

# EcoStruxure™ Control Expert Hardware Catalog Manager Operation Guide

(Original Document)

12/2018

EIO000002141.01

[www.schneider-electric.com](http://www.schneider-electric.com)



---

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

You agree not to reproduce, other than for your own personal, noncommercial use, all or part of this document on any medium whatsoever without permission of Schneider Electric, given in writing. You also agree not to establish any hypertext links to this document or its content. Schneider Electric does not grant any right or license for the personal and noncommercial use of the document or its content, except for a non-exclusive license to consult it on an "as is" basis, at your own risk. All other rights are reserved.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2018 Schneider Electric. All rights reserved.

---

# Table of Contents

---



	<b>Safety Information</b> .....	<b>5</b>
	<b>About the Book</b> .....	<b>7</b>
<b>Chapter 1</b>	<b>Hardware Catalog Manager Software Implementation</b> ..	<b>9</b>
1.1	Hardware Catalog Manager Overview .....	<b>10</b>
	Hardware Catalog Manager Description .....	<b>11</b>
	Hardware Catalog Manager Contents .....	<b>13</b>
1.2	Using the Hardware Catalog Manager .....	<b>16</b>
	How to Launch the Hardware Catalog Manager .....	<b>17</b>
	How to Add a Device to the Hardware Catalog Manager .....	<b>18</b>
	How to Add a Function on a Device .....	<b>22</b>
	Basic Configuration Parameters .....	<b>23</b>
	Expert Mode Configuration Parameters .....	<b>28</b>
	MFB Function for Expert Mode .....	<b>40</b>
	How to Copy or Delete a Function .....	<b>46</b>
	How to Import/Export or Delete One or Several User Devices .....	<b>47</b>
	How to Close the Hardware Catalog Manager .....	<b>50</b>
	Example of How to Create a Dedicated and Optimized STB Island ..	<b>51</b>
1.3	Troubleshooting .....	<b>52</b>
	Troubleshooting .....	<b>53</b>
	EDS/DCF Import Anomaly Code .....	<b>57</b>



---

# Safety Information

---



## Important Information

### NOTICE

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



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



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

## **DANGER**

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

## **WARNING**

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

## **CAUTION**

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

## **NOTICE**

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

---

## PLEASE NOTE

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

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

---

# About the Book

---



## At a Glance

### Document Scope

This manual describes the Hardware Catalog Manager implementation.

### Validity Note

This documentation is valid for EcoStruxure™ Control Expert 14.0 or later.

### Product Related Information

 <b>WARNING</b>
<b>UNINTENDED EQUIPMENT OPERATION</b> The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise are allowed to program, install, alter, and apply this product. Follow all local and national safety codes and standards. <b>Failure to follow these instructions can result in death, serious injury, or equipment damage.</b>



---

# Chapter 1

## Hardware Catalog Manager Software Implementation

---

### Subject of this chapter

This chapter describes the Hardware Catalog Manager software implementation.

### What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
1.1	Hardware Catalog Manager Overview	10
1.2	Using the Hardware Catalog Manager	16
1.3	Troubleshooting	52

# Section 1.1

## Hardware Catalog Manager Overview

---

### Subject of This Section

This section presents the Hardware Catalog Manager overview

### What Is in This Section?

This section contains the following topics:

Topic	Page
Hardware Catalog Manager Description	11
Hardware Catalog Manager Contents	13

## Hardware Catalog Manager Description

### Overview

The Hardware Catalog Manager is a specific software, installed with Control Expert, that allows management of CANopen devices in the Control Expert catalog database.

The Hardware Catalog Manager can be used to:

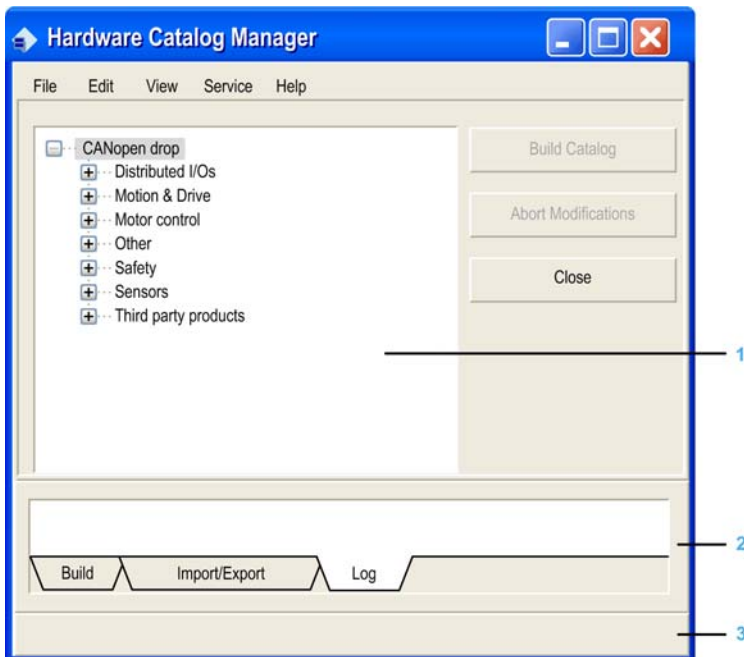
- Integrate third-party products.
- Add, remove, and configure access to CANopen devices on the fieldbus.
- Minimize the size of the CPU memory reserved for a given equipment.
- Customize the user interface.

The Control Expert user interface works on a catalog database which is installed with Control Expert, and uses some basic services from the Control Expert installation.

**NOTE:** An overview of the Hardware Catalog Manager in read-only mode is available in Control Expert through the **Hardware Catalog**.

The devices are added in the standard catalog in Control Expert and the devices can be used in projects as any devices that are provided with Control Expert.

The following illustration shows the Hardware Catalog Manager main screen.



- 1 Display window for device families, devices and functions.
- 2 **Output Window**
- 3 **Status Bar**

The **Output Window** has a context menu with the following information:

- **Build:** This is used to display information about the build progress.
- **Import/Export:** This is used to display information about import / export.
- **Log:** This is used to display information and operational status during **Add Device** or **Add Function**.

## WARNING

### RISK OF LOST APPLICATION

To avoid a complete reinstallation of Control Expert software and/or corrupting the STU files:  
Do not interrupt a build of the Hardware Catalog Manager.

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

**NOTE:** Only one instance of the Hardware Catalog Manager can be open at one time.

### Description

These following characteristics have to be taken into account during the development of the application:

- Multi-mapping is not authorized: you cannot map the same variable several times in many active PDOs.
- Maximum CANopen object length is 32 bits. Bit type object variables are not supported, although bit type parameters are supported.

**NOTE:** If the non-supported objects are not mapped in a PDO, they are deleted from the list and a warning message is generated. However, if these objects are mapped, the EDS file cannot be imported.

**NOTE:** The Manufacturer specific Datatypes are not supported : 40H, 41H, 44H, 50H, 51H, 52H, 54H, and 59H.

- A R/W variable must be mapped only once, either on a RX PDO or on a TX PDO.
- The CANopen module with profile V2.0B (coded COB-ID on 29 bits) is not supported. Profile V2.0A (coded COB-ID on 11 bits) is supported.
- Bitmapping: for an Input/Output module, you can configure it like an FTB device: each Input/Output is linked to a channel which can be associated to a topological address (bit). Therefore, it is possible to retrieve and use the signal directly on this bit instead of extracting it from the %IW.

## Hardware Catalog Manager Contents

### Overview

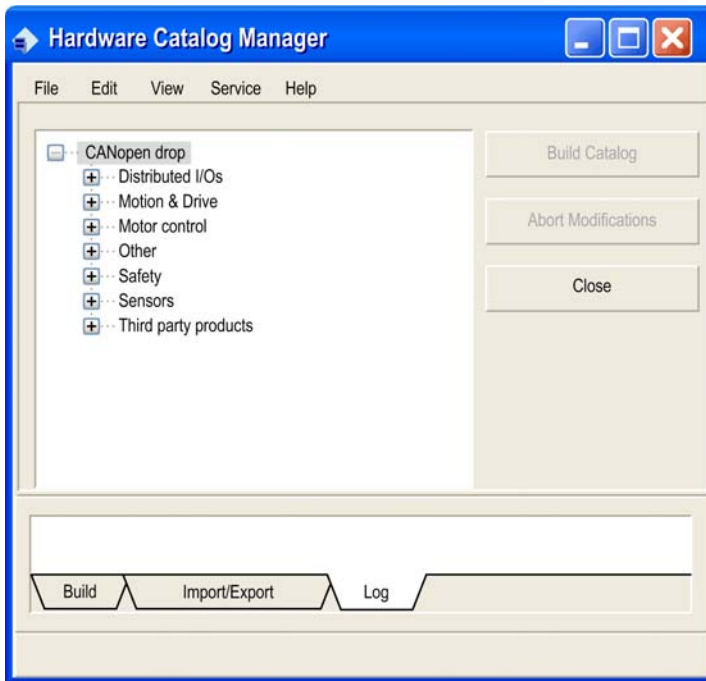
The Hardware Catalog Manager is composed of three levels of data:

- Device families
- Devices
- Functions

### Devices Families

The device families contain all the devices specific to each family: **Distributed I/Os, Motion & Drive, Motor Control, Other, Safety, Sensors, and Third party products.**

The following illustration shows the different device families:



The menu **View** is a context menu with the following information:

- **Status Bar:** Show/Hide **Status Bar**.
- **Output Window:** Show/Hide **Output Window**.
- **View Function:** Display the selected function.

## Devices

Devices are individual external units that may offer one or several different functions.

A device is identified in the catalog by its name. The default name is taken from the EDS file although this can be modified. The device name must be unique in the catalog.

There are two different kinds of device:

- Pre-programmed devices



Information about these devices is included with Control Expert software.

The user cannot modify the list of pre-programmed devices. An icon with a Schneider logo is shown on the left of a Schneider device name.

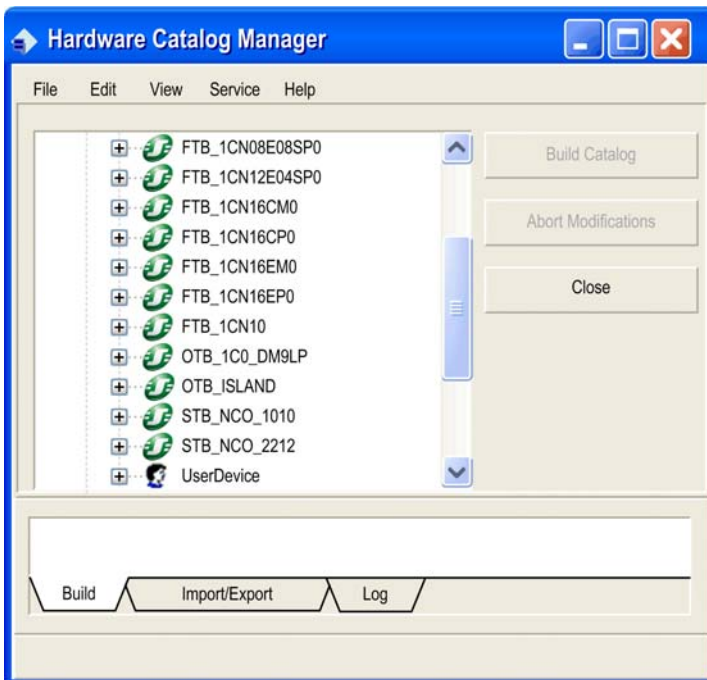
**NOTE:** When clicking on a CANopen device, the list of PLC which support this CANopen device is given at the end of the displayed text in the Status Bar. The same information is given by a tooltip when the mouse pointer rests on a CANopen device.

- User devices

Any device which is not part of the Schneider offer is considered to be a user device.

User devices can be deleted or reconfigured in the Hardware Catalog Manager. User devices that have been added but not yet built are shown with a  icon. Devices that have been built are indicated with a  icon.

The following illustration shows a list of devices and one user device in the family **Distributed I/Os**:



## Functions

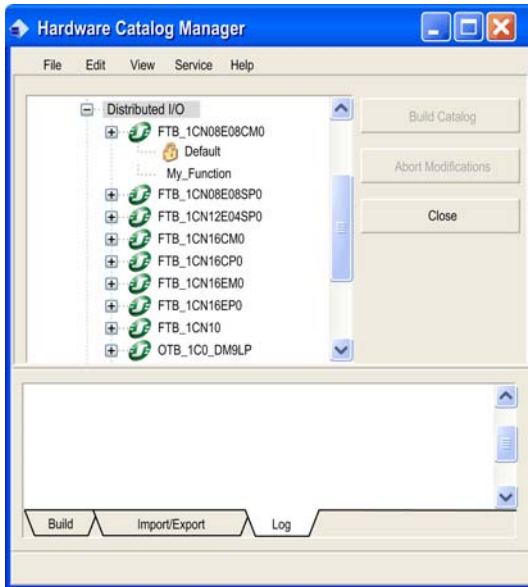
A function is a sub-category within a device. The user can create functions in order to use only a limited subset of capabilities for that device.

A default name is proposed for the first function although it can be modified. The function name must be unique within a given device. Each function has its own IODDT/DDDT.

There are two different kinds of function:

- Pre-programmed function  
Information about this function is included with Control Expert software.  
The user cannot modify a pre-programmed function, this is indicated by a padlock on the left of the function.
- User function  
Any function which is not part of the Schneider offer is considered to be a user function.  
User functions can be deleted or reconfigured in the Hardware Catalog Manager.

The following illustration shows a pre-programmed device with both pre-programmed and user functions:



Example:

Device family: **Distributed I/Os**

Device: **FTB\_1CN08E0BCMO**

Functions:

**Default** (write protected Schneider functions)

**My\_Function** (User function)

# Section 1.2

## Using the Hardware Catalog Manager

---

### Subject of this Section

This section presents the different steps to use the Hardware Catalog Manager.

### What Is in This Section?

This section contains the following topics:

Topic	Page
How to Launch the Hardware Catalog Manager	17
How to Add a Device to the Hardware Catalog Manager	18
How to Add a Function on a Device	22
Basic Configuration Parameters	23
Expert Mode Configuration Parameters	28
MFB Function for Expert Mode	40
How to Copy or Delete a Function	46
How to Import/Export or Delete One or Several User Devices	47
How to Close the Hardware Catalog Manager	50
Example of How to Create a Dedicated and Optimized STB Island	51

## How to Launch the Hardware Catalog Manager

### At a Glance

This is the procedure to launch the Hardware Catalog Manager.

**NOTE:** Hardware Catalog Manager and Control Expert software cannot both be running at the same time.

### Procedure

The table below shows the procedure to launch the Hardware Catalog Manager

Step	Action
1	Verify that Control Expert software is not running <ul style="list-style-type: none"><li>• if Control Expert Software is running, close it</li><li>• if Control Expert Software is not running, go to the next step</li></ul>
2	Select: <b>Start → Programs → EcoStruxure Control Expert → Hardware Catalog Manager</b>

## How to Add a Device to the Hardware Catalog Manager

### At a Glance

This is the procedure to add a device to the Hardware Catalog Manager.

 <b>WARNING</b>
--------------------------------------------------------------------------------------------------

<b>UNINTENDED EQUIPMENT OPERATION</b>
---------------------------------------

Verify with your device supplier that the EDS or DCF file is compatible with the firmware version of the product.
-------------------------------------------------------------------------------------------------------------------

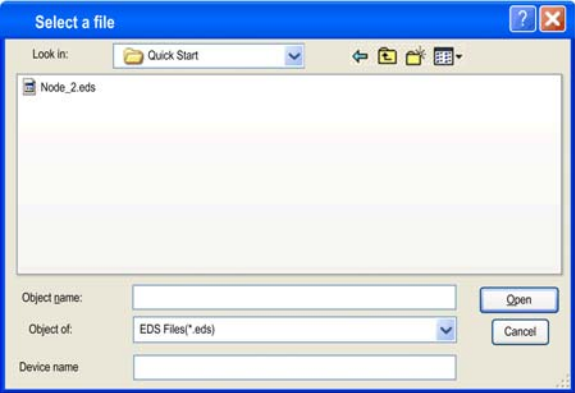
For instance, manufacturer specific Datatypes are not supported: 40H, 41H, 44H, 50H, 51H, 52H, 54H, and 59H.
--------------------------------------------------------------------------------------------------------------

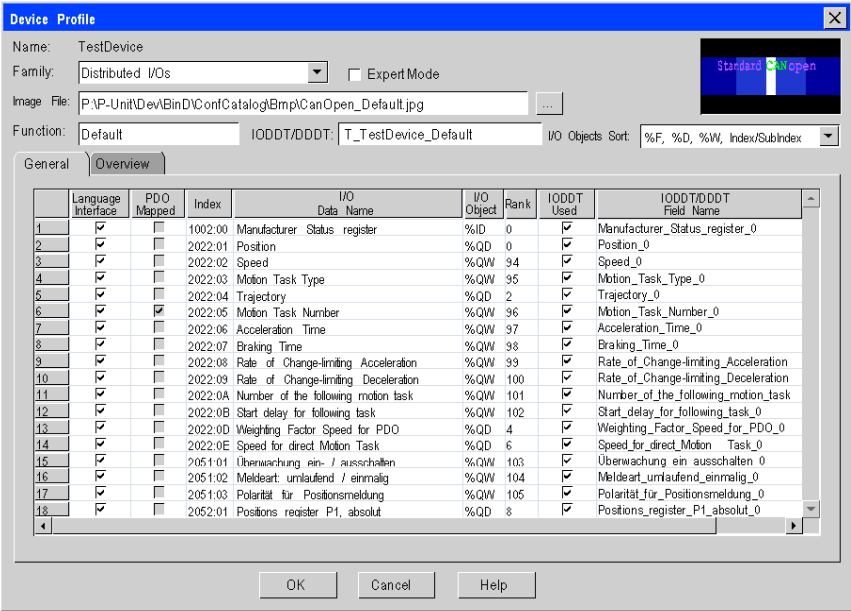
<b>Failure to follow these instructions can result in death, serious injury, or equipment damage.</b>
-------------------------------------------------------------------------------------------------------

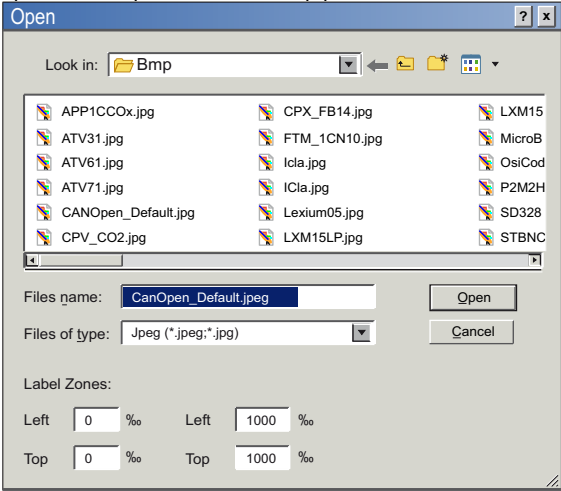

## Procedure

New devices are added to the Hardware Catalog Manager using EDS or DCF files (a DCF file can be used in certain cases, for example, for Modicon TM5 IP20 and TM7 IP67 Distributed Performance I/O).

Before starting the procedure in the following table, it is necessary to know the name and location of the EDS or DCF file corresponding to the new device to be added.

Step	Action
1	Open the Hardware Catalog Manager.
2	<ul style="list-style-type: none"> <li>● Select <b>Edit → Add Device</b></li> </ul> OR <ul style="list-style-type: none"> <li>● Right-click on a device family.</li> <li>● Select <b>Add Device</b></li> </ul>
3	A windows dialog box appears to select the EDS or DCF file to import ( <i>see page 57</i> ): 
4	<ul style="list-style-type: none"> <li>● Select the EDS or DCF file name corresponding to the device that is to be added</li> <li>● Enter a unique name for the device (optional).</li> </ul> Only one EDS or DCF file can be selected at a time - multiple selections are not permitted. By default, the name of the EDS or DCF file is used as device name although it can be modified. The device name cannot be changed after this step.

Step	Action
5	<p>A screen appears showing the new device parameters. An example is shown here:</p>  <p>The screenshot shows a 'Device Profile' window for 'TestDevice'. It includes fields for Name, Family (Distributed I/Os), Image File, and Function. Below these is a table of I/O objects. The table has columns: Language Interface, PDO Mapped, Index, I/O Data Name, I/O Object, Rank, IODDT Used, and IODDT/DDDT Field Name. The table lists 18 objects with various parameters and checkmarks.</p> <p>In this tab, you can sort the objects in different ways: either clicking on the column head or ticking the check boxes then clicking on the column head again.</p>

Step	Action
6	<p>Some parameters on the device profile screen can be changed by the user:</p> <ul style="list-style-type: none"> <li>● <b>Family:</b> select the device family from the list box.</li> <li>● <b>Image File:</b> select the image file (BMP or JPG format) associated to the device which will be displayed in the graphical Hardware Catalog Manager Configurator Editor. There is no limit to the size of the image. In the boxes included in the label zone, you define the device name position compared to the bitmap position.</li> </ul>  <ul style="list-style-type: none"> <li>● <b>Expert Mode:</b> Expert Mode Configuration Parameters (<a href="#">see page 28</a>).</li> </ul> <p>To configure the function parameters, Basic Configuration Parameters (<a href="#">see page 23</a>) and Expert Mode Configuration Parameters (<a href="#">see page 28</a>).</p> <p>When all configuration parameters are set as required, click OK.</p>
7	<p>Return to the Hardware Catalog Manager main screen and confirm with <b>Build Catalog</b> in order to save any changes, and to load the new device into the Hardware Catalog Manager database. The <b>Output Window</b> displays the progress of the catalog build.</p> <p>When the build is complete, the device is indicated with a  icon.</p>

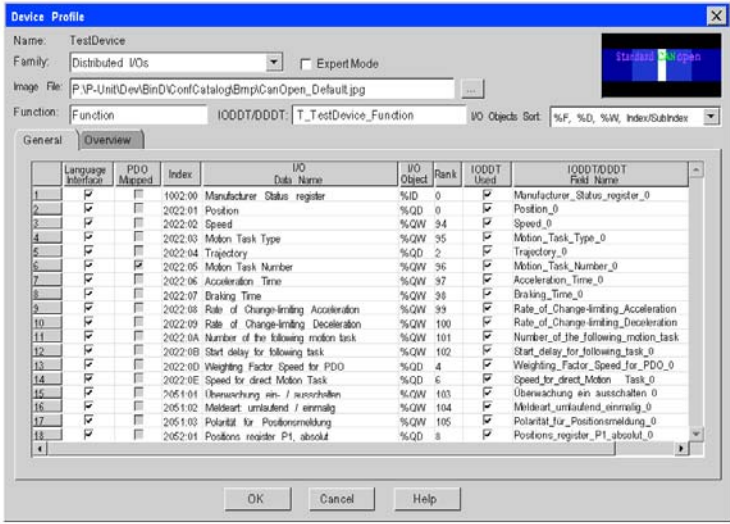
## How to Add a Function on a Device

### At a glance

This procedure describes how to add a function to an existing device in the Hardware Catalog Manager. Adding a function on a device allows the user to access a limited subset of capabilities on the device.

### Procedure

The table below shows the procedure to add a function and access the configuration screen:

Step	Action
1	Open the Hardware Catalog Manager.
2	<p>There are different ways to access the configuration screen:</p> <ul style="list-style-type: none"> <li>Right-click on a device. Select <b>Add Function</b></li> <li>OR</li> <li>Select a device in the Hardware Catalog Manager. Select <b>Edit → Add Function</b></li> </ul>
3	<p>The window below appears showing the basic configuration parameters:</p>  <p>To configure the function parameters, Basic Configuration Parameters (<a href="#">see page 23</a>), Expert Mode Configuration Parameters (<a href="#">see page 28</a>).</p>
4	<p>When configuration is complete, click OK.</p> <p>Return to the Hardware Catalog Manager main screen and click on the button <b>Build Catalog</b> in order to save any changes and to load the new function into the Hardware Catalog Manager database.</p>

## Basic Configuration Parameters

### At a Glance

The basic configuration parameters can be set in the **Device Profile** screen, using the **General** and **Overview** tabs.

### Function Parameters

The user can customize the function which has been added to the Hardware Catalog Manager, in order to select the specific capabilities required.

Before starting the configuration, be aware of the following rules:

- Each function name must be unique within a device. The name of the default function is **"Default"** although it can be modified by the user.
- An IODDT/DDDT is generated for the function if at least one of the variables in the grid is checked for **IODDT used**.
- The IODDT/DDDT name must be unique in the whole catalog. By default, it is derived by combining the device name and the function name preceded by "T\_". The user can modify the IODDT/DDDT name, but the new name must be unique.
- Predefined or user defined IODDT/DDDTs cannot be reused. However, they can be duplicated with new names.

The configuration parameters can be sorted using the **I/O Objects Sort** box. The list of available sort modes are:

- Object sort 1 - **%F, %D, %D, Index/SubIndex**
- Object sort 2 - **%F, %D, %D, PDOs order**
- Object sort 3 - **Index/SubIndex**

In this example table, the I/O objects are sorted by type (%F,%D,%W) and by index and subindex in each type:

Sort 1		
I/O Object	Type	PDO
0x3000.04	%IF	0
0x3000.03	%ID	4
0x2004.06	%IW	6
0x2004.07	%IW	7
0x3000.02	%QF	0
0x3000.01	%QD	4
0x2008.01	%QW	6
0x2008.05	%QW	7

In this example table, the I/O objects are sorted by type (%F,%D,%W) and by PDO order in each type for unmapped object:

Sort 2		
I/O Object	Type	PDO
0x2004.06	%IW	0
0x2004.07	%IW	1
0x3000.04	%IF	2
0x3000.03	%ID	6
0x2008.01	%QW	0
0x2008.05	%QW	1
0x3000.02	%QF	2
0x3000.01	%QD	6

In this example table, the I/O objects are sorted by type (%F,%D,%W) and by PDO order in each type for unmapped object:

Sort 3		
I/O Object	Type	PDO
0x2004.06	%IW	0
0x2004.07	%IW	1
0x3000.04	%IF	2
0x3000.03	%ID	6
0x2008.01	%QW	0
0x2008.05	%QW	1
0x3000.02	%QF	2
0x3000.01	%QD	6

The following table shows an example of I/O object allocation for the three sort types listed above:

Object Dictionary	Type	PDO Mapping
0x2004.06	%IW	Tx1.1
0x2004.07	%IW	Tx1.2
0x2008.01	%QW	Rx1.1
0x2008.05	%QW	Rx1.2
0x3000.02	%QF	Rx4.1
0x3000.01	%QD	Rx4.2

Object Dictionary	Type	PDO Mapping
0x3000.04	%IF	Tx4.1
0x3000.03	%ID	Tx4.2

PDO Mapping Legend:

- Tx 1.2: Object mapped on transmission PDO number 1, position 2
- Rx 1.2: Object mapped on reception PDO number 1, position 2

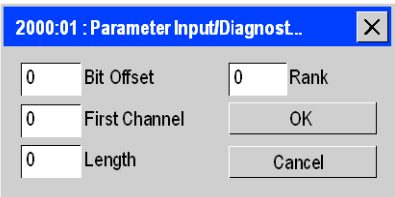
## General Tab

The **General** tab shows the list of all mappable variables found in the EDS file.

	Language Interface	PDO Mapped	Index	I/O Data Name	I/O Object	Rank	IODDT Used	IODDT/DDDT Field Name
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2000:01	Parameter Input/Diagnostic at Pin 2	%QW	0	<input checked="" type="checkbox"/>	Parameter_Input/Diagnostic_at_Pin 2_0
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2001:01	Parameter Input/Output at Pin 4	%QW	1	<input checked="" type="checkbox"/>	Parameter_Input/Output_at_Pin 4_0
3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3000:01	Common diagnosis	%IW	0	<input checked="" type="checkbox"/>	Common_diagnosis_0
4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3000:02	Sensor short circuit	%IW	1	<input checked="" type="checkbox"/>	Sensor_short_circuit_0
5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3000:03	Actuator shutdown Pin 4	%IW	2	<input checked="" type="checkbox"/>	Actuator_shutdown_Pin_4_0
6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3000:05	Actuator warning Pin 4	%IW	3	<input checked="" type="checkbox"/>	Actuator_warning_Pin 4_0
7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	3000:07	Desina inputs	%IW	4	<input checked="" type="checkbox"/>	Desina_inputs_0
8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6000:01	Digital Input 8 Bits Pin4	%IW	5	<input checked="" type="checkbox"/>	Digital_Input_8_Bits_Pin4_0
9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6000:02	Digital Input 8 Bits Pin2	%IW	6	<input checked="" type="checkbox"/>	Digital_Input_8_Bits_Pin2_0
10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	6100:01	Digital Input 16 Bits	%IW	7	<input checked="" type="checkbox"/>	Digital_Input_16_Bits_0
11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	6102:01	Polarity 16 Inputs	%QW	2	<input checked="" type="checkbox"/>	Polarity_16_Inputs_0
12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6200:01	Write Outputs 1 to 8	%QW	3	<input checked="" type="checkbox"/>	Write_Outputs_1_to_8_0
13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	6300:01	Digital Output 16 Bits	%QW	4	<input checked="" type="checkbox"/>	Digital_Output_16_Bits_0

**NOTE:** A left click on the header of each column can be used to sort the grid (first click ascending order/second click descending order).

The following information is listed for each variable:

Name	Description
Line number	<p>The user can have more information via a right-click on the "Line Number" column:</p> <ul style="list-style-type: none"> <li>● <b>Set As Parameter:</b> This converts the variable to a parameter.</li> <li>● <b>Set Bit Mapping:</b> This opens a dialog box to define the bit mapping. The bit mapping creates a boolean view of the CANopen object mapped on the %I or %Q topological variable.</li> </ul>  <ul style="list-style-type: none"> <li>● <b>Reset Bit Mapping:</b> This resets bit mapping of the variable.</li> </ul>
<b>Language Interface</b>	<p>If checked, the variable will have a language interface. Therefore, this variable can be used in the program and its value is displayed in the debug screen. If unchecked, the variable will be unavailable.</p> <p>To minimize the amount of memory needed for the function, it is useful to uncheck variables which are not required by the user. All variables are checked by default.</p> <p>The user can check or uncheck all variables in the column by a right-click.</p> <p>All "Language Interface" variables have a static language interface defined in the catalog.</p>
<b>PDO Mapped</b> (not modifiable)	<p>This indicates if the variable is currently mapped to a PDO. It can be changed in the expert mode (refer to expert mode configuration parameters).</p> <p>The default function uses the PDO mapping defined in the EDS file.</p> <p>The check box for <b>PDO Mapped</b> variables is not directly modifiable by the user but it is refreshed if the user removes a variable by unchecking it from the <b>Language Interface</b> list, the <b>PDO Mapped</b> check box is unchecked.</p>
<b>Index</b> (not modifiable)	<p>This indicates the CANopen index parameter.</p>
<b>I/O Data Name</b> (not modifiable)	<p>This is the parameter name found in the EDS file.</p>
<b>I/O Object</b>	<p>Access type (%I, %Q), data type (W, D, F) and rank of the I/O data in topological syntax. The rank is calculated from the list of language interface variables and is updated each time the list of language interface variables changes. The <b>I/O Object</b> field is empty if the <b>Language Interface</b> check box is not checked.</p>

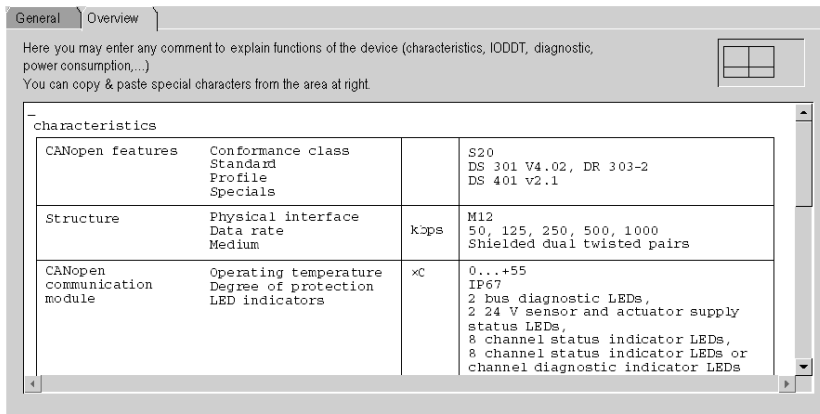
Name	Description
<b>IODDT Used</b>	If checked this data will be a field of the related IODDT/DDDT. This check box is modifiable only if the <b>Language Interface</b> check box is checked. Otherwise it is unchecked. The user can select or deselect all variables in the column by a right-click.
<b>IODDT/DDDT Field Name</b>	By default, this is the <b>I/O Data Name</b> with spaces and special characters replaced by underscores. Numbers may be added if the name is not unique in the IODDT/DDDT. This name is displayed and can be modified by the user only if <b>IODDT Used</b> is checked.

**NOTE:** When the **Device Function** screen opens, all PDO-mappable variables are checked as **Language Interface** and as **IODDT used**. Only variables that are mapped by default in the EDS file are checked as **PDO Mapped**.

### Overview Tab

In this screen, the user can enter a text description of the device and the related functions. It will be displayed in the module Editor.

The following illustration shows an example Overview tab:



The entry field can be filled by the user. It is possible to draw grid lines by copying special characters from the top right area in the windows.

## Expert Mode Configuration Parameters

### At a Glance

This paragraph speaks about the expert mode configuration parameters.

The **Expert Mode** check box has to be selected to get the expert mode configuration parameters.

The expert mode configuration parameters are used to:

- Remove variables
- Change variables mapping
- Modify variables parameters
- Substitute the standard bootup procedure for devices which do not comply with CANopen standards
- Display in read-only mode the content of the EDS used to build the device

The expert mode configuration parameters are composed of five tabs:

- **PDO Mapping** tab
- **Parameters** tab
- **Bootup Procedure** tab
- **Object Dictionary** tab
- **EDS** tab

**NOTE:** **Bootup Procedure** and **Object Dictionary** tabs are provided for CANopen expert users.

Configuration changes via these tabs can result in incorrect configuration of the device, which can disturb the CANopen bus. This tabs are not available for CANopen devices configuration in Control Expert with **BMX P34 2010** and **BMX P34 2030** M340 CPUs.

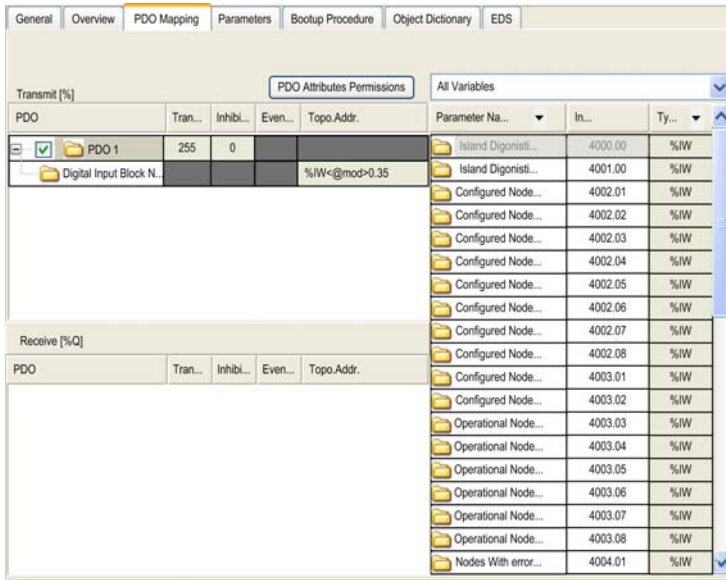
## PDO Mapping

The **PDO Mapping** tab opens a CANopen expert screen for PDO configuration.

The **PDO Mapping** configuration is composed of three parts:

- **Transmit PDOs.**
- **Receive PDOs.**
- **Variables.**

The following illustration describes the **PDO Mapping** tab:



Parameters for transmit and receive PDOs are initialized with the default PDO mapping found in the EDS file.

This default configuration can be changed by the user:

- PDOs can be enabled or disabled (by checking / unchecking them).
- **Variables** can be mapped or unmapped using drag-and-drop mechanism between PDOs and the **Variables** windows.
- **Transmission Type, Inhibit Time, Event time** of each PDO can be changed. If the properties are not available for the PDO, cells are painted in dark-grey and are not editable. A message box is displayed if the user enters a value of transmission not authorized. Implicit rules of transmission are:
  - Values between 241-251 are not available since they are reserved value.
  - Values 252/253 are not supported.
  - Values between 0-240 are not available if the device does not support synchronous communication.

**NOTE:** **Inhibit Time** and **Event time** are always disabled for PDO in reception.

The **Topo.Adrr (IODDT)** field cannot be changed by the user.

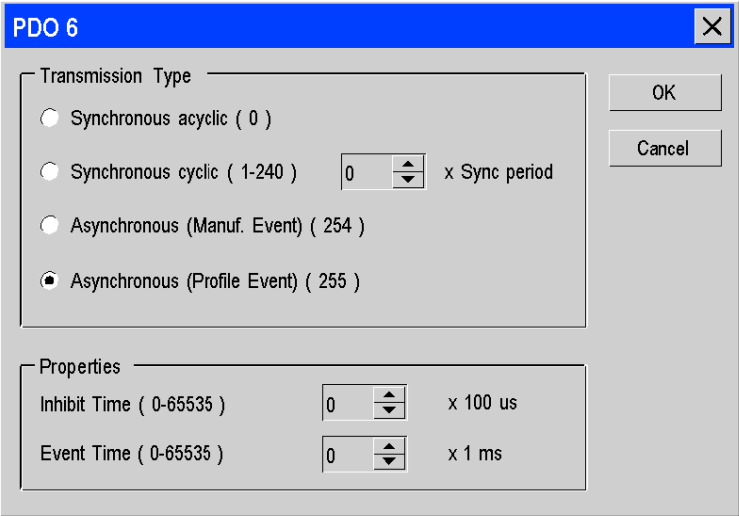
The Hardware Catalog Manager calculates the fixed part of the topological address for mapped variables:

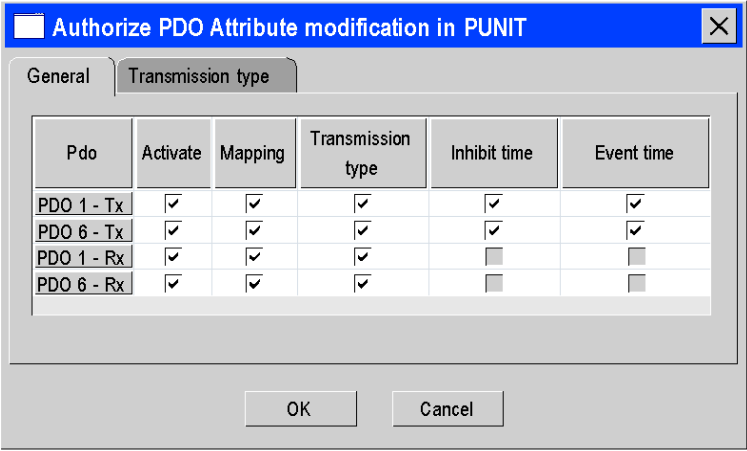
- **Access type.**
- **Data type.**
- **Channel number.**
- **Rank.**

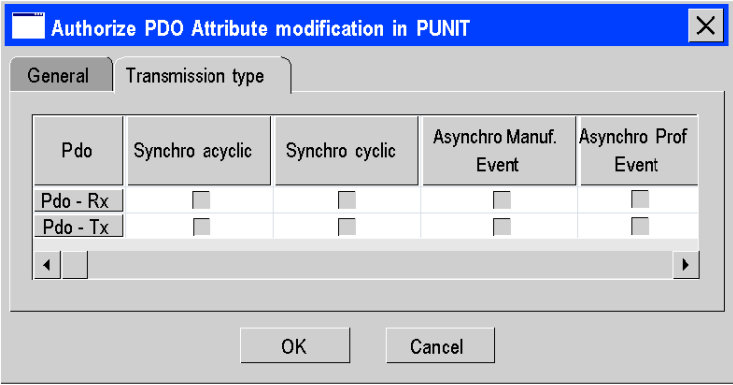
The module address is known only when the device is instantiated in an application. Therefore the catalog expert screen shows "<@mod>" instead of the real module address.

**NOTE:** Variables can be mapped to a PDO even if it is disabled. In this case, these variables still have a memory address (static memory mapping) but they will not exchange on the bus and a warning message is shown on analyzing the application.

The user can access to different menus on the **PDO Mapping** tab:

Name	Description
1. Drag-and-drop of variables	<p>Drag-and-drop of variables on the <b>PDO Mapping</b> tab is possible:</p> <ul style="list-style-type: none"> <li>● Inside the same PDO to another position.</li> <li>● From one PDO to another of the same type (<b>Transmit</b> and <b>Receive</b>).</li> <li>● From a PDO to the variables window (unmap this variable)</li> <li>● From the variables windows to a PDO if access type and PDO type are compatible and if the amount of the memory needed for the PDO will not exceed the limit of 8 bytes.</li> <li>● Inside the same PDO to another position.</li> </ul>
2. <b>Transmission Type, Inhibit Time, Event time</b>	<p>A double click on these columns opens the following screen:</p>  <p><b>Inhibit Time</b> and <b>Event time</b> are accessible only if an asynchronous transmission type has been chosen.</p> <p>The unauthorized values of <b>Transmission Type</b> must be disabled according to the PDO attributes.</p>

Name	Description
<b>3. PDO Attributes Permissions</b>	<p>The <b>General</b> tab is used to define general access right attributes of PDO. These attributes are used in the Control Expert and in the Hardware Catalog Manager to authorize or not the modification of PDO.</p> <p>The following illustration shows the general <b>PDO attributes</b> screen:</p>  <p>The grid is initialized with the list of all PDO found in the device and the information is displayed for each PDO:</p> <ul style="list-style-type: none"> <li>● <b>PDO</b>: This is the list of available PDO (TX and RX) and it is not modifiable.</li> <li>● <b>Activate</b>: If checked then the PDO can be activated. Thus associated PDO can be enabled or disabled in Control Expert.</li> <li>● <b>Mapping</b>: If checked then mapping is editable in Control Expert.</li> <li>● <b>Transmission Type</b>: If checked, then <b>Transmission Type</b> value will be editable in <b>Transmission Type</b> tab. Other constraints can be defined in the <b>Transmission Type</b> tab.</li> <li>● <b>Inhibit Time</b>: If checked, then <b>Inhibit Time</b> value will be editable in Control Expert.</li> <li>● <b>Event time</b>: If checked, then <b>Event time</b> value will be editable in Control Expert.</li> </ul>

Name	Description
<p>4. <b>Transmission Type</b> range of PDO-Tx and PDO-Rx</p>	<p><b>Transmission Type</b> range of Tx PDO and Rx PDO can be defined by the user. the following illustration shows the <b>Transmission Type</b> screen for PDO attributes:</p>  <p>The grid is initialized with attributes of Tx PDO and Rx PDO and the information is displayed for each PDO:</p> <ul style="list-style-type: none"> <li>● <b>PDO</b>: This is the RX PDO and Tx PDO. It is not modifiable.</li> <li>● <b>Synchronous acyclic</b>: If checked then synchronous acyclic Transmission Type range is available in Control Expert.</li> <li>● <b>Synchronous cyclic</b>: if checked then synchronous cyclic Transmission Type range is available in Control Expert.</li> <li>● <b>Asynchronous Manuf Event</b>: if checked then asynchronous manuf. Event Transmission Type range is available in Control Expert.</li> <li>● <b>Asynchronous Profile Event</b>: if checked then asynchronous profile event Transmission Type range is available in Control Expert.</li> </ul>

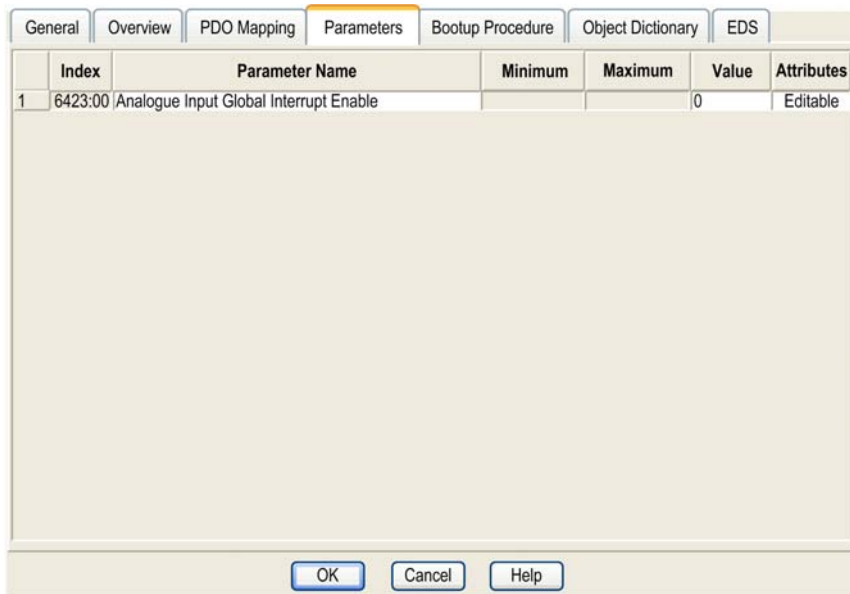
### Parameters Tab

The **Parameters** tab is used to modify the properties of parameters.

The properties of parameters are:

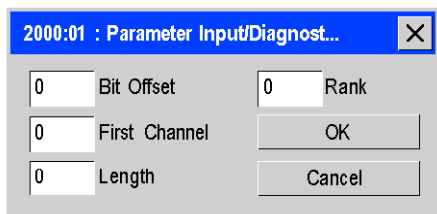
- **Index**: indicates the index and the subindex of the parameter (not editable).
- **Parameter Name**: indicates the name of the parameter (not editable).
- **Minimum/Maximum**: indicates the range of authorized value (not editable).
- **Value**: indicates the parameter value (editable) and is initialized with the default value find in the EDS.
- **Attributes**:
  - **Editable**: The parameter is editable in Control Expert (default value).
  - **Read Only**: The parameter is visible in Control Expert but not editable.
  - **Hide**: The parameter is not visible in Control Expert but the value is sent to the device.

The following illustration shows the **Parameters** tab:



The user can have more information via a right-click on the "Line Number":

- **Set As Variable:** This converts the parameter to a variable.
- **Set BitMapping:** This opens a dialog box to define the bit mapping. The bit mapping creates a boolean view of the CANopen object mapped on the %I or %Q topological variable.



- **Reset BitMapping:** This resets bit mapping of the variable.
- **Move Up** and **Move Down:** This can be used to sort the parameters in the Control Expert configuration screen.

## Bootup Procedure Tab

The goal of **Bootup Procedure** tab is to bypass the standard bootup procedure for devices which do not comply with CANopen standards

### ⚠ WARNING

#### UNEXPECTED EQUIPMENT OPERATION

Manually verify all deactivated standard checks on the device before operating the system.

Changing the default parameters of the **Bootup Procedure** tab will bypass standard system checks.

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

The following illustration describes the **Bootup Procedure** tab:

The screenshot shows the 'Bootup Procedure' tab in the Hardware Catalog Manager. The interface is divided into several sections, each with a set of options and a corresponding description in a text box on the right.

- Restore:**
  - No Restore
  - Restore communication parameters
  - Restore application parameters
  - Restore all parameters

Restore option is used to define the allowed restore procedure for a CANopen device during startup.  
Communication parameters: 0x1000 -> 0x1FFF  
Application parameters: 0x6000 -> 0x9FFF
- Reset:**
  - Reset communication parameters
  - Reset node

Reset option is used to configure the type of reset sent to the device.  
Communication parameters: 0x1000 -> 0x1FFF  
Application parameters: 0x6000 -> 0x9FFF
- Check node:**
  - Device type
  - Device identity

Check node option is used to bypass the device type (0x1000) or the device identity (0x1018) test.
- Download Configuration:**
  - Force communication parameters
  - Force application parameters

Download configuration option is used to force the parameters to be sent even if they are equal to the default value.
- Start:**
  - Start node

Start option is used to configure whether the device shall be started automatically by the master or not.

At the bottom of the window, there are three buttons: OK, Cancel, and Help.

- The type of restore:
  - **No Restore**: enabled by default.
  - **Restore communication parameters**: enabled option according to the object 0x1011sub02. If the option is checked, all parameters between 0x1000 to 0x1FFF are restored.
  - **Restore application parameters**: enabled option according to the object 0x1011sub03. If the option is checked and if the device correctly implements the service, all application parameters are restored.
  - **Restore all parameters**: enabled option according to the object 0x1011sub01. If the option is checked, all parameters are restored (default value).
- The type of reset:
  - **Reset communication parameters**: option always enabled. If the option is checked, all communication parameters are reset.
  - **Reset node** (default value): option always enabled. If the option is checked, all parameters are reset.
- The check boxes **Device type** and **Device identity** (checked by default):
  - If the device type identification value for the slave in object dictionary 0x1F84 is not 0x0000 ("don't care"), compare it to the actual value.
  - If the configured Vendor ID in object dictionary 0x1F85 is not 0x0000 ("don't care"), read slave index 0x1018, subindex 1 and compare it to the actual value.
  - The same comparison is done with ProductCode, RevisionNumber and SerialNumber with the according objects 0x1F86-0x1F88.

**NOTE:** Unchecked option **Device type** forces the object dictionary 0x1F84 to 0x0000.

**NOTE:** Unchecked option **Device identity** forces the object dictionary 0x1F86-0x1F88 (sub device nodeID) to 0x0000.

- Force the download of communication or application parameters (unchecked by default). If option is checked, it forces all the corresponding objects to be downloaded. If the option is unchecked, you must follow these standard rules:
  - Parameters are downloaded if they are different from the default value.
  - Parameters are downloaded if they are forced in the object dictionary.
  - Parameters are not downloaded in the other cases.
- **The Start node:**

If option is checked (default value), the CANopen master starts automatically the device after the bootup procedure.

If option is unchecked, the device stays in pre-operational state after bootup procedure. In this case, the device must be started by the application program.

## Object Dictionary Tab

**⚠ WARNING**

**UNEXPECTED EQUIPMENT OPERATION**

Manually verify all Object Dictionary values.

Changing the default values of the Object Dictionary table will generate non-standard of the equipment.

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

The **Object Dictionary** screen is a CANopen Expert editor, and allows to:

- Force parameters to be transmitted even if they are unchanged.
- Suppress parameters that do not need to be sent to the device.
- Set objects to a specific value just before (prologue), or just after (epilogue) the standard bootup procedure.
- Modify the current value of an object (except read-only objects).

The following illustration describes the **Object Dictionary** tab:

Index:Subindex	Name	Value	Default Value	Access	Type	Min	Max

You can select two filters to reduce the number of displayed objects on the grid:

<b>Area filter</b>	
<b>All</b>	show all area
<b>Prolog / Epilog</b>	show only prologue and epilogue projects
[XXXX...XXXX]	show only objects between XXXX to XXXX
<b>Status filter</b>	
<b>All</b>	show all objects
<b>Configured</b>	show only transmitted objects to the device during bootup
<b>Not configured</b>	show only not transmitted objects to the device
<b>Modified</b>	show only objects from which values are different from default values

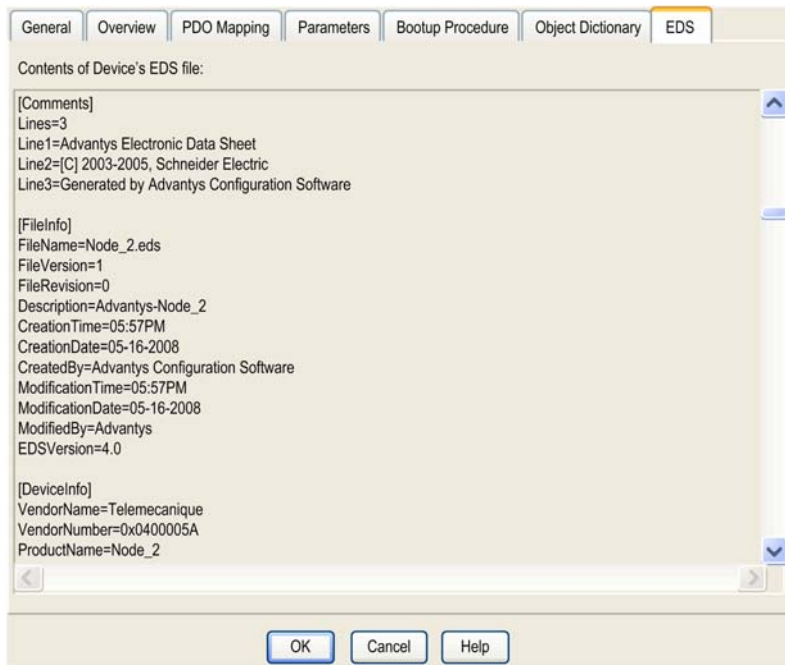
You can right-click on an object to execute function:

Right-click on an object in the prologue and epilogue sections	
<b>Cut</b>	Cut the row and copy the object in the clipboard
<b>Copy</b>	Copy the object in the clipboard
<b>Paste</b>	Paste the object in the selected row
<b>Delete</b>	Delete the selected object
<b>Move up</b>	Used to manage the order of the list
<b>Move down</b>	Used to manage the order of the list
<b>Configured</b>	If checked, the object is transmitted to the device
<b>Expand all</b>	Expand all nodes of the tree
<b>Collapse all</b>	Collapse all nodes of the tree
Right-click on an object in the standard sections	
<b>Copy</b>	Copy the object in the clipboard
<b>Configured</b>	If checked, the object is transmitted to the device
<b>Expand all</b>	Expand all nodes of the tree
<b>Collapse all</b>	Collapse all nodes of the tree

## EDS Tab

The **EDS** tab displays in read-only mode the content of the EDS used to build the device.

The following illustration shows the **EDS** tab:



## MFB Function for Expert Mode

### MFB Function Creation

Before configuring your MFB function, create it by following this guideline:

Step	Action
1	In the Hardware Catalog Manager, expand the <b>Motion &amp; Drive</b> CANopen family and select the drive you wish, ATV71_V1_1 for example.
2	Expand the ATV71_V1_1 drive and select the MFB function. Right-click the selected function and then click <b>Copy</b> .
3	Right-click the selected drive (ATV71_V1_1) and then click <b>Paste Function</b> .

### IODDT/DDDT Creation

After the paste function, the **Device Profile ATV71\_V1\_1** dialog box appears:

Step	Action
1	Tick <b>Expert Mode</b> check box.
2	Select <b>%F, %D, %W, PDOs order</b> in the <b>I/O Objects Sort</b> box.
3	Type the function name you wish in the <b>Function</b> box, MFB_My_Function for instance. <b>NOTE:</b> The function name must begin with MFB_
4	The IODDT/DDDT with the name T_ATV71_V1_1_MFB_My_Function is automatically created. It is shown on the <b>IODDT/DDDT</b> box.

## Objects Selection

Now, you have to select your objects:

Step	Action
1	Select the <b>General</b> tab in the <b>Device Profile ATV71_V1_1</b> dialog box.
2	Select the objects you wish by ticking them in the <b>Language Interface</b> column.

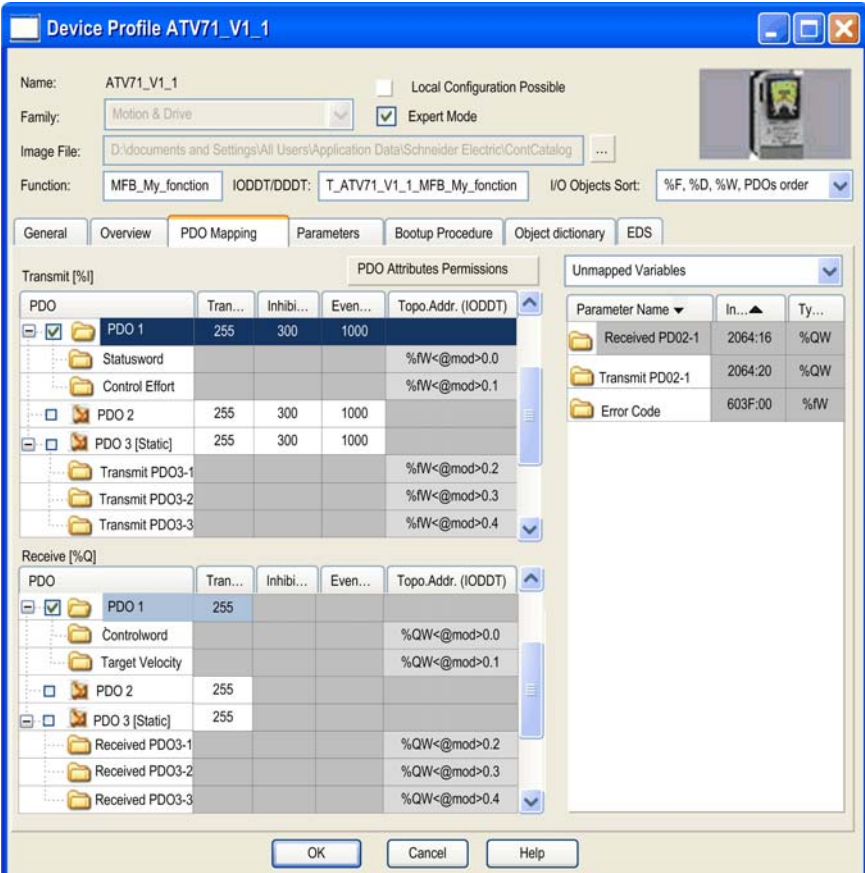
  

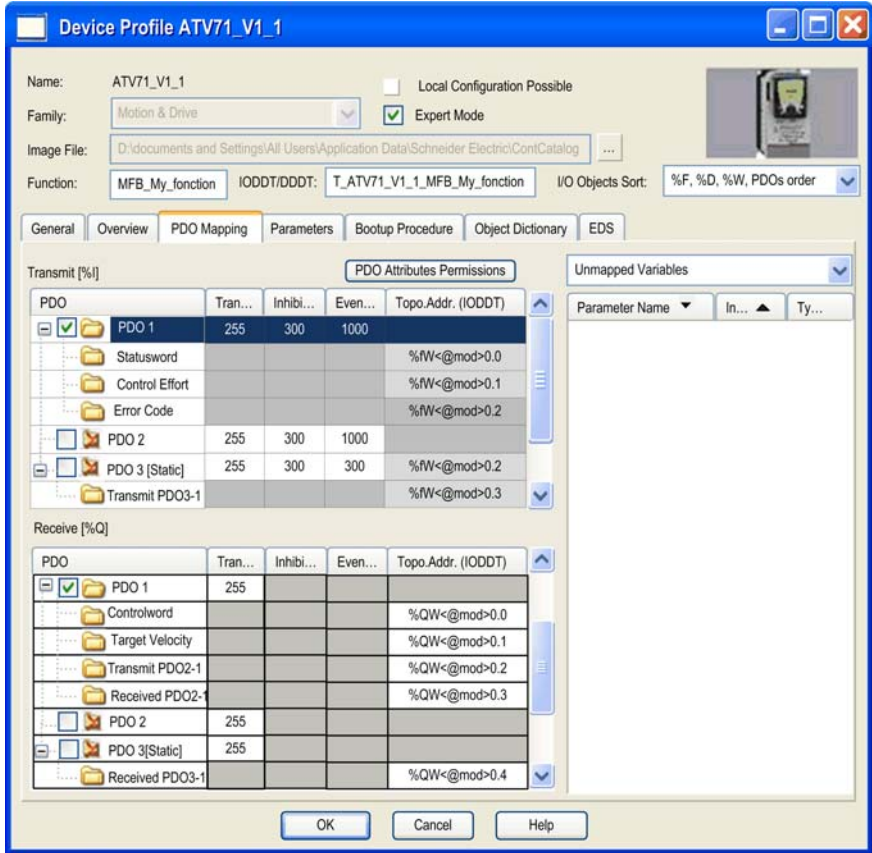
The screenshot shows the 'Device Profile ATV71\_V1\_1' dialog box with the 'General' tab active. The 'Language Interface' column in the table below is checked for rows 1259, 1260, 1261, and 1262.

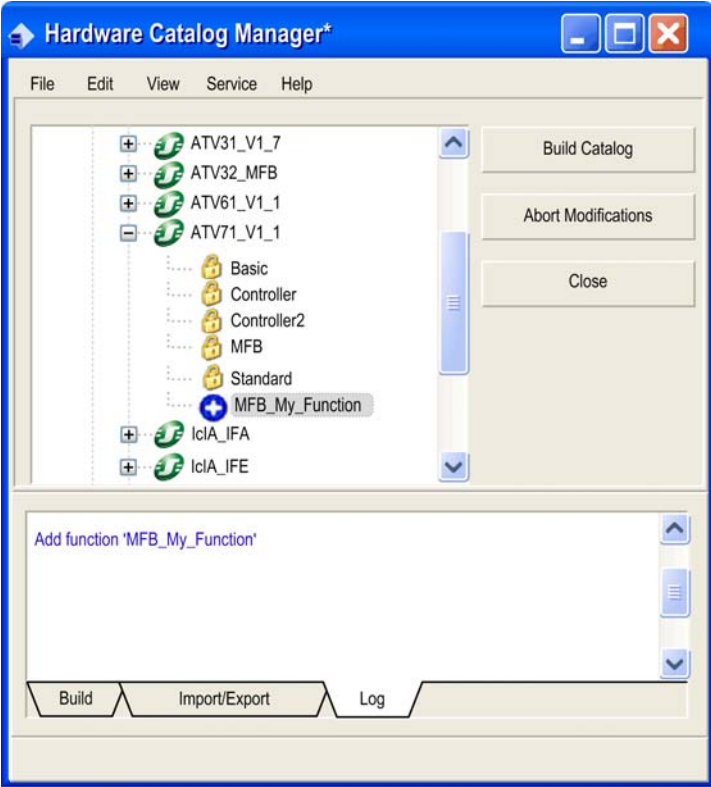
	Language Interface	PDO Mapped	Index	I/O Data name	I/O object	Rank	IODDT Used	IODDT/DDDT Field Name
1252	<input type="checkbox"/>	<input type="checkbox"/>	2070:04	Freq.const power			<input type="checkbox"/>	Freq.Const power
1253	<input type="checkbox"/>	<input type="checkbox"/>	2071:02	Load sharing			<input type="checkbox"/>	Load sharing
1254	<input type="checkbox"/>	<input type="checkbox"/>	2071:03	Load Correction			<input type="checkbox"/>	Load correction
1255	<input type="checkbox"/>	<input type="checkbox"/>	2071:04	Correction min spd			<input type="checkbox"/>	Correction min spd
1256	<input type="checkbox"/>	<input type="checkbox"/>	2071:05	Correction max spd			<input type="checkbox"/>	Correction Max.spd
1257	<input type="checkbox"/>	<input type="checkbox"/>	2071:06	Torque offset			<input type="checkbox"/>	Torque offset
1258	<input type="checkbox"/>	<input type="checkbox"/>	2071:07	Sharing filter			<input type="checkbox"/>	Sharing folder
1259	<input checked="" type="checkbox"/>	<input type="checkbox"/>	603F:00	Error code	%W	6	<input checked="" type="checkbox"/>	Error code
1260	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6040:00	Control word	%QW	0	<input checked="" type="checkbox"/>	Control word
1261	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6040:01	Status word	%W	0	<input type="checkbox"/>	
1262	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6042:00	Target velocity	%QW	1	<input checked="" type="checkbox"/>	Target velocity
1263	<input type="checkbox"/>	<input type="checkbox"/>	6043:00	Velocity demand			<input type="checkbox"/>	Velocity demand
1264	<input type="checkbox"/>	<input checked="" type="checkbox"/>	6044:00	Control effort	%W	1	<input type="checkbox"/>	
1265	<input type="checkbox"/>	<input type="checkbox"/>	6046:01	Velocity min amount			<input type="checkbox"/>	
1266	<input type="checkbox"/>	<input type="checkbox"/>	6046:02	Velocity max amount			<input type="checkbox"/>	
1267	<input type="checkbox"/>	<input type="checkbox"/>	6048:01	Delta speed			<input type="checkbox"/>	
1268	<input type="checkbox"/>	<input type="checkbox"/>	6048:02	Delta time			<input type="checkbox"/>	
1269	<input type="checkbox"/>	<input type="checkbox"/>	6049:01	Delta speed			<input type="checkbox"/>	
1270	<input type="checkbox"/>	<input type="checkbox"/>	6049:02	Delta time			<input type="checkbox"/>	

**PDO Mapping**

To map your PDOs, click-on the **PDO Mapping** tab and do the following guideline:

Step	Action
1	<p>To see your new selected objects in the <b>Parameter Name</b> Window, select <b>Unmapped Variables</b> in the list which is above the <b>Parameter Name</b> Window.</p> 
2	<p>Drag-and-drop your %IW objects from the <b>Parameter Name</b> window to <b>Transmit</b> window.</p> <p><b>NOTE:</b> Drop your selected objects in the activated PDOs, i.e after the already mapped objects in the PDOs. 4 words are authorized by PDO.</p> <p><b>NOTE:</b> If a few PDOs are already activated, you must choose the last activated PDO to insert your objects.</p> <p><b>NOTE:</b> The topological addresses of the MFB objects must be kept.</p>

Step	Action																																																																																					
3	<p>Do the step 2 for your %QW objects, from <b>Parameter Name</b> Window to <b>Receive</b> Window.</p>  <p>The screenshot shows the 'Device Profile ATV71_V1_1' configuration window. The 'PDO Mapping' tab is selected. The 'Transmit [%I]' table is as follows:</p> <table border="1" data-bbox="381 521 879 737"> <thead> <tr> <th>PDO</th> <th>Tran...</th> <th>Inhibi...</th> <th>Even...</th> <th>Topo.Addr. (IODDT)</th> </tr> </thead> <tbody> <tr> <td>PDO 1</td> <td>255</td> <td>300</td> <td>1000</td> <td></td> </tr> <tr> <td>Statusword</td> <td></td> <td></td> <td></td> <td>%fW&lt;@mod&gt;0.0</td> </tr> <tr> <td>Control Effort</td> <td></td> <td></td> <td></td> <td>%fW&lt;@mod&gt;0.1</td> </tr> <tr> <td>Error Code</td> <td></td> <td></td> <td></td> <td>%fW&lt;@mod&gt;0.2</td> </tr> <tr> <td>PDO 2</td> <td>255</td> <td>300</td> <td>1000</td> <td></td> </tr> <tr> <td>PDO 3 [Static]</td> <td>255</td> <td>300</td> <td>300</td> <td>%fW&lt;@mod&gt;0.2</td> </tr> <tr> <td>Transmit PDO3-1</td> <td></td> <td></td> <td></td> <td>%fW&lt;@mod&gt;0.3</td> </tr> </tbody> </table> <p>The 'Receive [%Q]' table is as follows:</p> <table border="1" data-bbox="381 776 879 1019"> <thead> <tr> <th>PDO</th> <th>Tran...</th> <th>Inhibi...</th> <th>Even...</th> <th>Topo.Addr. (IODDT)</th> </tr> </thead> <tbody> <tr> <td>PDO 1</td> <td>255</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Controlword</td> <td></td> <td></td> <td></td> <td>%QW&lt;@mod&gt;0.0</td> </tr> <tr> <td>Target Velocity</td> <td></td> <td></td> <td></td> <td>%QW&lt;@mod&gt;0.1</td> </tr> <tr> <td>Transmit PDO2-1</td> <td></td> <td></td> <td></td> <td>%QW&lt;@mod&gt;0.2</td> </tr> <tr> <td>Received PDO2-1</td> <td></td> <td></td> <td></td> <td>%QW&lt;@mod&gt;0.3</td> </tr> <tr> <td>PDO 2</td> <td>255</td> <td></td> <td></td> <td></td> </tr> <tr> <td>PDO 3 [Static]</td> <td>255</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Received PDO3-1</td> <td></td> <td></td> <td></td> <td>%QW&lt;@mod&gt;0.4</td> </tr> </tbody> </table>	PDO	Tran...	Inhibi...	Even...	Topo.Addr. (IODDT)	PDO 1	255	300	1000		Statusword				%fW<@mod>0.0	Control Effort				%fW<@mod>0.1	Error Code				%fW<@mod>0.2	PDO 2	255	300	1000		PDO 3 [Static]	255	300	300	%fW<@mod>0.2	Transmit PDO3-1				%fW<@mod>0.3	PDO	Tran...	Inhibi...	Even...	Topo.Addr. (IODDT)	PDO 1	255				Controlword				%QW<@mod>0.0	Target Velocity				%QW<@mod>0.1	Transmit PDO2-1				%QW<@mod>0.2	Received PDO2-1				%QW<@mod>0.3	PDO 2	255				PDO 3 [Static]	255				Received PDO3-1				%QW<@mod>0.4
PDO	Tran...	Inhibi...	Even...	Topo.Addr. (IODDT)																																																																																		
PDO 1	255	300	1000																																																																																			
Statusword				%fW<@mod>0.0																																																																																		
Control Effort				%fW<@mod>0.1																																																																																		
Error Code				%fW<@mod>0.2																																																																																		
PDO 2	255	300	1000																																																																																			
PDO 3 [Static]	255	300	300	%fW<@mod>0.2																																																																																		
Transmit PDO3-1				%fW<@mod>0.3																																																																																		
PDO	Tran...	Inhibi...	Even...	Topo.Addr. (IODDT)																																																																																		
PDO 1	255																																																																																					
Controlword				%QW<@mod>0.0																																																																																		
Target Velocity				%QW<@mod>0.1																																																																																		
Transmit PDO2-1				%QW<@mod>0.2																																																																																		
Received PDO2-1				%QW<@mod>0.3																																																																																		
PDO 2	255																																																																																					
PDO 3 [Static]	255																																																																																					
Received PDO3-1				%QW<@mod>0.4																																																																																		
4	Validate your configuration by clicking <b>OK</b>																																																																																					

Step	Action
5	<p>Your function and its mapping are created.</p> 

**⚠ WARNING**

**UNINTENDED EQUIPMENT OPERATION**

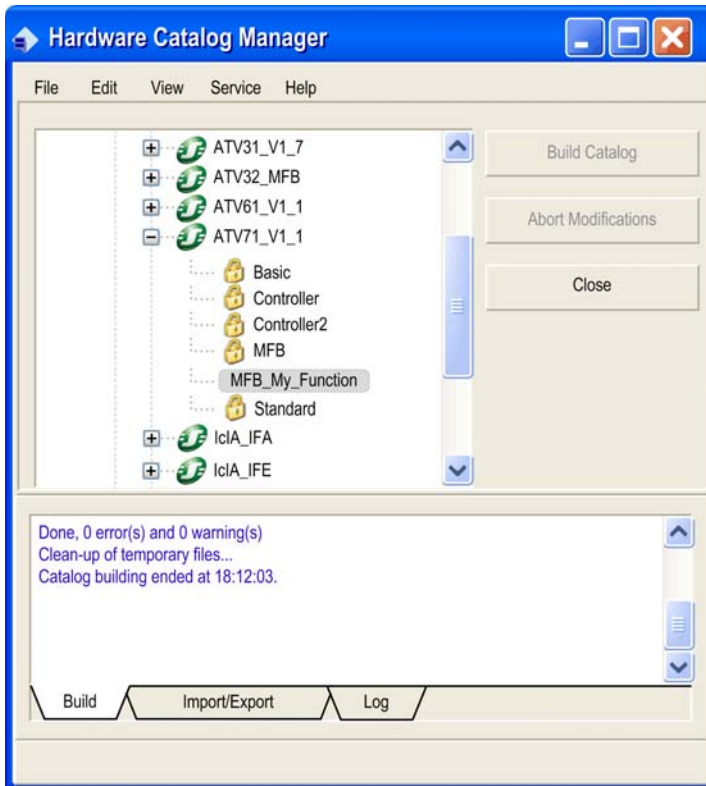
Do not remove or change the original MFB Objects addresses.

You must keep the same topological addresses, e.g Status Word %IW<@mod>0.5.

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

## Build the Catalog

To integrate this new function in the Hardware Catalog of Control Expert, click **Build Catalog**. Therefore, the new function appears in the Hardware Catalog Manager.



## How to Copy or Delete a Function

### At a Glance

Functions can be copied from one device to another, or deleted from the Hardware Catalog Manager database.

### Copy

The **Copy** function is only available if a "Function" item on a device is selected in the Hardware Catalog Manager main screen.

Follow one of the instructions below to copy a function:

- Select **Edit → Copy**
- Right-click on the function name and select **Copy**.

The function parameters are stored in the clipboard and can then be pasted to a compatible device.

### Paste

The **Paste** function is only available if a "Function" is in the Clipboard and if a device is selected in the Hardware Catalog Manager main screen.

Follow one of the instructions below to paste a function:

- Select **Edit → Paste**
- Right-click on the device and select **Paste**.

There is no need to specify the EDS file once more. The complete EDS file is already stored with the device.

The **Device Function** screen appears after pasting a function. The new function is initialized with the data of the source function. The same name is proposed for the new function and must be modified by the user before saving.

All other information can be modified and saved for the new function.

### Delete

Functions can be deleted by the user:

Follow the instructions below to delete a function:

- Select a function from one device.
- Right-click on the function name and select **Delete**.  
OR  
Select **Edit → Delete**
- A message will appear to request confirmation, click OK to continue.
- To save changes, click on the button **Build Catalog**.

## How to Import/Export or Delete One or Several User Devices

### At a Glance

This section describes how to import, export or delete one or several user devices in the Hardware Catalog Manager.

### Export User Devices

**Export User Devices** is enabled only:

- If nothing has been modified or if all modifications are already built in the catalog database.
- If at least one user device exists in the catalog.

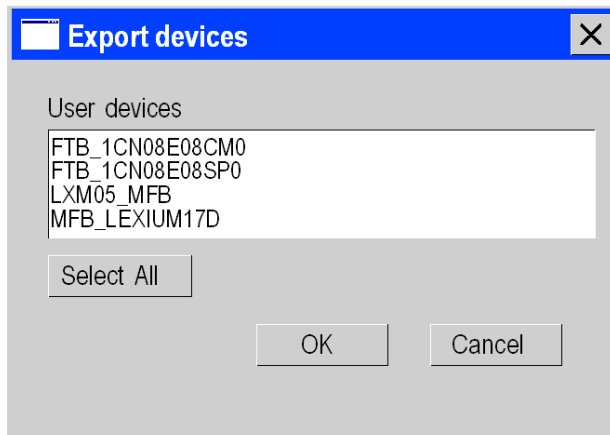
This is independent of the current item selection in the tree control.

The user must follow the instructions below to export user devices:

Select **File** → **Export User Devices**

A screen displays the list of devices with user function (user device and preprogrammed device with user function). Listbox supports multi-selection. Devices to export are selected in usual Windows-manner by Ctrl-Click or Shift-click. Alternatively all devices can be selected by clicking the **Select All** button.

The following illustration shows the **Export devices** screen:



On validation of this dialog and if one or more devices are selected, a standard windows **Save** dialog is shown allowing the user to choose file name and location of the export file.

The extension of the file name is .cpx.

The export file is a zip file composed of all catalog source files of the exported devices.

The user cannot export only one function; he must export all user functions from a device.

On validation of this dialog the \*.cpx catalog source file is extracted from the database and saved.

## Import User Devices

**Import User Devices** is enabled only:

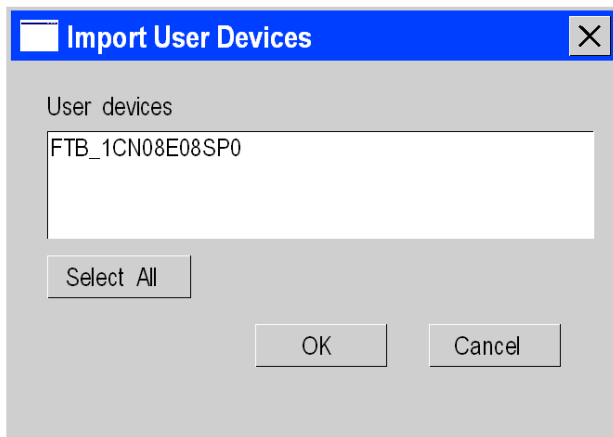
- If nothing has been modified or if all modifications are already built in the catalog database.
- It is independent of the current item selection in the tree control.

Follow the instructions below to import user devices:

Select **File** → **Import User Devices**

A standard Windows « Open » screen appears. Only \*.cpx files are accepted.

On opening the cpx file the following modal dialog is displayed allowing the user to choose a subset of devices in cpx files:



This dialog works in the same way as for **Export User Devices** (multi-selection listbox).

On validation all necessary catalog source files are extracted from the \*.cpx file and a new catalog database is built.

This new database contains:

- All Schneider core devices present in the old database.
- All User devices present in the old database.
- All User devices selected to import in this dialog.

All user functions must be imported from a device, it is not possible to import only one function.

In case of conflicts, (a device or a function already present in the database with the same name but without the same ID) the device will not be imported and a message box will appear.

Other import messages are displayed in the output window.

Please refer to chapter *Troubleshooting* ([see page 52](#)) for more details on possible conflicts and how they are handled.

At the end of the Import function, the main window is refreshed from the new database.

## Delete User Devices

User devices can be deleted by the user.

Follow the instructions below to delete user devices:

- Select a user device from one device family.
- Right-click the selected user device and then click **Delete**.
- A message box will appear for confirmation, then click **OK**.
- To save any changes, click **Build Catalog**.

## How to Close the Hardware Catalog Manager

### At a Glance

This is the procedure to close the Hardware Catalog Manager.

### Procedure

The table shows the procedure to close the Hardware Catalog Manager:

Step	Action
1	Verify that your changes have been saved: <ul style="list-style-type: none"><li>● If there is changes, click <b>Build Catalog</b>. A window appears showing the progress of the catalog build. When the build is complete, go to the next step</li><li>● If there are not any changes, go to the next step.</li></ul>
2	Select <b>File → Exit</b> OR Select <b>Close</b> on the main windows.

## Example of How to Create a Dedicated and Optimized STB Island

### At a Glance

The following procedure describes how to create a dedicated and optimized STB Island

Step	Action
1	Launch Advantys software.
2	Create your optimized STB Island with the configuration that you want.
3	Export the EDS file. The following illustration describes the 'Export EDS file' step:
4	Open the Hardware Catalog Manager software.
5	Add a device ( <i>see page 18</i> ) to the Hardware Catalog Manager

# Section 1.3

## Troubleshooting

---

### Subject of This Section

This section presents the troubleshooting of the Hardware Catalog Manager.

### What Is in This Section?

This section contains the following topics:

Topic	Page
Troubleshooting	53
EDS/DCF Import Anomaly Code	57

## Troubleshooting

### At a Glance

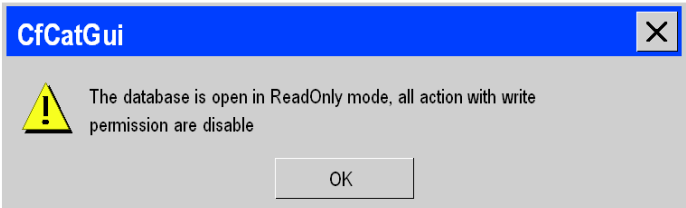
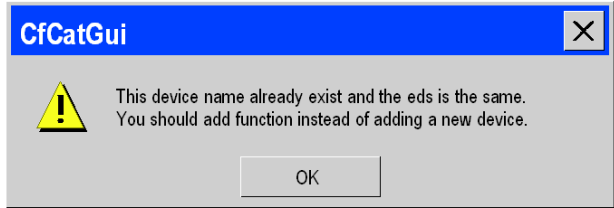
Use this section to find solutions to any difficulties that may be encountered when using the Hardware Catalog Manager.

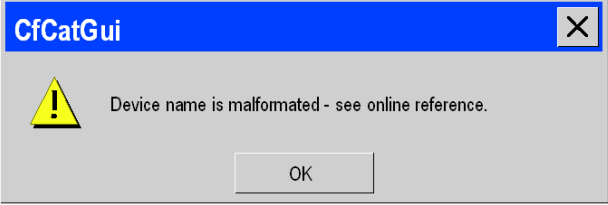

Messages that need to be acknowledged by the user are shown in message boxes. All other messages are shown in the **Output window** using a text color corresponding to the type of message:


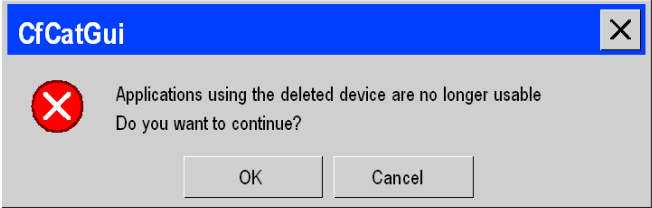
- Red for detected errors during analysis
- Orange for important information that must be checked
- Blue for information messages

## Troubleshooting

The following table describes the troubleshooting for the Hardware Catalog Manager:

Anomaly	Solution
<p><b>Only available in read-only mode</b></p>	<p>If Control Expert is open, then the Hardware Catalog Manager will be opened in read mode. Therefore, all the functions which modify the Hardware Catalog Manager will be disabled.</p> <p>A message prompts to inform the user about the behavior:</p>  <p>Follow these instructions to solve this issue:</p> <ul style="list-style-type: none"> <li>● CloseControl Expert before opening the Hardware Catalog Manager. Control Expert and the Hardware Catalog Manager can not be launched at the same time.</li> </ul>
<p><b>The device name already exists</b></p>	<p>The EDS file is required when the user wants to add or to import a device on the Hardware Catalog Manager database.</p> <ul style="list-style-type: none"> <li>● If a device name already exists in the Catalog. Then the user is asked if he wants to add a new function to the device. <ul style="list-style-type: none"> <li>○ If yes, then the existing device becomes selected in the main screen and the <b>Add Function</b> service is called automatically.</li> <li>○ If no, then the <b>Add Device</b> service is refused.</li> </ul> </li> </ul> <p>Follow these instructions to solve the anomalies:</p> <ul style="list-style-type: none"> <li>○ Rename the device.</li> <li>○ If the EDS file is refused then the reason is shown in a message and the 'Open' screen remains open. The user can select another EDS file or cancel the action.</li> </ul> <p>A message prompts to inform the user about the behavior:</p> 

Anomaly	Solution
<p><b>The device name is malformed</b></p>	<p>The name doesn't have the right format:</p> <ul style="list-style-type: none"> <li>● The number of characters is greater than 24.</li> <li>● The name does not comply with the rules of naming the Control Expert variables. Characters must be within the ranges A-Z, a-z or 0-9 and underline. Please note that the Hardware Catalog Manager creates a device name by default, by taking the EDS name and replacing prohibited characters by authorized ones. If you modify the device name with prohibited characters, the message appears. Follow these instructions to solve the anomalies: <ul style="list-style-type: none"> <li>○ Reduce the number of characters and use the authorized characters as above.</li> </ul> </li> </ul> <p>A message prompts to inform the user about the behavior:</p> 
<p><b>The file is not supported. Do you want a default image?</b></p>	<p>The file is not found or the format is not supported. Only the .bmp and .jpg format are supported.</p>
<p><b>XML file format is not correct</b></p>	<p>The following messages can appear on the output windows:</p> <ul style="list-style-type: none"> <li>● The syntax of the xml files is not correct.</li> <li>● Incoherent xml.</li> <li>● Wrong generation of the database.</li> </ul> <p>Follow these instructions to solve the anomaly:</p> <ul style="list-style-type: none"> <li>○ How to add a device to the Hardware Catalog Manager (<a href="#">see page 18</a>).</li> <li>○ How to add a function on a device (<a href="#">see page 22</a>).</li> </ul>
<p><b>A function with this name already exists</b> or <b>An IODDT/DDDT with this name already exists</b></p>	<p>The following messages can appear on the output windows. Follow these instructions to solve the anomalies:</p> <ul style="list-style-type: none"> <li>● <b>The function name already exists.</b> The function name must be renamed to solve it.</li> <li>● <b>The IODDT/DDDT name already exists.</b> The IODDT/DDDT must be renamed to solve it.</li> </ul> <p>The following message appears:</p> 

Anomaly	Solution
<b>Wrong File Format</b>	<p>The following message can appears on the <b>Export User Devices</b> action:</p> <ul style="list-style-type: none"> <li>● <b>Wrong file format.</b> Follow these instructions to resolve this situation: <ul style="list-style-type: none"> <li>○ How to import/export or delete one or several user devices (<i>see page 47</i>).</li> </ul> </li> </ul>
<b>Delete Device- Applications using the deleted device are no longer usable</b>	<p>This function is enabled only if the list of selected items in the tree control contains only <b>User Device</b> item. When all conditions for the <b>Delete Device(s)</b> function are fulfilled:- Main menu command <b>Edit → Delete</b> or- context menu <b>Delete</b> on a selection of one or more user devices. The user is asked for confirmation of this action. The confirmation message informs about possible consequences: Applications using the deleted device are no longer usable. They cannot be opened.</p> <p>The following message appears:</p>  <p>Follow these instructions to resolve this situation:</p> <ul style="list-style-type: none"> <li>● How to import/export or delete one or several user devices (<i>see page 47</i>).</li> </ul>
<b>Delete Function- Applications using the deleted device are no longer usable</b>	<p>This function is enabled only if the list of selected items in the tree control contains only "User Function" item. The default function of a device cannot be deleted. When all conditions for the <b>Delete Function(s)</b> service are fulfilled:- Main menu command <b>Edit → Delete</b> or- context menu <b>Delete</b> on a selection of one or more user functions. The user is asked for confirmation of this action. The confirmation message informs about possible consequences: Applications using the deleted function are no longer usable. They cannot be opened.</p> <p>The following message appears:</p>  <p>Follow these instructions to resolve this situation:</p> <ul style="list-style-type: none"> <li>● How to copy or delete a function (<i>see page 46</i>).</li> </ul>

## EDS/DCF Import Anomaly Code

### Table

The following table describes the EDS/DCF Import Anomalies:

Anomaly Name	Check performed Anomaly condition	Context delivered	Severity
W_IMP_ALREADYEXISTS	EDS/DCF already present.. Delivering existing instance.	ecEmpty	Monition
F_IMP_COBDNOTFOUND	COBID Profile Database not found.	ecCiAprofile Profile number	Fatal Anomaly
E_IMP_MISSINGMAND	Mandatory object is missing.Mandatory objects are [1000],[1001].	ecObject- Main/Subindex	Anomaly
E_IMP_ILLDATATYPE	Illegal object datatype.	ecObject- Main/Subindex	Anomaly
E_IMP_MISSINGSYNCCYCLE	Object [1006] is mandatory for Sync Producer. Sync Producer is [1005] Bit 30 == 1.	ecEmpty	Anomaly
E_IMP_MISSINGSYNC	Object [1006] resp [1007] cannot exist without Object [1005].	ecEmpty	Anomaly
W_IMP_MISSINGPDOCOMM	PDO communication parameter (20h) missing. PDO ignored.	ecPDO PDO type (In/Out); PDO number	Monition
W_IMP_MISSINGPDOMAP	PDO mapping parameter (21h) missing. PDO ignored.	ecPDO PDO type (In/Out); PDO number	Monition
E_IMP_MISSINGPDOCOBID	PDO communication parameter (21h) element COBID missing.	ecPDO PDO type (In/Out); PDO number	Anomaly
E_IMP_COBID	COBID check for objects [1005],[1012], [1014] – COBID must be in range 1h..7FFh if 11bits CAN Identifiers are used; COBID must be in range 1h..1FFFFFFF if 29bit CANIdentifiers are used.	ecCOBID COBID as number.	Anomaly

E_IMP_PDO_COBID	PDO communication parameter COBID check for objects [14xxsub1] and [18xxsub1] – COBID must be in range 1h..7FFh if 11bits CAN Identifiers are used; COBID must be in range 1h..1FFFFFFF if 29bitCAN Identifiers are used.	ecPDOCOBID PDO type (In/Out); PDO number; COBID as number.	Anomaly
E_IMP_PDO_TTYPE_NOSYNC	PDO communication parameter transmission type check for objects [14xxsub2] and [18xxsub2] – values 0..252 are allowed only if [1005] exists.	ecPDO PDO type (In/Out); PDO number	Anomaly
E_IMP_MAPP_NOENTRY	PDO mapping entry check for objects [16xx] and [1Axx] – Mapped object must exist (excluding NWVs).	ecPDOMappPDO PDO type (In/Out); PDO number; Mapping entry number.	Anomaly
E_IMP_MAPP_NOTMAPPABLE	PDO mapping entry check for objects [16xx] and [1Axx] – Mapped object must be mappable (PDOMapping=1).	ecPDOMappPDO PDO type (In/Out); PDO number; Mapping entry index.	Anomaly
E_IMP_MAPP_INVALIDDATA_TYPE	PDO mapping entry check for objects [16xx] and [1Axx] – Mapped object not mappable due to datatype.	ecPDOMappPDO PDO type (In/Out); PDO number; Mapping entry index.	Anomaly
E_IMP_MAPP_LENGTHDATA_TYPE_MISMATCH	PDO mapping entry check for objects [16xx] and [1Axx] – PDO mapping entry bitlength must match datatype of object to map.	ecPDOMappPDO PDO type (In/Out); PDO number; Mapping entry index	Anomaly
E_IMP_MAPP_ACCESS_TYPE	PDO mapping entry check for objects [16xx] and [1Axx] – Object to map access type not suitable for PDO type (RPDO, TPDO).	ecPDOMappPDO PDO type (In/Out); PDO number; Mapping entry index	Anomaly
E_IMP_MAPP_PDOLNGTH	PDO mapping entry check for objects [16xx] and [1Axx] – PDO length exceeded.	ecPDOMappPDO PDO type (In/Out); PDO number; Mapping entry index	Anomaly

E_IMP_MAPP_GRANULARITY	PDO mapping entry check for objects [16xx] and [1Axx] – Minimum bitlength allowed is 8, Bitmapping not supported by manager.	ecPDOmappPDO type(In/Out); PDO number; Mapping entryindex	Anomaly
I_IMP_CORR_ADDMAP	Auto correction applied: Empty PDO Mapping entry added, corresponding section missing in the EDS/DCF.	ecObject-Main/Subindex	Info
I_IMP_CORR_OBSOLETE-COMMPAR	Auto correction applied: Removed obsolete PDO communication parameter.	ecObject-Main/Subindex	Info
I_IMP_CORR_ILLCOMMPAR	Auto correction applied: Removed illegal PDO communication parameter from RPDO.	ecObject-Main/Subindex	Info
I_IMP_CORR_DEFVALUE	Auto correction applied: Object default value not set, presuming 0.	ecObject-Main/Subindex	Info
I_IMP_CORR_DEFVALUEC-OMMPAR	Auto correction applied: PDO transmission type default value not set, presuming 255.	ecPDOPDO type (In/Out); PDO number	Info
I_IMP_CORR_INVALIDATEPDO	Auto correction applied: Deactivated PDO outside Predefined Connection Set.	ecPDOPDO type(In/Out); PDO number	Info

