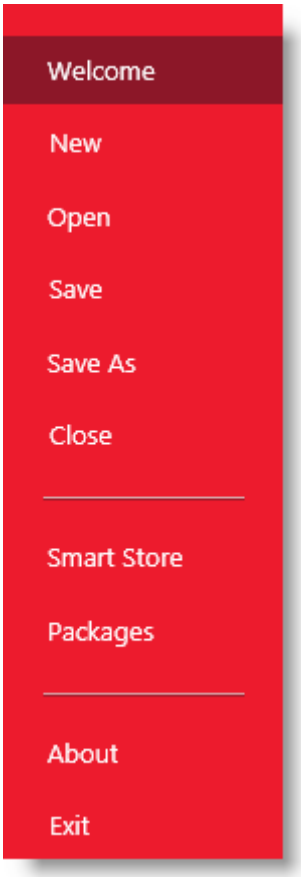
Pressing the backward arrow  in the top menu will open the main window of WARP, and in the ribbon you will find four tabs of selection:

- **File**: Will open the left menu pane
- **Home**: Normal workspace when building and editing a project
- **View**: Selection of dockable windows and functions
- **Tools**: The selection of extra functions and settings of the software




### File

Selection “File” will show the left main menu that include the links to Smart Store and Packages




- **Welcome**: Show the project welcome screen
- **New**: Create a new empty WARP project
- **Open**: Open a WARP-project (*.pproj) or a WARP-archive file (*.WARPArchive)
- **Save**: Saves current WARP-project (*.pproj) with all included project devices, attached documents and user settings in one directory.
- **Save As**: Browse a directory and *Save Project* with Devices and HMI/PLC projects in one directory (*.pproj) or use *Save Archive* to save in one file (*.WARPArchive) that easily can be sent to another programmer or end-user.

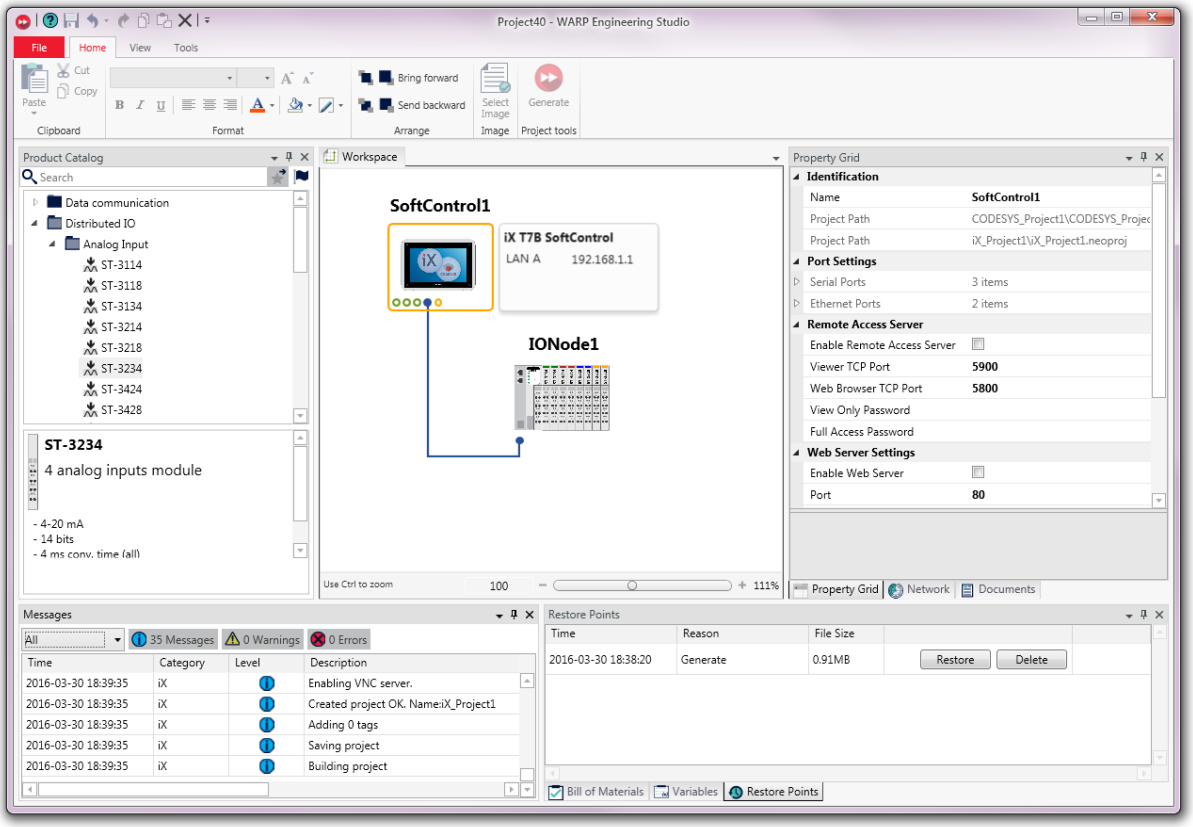
  **Important!** When saving to a WARP Archive file, all internal projects of the current WARP project (single / multiple CODESYS and iX Developer projects) of all devices in the workspace, all attached documents and settings will be included in the backup archive file.

- **Close:** Closes current open project
- **Smart Store:** Direct link to Smart Store, where you find all available installation files of hardware and software devices for the product catalog, and the software tools used by WARP.
- **Packages:** Installed software’s and packages (*.warppackage)
- **About:** Shows the build version of WARP
- **Exit:** Closes the WARP software and other open tools like CODESYS and iX Developer; saves open projects.

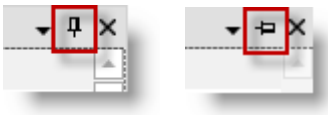
## Home

Home tab opens the main working area, where the user can create and edit WARP applications. All windows are dockable and can be dragged to any position for customized interface.

Below an example of the Workspace showing a SoftControl panel and its Property Grid. The other windows are Product Catalog to the left and Messages and Restore Points in the lower part.




Each window has the pin / unpin function for the user to show or hide windows individually.



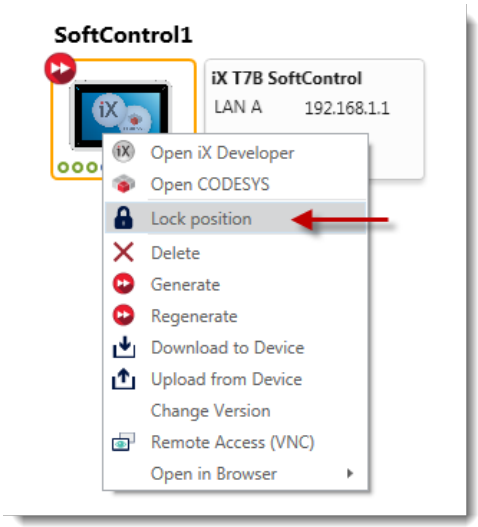
In the workspace the user has full access from the context menu of each device. The other software tools iX Developer, CODESYS and BCS Tools can be opened, generated and downloaded to a connected device; one device at a time.

Lock position, will fix the object to the workspace.

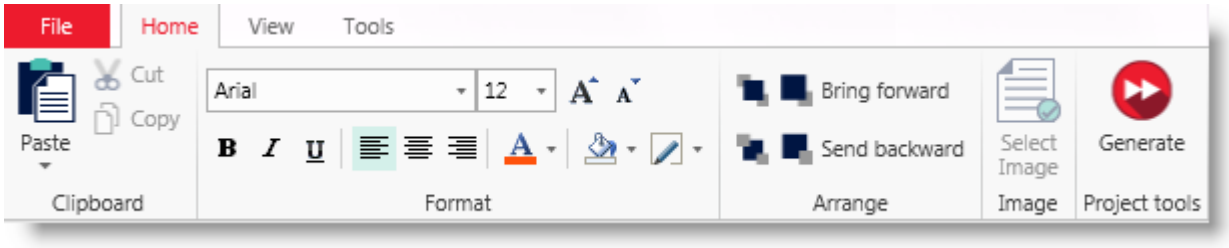
Note that the context menu may differ from object to object depending of the functionality available. For example the Remote Access (VNC) functionality is available just for iX devices like iX HMI, SoftControl and SoftMotion panels.

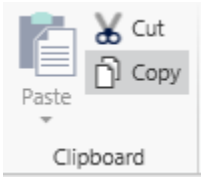
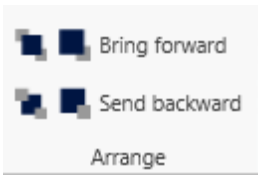
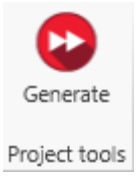
Open in Browser, will browse to the internal web server of a device when available.

Open External Resource can be anything from documents to external resource files for upgrading a device to other software tools like IOGuidePro, BFI Tools and BSD Tools.



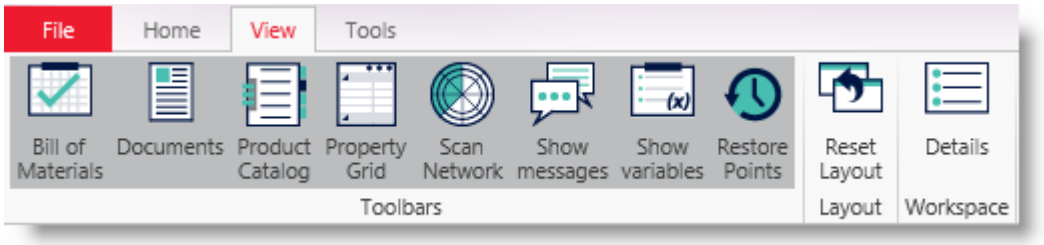
Ribbon “Home”

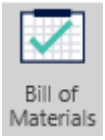
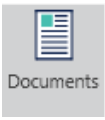
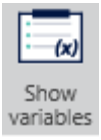
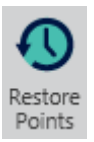


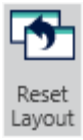
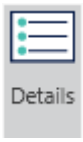

| | | |
|---|---|---|
| Clipboard | Function to Copy, Cut & Paste text objects in workspace (devices of type Drawing – Text)**<br><br>** In future updates of WARP this functionality will be available for all HW device objects. |  |
| Format | Font and character settings for the text objects in workspace. | |
| Arrange<br>- Bring forward<br>- Send backward | To bring objects on the workspace to foreground or to background.<br><br>Mark an object and then press button forward or backward to adjust the order of the objects on the workspace.<br>Four selections available: Bring to front, Sent to back, Bring forward and Send backward. |  |
| Image | Only available for a graphical drawing Image object in the Workspace. Mark the image in the workspace and press the button to change the graphical object in the workspace, for example a background screen. |  |
| Project tools<br>- Generate | Generate will create and build application project for the device in focus.<br><br>It will Save the WARP project and then start the Generate process opening the proper programming tool(s) (CODESYS / BCS Tools and iX Developer),<br><br>All files are stored in a structure of the WARP-project directory.<br><br>Any information, warning or error during project generation can be seen in the Message window, see ribbon View – Show Massage.<br><br>Note, more details about the Generate procedure in the Appendix of this document. |  |

## View

Ribbon “View” is where the user can activate dockable windows.




| | | |
|---|---|---|
| Bill of Material | All devices of the workspace are listed in the Bill of Materials.<br>Export to Excel format can be performed. |  |
| Documents | This window will be automatically search for documents in Beijer Electronics Help Online when computer is connected to Internet. Depending of device in workspace it will show a list of hits.<br>User can attach his own files and documents to the project. These files are saved with the WARP project in a separate folder, Documents. |  |
| Product Catalog | In the project catalog all hardware and software devices (smart objects) provided by the WARP package are displayed. From the product catalog user can drag&drop devices and smart objects to the workspace. |  |
| Property Grid | Each unique device in the workspace may have a property to adjust, this can be found in the property grid window. |  |
| Scan Network | Performing *Refresh* in the Network window will make an active search for iX devices in the network connected to your computer. Scan for iX TxX, SoftControl and SoftMotion panels. From this pane drag&drop the device to the workspace and the IP setting will be correctly set. |  |
| Show messages | Message window are showing the result during Generation and deploy to device. A history of *Last Generated*, *Current session* and *All messages* can be selected. Filter on *Information*, *Warning* and *Error*. |  |
| Show variables | In the Variables window user can create global variables for CODESYS applications and export / import from Excel.<br>Variables of Smart Objects dragged to a device in the workspace will be present in this window.<br>Setting for copy to iX, Datatype and scaling with RawMin, RawMax, EngMin and EngMax. |  |
| Restore Points | The window of all saved restore points.<br>At each “Generate” as well as “Download to Device”, a restore point will automatically be created, and saved with the WARP-project.<br>Important! A restore point saves away any customer modifications in the application of CODESYS and iX Developer before WARP make the generating changes. That means that all changes made in the project until this point will be backed up before the Generate is proceeded. |  |

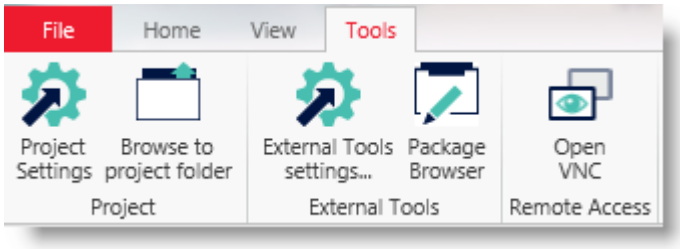
| | | |
|---|---|---|
| Reset Layout | Reset the layout of tool windows to the predefined default setting. |   |
| Details | Show device details in the workspace view.<br>For example the device name and IP address |   |

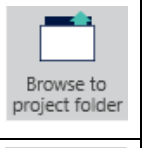
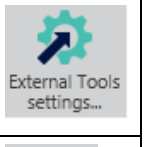
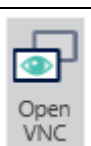
The example below shows the details of an iX T7B SoftControl. Note that the IP address can be set directly in the details view as it's an editable field. Similar if the device has a Station No.




## Tools

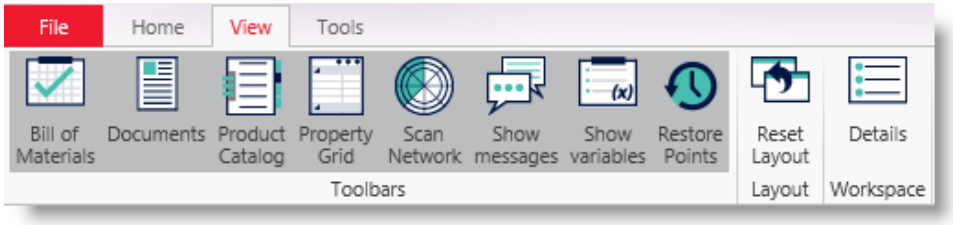
Ribbon "Tools" is mainly for special software and project settings, as well as for Remote access.




| | | |
|---|---|---|
| Project Settings | Special user setting of the project<br>- Workspace properties (width & height)<br>Default 1500*2121 = normal A4 |   |
| Browse to project folder | Show the total content of the WARP-project directory in a separate Browse window. It gives a good overview of the total projects and files created by WARP. |   |
| External Tools settings...<br>- External Tools | The external software tools settings, for configuration of program path. In the External Tools setting individual configuration can be made for the tools used by WARP |   |
| Package Browser<br>- External Tools | Link to Smart Store. |   |
| Open VNC<br>- Remote Access | Will open a new window with Remote Access VNC viewer. User can select any IP-address independent of present WARP-project and devices in the workspace. |   |

# 6 Functionality

In this chapter follows a more detailed description of each individual window and functionality in WARP. Note that software updates may affect the difference in the graphical appearance compared to screenshots in this document and the version of WARP that you are using.

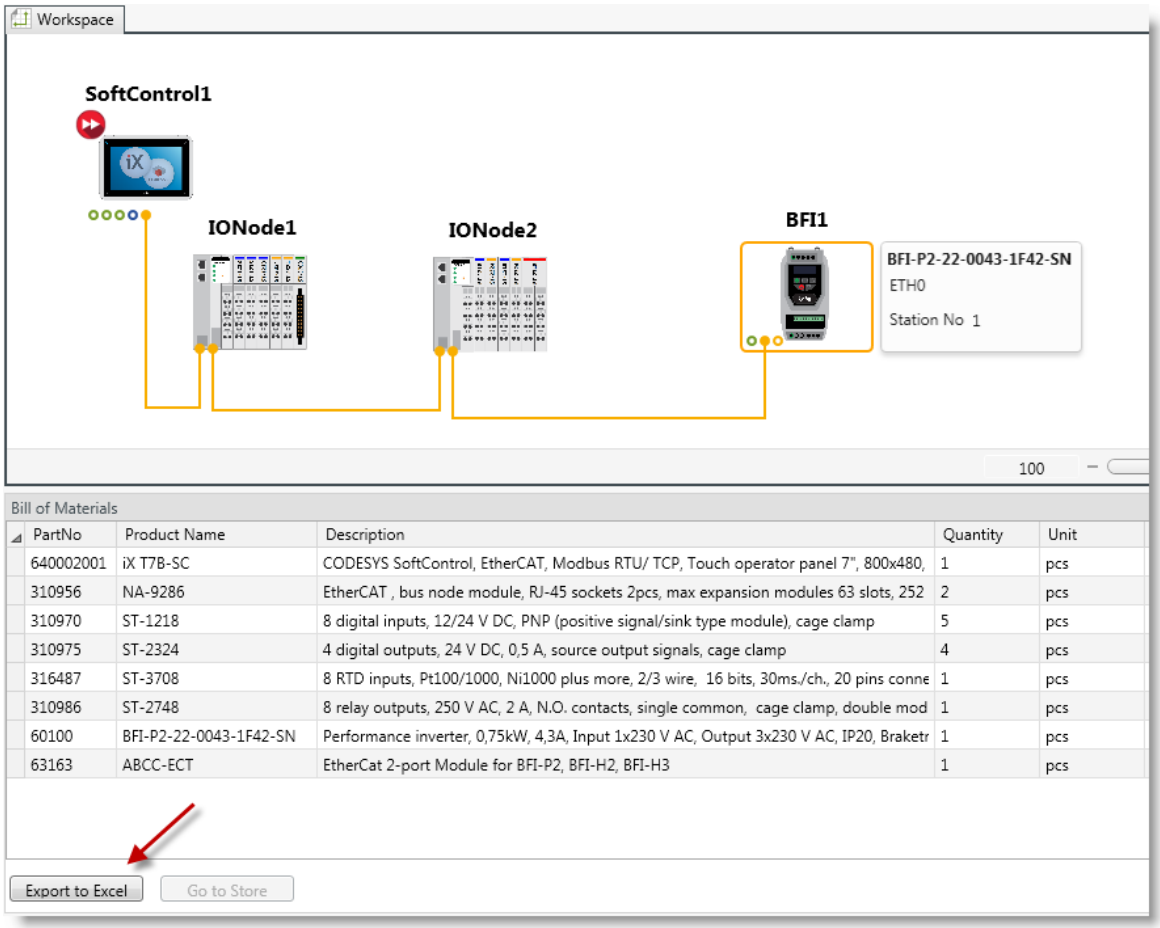



## 6.1 Bill of materials

The window *Bill of Materials*, include all devices in the current Workspace.

An export to Excel format can be performed, where the user or sales account manager can complete the hardware configuration with additional items like cables, EMC-filters, encoders and software licenses not included in the product catalog of WARP.

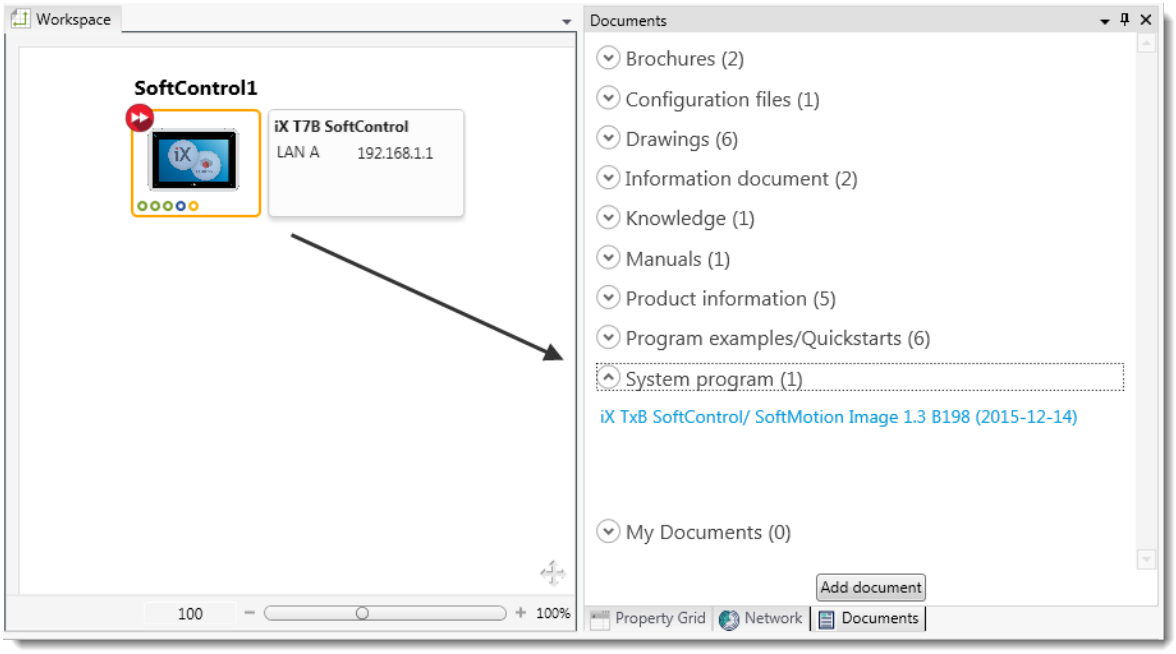
Connection to our eBusiness will be implemented in a future version (Go to Store).



## 6.2 Documents

WARP will automatically search for documents in the Beijer Electronics - Support - Help Online, when the computer is online connected to Internet, depending of the device in focus.

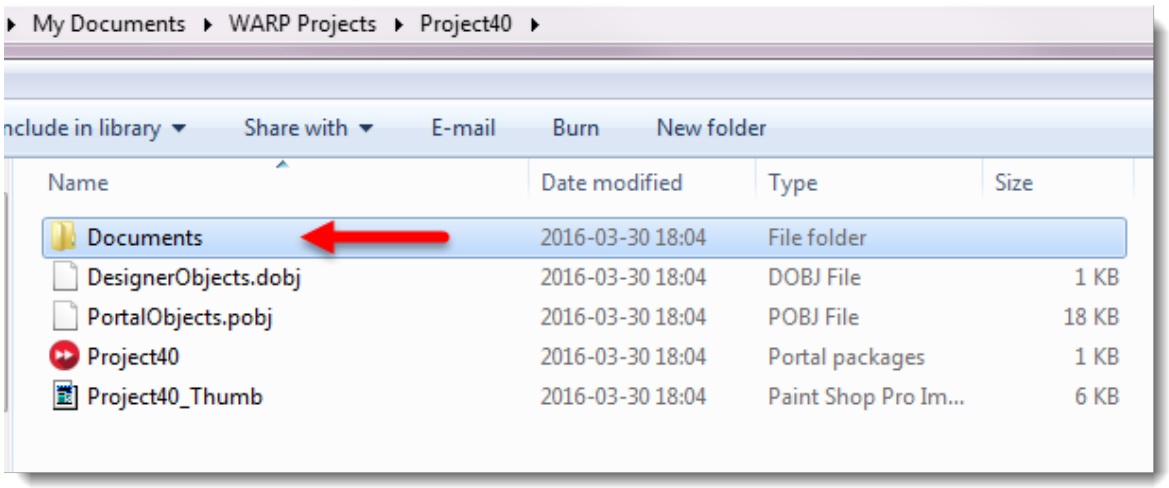
It will show hits in the same categories like Help Online.




In addition the user can attach own files and documents to the project by pressing button “Add document”. That can be any file accessible from the computer.

Attached files will open in the correct software if it’s associated in the computer.

The local documents are included in the Documents folder of the WARP-project.



## 6.3 Product catalog

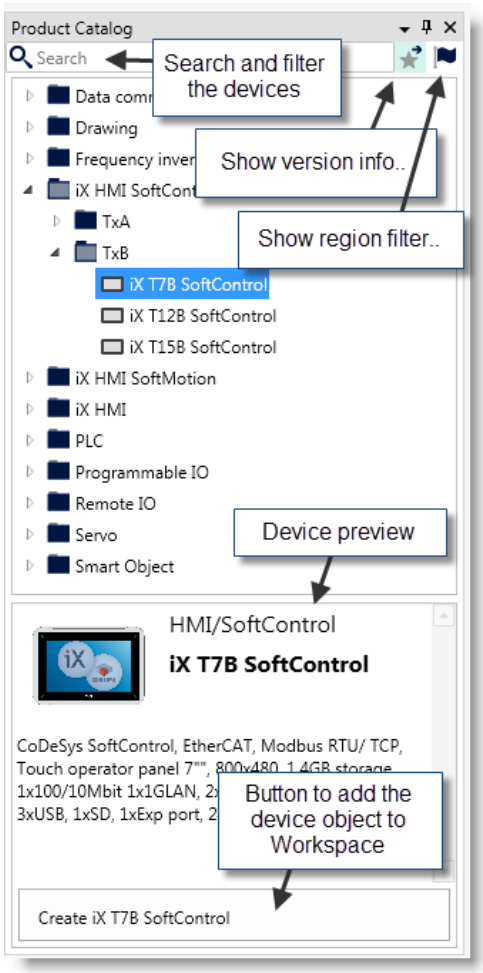
The product catalog of WARP includes all devices (hardware) and Smart objects (software) provided by the WARP-package downloaded from Smart Store, presented by a tree view.

### Search and filter the product catalog

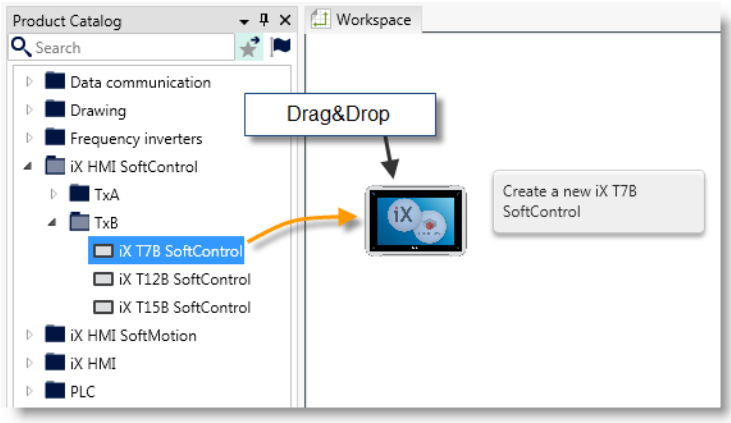
From the product catalog the user may search and filter for a specific items or just pick one by expanding the lower levels of the tree.

Example to the right shows the device iX SoftControl marked and a preview of the product description seen in the lower pane.

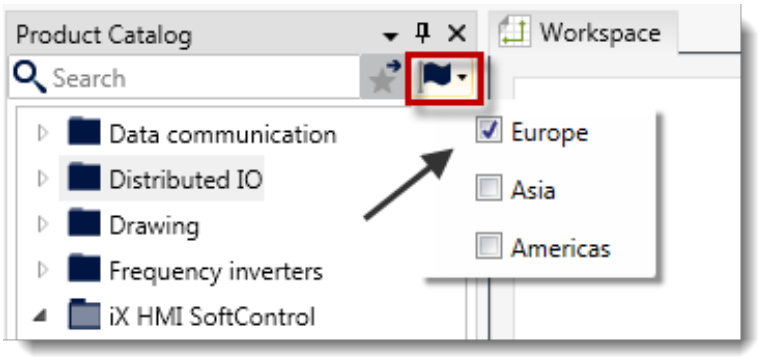
In the lower pane is a button to add the device to the workspace, by pressing *Create…*



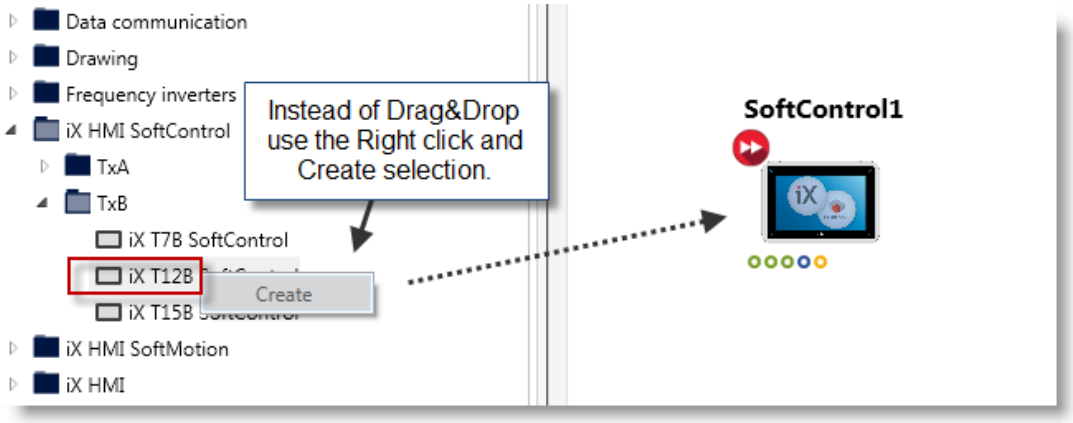

The normal procedure to add a devices or smart objects to the workspace, is to use *drag&drop*, as in the example below.




Example with filter for different regions: Europe – Asia - Americas



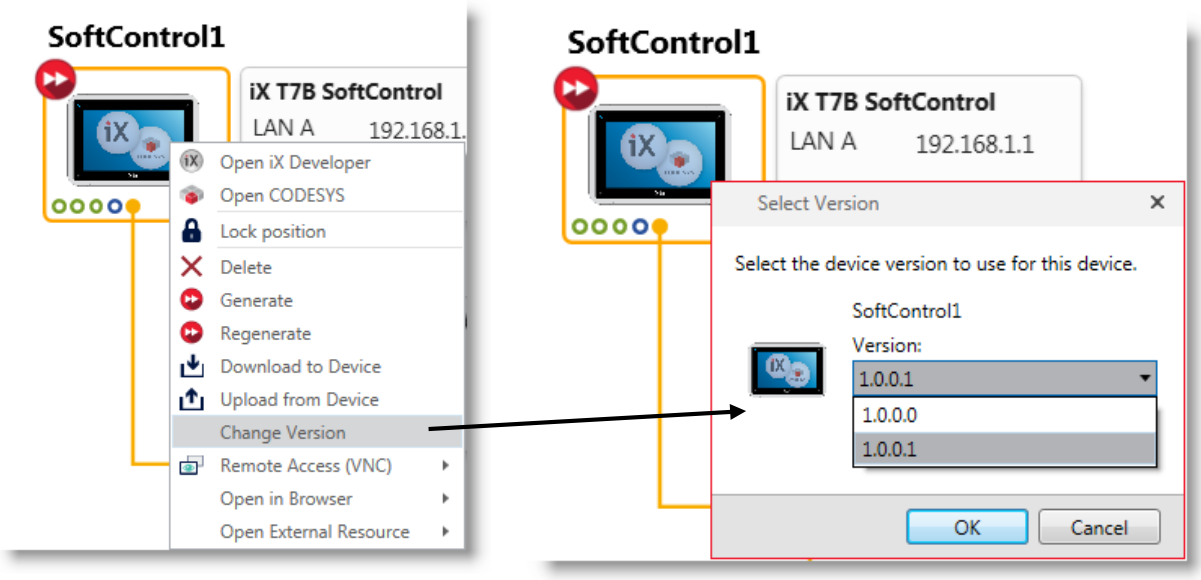
Instead of drag&drop, use the *Create* selection from context menu. This is available for most of the hardware devices except the IO modules of Distributed IO and Smart Objects.




## 6.3.1 Change version

If an object in the workspace for some reason need to be changed to another version, either delete the object and replace with a new one.

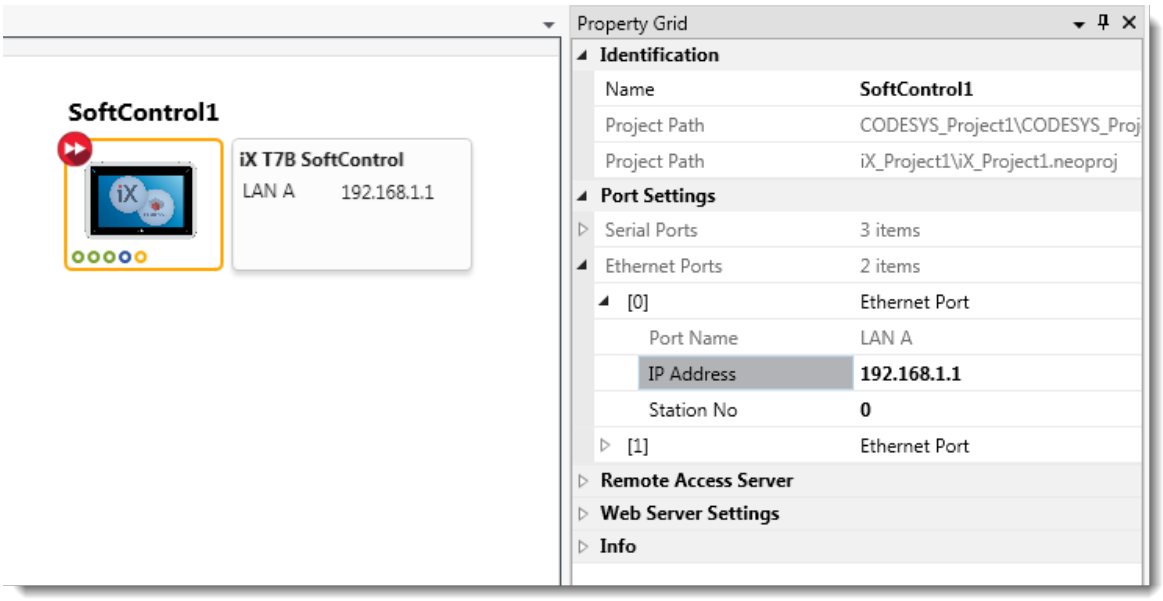
If the object is already generated, it is more convenient to use the *Change Version* function in the context menu of the device. The dialog that opens give access to all versions available in the current product catalog of this type of device, different selections in a dropdown list. Just pick the correct version and press OK. Now the project must be generated once more.



## 6.4 Property grid

Each unique device in the workspace may have a property to adjust, this can be found in the property grid window. Examples of settings can be Name, IP Address, Station No and other specific settings. It's the device included in the Product Catalog that hold the individual settings of each device.

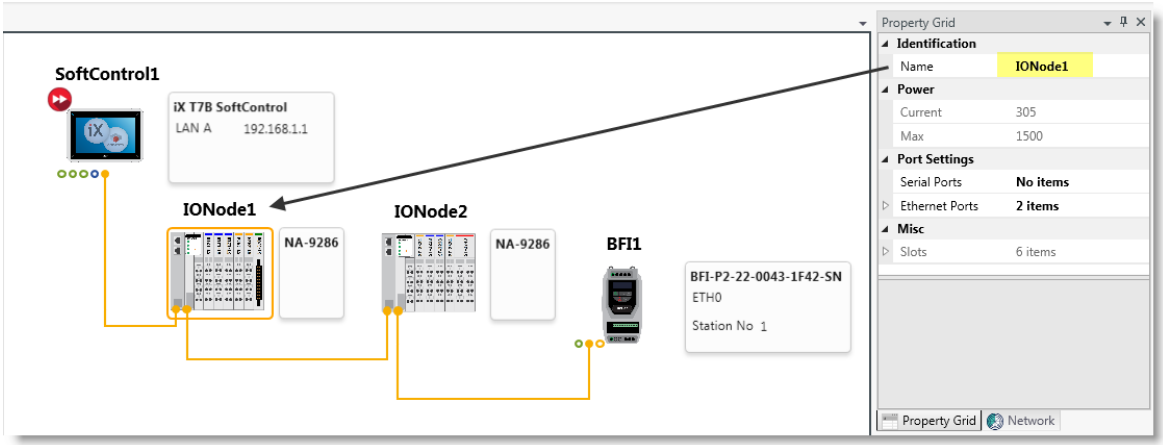
It's important to know that when downloading WARP package from Smart Store, devices will be installed that may open up for new features and settings without upgrading WARP itself. Therefore devices are given a version number for example 1.0.0.0, 1.0.0.1 and 1.0.0.2 etc.




## 6.4.1 Device identification name

The name of each object can manually be changed so that it better explains the function for a unique hardware of the workspace. This name will influence the code generated for CODESYS and iX Developer of this specific object.

Below example show an iX T7B SoftControl panel with three slaves, two Distributed IO adaptors NA-9286 and one BFI-P2 inverter. One of the IO adapters got the default name IONode1, and this name will be used as a part of the identifier of the object in the CODESYS application.




All distributed IO modules get a unique automatic variable name created for each slot and channel and the I/O mapping is automatically set up in the I/O Mapping table of CODESYS project. This is similar if you are using EtherCAT or Modbus RTU / TCP communication.

More examples are found in the chapter Project example.

## 6.4.2 Distributed IO properties

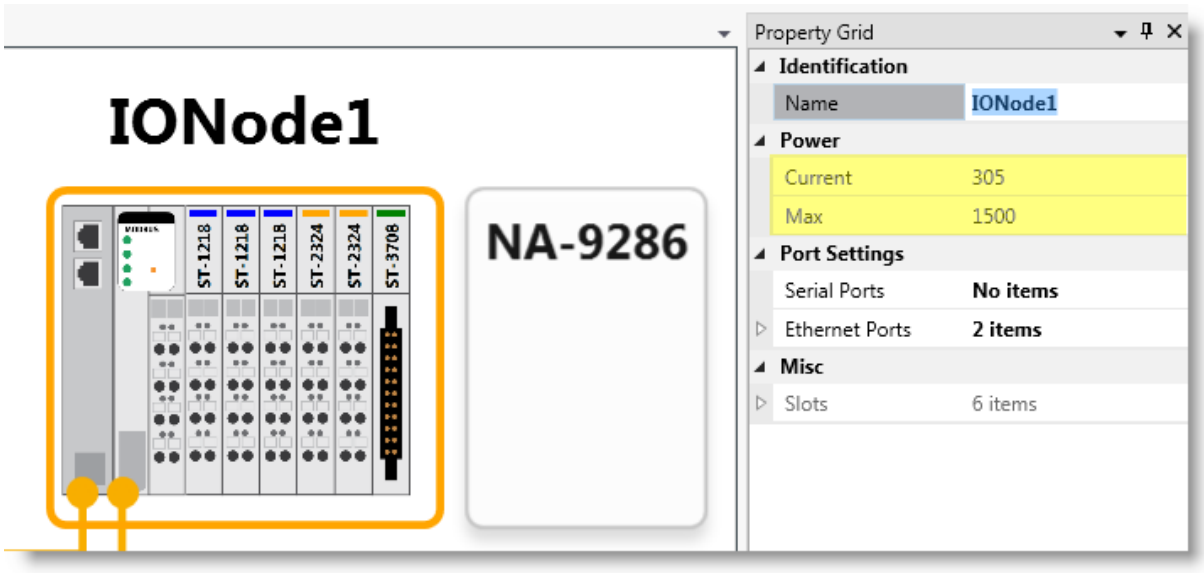
### Power

Crevis Distributed IO modules include information and settings for each slice.

By focusing on the Network adaptor level, the user gets the power consumption and available power for the entire I/O-slave node (in current, mA). Below a NA-9286 with 6 slots.

Current (mA): The total current consumption of the node, this example 305mA.

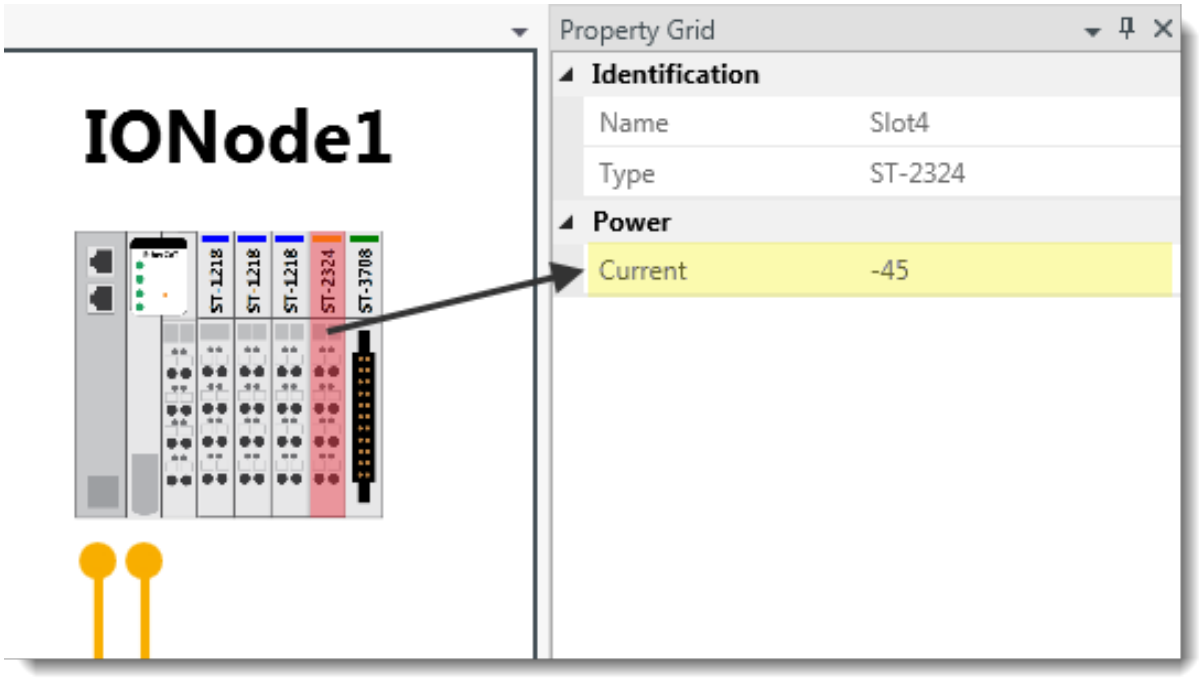
Max (mA): The maximum current from the Network and Power modules, 1500mA.




### IO modules power consumption

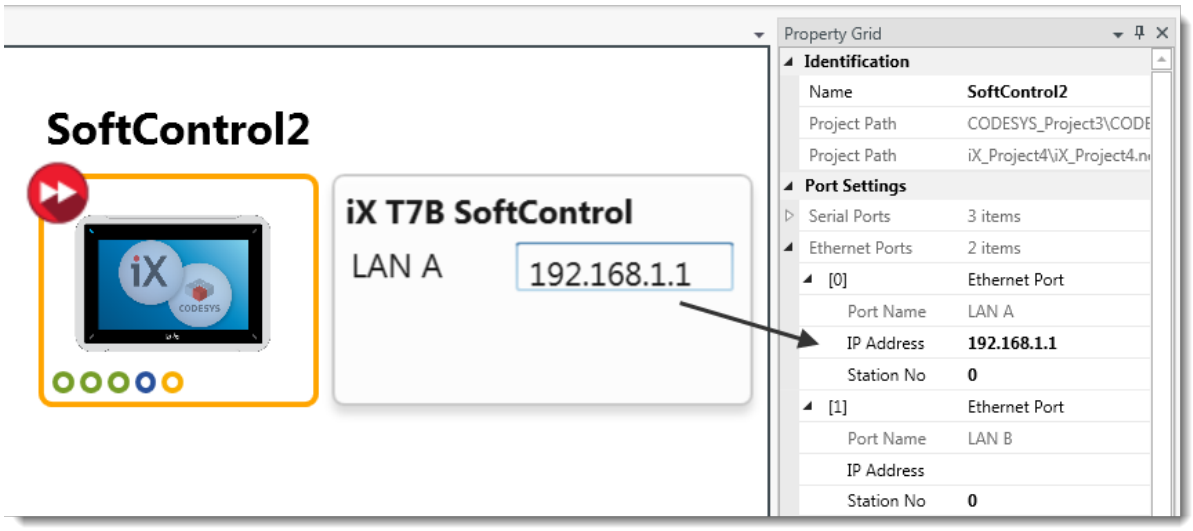
Click on each slice to get info and properties of this specific IO module.

For example the current consumption of a ST-2324 I/O slice, -45mA.



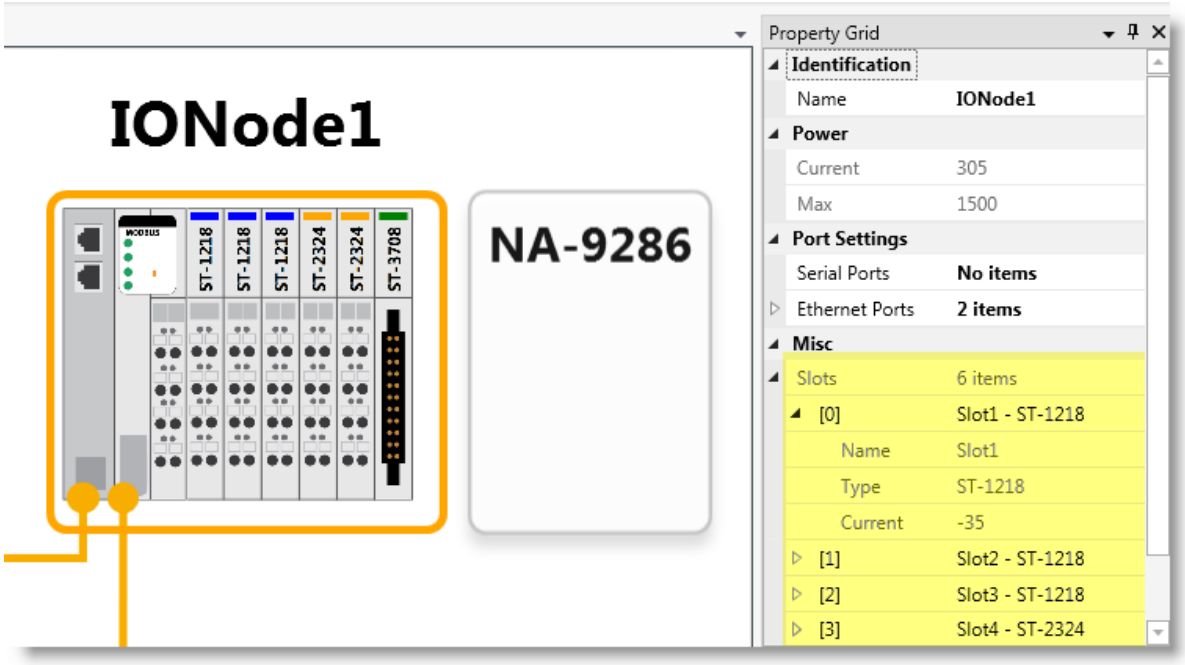
### Port settings

For serial and Ethernet ports, the user can make the necessary adjustments and settings. For example the default IP address of a SoftControl, 192.168.1.1.
IP settings can be changed in the workspace details box or in the property grid window.




### Misc. - slots

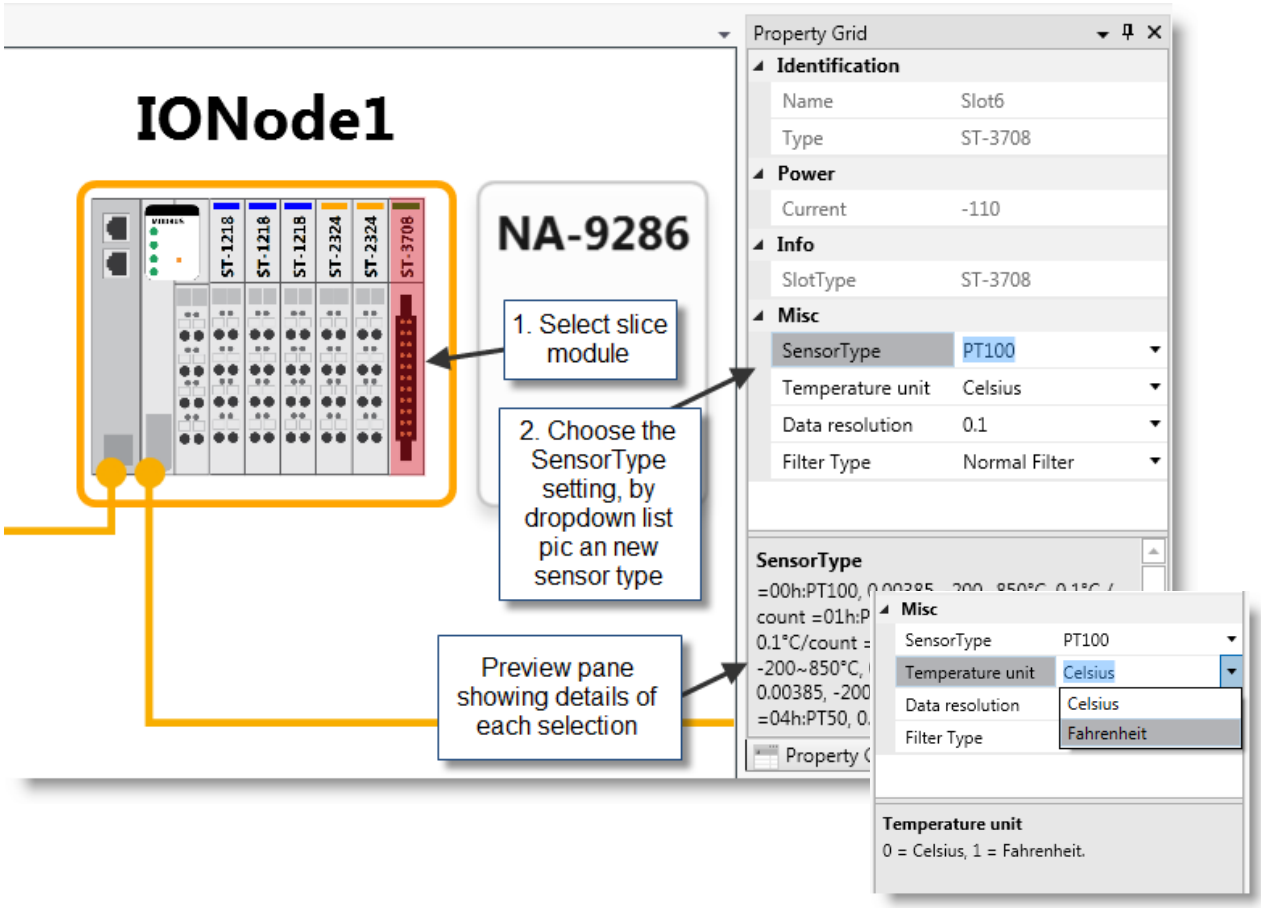
Details of each slot of the Network module, showing the name, type and current consumption.



### 6.4.3 Distributed IO modules initial settings

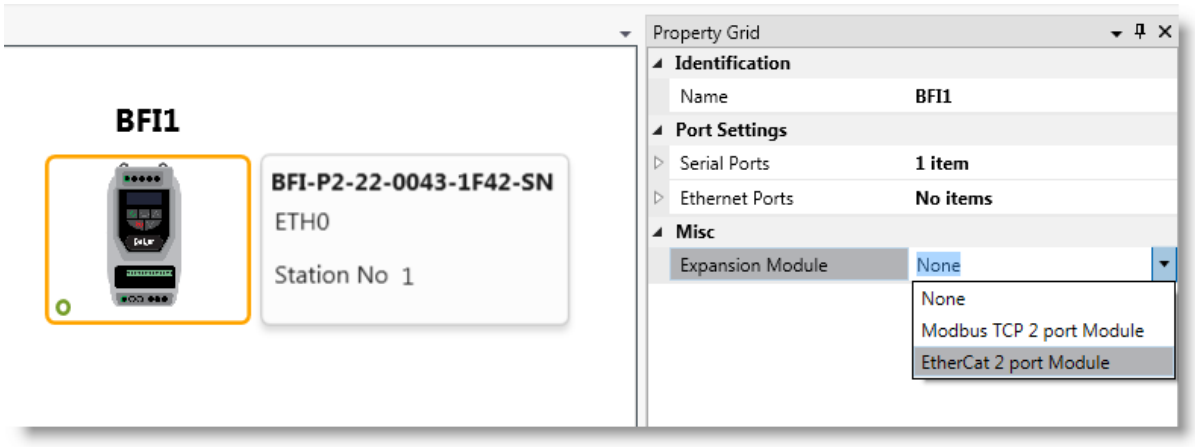
Crevis Distributed IO modules may have extra options for each slice, for example some for the Analog Input modules. In the Property Grid the user can pick the Sensor Type or select the Temperature unit. Now the initial parameters of Modbus or EtherCAT will adapt to these settings.

Settings made in this section of the properties will be converted to parameters of the communication driver of the slave, and written to the CODESYS project during Generate procedure.




### 6.4.4 BFI Inverter select Expansion Module

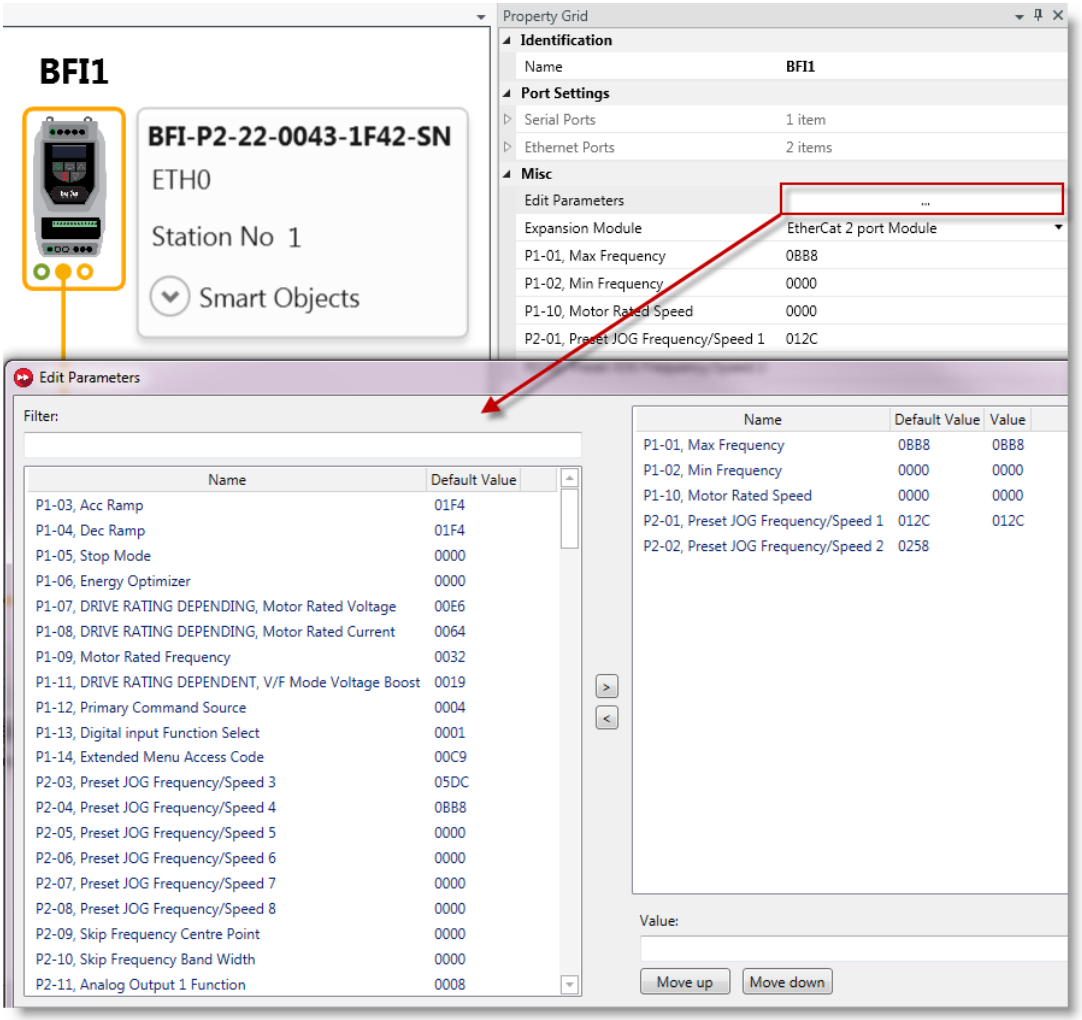
For BFI Inverters the property of using Expansion Module may be available depending of type of inverter. Select Modbus TCP 2 port or EtherCAT 2 port module and the graphics will change and make it possible to connect to the Master device.



## 6.4.5 BFI Inverter initial parameters

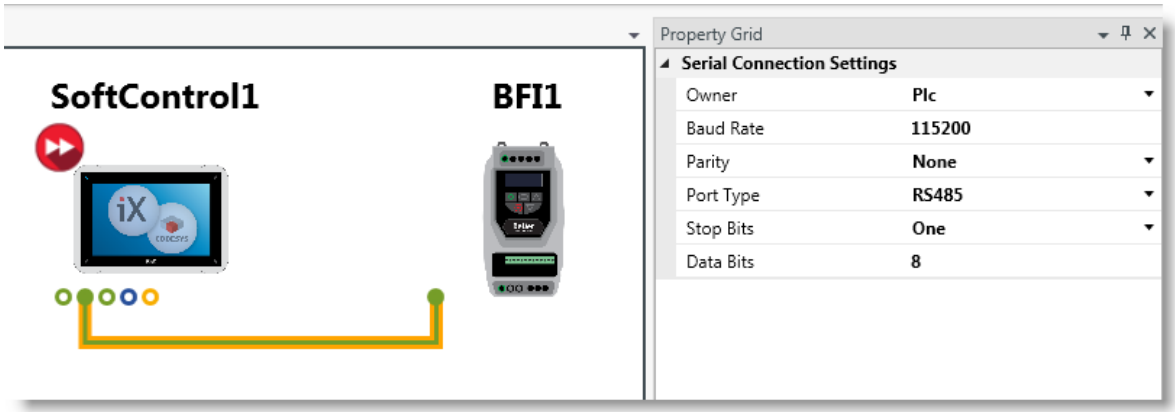
BFI Inverters using EtherCAT interface will get an extra option of editing initial parameters. Pressing the button […] will open window *Edit Parameters* where all possible parameters are listed.

Filter or mark them one by one and add to the right pane of the window. During Generate, the CODESYS configuration will take these initial settings for the BFI slave device.



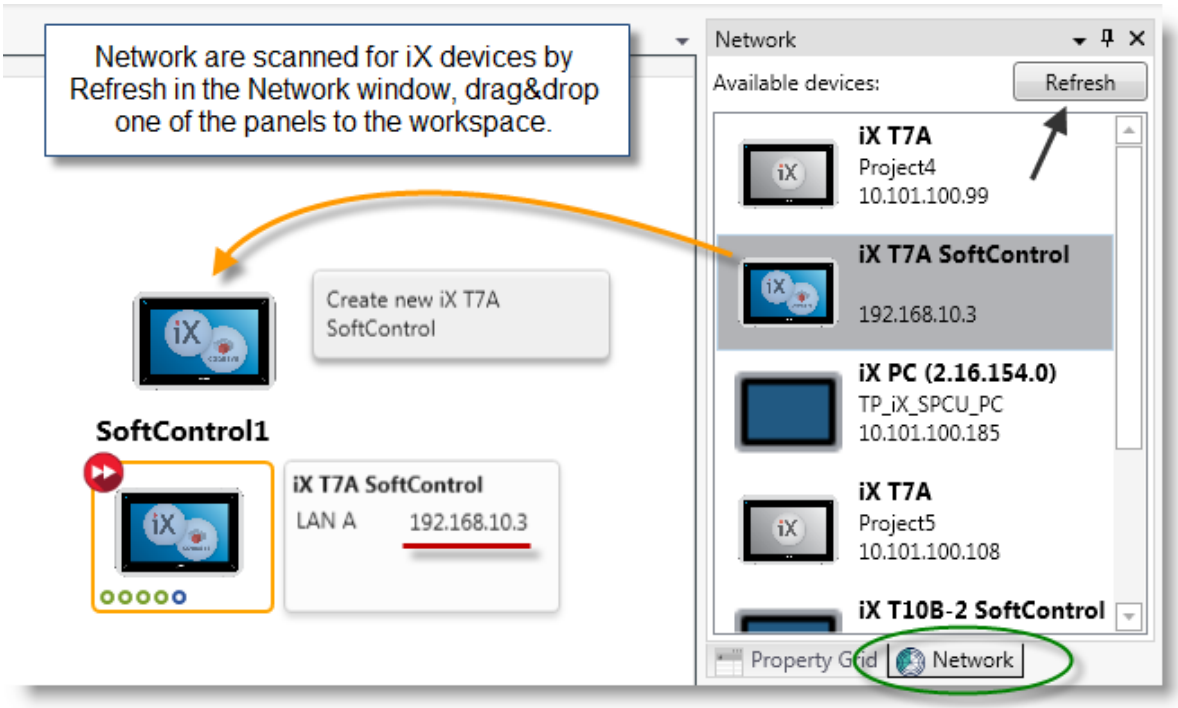

## 6.4.6 Serial communication settings

For serial communication the settings can be set on the line between the devices, in this example the settings of the SoftControl panel are adjusted.



## 6.5 Scan network

Performing a *Refresh* in the Network window will make an active search for iX devices in the Ethernet network connected to your computer. Scan for iX TxX, SoftControl and SoftMotion devices. From this pane drag&drop a device to workspace and the IP setting will be correctly set.

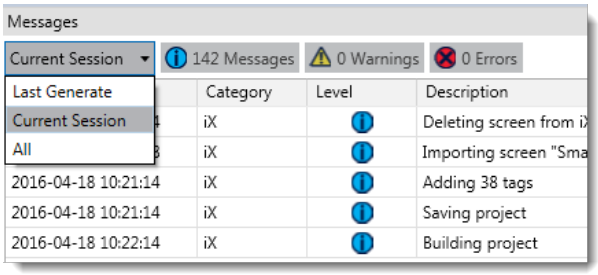


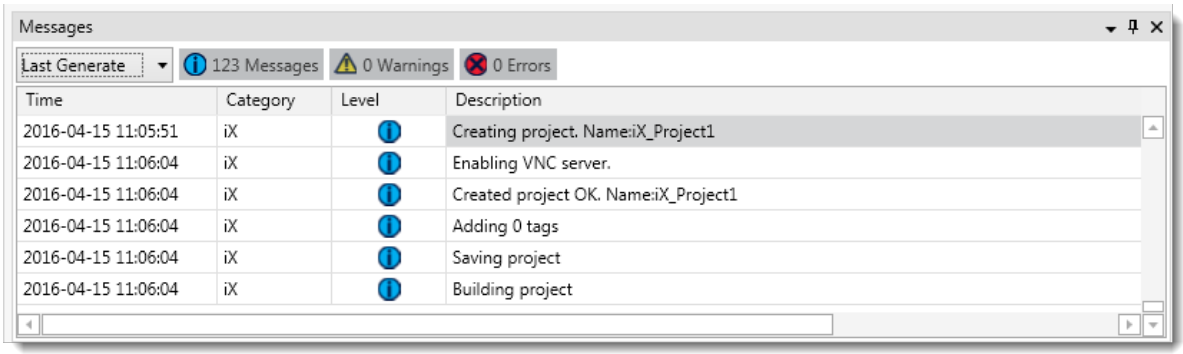

## 6.6 Show messages

Message window can show the result during Last Generate, Current Session and All messages of the open project.

This is a helpful history of finding warnings and errors during Generate.

Filter on Message, Warnings and Errors tabs.



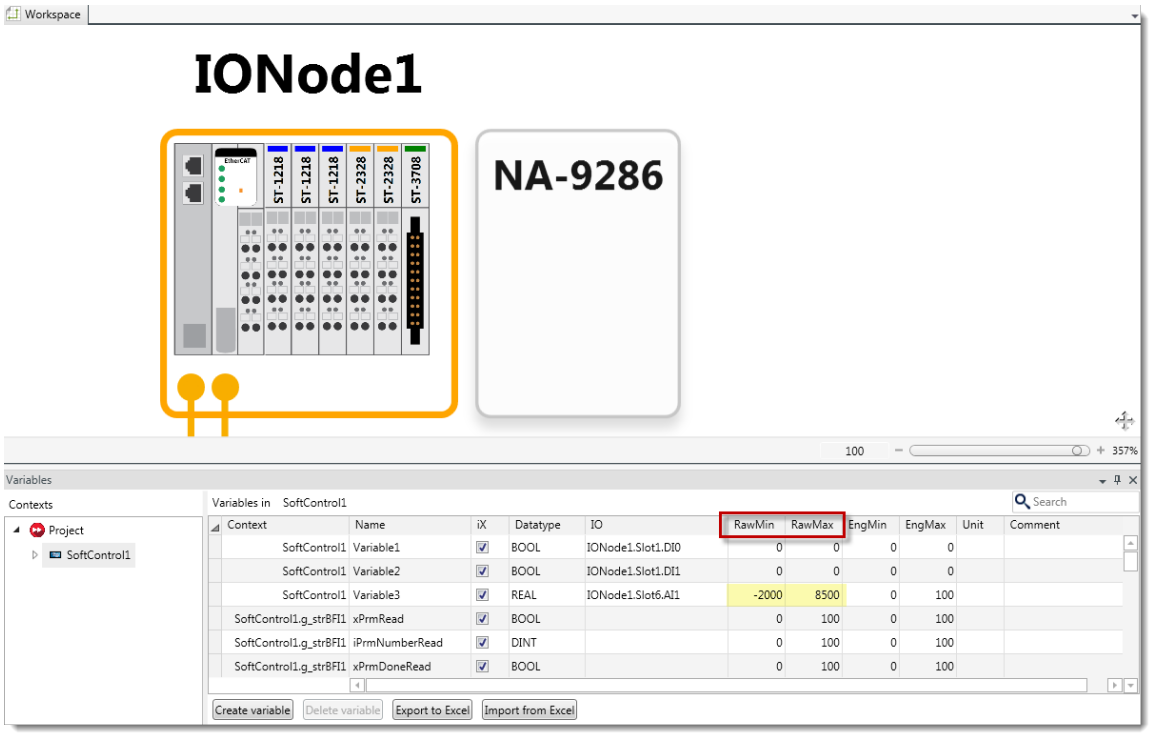



## 6.7 Show variables

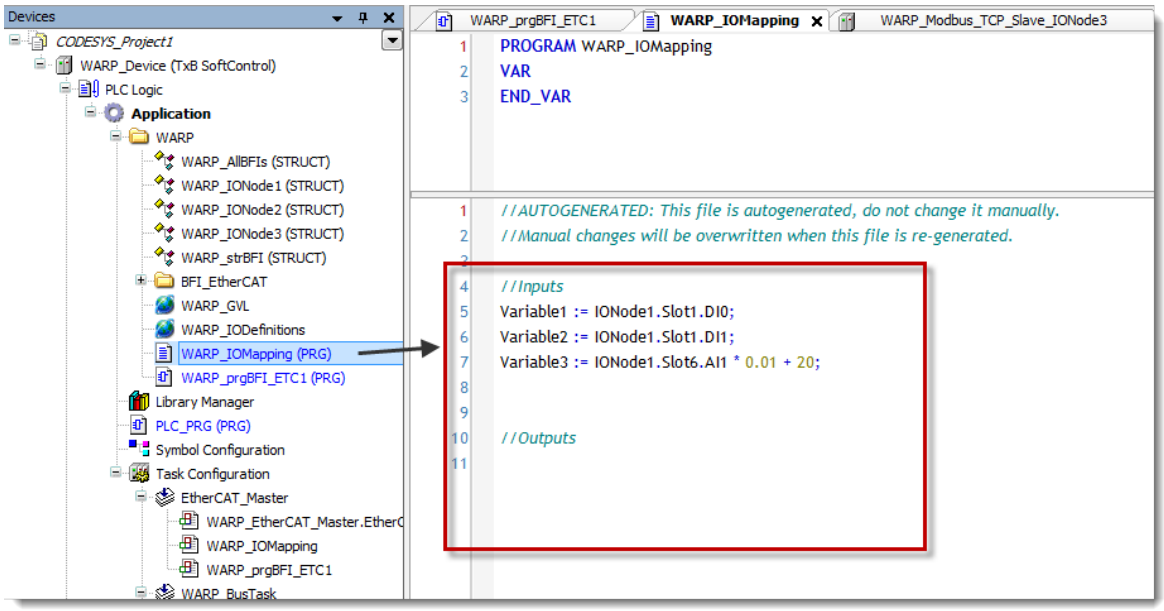
In the Variables window user can create user made variables and Export / Import from Excel format.

The list include settings for automatic copy to iX project and tag list, change datatype and scaling options with RawMin, RawMax, EngMin and EngMax, define Unit and add the a Comment text.

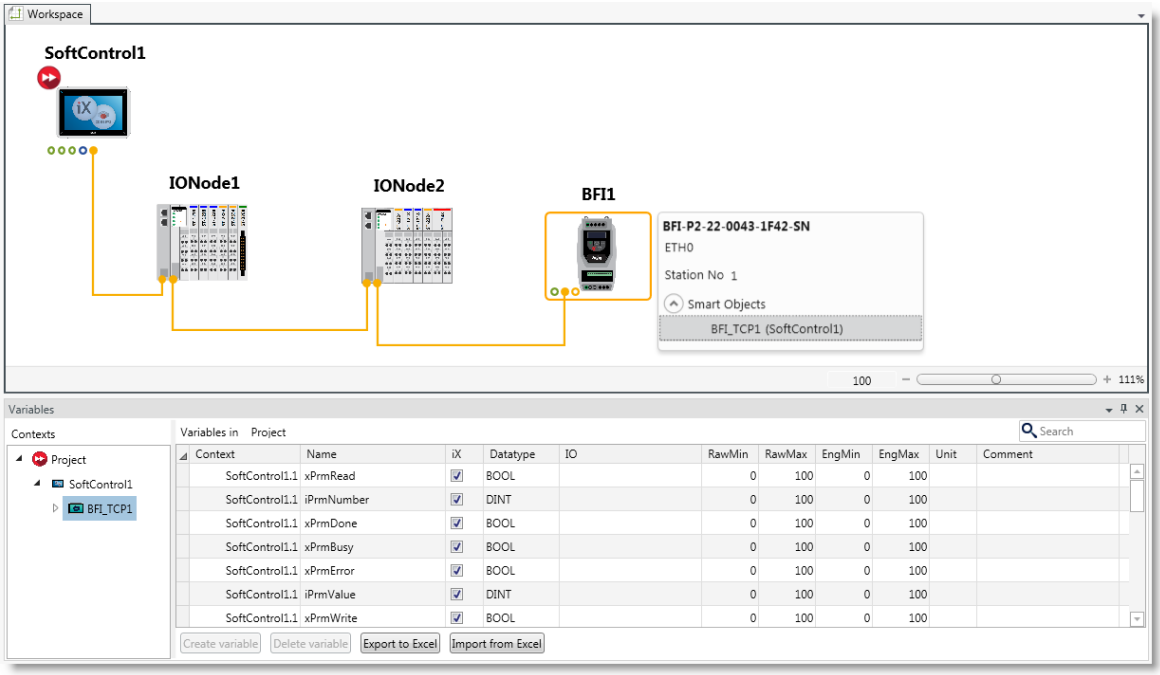
Note that values of RawMin / RawMax will get the default setting of that specific distributed I/O slice when an IO point are selected in the dropdown menu. Forexample ST-3708 have the default settings -2000 to 8500 in raw values, this will be scaled to 0-100 units.




The scaling and mapping from physical I/O to user variable names are generated by WARP and inserted to the global variable list *WARP_IOMapping* in the application of CODESYS.



Variables of Smart Objects dragged to a device in the workspace will be included in this window as well. Below are the variables of a BFI smart object including the tags for the iX application.

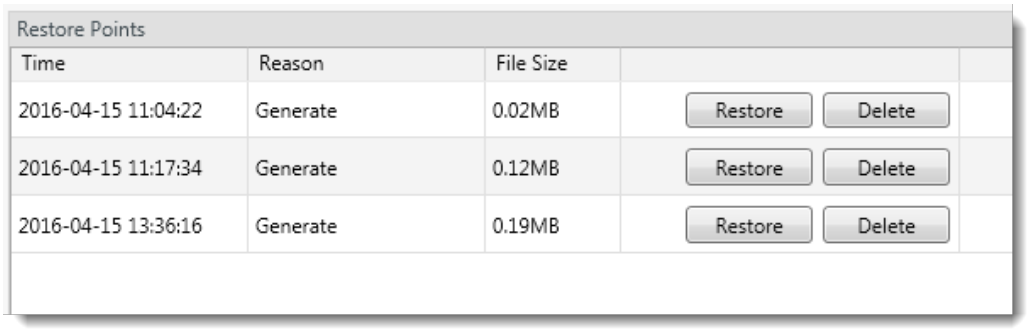



## 6.8 Restore points

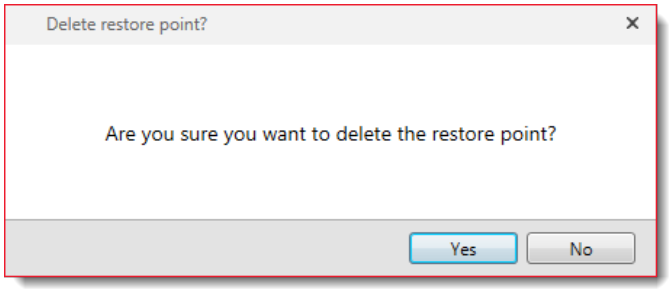
In the window of restore points, each Generate will always create a new restore point automatically before generating procedure are started, and it's saved with the WARP-project.

Note, a restore point saves away any customer modifications in the underlying application (iX Developer, CODESYS and BCS Tools) before WARP makes any generating changes. Meaning that all changes made in the project until this point, will be backed up before the generate is proceeded.

By pressing the button *Restore* a specific restore point can be loaded.




Each saved restore point can be removed by pressing *Delete* and then confirm the question by *Yes*.



## 6.9 Auto resolve

During editing in the workspace, WARP will guide the user to make the correct connections and IO configurations, this is called *Auto Resolve*.

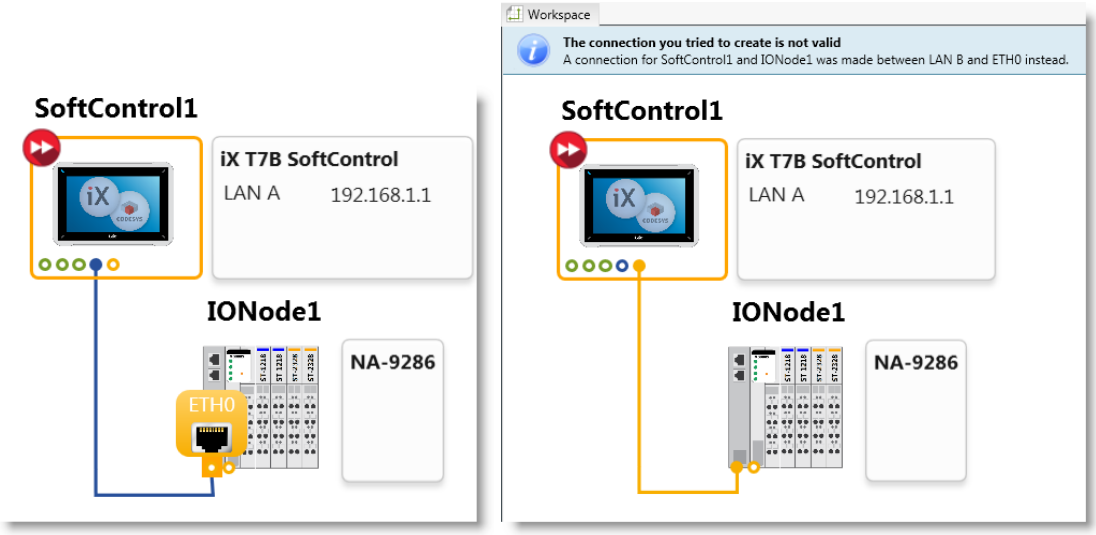
For the Crevis Distributed IO and Programmable IO it will help the user to arrange slices in a correct order. WARP will limit the amount of modules depending of what Network module or Programmable IO type are being used. And it notifies the user when it's time to add a power module to the node.

This is done to prevent unnecessary faults of mixing slices in incorrect order and to avoid misconfigurations!

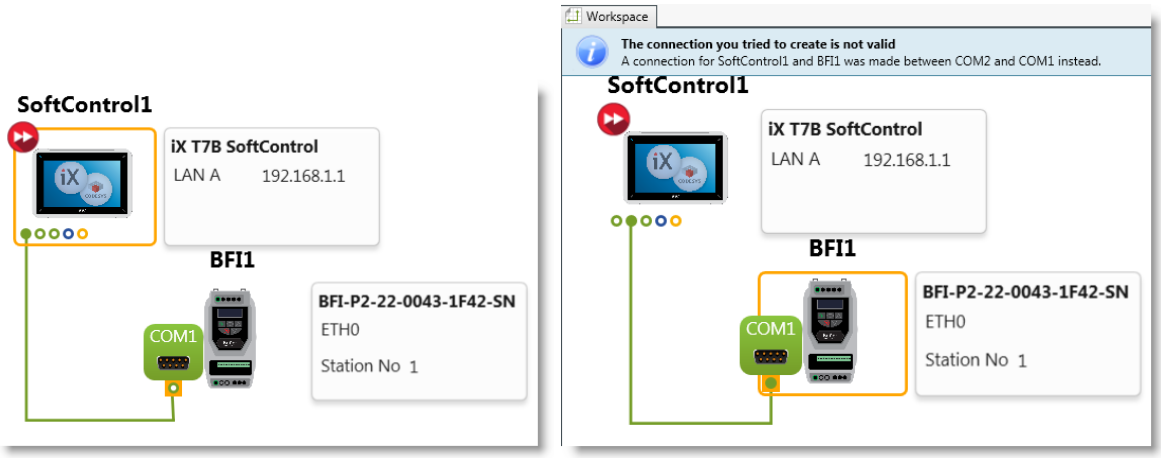
### 6.9.1 Correcting connections between devices

If the user by mistake tries to connect one device to another using faulty connections (not the same type) or connections not supporting the same protocol, WARP will change this directly by Auto Resolve. No need to do anything just check the message in the top of the screen.

**Example1:** If you try to connect a blue Ethernet LAN A from SoftControl1 device (iX T7B SoftControl) to ETH0 of IONode1 slave device (NA-9286), you are trying to establish a connection that is not valid! WARP will automatically change connection to a yellow EtherCAT line from LAN B of SoftControl1 to ETH0 of IONode1. An information message about this change is displayed in the top of the workspace window.




**Example2:** Incorrect serial connection that might be difficult for the user to predict.

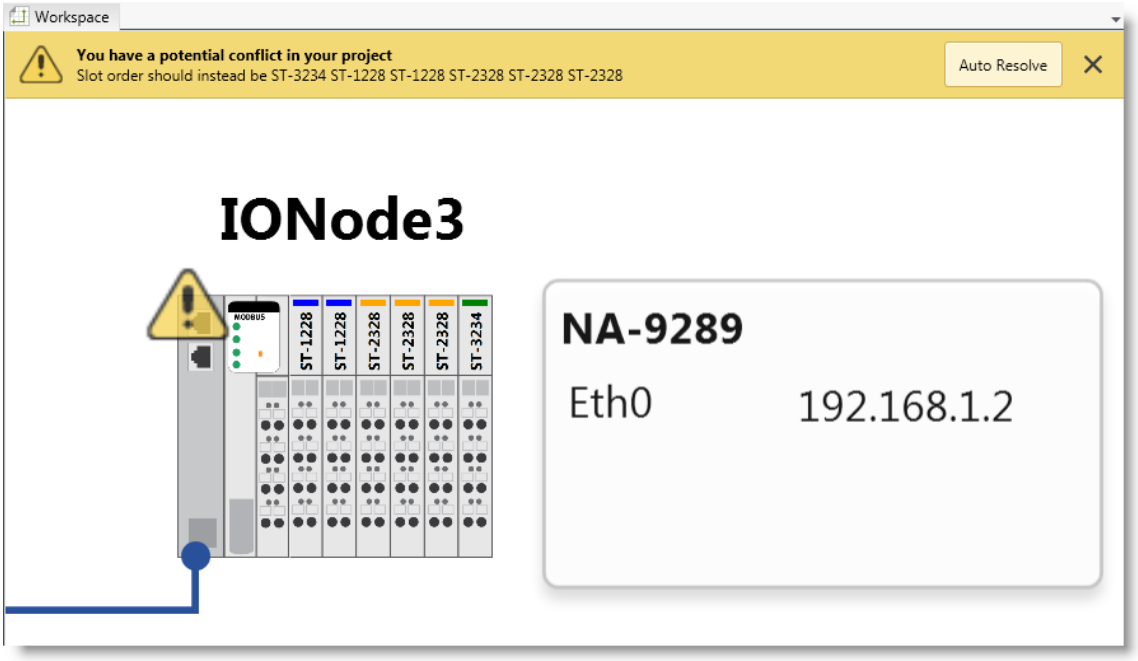


## 6.9.2 Correcting the order of I/O modules

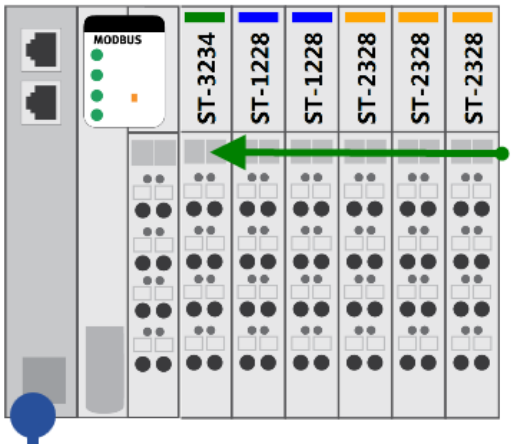
For example a Modbus TCP node NA-9289 is added an analog input slice (ST-3234) as the last module in the node. This is not a good way of configure Crevis Modbus IO nodes because of the byte oriented Modbus messages that can split the two bytes of the analog word.

Instead, the user gets a warning and a chance to correct the mistake just by pressing Auto Resolve.

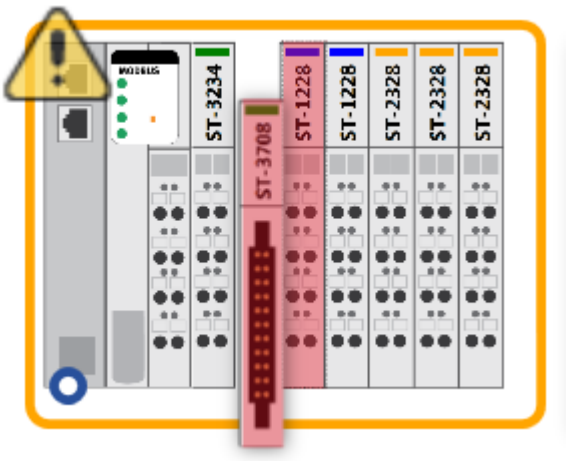
The slave node has the warning sign as long as WARP finds a possible risk of this configuration.




When pressing Auto Resolve the potential faulty position of the analog module will be corrected and the module is moved to the first position before the Digital IO's.




The next analog module ST-3708 will first be placed to the far right. Then user can either press Auto Resolve that moves the module to the second analog position, or by manually moving it in a new position by drag&drop. When corrected the Warning sign and the message on the top will disappear.



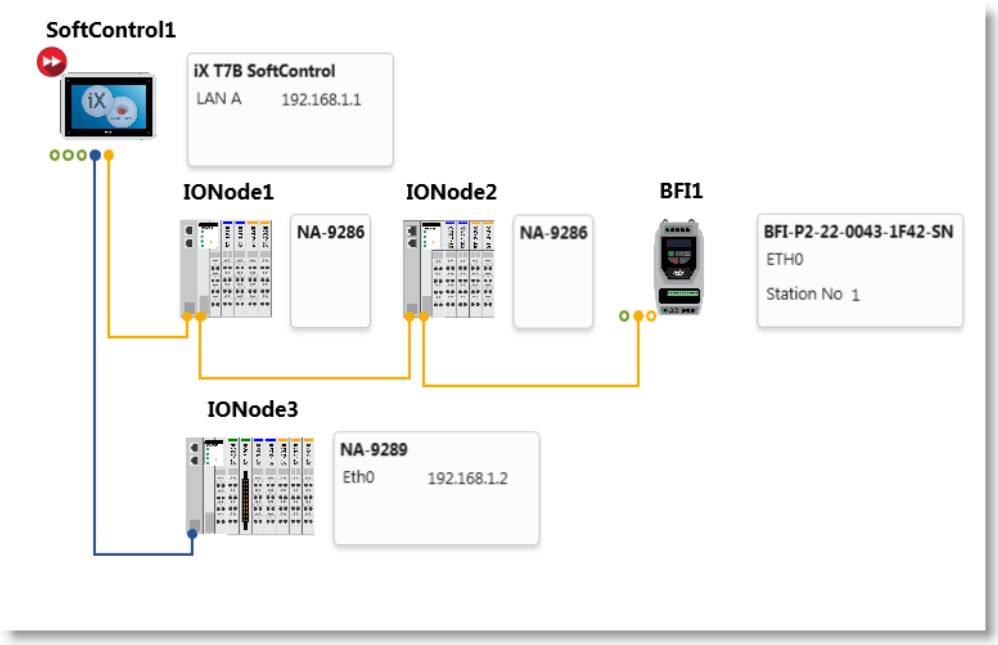
# 7 Project example

Create an iX SoftControl project with distributed IO and a BFI inverter connected via EtherCAT and Modbus TCP.

One **iX T7B SoftControl**, 2 pcs of **NA-9286** EtherCAT network adaptor with I/O modules, and last one **BFI-P2** inverter with EtherCAT interface.

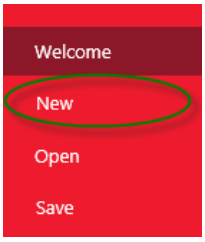
The second bus with Modbus TCP connects to one **NA-9289** network adaptor with I/O modules.

Add a Smart object for the BFI inverter to get our template control function with PLC code and connection to a HMI screen in the SoftControl application.



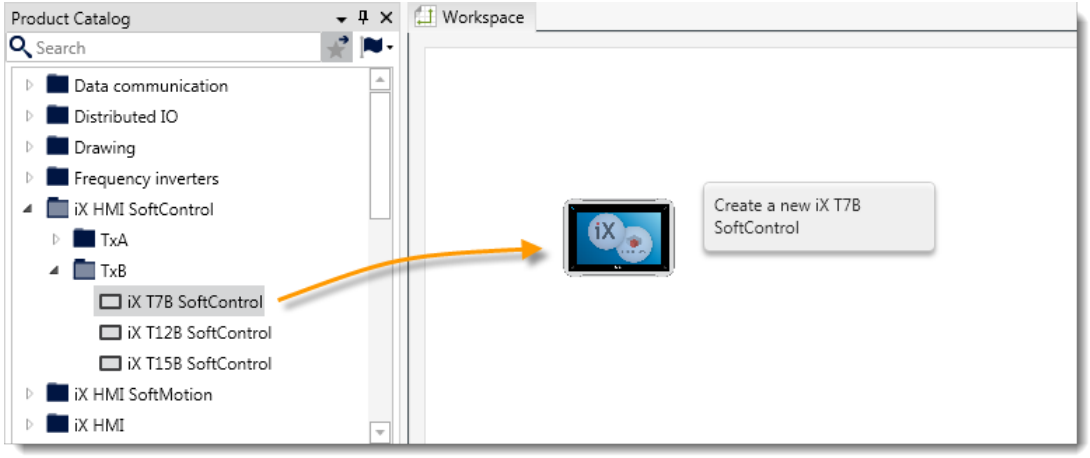

## 7.1 New project

Create a new WARP project using
File – New selection in the Main menu.



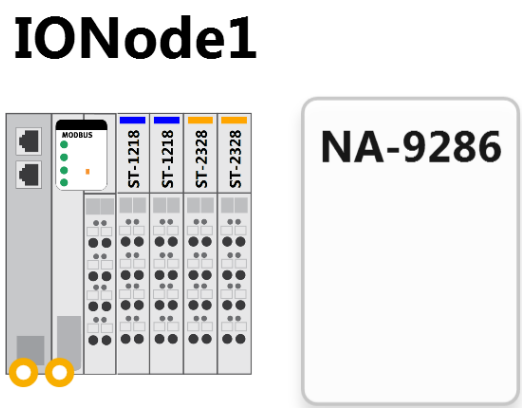

## 7.2 Add devices to Workspace

First place an iX T7B SoftControl on the workspace by drag&drop.



## 7.3 Build up the network modules

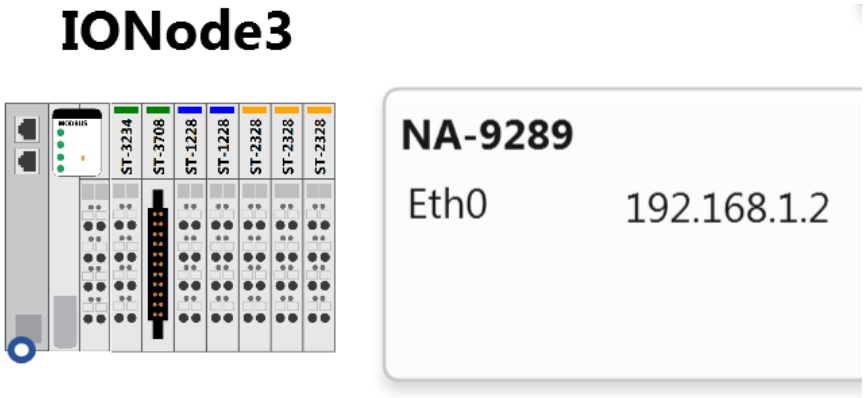
Selecting input and output modules from the Product Catalog, we now expand the slave IO nodes. Both **IONode1** and **IONode2** are identical, each **NA-9286** consist of:
2 pcs of **ST-1218** and 2 pcs of **ST-2328**.



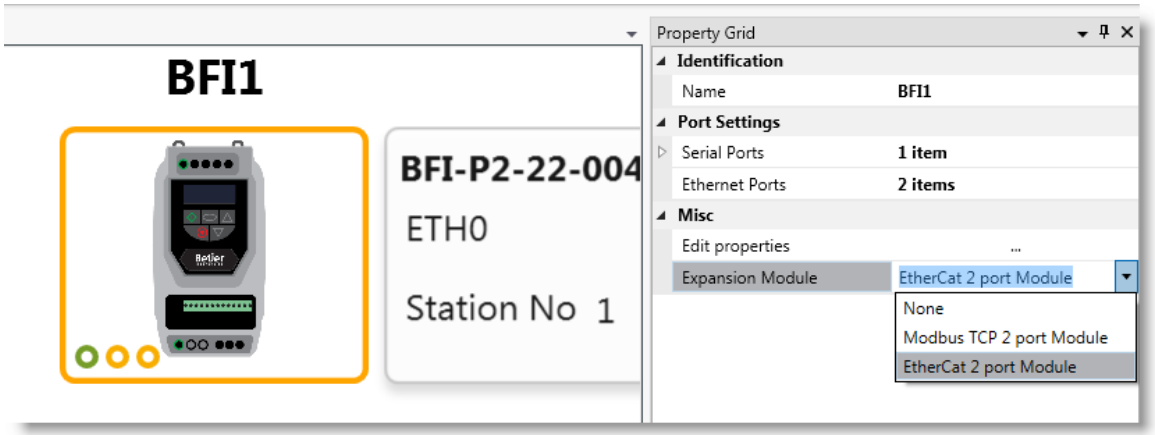

**IONode3** is a **NA-9289** network adaptor and consist of:
2 pcs of **ST-1218** and 2 pcs of **ST-2328** and the analog modules **ST-3234** and **ST-3708**.

Try to drag and drop slices in different positions to see the warning message in WARP and use the Auto Resolve to fix the issue.

Change the IP address to **192.168.1.2**.

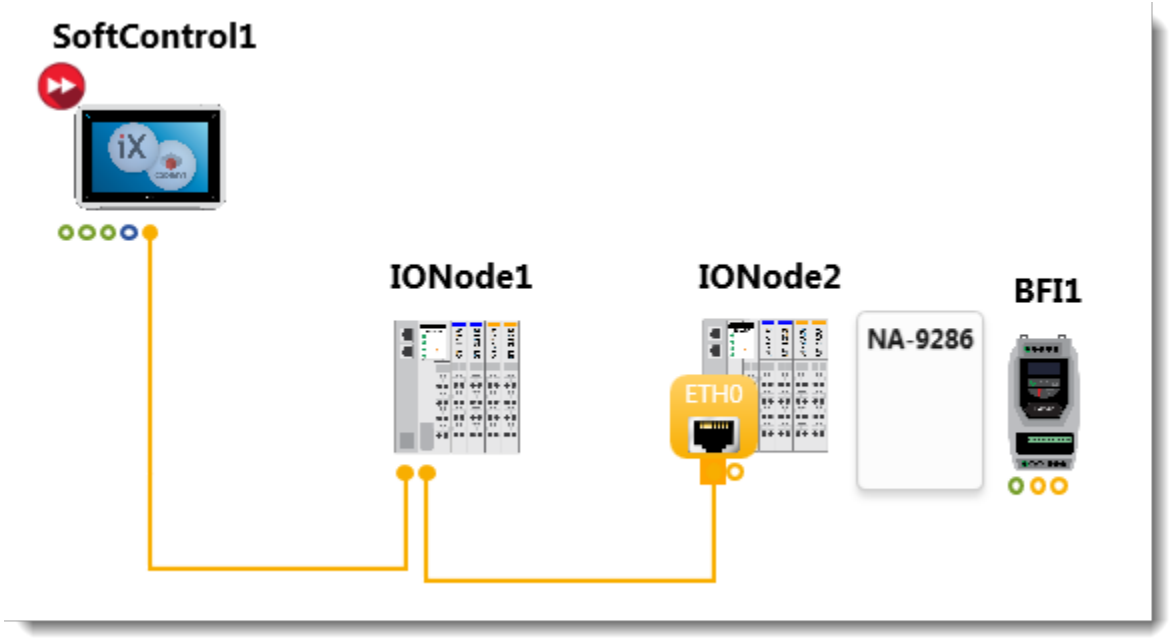



Add a **BFI-P2** inverter that becomes the **BFI1** object, then change properties to EtherCAT interface as shown in the picture below. The EtherCAT module will be included in the Bill of Materials.

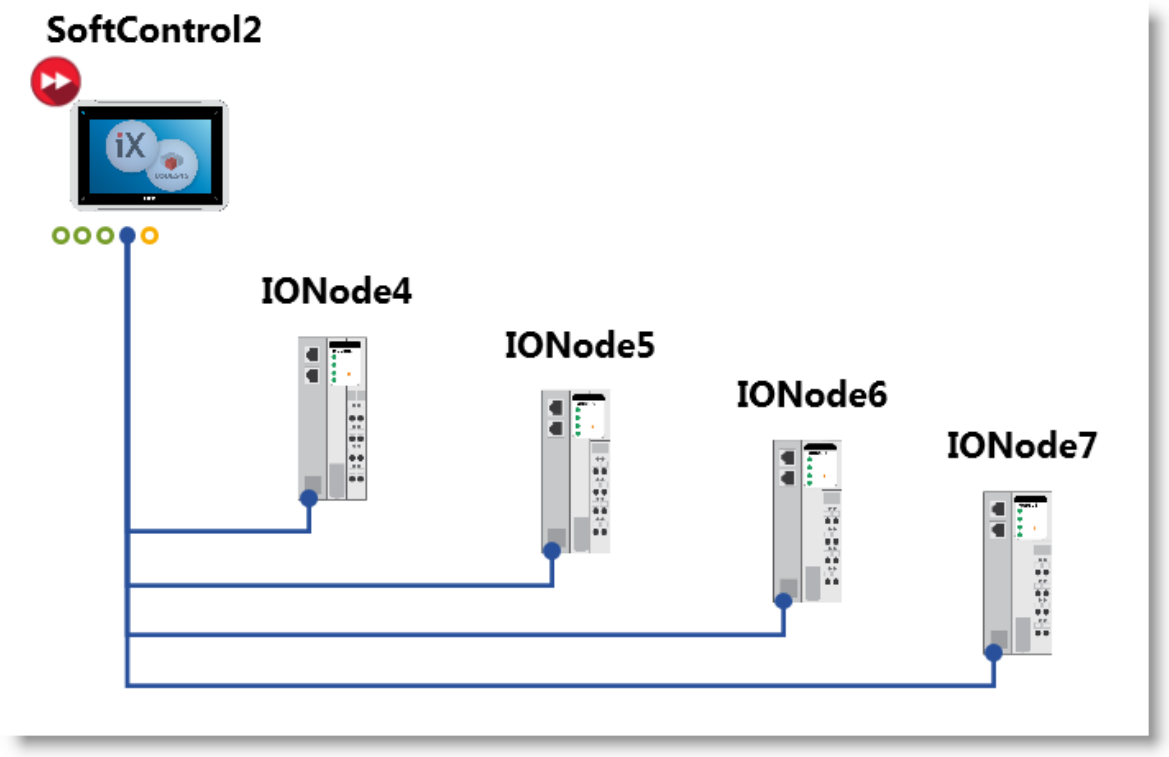


## 7.4 Connect the device in the workspace

In EtherCAT the correct physical order of the slaves is very important, or else the communication will not start. Therefore the lines of EtherCAT bus are connected first from the Master device to the first Slave input connector. Then from the output connector of a slave to the next input connector of next slave device to build the complete bus structure.




**Note!**
In the current implementation of WARP, all Modbus (RTU/TCP) connections are logical. Connect from the Master/Client to each Slave/Server, according to example below:

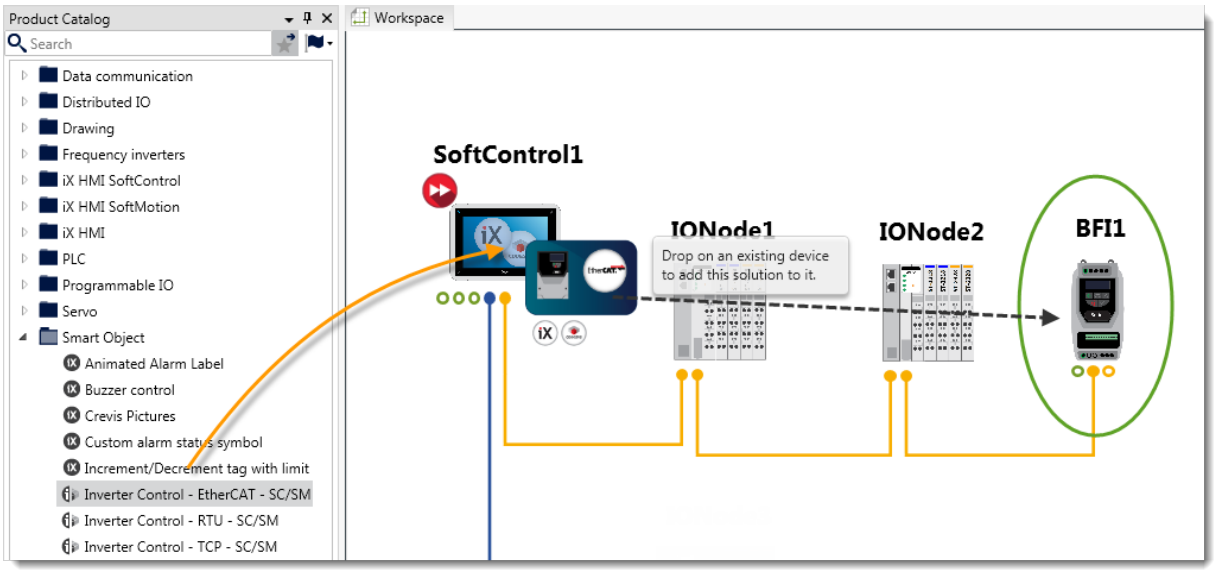


## 7.5 Add a smart object to the SoftControl

A great feature of WARP Engineering Studio is the introduction of smart objects. A smart object includes ready-made, embedded functionality such as PLC code, HMI screens, Tags, Alarms and even C# scripting. Drag a smart object into your work space and just drop it on a device. All embedded code is then injected into the targeted devices. Smart and time-saving.

In the example below the smart object *Inverter Control* will add on both PLC and HMI application parts to the BFI1 object. For example function blocks, program code and global variables of CODESYS connected to tags and a screen in the iX Developer application.

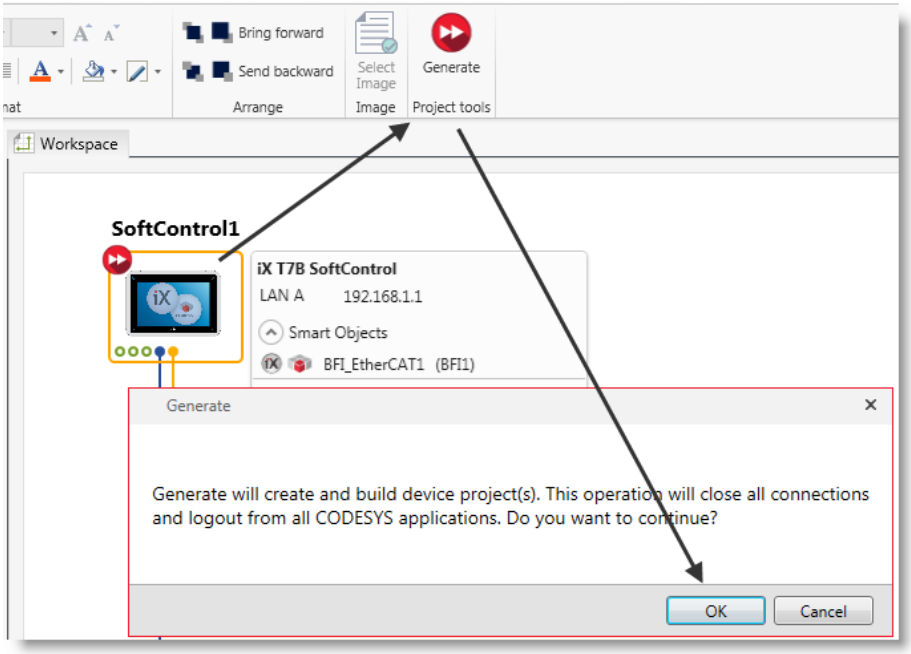
Drag the smart object from the product catalog and drop on the SoftControl1 or BFI1 object in the workspace to add the new functionality.




## 7.6 Generate project

By pressing Generate in the ribbon or context menu, an automatic procedure will start to create the device project and build the code, screens, and communication settings and tag list etc.

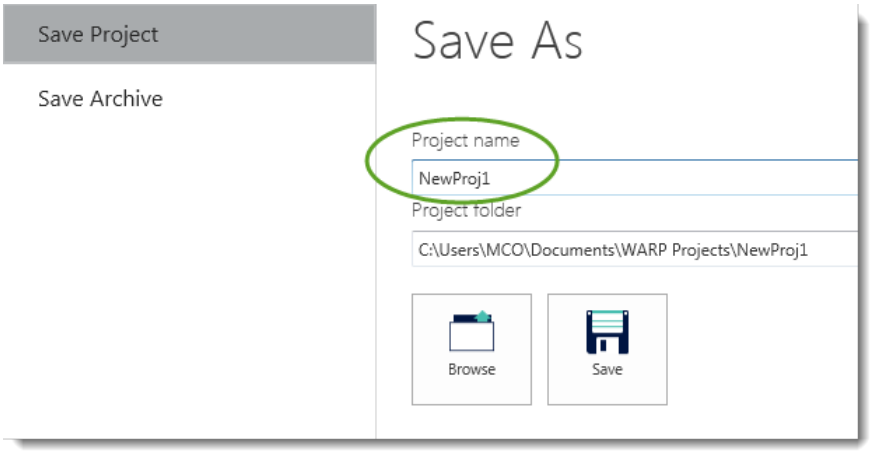
WARP will automatically ask the user to save the WARP project before generation, and start both CODESYS and iX Developer after pressing OK.



### 7.6.1 Saving the new project - Save As

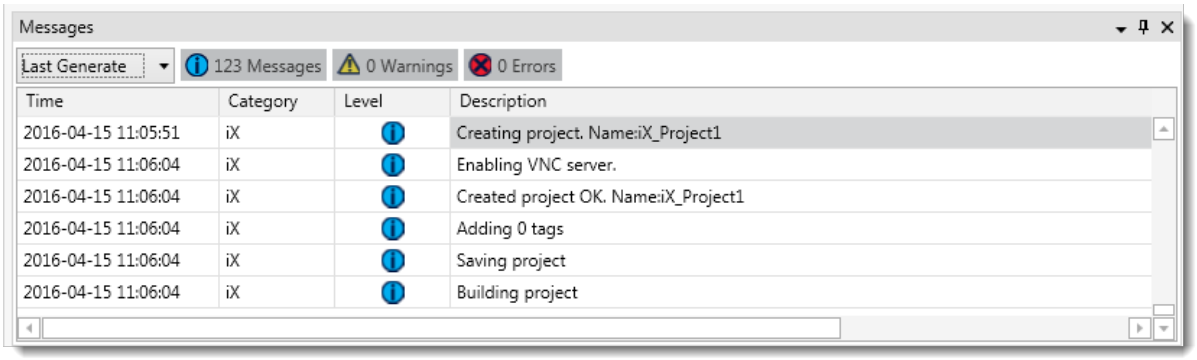
Project can be saved with a default name so change this name and Save with new name, NewProj1.

Note that project path can be change by pressing Browse before saving.



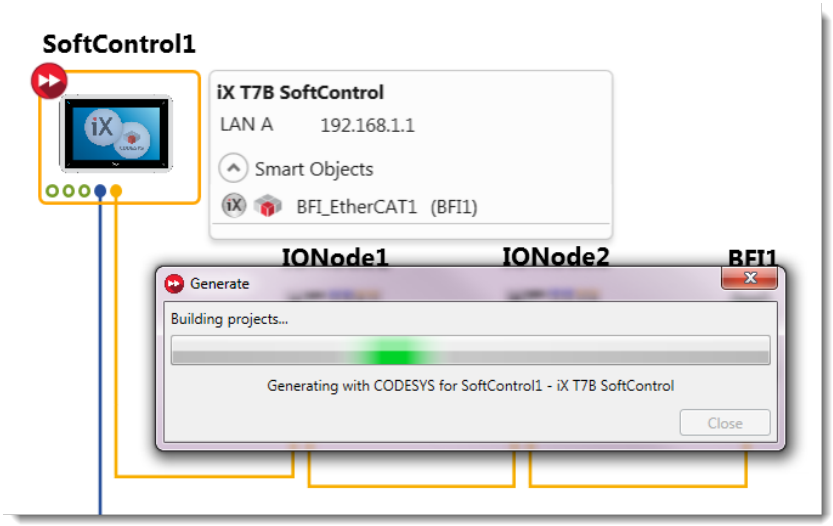

## 7.7 Messages during generate

In the separate Message window all Info / Warning / Errors during the generate procedure are logged so it easily can be checked later. Each row contain time stamp, category and level of importance. The description text is the same as seen in the message box during generating process.




### 7.7.1 Generate message box

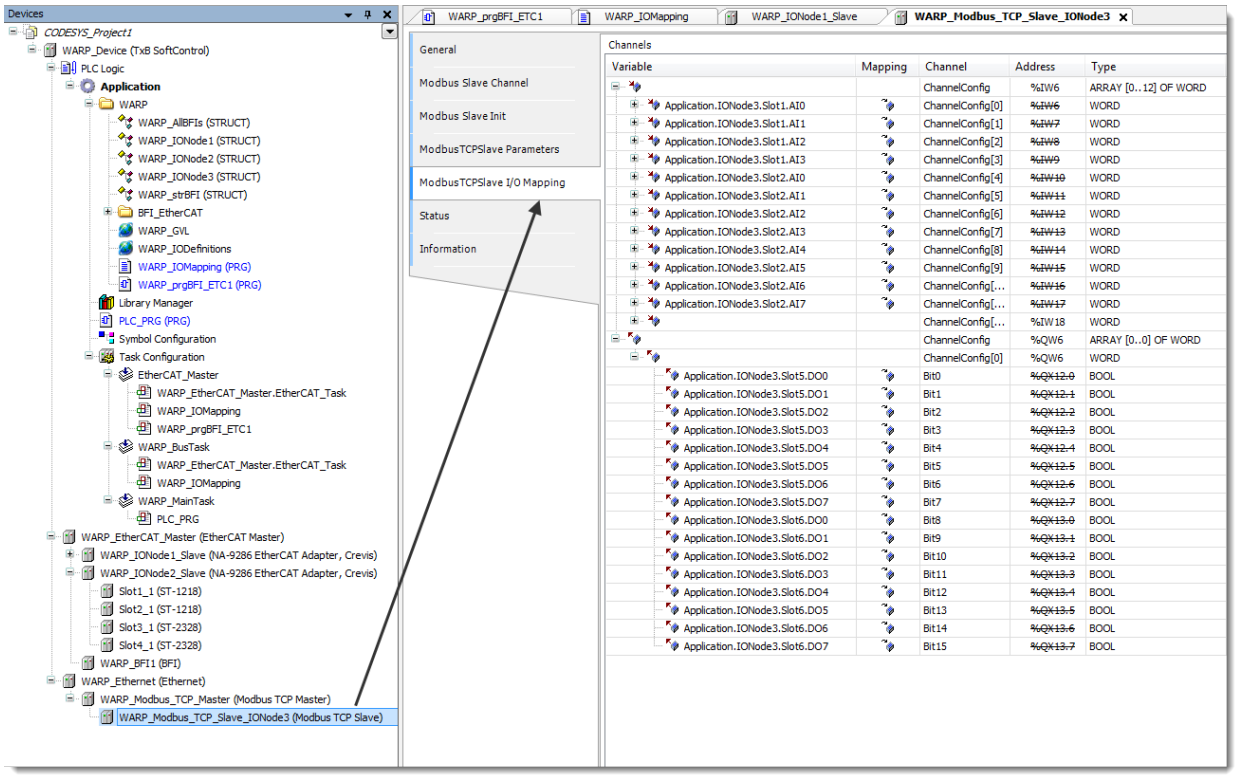
During the generate procedure, each step are shown in a message box.



## 7.8 Variable declarations and WARP objects in CODESYS

Objects generated by WARP will be marked with a prefix "*WARP_*". This makes it more easy to identify what has been automatically generated by WARP or customer specific code made by the user direct in CODESYS editors. For example a WARP directory is generated with all specific program code and FB's and variable declarations with scaling from the settings in *Variables* window.

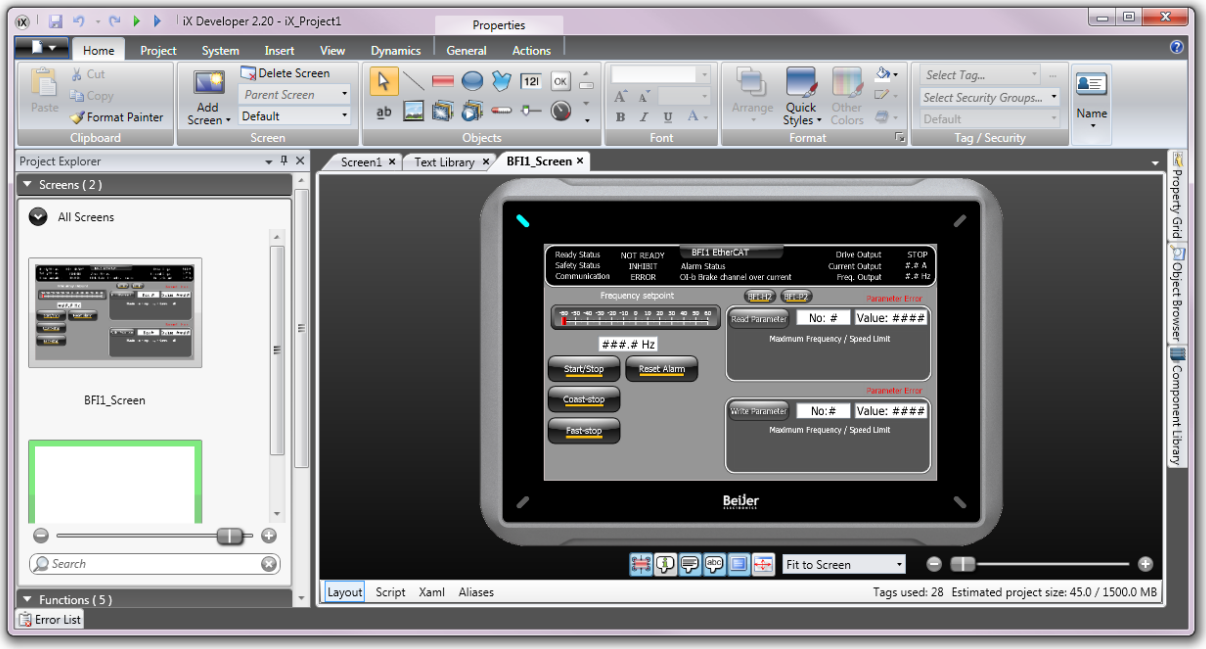
Both the *EtherCAT I/O Mapping* and the *Modbus TCP Slave I/O Mapping* are ready to be used with automatic generated variables. The picture below shows the I/O Mapping structure of IONode3.



## 7.9 Generate the HMI project

The HMI project with a BFI screen is created and the complete tag list of the BFI EtherCAT smart object is automatically exported from CODESYS global variables via the Symbol configuration, to connect the HMI and PLC part of the SoftControl.

Note that the user can add more code and screens manually in CODESYS and iX Developer. Make sure to save the project each time you have made any changes!



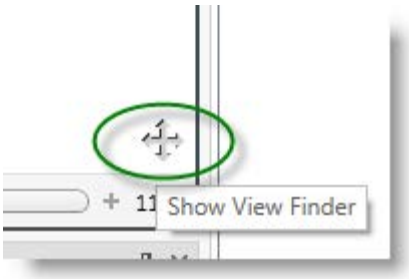

## 7.10 Download to device

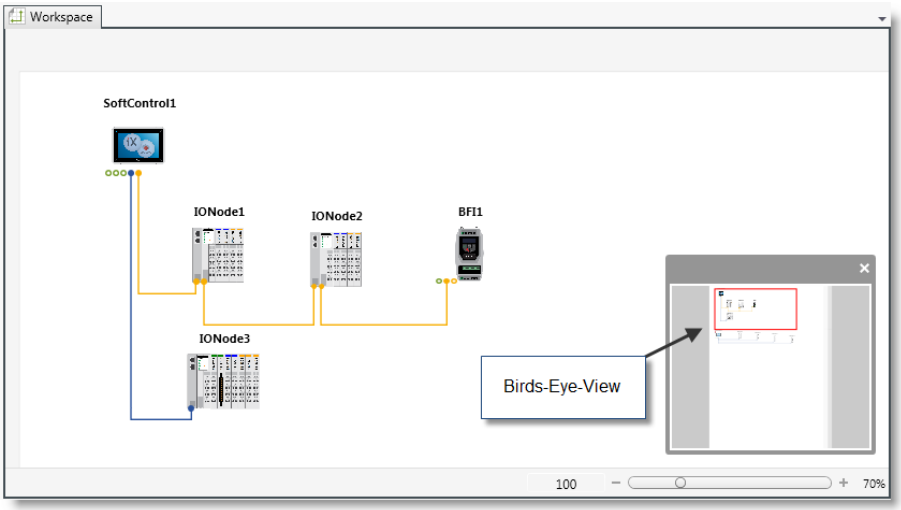
If the hardware is connected, the user may also press Download to Device from the context menu of the main device. Then the generate procedure will be made according to above and finally the PLC and HMI projects are being downloaded automatically, and the CPU set into run mode if acknowledged by the user.

## 7.11 Zoom in workspace

There are three ways to zoom in and out in the Workspace, using [Ctrl] +, or specify a square with the mouse round the required zoom area. Use [Ctrl] + [Scroll mouse].

By pressing the *Show View Finder* symbol will open the Birds-Eye-View.

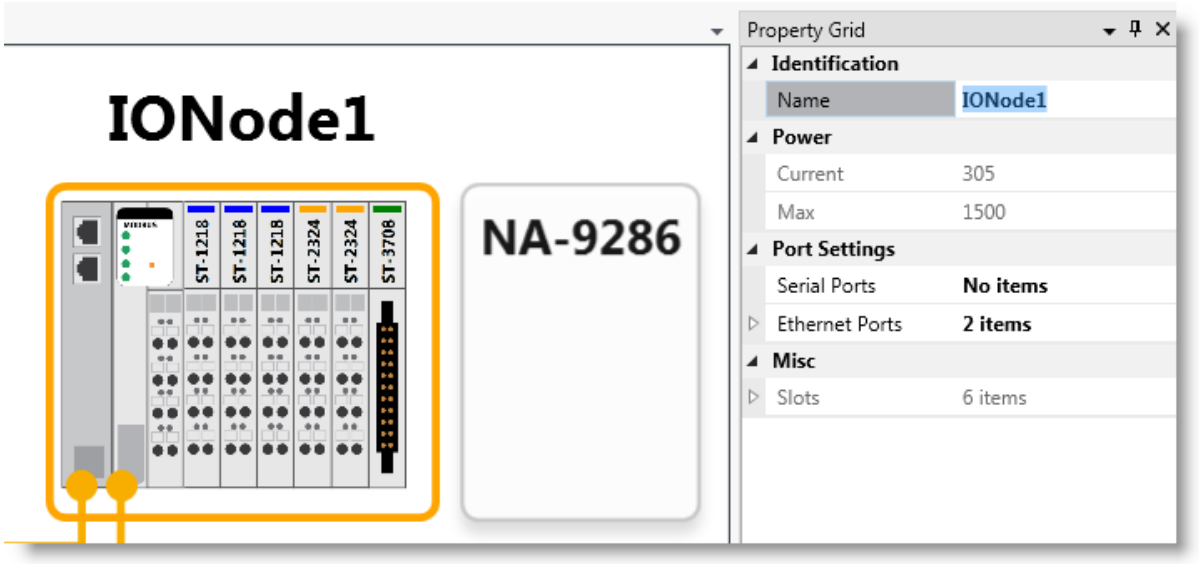


## 7.12 Device identification name

The name of each object can be changed manually to explain the solution for a unique project better. This name will influence the object code generated by CODESYS and iX Developer.

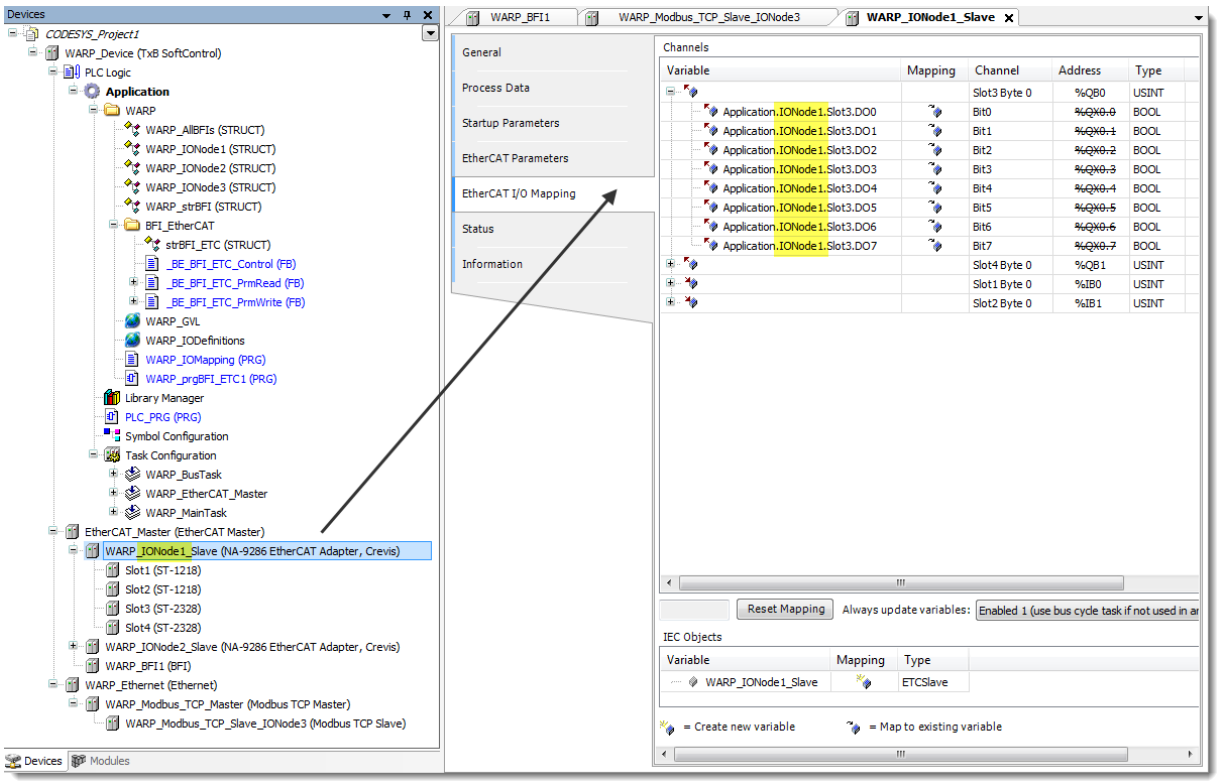
This example show a Crevis Network Adaptor NA-9286, that has the default name **IONode1**.




## 7.13 Making name change before generate

After changing name of the device object, this new name will be used as part of the identifier and object name in the CODESYS application.

All the distributed IO modules will get unique automatic variable names created for each slot. This mapping is set up automatically in the I/O Mapping table. It is similar if you are using EtherCAT or Modbus RTU / TCP as communication interface.



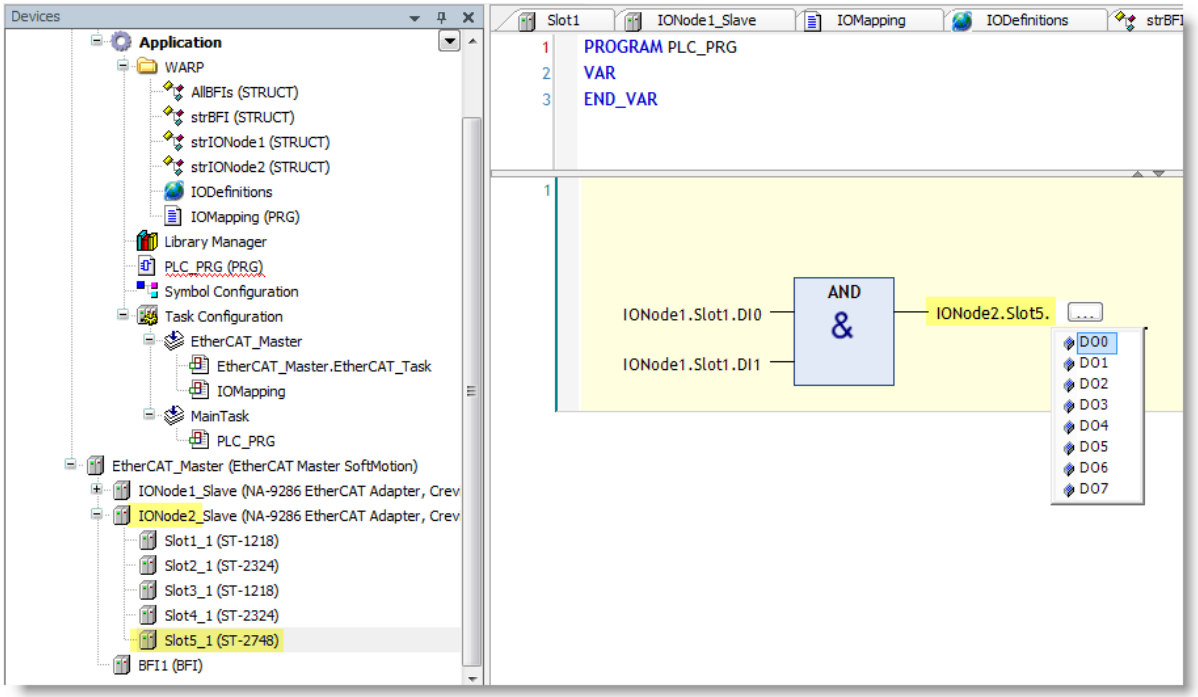



KI_eng_130531.dot

## 7.14 Use the variables in PLC code

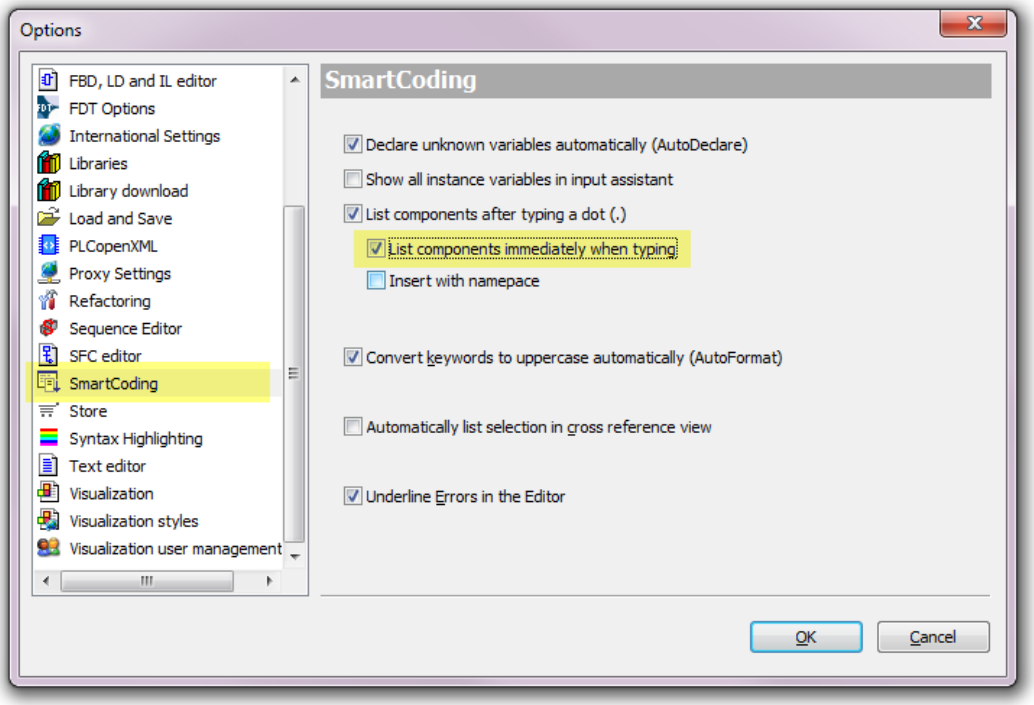
The automatic generated variable names can directly be used in the PLC code.

This makes it very easy to build up an application that uses both the input and output names from WARP without having to declare own global variables!




## 7.15 Tip! SmartCoding in CODESYS

To get the dropdown selection dialog in the editor of CODESYS 3.5, you need to activate the option SmartCoding and check *List components immediately when typing* (Tools – Option).



# 8 Project backup

## 8.1 Save project

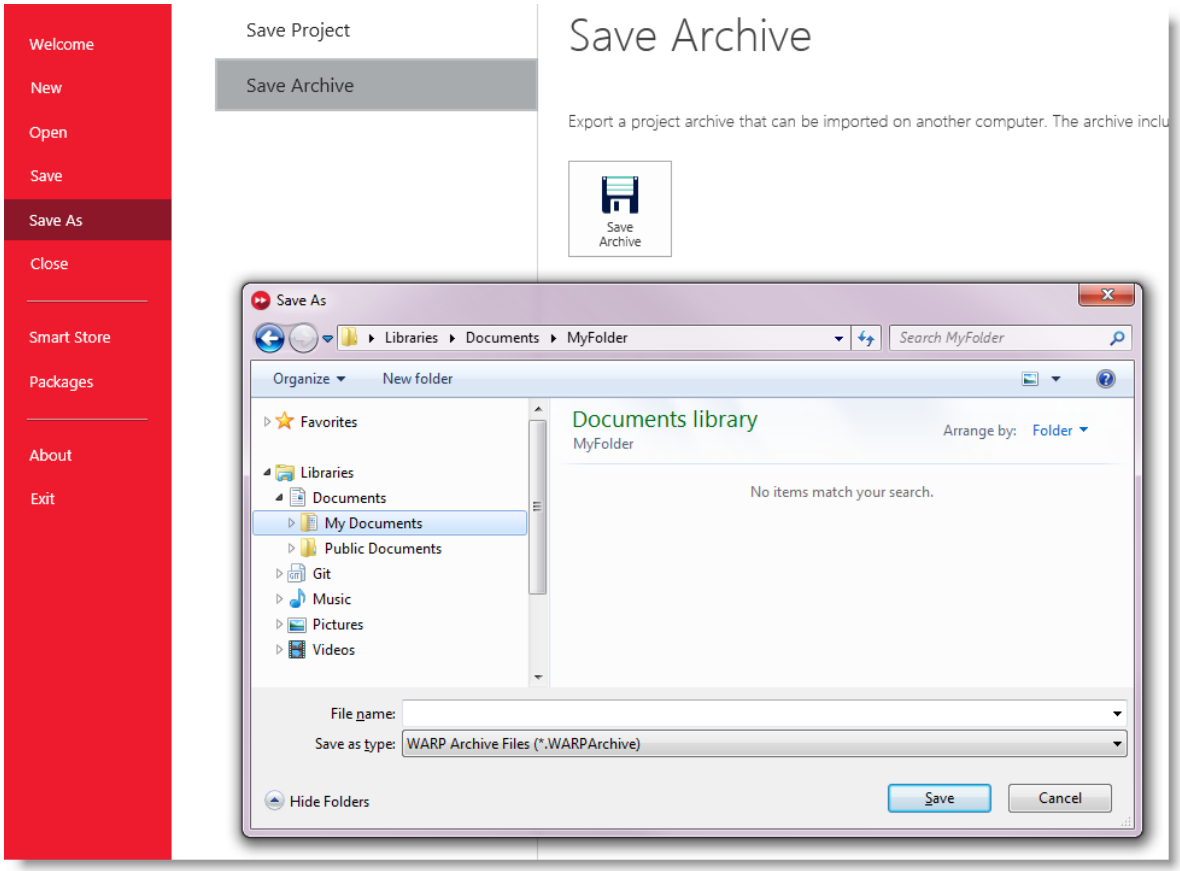
All devices and the projects of iX Developer, CODESYS and BCS Tools are saved in one directory of the WARP Engineering Studio applications. WARP project file has extension *.pproj.

The default path are C:\Users\User\Documents\WARP Projects\MyProjDir\ <projectname>. pproj.

## 8.2 Project archive

The best way to backup all components from a WARP application and all of its devices, objects and attachments is to make a '**Save Archive**'. That will save and pack all files referenced and used within the currently opened project in to one archive file (*WARPArchive).

The archive file can easily be unpacked by use menu Open.

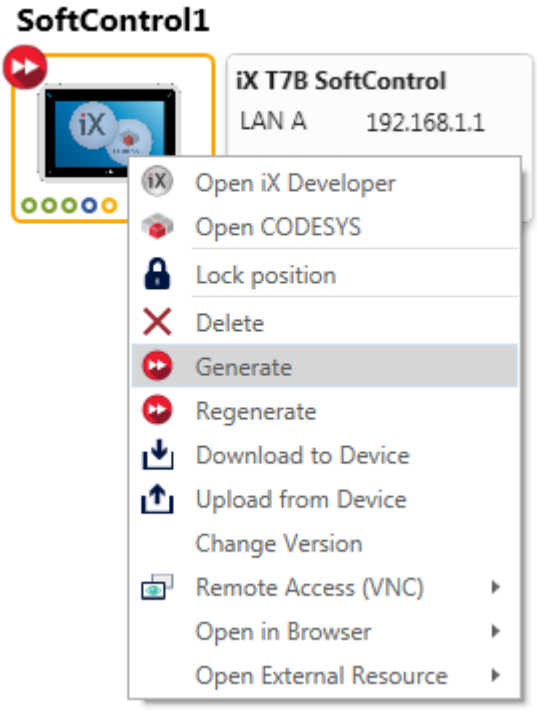



*Select Save Archive in the menu*

# 9 Appendix

## 9.1 Notes regarding Generate, Regenerate, Download and Upload

The functionality in version 1.0 of Generate, Regenerate, Download to Device (Deploy) and Upload from Device (Retrieve) are:

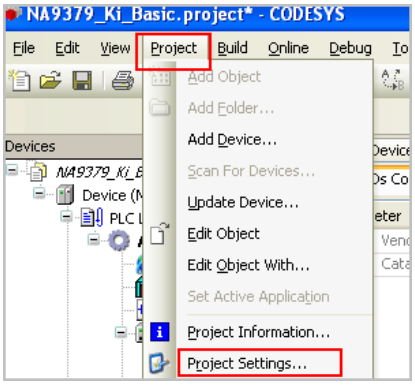


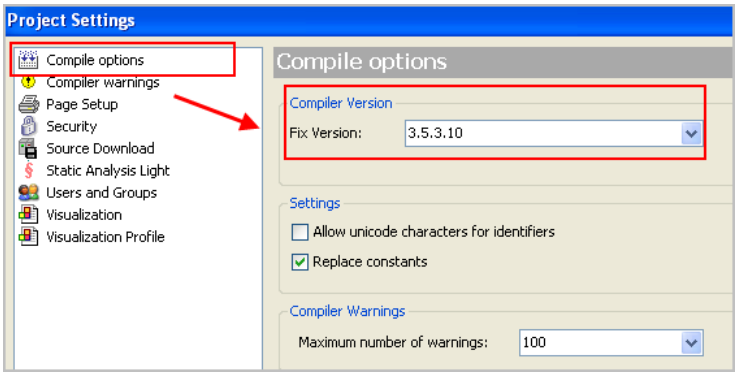

- **Generate**
  - WARP builds a project of the changes made in the workspace after last generation. Before the generate procedure a restore point will be saved.
  - Some parts will always be regenerated, for example the EtherCAT / PIO HW tree will always be built again.
- **Regenerate**
  - Regenerate will rebuild the unique WARP objects created by the generation procedure. The user specific code directly in CODESYS or iX will not be removed.
- **Download to Device (Deploy)**
  - iX Developer: If the red WARP symbol can be seen on the device after manual modification of an iX project, then make a manual Build in iX Developer before downloading. Else, the user made changes in iX Developer will not be included.
  - If the red WARP symbol does not show, the Deploy will only do a normal download of iX-project to the panel.
  - CODESYS is always compiled at download. Download is used that mean the PLC will stop, and a question will pop-up if the CODESYS should be started.
- **Upload from Device (Retrieve)**
  - For iX projects, WARP will upload the zip file including the iX project from the panel. This is the file that is downloaded automatically to the panel by iX Developer.
  - This function is not yet supported for CODESYS application.

## 9.2 How to check compiler and library versions in CODESYS 3.5

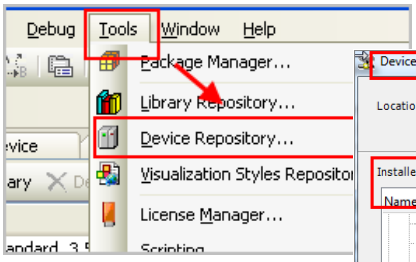
**Note!** The pictures below might include older device / details / compiler-versions and can differ from device to device The examples are only intended to show where to find the setting in CODESYS.

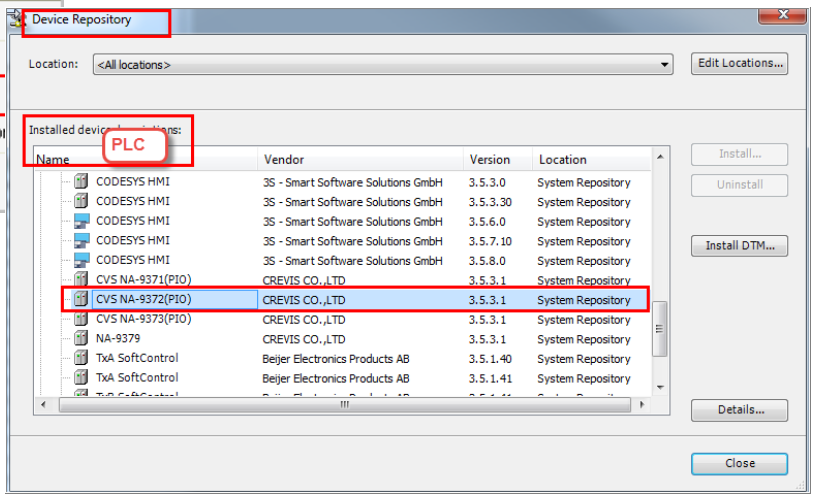
### Check CODESYS compiler version



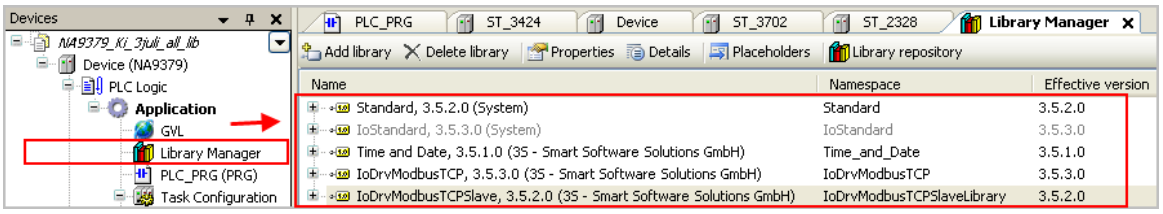




### Check a device version




### Check CODESYS library version

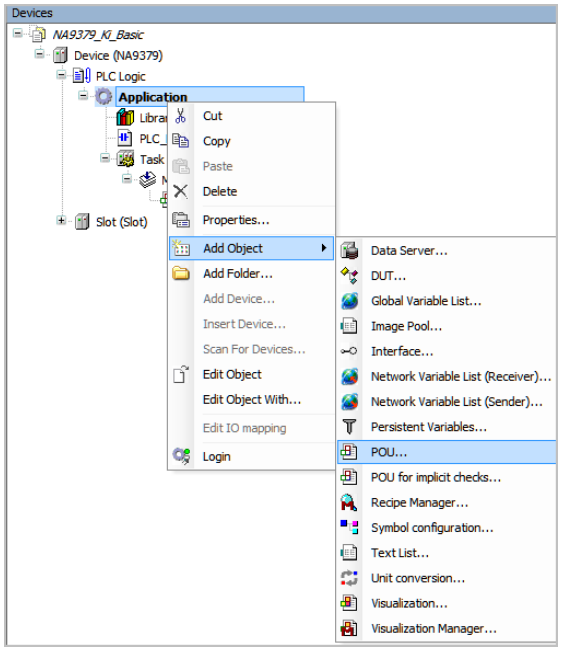


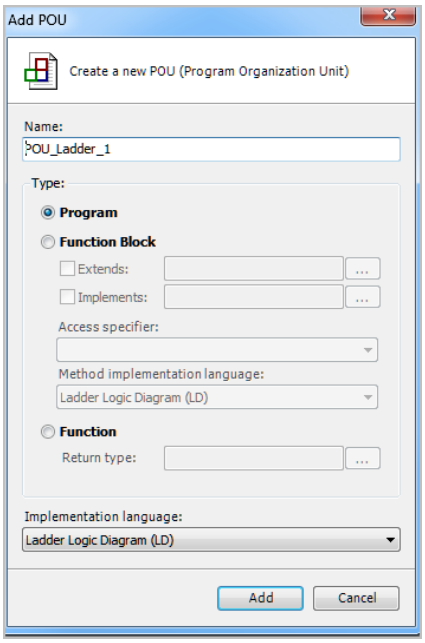
## 9.3 CODESYS project objects

**Note!** The pictures below include older device / details / compiler-versions and differ from device to device. The examples are only intended to show where to find the menu and settings in CODESYS.

### How to add a POU

- Highlight the “Application” option in the “Device” window and right click
- Select “Add Object” and choose “POU”.
  Customize the POU according to desired characteristics.

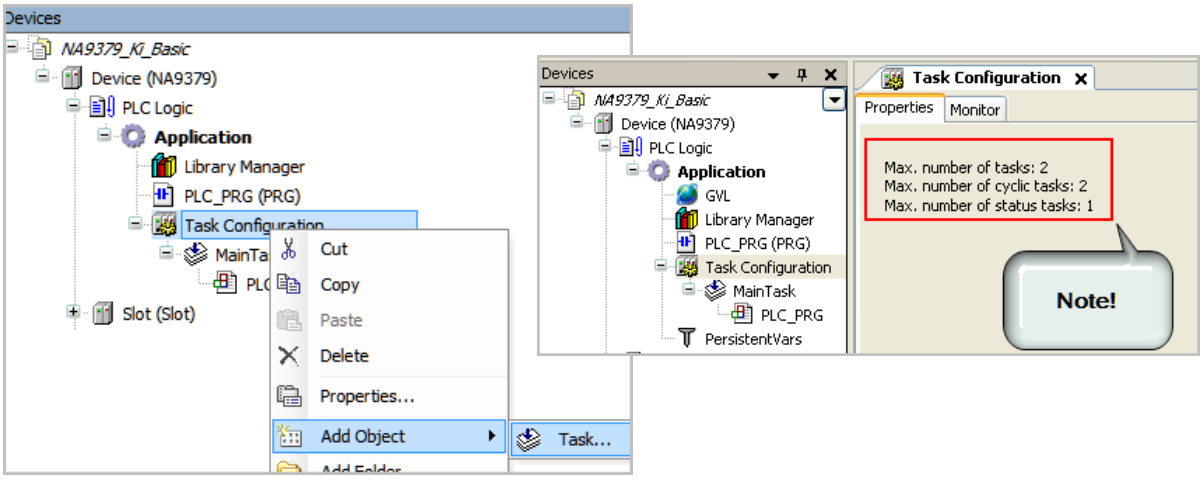



### How to add a Task

When a “Standard project” is selected, it automatically creates a "Main Task"

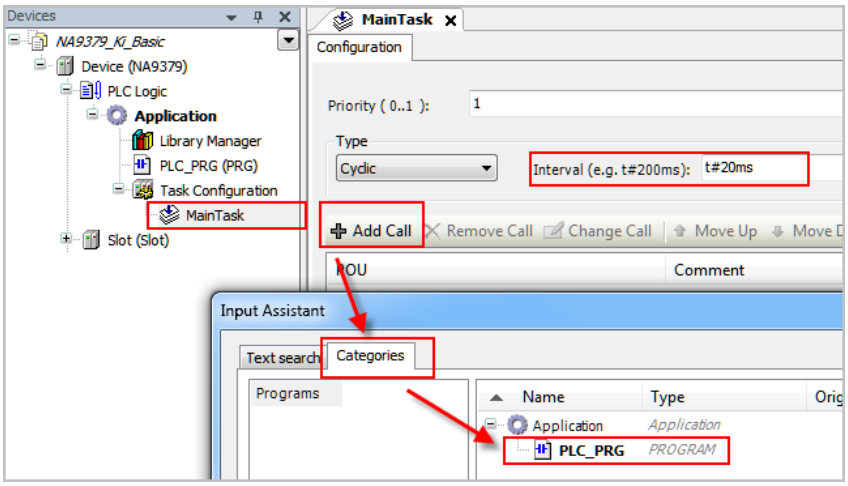
- Highlight the “Application” option (to make a “Task Configuration”) or highlight the “Task Configuration” (to make a “Task”)
- Select “Add Object” and choose “Task Configuration” or “Task”
- Click “Open”, and a task named “Task” is automatically created



### How to add a POU to a Task

Connect the created POU to the task in order for the IEC code in the POU to be executed.

- Open the “MainTask”
- Under “POUs”, select “Add POU”
- In the “Items” window, under “Application”, highlight the desired “POU” and click “OK”




## 9.4 Trouble shooting SW use logfiles

Log files created by WARP and iX Developer in case of error in the software generate procedure are saved in separate directories, see below paths:

**WARP (general):** C:\Users*User*\AppData\Roaming\Beijer Electronics AB*version***WARP.log**

**WARP (project):** C:\Users*User*\Documents\WARP Projects*projectname***messages.log**

**iX Developer:** C:\Users*User*\AppData\Local\Temp\NeoIde*date***NeoIDELog.log**

## 9.5 Trouble shooting in the HW communication

How to diagnose when a device cannot communicate with the network

- **Inspect wrong or missing cable connection**
  - Check status of cable connection for each node
  - Check that all color matches between connector and cable
  - Check wire omission
- **Terminator resistor**
  - If terminator resistor is not installed, install terminator resistor
  - Check location of terminator resistor
- **Configuration of Node address**
  - Check duplication of node address
- **Configuration of Master**
  - Check configuration of master
  - Check whether to do download or don’t
  - Check composition is right
    - Configuration of communication baud rate
    - I/O size
    - Configuration of each node
- **Ground and environment**
  - Check ground is contacted
  - Check environment factor (temperature, humidity, etc.) is in less than regular limit