



**Operating Instruction Manual**  
**DTM for Hilscher CANopen Slave device**  
**Configuration of Hilscher Slave Devices**

**Hilscher Gesellschaft für Systemautomation mbH**

**[www.hilscher.com](http://www.hilscher.com)**

DOC091002OI04EN | Revision 4 | English | 2010-10 | Released | Public

# Table of Contents

1	INTRODUCTION.....	5
1.1	About this Manual.....	5
1.1.1	Descriptions of the Dialog Panes.....	6
1.1.2	Online Help.....	6
1.1.3	List of Revisions.....	7
1.1.4	Conventions in this Manual.....	7
1.2	Legal Notes.....	8
1.2.1	Copyright.....	8
1.2.2	Important Notes.....	8
1.2.3	Exclusion of Liability.....	9
1.2.4	Warranty.....	9
1.2.5	Export Regulations.....	10
1.2.6	Registered Trademarks.....	10
1.3	About CANopen Slave DTM.....	11
1.3.1	Requirements.....	12
1.4	Dialog Structure of the CANopen Slave DTM.....	13
1.4.1	General Device Information.....	14
1.4.2	Navigation Area.....	14
1.4.3	Dialog Panes.....	15
1.4.4	OK, Cancel, Apply and Help.....	16
1.4.5	Table Lines.....	16
1.4.6	Status Bar.....	17
2	GETTING STARTED AND STEP-INSTRUCTIONS.....	18
2.1	Overview Configuration Steps.....	18
2.1.1	Slave DTM at the Root-Line (Stand-Alone Slave).....	18
2.1.2	Slave DTM at the Master busline.....	20
2.1.3	Setting Driver and Selecting Device.....	22
2.1.4	Configuring Slave Parameters.....	23
3	SETTINGS.....	24
3.1	Overview Settings.....	24
3.2	Driver.....	25
3.2.1	Selecting the Driver.....	25
3.2.2	Configuring the Driver.....	26
3.2.3	cifX Device Driver.....	27
3.2.4	netX Driver.....	28
3.3	Device Assignment.....	35
3.3.1	Scanning for Devices.....	36
3.3.2	Selecting the Device (with or without firmware).....	39
3.3.3	Selecting the Device once more (with Firmware).....	40
3.4	Firmware Download.....	42

4	CONFIGURATION .....	47
4.1	Overview Configuration .....	47
4.2	General Settings .....	48
4.3	Special Function Objects .....	50
4.4	Object Dictionary .....	52
4.5	Process Data Objects .....	56
4.5.1	PDO Properties .....	56
4.5.2	PDO Mapping .....	59
4.6	Address Table.....	61
4.6.1	Description of the Paramters.....	61
4.6.2	Display Mode.....	62
5	DEVICE DESCRIPTION .....	63
5.1	About Device Description .....	63
5.2	Device.....	64
5.3	EDS .....	64
6	DIAGNOSIS .....	65
6.1	Overview Diagnosis .....	65
6.2	General Diagnosis .....	66
6.3	Firmware Diagnosis.....	69
7	EXTENDED DIAGNOSIS .....	70
7.1	Overview Extended Diagnosis.....	70
7.2	Task Information .....	71
7.3	IniBatch Status.....	72
7.3.1	CANOPEN_SLAVE .....	73
7.3.2	CANOPEN_APS.....	76
8	TOOLS .....	78
8.1	Overview Tools .....	78
8.2	Packet Monitor .....	79
8.2.1	Sending Packet .....	80
8.2.2	Receiving Packet.....	81
8.3	IO Monitor .....	83
9	ONLINE FUNCTIONS .....	84
9.1	Connecting/Disconnecting Device .....	84
9.2	Download Configuration .....	86
10	ERROR CODES.....	87
10.1	Error Code Definition .....	87
10.2	Overview Error Codes .....	88

10.3	General Hardware Error Codes .....	89
10.3.1	RCX General Task Errors .....	89
10.3.2	RCX Common Status & Errors Codes .....	90
10.3.3	RCX Status & Error Codes .....	91
10.4	ODM Error Codes .....	92
10.4.1	General ODM Error Codes .....	92
10.4.2	General ODM Driver Error Codes .....	93
10.4.3	cifX Driver Specific ODM Error Codes .....	94
10.5	Error Codes cifX Device Driver and netX Driver .....	97
10.5.1	Generic Error Codes.....	97
10.5.2	Generic Driver Error Codes.....	98
10.5.3	Generic Device Error Codes .....	99
10.6	Error Codes netX Driver .....	100
10.6.1	CIFX API Transport Error Codes.....	100
10.6.2	CIFX API Transport Header State Error Codes .....	100
10.7	ODM Error Codes DBM V4.....	101
11	APPENDIX .....	105
11.1	COB-ID (Predefined Connection Set).....	105
11.2	User Rights .....	106
11.2.1	Settings.....	106
11.2.2	Configuration .....	106
11.3	References .....	107
12	LISTS .....	108
12.1	List of Figures .....	108
12.2	List of Tables .....	110
13	GLOSSARY.....	112
13.1	Contacts.....	113

# 1 Introduction

## 1.1 About this Manual

This manual provides information on how to set and configure the device parameters of a CANopen Slave within a FDT Framework using the CANopen Slave DTM, and what can be read from the diagnosis panes.

In network project the CANopen Slave DTM can be dropped:

- to the root line (Stand-Alone Slave),
- or to the Master busline of a CANopen Master DTM netX.



---

**Note:** This manual edition first provides information for the case, if the CANopen Slave DTM is dropped to the root line (Stand-Alone Slave) and in addition important information for the case, if the CANopen Slave DTM is dropped to the Master busline of a CANopen Master DTM netX.

---

## 1.1.1 Descriptions of the Dialog Panes

The table below gives an overview for the individual dialog panes descriptions:

Section	Subsection	Manual Page
<i>Settings</i>	<i>Overview Settings</i>	24
	<i>Driver</i>	25
	<i>Device Assignment</i>	35
	<i>Firmware Download</i>	42
<i>Configuration</i>	<i>Overview Configuration</i>	47
	<i>General Settings</i>	48
	<i>Special Function Objects</i>	50
	<i>Object Dictionary</i>	52
	<i>Process Data Objects</i>	56
	<i>PDO Properties</i>	56
	<i>PDO Mapping</i>	59
<i>Device Description</i>	<i>Address Table</i>	61
	<i>Device</i>	64
	<i>EDS</i>	64
	<i>Overview Diagnosis</i>	65
<i>Diagnosis</i>	<i>General Diagnosis</i>	66
	<i>Firmware Diagnosis</i>	69
	<i>Overview Extended Diagnosis</i>	70
<i>Tools</i>	<i>Overview Tools</i>	78
	<i>Packet Monitor</i>	79
	<i>IO Monitor</i>	83

Table 1: Descriptions Dialog Panes

## 1.1.2 Online Help

The CANopen Slave DTM contains an integrated online help facility.

- To open the online help, click on the **Help** button or press the **F1** key.

### 1.1.3 List of Revisions

Index	Date	Version	Component	Chapter	Revision
1	23.10.09	1.0.0.4759 1.0.0.4759	CANoSlaveDTM.dll CANoSlaveGUI.ocx	All	created
2	09.08.10	1.2.x.x 1.0.x.x	CANoSlaveDTM.dll CANoSlaveGUI.ocx	3.2.4, 3.3, 10	Section <i>netX Driver</i> added, Section <i>Device Assignment</i> actualized, Chapter <i>Error Codes</i> actualized and completed.
3	12.10.10	1.3.x.x 1.0.x.x	CANoSlaveDTM.dll CANoSlaveGUI.ocx	8.2	Section <i>Packet Monitor</i> actualized.
4	19.10.10	1.3.x.x 1.0.x.x	CANoSlaveDTM.dll CANoSlaveGUI.ocx	8.2	Section <i>Packet Monitor</i> actualized.

### 1.1.4 Conventions in this Manual

Operation instructions, a result of an operation step or notes are marked as follows:

#### **Operation Instructions:**

➤ <instruction>

Or

1. <instruction>

2. <instruction>

#### **Results:**

⇒ <result>

#### **Notes:**




---

**Important:** <important note>

---




---

**Note:** <note>

---




---

<note, were to find further information>

---

## 1.2 Legal Notes

### 1.2.1 Copyright

© 2008-2010 Hilscher Gesellschaft für Systemautomation mbH

All rights reserved.

The images, photographs and texts in the accompanying material (user manual, accompanying texts, documentation, etc.) are protected by German and international copyright law as well as international trade and protection provisions. You are not authorized to duplicate these in whole or in part using technical or mechanical methods (printing, photocopying or other methods), to manipulate or transfer using electronic systems without prior written consent. You are not permitted to make changes to copyright notices, markings, trademarks or ownership declarations. The included diagrams do not take the patent situation into account. The company names and product descriptions included in this document may be trademarks or brands of the respective owners and may be trademarked or patented. Any form of further use requires the explicit consent of the respective rights owner.

### 1.2.2 Important Notes

The user manual, accompanying texts and the documentation were created for the use of the products by qualified experts, however, errors cannot be ruled out. For this reason, no guarantee can be made and neither juristic responsibility for erroneous information nor any liability can be assumed. Descriptions, accompanying texts and documentation included in the user manual do not present a guarantee nor any information about proper use as stipulated in the contract or a warranted feature. It cannot be ruled out that the user manual, the accompanying texts and the documentation do not correspond exactly to the described features, standards or other data of the delivered product. No warranty or guarantee regarding the correctness or accuracy of the information is assumed.

We reserve the right to change our products and their specification as well as related user manuals, accompanying texts and documentation at all times and without advance notice, without obligation to report the change. Changes will be included in future manuals and do not constitute any obligations. There is no entitlement to revisions of delivered documents. The manual delivered with the product applies.

Hilscher Gesellschaft für Systemautomation mbH is not liable under any circumstances for direct, indirect, incidental or follow-on damage or loss of earnings resulting from the use of the information contained in this publication.

### 1.2.3 Exclusion of Liability

The software was produced and tested with utmost care by Hilscher Gesellschaft für Systemautomation mbH and is made available as is. No warranty can be assumed for the performance and flawlessness of the software for all usage conditions and cases and for the results produced when utilized by the user. Liability for any damages that may result from the use of the hardware or software or related documents, is limited to cases of intent or grossly negligent violation of significant contractual obligations. Indemnity claims for the violation of significant contractual obligations are limited to damages that are foreseeable and typical for this type of contract.

It is strictly prohibited to use the software in the following areas:

- for military purposes or in weapon systems;
- for the design, construction, maintenance or operation of nuclear facilities;
- in air traffic control systems, air traffic or air traffic communication systems;
- in life support systems;
- in systems in which failures in the software could lead to personal injury or injuries leading to death.

We inform you that the software was not developed for use in dangerous environments requiring fail-proof control mechanisms. Use of the software in such an environment occurs at your own risk. No liability is assumed for damages or losses due to unauthorized use.

### 1.2.4 Warranty

Although the hardware and software was developed with utmost care and tested intensively, Hilscher Gesellschaft für Systemautomation mbH does not guarantee its suitability for any purpose not confirmed in writing. It cannot be guaranteed that the hardware and software will meet your requirements, that the use of the software operates without interruption and that the software is free of errors. No guarantee is made regarding infringements, violations of patents, rights of ownership or the freedom from interference by third parties. No additional guarantees or assurances are made regarding marketability, freedom of defect of title, integration or usability for certain purposes unless they are required in accordance with the law and cannot be limited. Warranty claims are limited to the right to claim rectification.

## 1.2.5 Export Regulations

The delivered product (including the technical data) is subject to export or import laws as well as the associated regulations of different countries, in particular those of Germany and the USA. The software may not be exported to countries where this is prohibited by the United States Export Administration Act and its additional provisions. You are obligated to comply with the regulations at your personal responsibility. We wish to inform you that you may require permission from state authorities to export, re-export or import the product.

## 1.2.6 Registered Trademarks

Windows® 2000 and Windows® XP are registered trademarks of Microsoft Corporation.

CANopen® is a registered trademark of CAN in AUTOMATION - International Users and Manufacturers Group e.V (CiA), Nürnberg.

All other mentioned trademarks are property of their respective legal owners.

## 1.3 About CANopen Slave DTM

You can use the CANopen Slave DTM to configure CANopen Slave devices within a FDT Framework and to view the device diagnosis.

In network project the CANopen Slave DTM can be dropped:

- to the root line (Stand-Alone Slave),
- or to the Master busline of a CANopen Master DTM netX.

### **Slave DTM at the Root-Line (Stand-Alone Slave)**

Insert the CANopen Slave DTM to the root busline, if you intend to configure only one Slave device.

### **Slave DTM at the Master busline**

Insert the CANopen Slave DTM to the Master busline of the CANopen Master DTM netX, if within the network project the Slave device and the Master device are used, i. e., if you intend to configure the Slave device and the Master device.

## 1.3.1 Requirements

### System Requirements

- Windows® 2000/ Windows® XP
- CD ROM drive
- Graphic resolution: min. 1024 x 768 pixel
- Keyboard and Mouse

### Requirements CANopen Slave DTM

To configure a CANopen Slave device with a DTM the following requirements have to be accomplished:

- Completed hardware installation of a DTM-compatible CANopen Slave device, inclusive loaded firmware and loaded configuration file
- Installed FDT/DTM V 1.2 compliant frame application
- Installed CANopen Master DTM (If Slave DTM is dropped to the Master busline.)
- Loaded DTM in the Device Catalog of the FTD Framework



---

For more information to the hardware installation, please refer to the corresponding user manual of your device.

---

## 1.4 Dialog Structure of the CANopen Slave DTM

The graphical user interface of the DTM is composed of different areas and elements listed hereafter:

1. A header area containing the **General Device Information**,
2. The **Navigation Area** (area on the left side),
3. The **Dialog Pane** (main area on the right side),
4. **OK, Cancel, Apply, Help**,
5. The **Status Line** containing information e. g. the online-state of the utility.

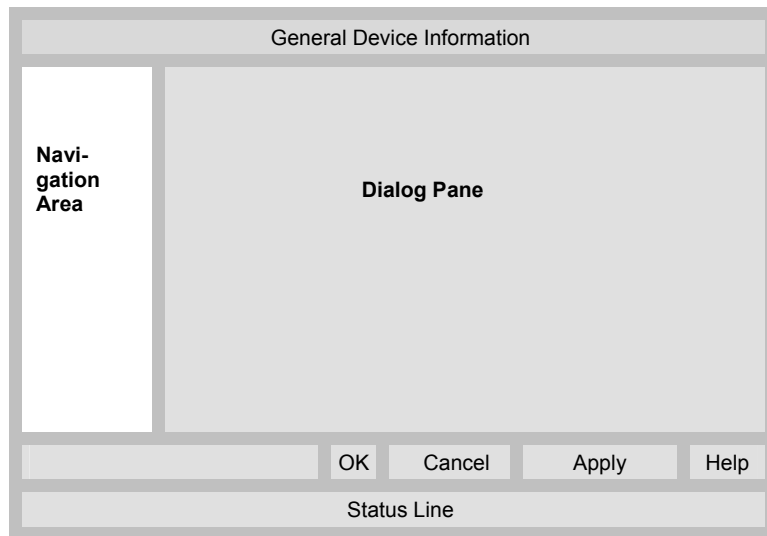


Figure 1: Dialog Structure of the CANopen Slave DTM

## 1.4.1 General Device Information

Parameter	Meaning
IO Device	Name of the device
Vendor	Vendor name of the device
Device ID	Identification number of the device
Vendor ID	Identification number of the vendor

Table 2: General Device Information

## 1.4.2 Navigation Area

The **Navigation Area** contains folders and subfolders to open the dialog panes of the DTM.

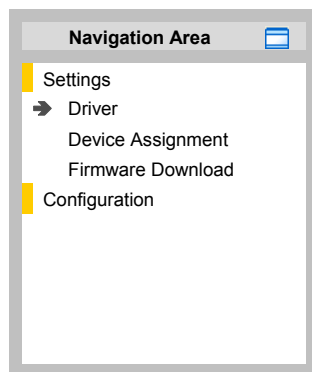




Figure 2: Navigation Area

- Select the required folder and subfolder.
- The corresponding Dialog pane is displayed.

### Hide / display Navigation

	Hiding the navigation area (above right side).
 Show navigation area	<b>Opening</b> the navigation area (below left side).

### 1.4.3 Dialog Panes

Settings	
Driver	On the pane Driver you can select a driver from the driver list. For further information, refer to section <i>Driver</i> on page 25.
Device Assignment	On the pane Device Assignment you select the device and assign the device to the driver. For further information, refer to section <i>Device Assignment</i> on page 35.
Firmware Download	The dialog on the pane Firmware Download is used to load a new firmware into the device. A detailed description can be found in section <i>Firmware Download</i> on page 42.
Configuration	
General	The dialog General Settings displays EDS file data. For further information see section <i>General Settings</i> on page 48.
Special Function Objects	The dialog Special Function Objects displays data of the synchronization, time stamp and emergency telegram. For further information see section <i>Special Function Objects</i> on page 50.
Object Dictionary	The dialog Object Dictionary represents the object dictionary of the device. The display shows data read out from the EDS file. For further information see section <i>Object Dictionary</i> on page 52.
Process Data Objects	<p>PDO Properties:</p> <p>In the dialog PDO Properties the transmit and the receive PDOs are displayed. For further information see section <i>PDO Properties</i> on page 56.</p>
	<p>PDO Mapping:</p> <p>The dialog PDO Mapping permits to map the contents of a PDO. For further information see section <i>PDO Mapping</i> on page 59.</p>
Address Table	The Address Table shows a list of all dpram address used in the process data image. For further information, refer to section <i>Address Table</i> on page 61.
Device Description	
Device	The Device Info pane contains the manufacturer information about the device. Further information to this you find in section <i>Device</i> on page 64.
EDS	By use of the EDS-Viewer a EDS file can be searched through. Further information to this you find in section <i>EDS</i> on page 64.
Diagnosis	
Diagnosis/Extended Diagnosis	At the Diagnosis panes diagnosis information can be read. For further information, refer to section <i>Overview Diagnosis</i> on page 65.
Tools	
Packet Monitor/IO Monitor	Under Tools the Packet Monitor and the IO Monitor are provided for test and diagnosis purposes. For further information, refer to section <i>Packet Monitor</i> on page 79 or section <i>IO Monitor</i> on page 83.

Table 3: Overview Dialog Panes



**Note:** Accessing the **Diagnosis** panes of the CANopen Slave DTM requires an online connection from the CANopen Slave DTM to the CANopen Slave DTM.



For further information, refer to section *Connecting/Disconnecting Device* on page 84.

## 1.4.4 OK, Cancel, Apply and Help

**OK, Cancel, Apply and Help** you can use as described hereafter.

	Meaning
<b>OK</b>	To confirm your latest settings, click <b>OK</b> . All changed values will be applied on the frame application database. <i>The dialog then closes.</i>
<b>Cancel</b>	To cancel your latest changes, click <b>Cancel</b> . Answer to the safety query <b>Configuration data has been changed. Do you want to save the data?</b> by <b>Yes, No</b> or <b>Cancel</b> . <b>Yes:</b> The changes are saved or the changed values are applied on the frame application database. <i>The dialog then closes.</i> <b>No:</b> The changes are <u>not</u> saved or the changed values are not applied on the frame application database. <i>The dialog then closes.</i> <b>Cancel:</b> <i>Back to the DTM.</i>
<b>Apply</b>	To confirm your latest settings, click <b>Apply</b> . All changed values will be applied on the frame application database. <i>The dialog remains opened.</i>
<b>Help</b>	To open the DTM online help, click <b>Help</b> .

Table 4: OK, Cancel, Apply and Help

## 1.4.5 Table Lines

In the DTM dialog pane table lines can be selected, inserted or deleted.

	Meaning
	To select the first line of a table use <b>First Line</b> .
	To select the previous line of a table use <b>Previous Line</b> .
	To select the next line of a table use <b>Next Line</b> .
	To select the last line of a table use <b>Last Line</b> .
	<b>Create a new Line</b> inserts new lines into the table.
	<b>Delete selected Line</b> deletes the selected line from the table.

Table 5: Selecting, inserting, deleting Table Line

### 1.4.6 Status Bar

The **Status Bar** displays information about the current state of the DTM. The current activity, e.g. download, is signaled graphically via icons in the status bar.

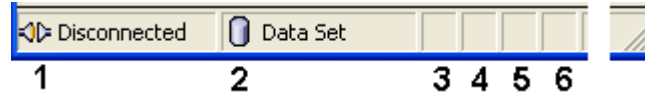


Figure 3: Status Bar – Status Fields 1 to 6

Status Field	Icon / Meaning	
<b>1</b>	<b>DTM Connection States</b>	
		<b>Connected:</b> Icon closed = Device is online
		<b>Disconnected:</b> Icon opened = Device is offline
<b>2</b>	<b>Data Source States</b>	
		<b>Data set:</b> The displayed data are read out from the instance data set (database).
		<b>Device:</b> The displayed data are read out from the device.
<b>3</b>	<b>States of the instance Date Set</b>	
		<b>Valid Modified:</b> Parameter is changed (not equal to data source).
<b>4</b>	<b>Changes directly made on the Device</b>	
		Load/configure diagnosis parameters: Diagnosis is activated.
<b>6</b>	<b>Device Diagnosis Status</b>	
		<b>Save operation succeeded:</b> The save operation has been successful. Further messages due to successful handling of device data.
		<b>Firmware Download:</b> Firmware Download is running
		<b>Save operation failed:</b> The save operation has failed. Further fail operation messages due to incorrect communication due to malfunction in the field device or its peripherals.

Table 6: Status Bar Icons [1]

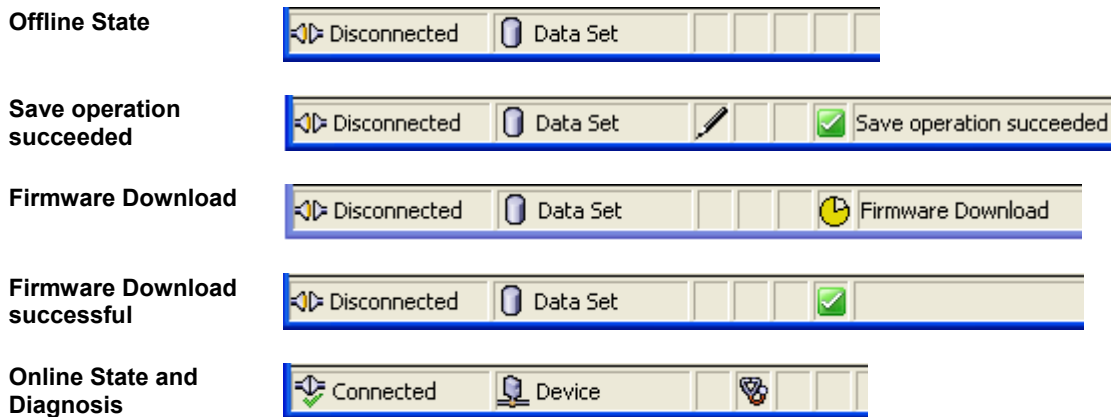


Figure 4: Status Bar Display Examples

## 2 Getting Started and Step-Instructions

### 2.1 Overview Configuration Steps

The following tables describe the steps to configure a CANopen Slave device with the CANopen Slave DTM for the two cases listed hereafter, as it is typical for many cases:

- Slave DTM at the Root-Line (Stand-Alone Slave)
- and Slave DTM at the Master busline.

At this time it is presupposed that the hardware installation was done.


#### 2.1.1 Slave DTM at the Root-Line (Stand-Alone Slave)

#	Step	Short Description	For detailed information see section	Page
1	Load device catalog	Depending of the FDT Container: For netDevice: - select <b>Network &gt; Device Catalog</b> , - select button <b>Reload Catalog</b> .	<i>(See User Manual netDevice and netProject)</i>	-
2	Create new project / Open existing project	Depending of the frame application. For the configuration software: - select <b>File &gt; New</b> or <b>File &gt; Open</b> .	<i>(See User Manual of the Frame Application)</i>	-
3	Insert Slave into configuration	Depending of the FDT Container: For netDevice: - in the Device Catalog click to the Slave, - and insert the Slave via drag and drop <b>to the root line</b> in the network view.	<i>(See User Manual netDevice and netProject)</i>	-
4	Open the Slave DTM configuration dialog	Open the Slave DTM configuration dialog. - Double click to the device icon of the Slave. - The Slave DTM configuration dialog is displayed.	-	-
5	Select driver	In the Slave DTM configuration dialog: - select <b>Settings &gt; Driver</b> , - select a driver, - therefore check the appropriate checkbox.	<i>Setting Driver and Selecting Device</i> <i>Driver</i>	22 25
6	Configure Driver	If necessary, configure the driver settings. For netX Driver and communication via TCP/IP set the IP address of the device: - Select <b>Settings &gt; Driver &gt; netX Driver &gt; TCP Connection</b> . - Via <input type="checkbox"/> add an IP range. - Under <b>IP Address</b> enter the IP Address of the device or an IP range. - Click <b>Save</b> .	<i>netX Driver</i>	28
7	Assign Slave device (with or without firmware)	Assign the device to this driver. In the Slave DTM configuration dialog: - select <b>Settings &gt; Device Assignment</b> , - select a Slave device (with or without firmware), - therefore check the appropriate checkbox, - select the button <b>Apply</b> .	<i>Selecting the Device (with or without firmware)</i>	39
8	Select and download firmware	If not yet a firmware was loaded to the device. In the Slave DTM configuration dialog: - select <b>Settings &gt; Firmware Download</b> , - select the button <b>Browse...</b> , - select a firmware file, - select the button <b>Open</b> , - select the buttons <b>Download</b> and <b>Yes</b> .	<i>Firmware Download</i>	42

#	Step	Short Description	For detailed information see section	Page
9	Assign Slave device once more (with firmware)	In the Slave DTM configuration dialog: <ul style="list-style-type: none"> <li>- select <b>Settings &gt; Device Assignment</b>,</li> <li>- select the button <b>Scan</b>,</li> <li>- select the Slave device (with loaded firmware),</li> <li>- therefore check the appropriate checkbox,</li> <li>- select the button <b>Apply</b>,</li> <li>- close the Slave DTM configuration dialog via the button <b>OK</b>.</li> </ul>	<i>Selecting the Device once more (with Firmware)</i>	40
10	Configure Slave device	Configure the Slave device. <ul style="list-style-type: none"> <li>- Double click to the device icon of the Slave.</li> <li>- The Slave DTM configuration dialog is displayed.</li> </ul> In the Slave DTM configuration dialog: <ul style="list-style-type: none"> <li>- select <b>Configuration &gt; Object Dictionary</b>,</li> <li>- define the object filters,</li> <li>- select <b>Configuration &gt; Process Data Objects &gt; PDO Properties</b>,</li> <li>- configure the PDO to be used for the communication,</li> <li>- select <b>Configuration &gt; Process Data Objects &gt; PDO Mapping</b>,</li> <li>- configure the list of the mappable or the list of the mapped objects each,</li> <li>- close the Slave DTM configuration dialog via the button <b>OK</b>.</li> </ul>	<i>Configuring Slave Parameter</i>  <i>Object Dictionary</i>  <i>PDO Properties</i>  <i>PDO Mapping</i>	23  52  56  59
11	Save project	Depending of the frame application. For the configuration software: <ul style="list-style-type: none"> <li>- select <b>File &gt; Save</b>.</li> </ul>	<i>(See User Manual of the Frame Application)</i>	-
12	Connect Slave device	Depending of the FDT Container. For netDevice: <ul style="list-style-type: none"> <li>- right click to the device icon of the Slave,</li> <li>- select context menu entry <b>Connect</b>.</li> </ul>	<i>Connecting/Disconnecting Device</i>	84
13	Download Configuration	Depending of the FDT Container. For netDevice: <ul style="list-style-type: none"> <li>- right click to the device icon of the Slave,</li> <li>- select context menu entry <b>Download</b>.</li> </ul>	<i>Download Configuration</i>	86
14	Diagnosis	Depending of the FDT Container. For netDevice: <ul style="list-style-type: none"> <li>- right click to the device icon of the Slave,</li> <li>- select context menu entry <b>Diagnosis</b>.</li> <li>- The Slave DTM diagnosis dialog is displayed.</li> <li>- Continue with further device diagnosis,</li> <li>- close the Slave DTM diagnosis dialog via the button <b>OK</b>.</li> </ul>	<i>Overview Diagnosis</i>	65
15	IO Monitor	Depending of the FDT Container: For netDevice: <ul style="list-style-type: none"> <li>- right click to the device icon of the Slave,</li> <li>- select context menu entry <b>Diagnosis</b>,</li> <li>- select <b>Tools &gt; IO Monitor</b>.</li> <li>- Check the input or output data.</li> </ul>	<i>IO Monitor</i>	83
16	Disconnect	Depending of the FDT Container. For netDevice: <ul style="list-style-type: none"> <li>- right click to the device icon of the Slave,</li> <li>- select <b>context menu entry Disconnect</b>.</li> </ul>	<i>Connecting/Disconnecting Device</i>	84

Table 7: Getting Started - Configuration Steps (Slave DTM at the Root-Line (Stand-Alone Slave))

## 2.1.2 Slave DTM at the Master busline

#	Step	Short Description	For detailed information see section	Page
1	Load device catalog	Depending of the FDT Container: For netDevice: - select <b>Network &gt; Device Catalog</b> , - select button <b>Reload Catalog</b> .	(See User Manual netDevice and netProject)	-
2	Create new project / Open existing project	Depending of the frame application. For the configuration software: - select <b>File &gt; New</b> or <b>File &gt; Open</b> .	(See User Manual of the Frame Application)	-
3	Insert Master or Slave into configuration	For netDevice: - in the Device Catalog click to the Master, - and insert the device via drag and drop <b>to the root line</b> in the network view, - in the Device Catalog click to the Slave, - and insert the device via drag and drop <b>to the Master bus line</b> in the network view.	(See User Manual netDevice and netProject)	-
4	Open the Slave DTM configuration dialog	Open the Slave DTM configuration dialog. - Double click to the device icon of the Slave. - The Slave DTM configuration dialog is displayed.	-	-
5	Select driver	In the Slave DTM configuration dialog: - select <b>Settings &gt; Driver</b> , - select a driver, - therefore check the appropriate checkbox. - if necessary, configure the driver settings.	<i>Setting Driver and Selecting Device</i> <i>Driver</i>	22 25
6	Configure Driver	If necessary, configure the driver settings. For netX Driver and communication via TCP/IP set the IP address of the device: - Select <b>Settings &gt; Driver &gt; netX Driver &gt; TCP Connection</b> . - Via  add an IP range. - Under <b>IP Address</b> enter the IP Address of the device or an IP range. - Click <b>Save</b> .	<i>netX Driver</i>	<b>28</b>
7	Assign Slave device (with or without firmware)	Assign the device to this driver. In the Slave DTM configuration dialog: - select <b>Settings &gt; Device Assignment</b> , - select a Slave device (with or without firmware), - therefore check the appropriate checkbox, - select the button <b>Apply</b> .	<i>Selecting the Device (with or without firmware)</i>	39
8	Select and download firmware	If not yet a firmware was loaded to the device. In the Slave DTM configuration dialog: - select <b>Settings &gt; Firmware Download</b> , - select the button <b>Browse..</b> , - select a firmware file, - select the button <b>Open</b> , - select the buttons <b>Download</b> and <b>Yes</b> .	<i>Firmware Download</i>	42
9	Assign Slave device once more (with firmware)	In the Slave DTM configuration dialog: - select <b>Settings &gt; Device Assignment</b> , - select the button <b>Scan</b> , - select the Slave device (with loaded firmware), - therefore check the appropriate checkbox, - select the button <b>Apply</b> , - close the Slave DTM configuration dialog via the button <b>OK</b> .	<i>Selecting the Device once more (with Firmware)</i>	40

#	Step	Short Description	For detailed information see section	Page
10	Configure Slave device	Configure the Slave device. - Double click to the device icon of the Slave. - The Slave DTM configuration dialog is displayed. In the Slave DTM configuration dialog: - select <b>Configuration &gt; Object Dictionary</b> , - define the object filters, - select <b>Configuration &gt; Process Data Objects &gt; PDO Properties</b> , - configure the PDO to be used for the communication, - select <b>Configuration &gt; Process Data Objects &gt; PDO Mapping</b> , - configure the list of the mappable or the list of the mapped objects each, - close the Slave DTM configuration dialog via the button <b>OK</b> .	<i>Configuring Slave Parameter</i>  <i>Object Dictionary</i>  <i>PDO Properties</i>  <i>PDO Mapping</i>	23  52  56  59
11	Configure Master device	Configure the Master device via the CANopen Master DTM netX.	<i>(See User Manual DTM for CANopen Master devices)</i>	-
12	Save project	Depending of the frame application. For the configuration software: - select <b>File &gt; Save</b> .	<i>(See User Manual of the Frame Application)</i>	-
13	Connect Slave device	Depending of the FDT Container. For netDevice: - right click to the device icon of the Slave, - select context menu entry <b>Connect</b> .	<i>Connecting/Disconnecting Device</i>	84
14	Download Configuration	Depending of the FDT Container. For netDevice: - right click to the device icon of the Slave, - select context menu entry <b>Download</b> .	<i>Download Configuration</i>	86
15	Diagnosis	Depending of the FDT Container. For netDevice: - right click to the device icon of the Slave, - select context menu entry <b>Diagnosis</b> . - The Slave DTM diagnosis dialog is displayed. - Continue with further device diagnosis, - close the Slave DTM diagnosis dialog via the button <b>OK</b> .	<i>Overview Diagnosis</i>	<b>65</b>
16	IO Monitor	Depending of the FDT Container: For netDevice: - right click to the device icon of the Slave, - select context menu entry <b>Diagnosis</b> , - select <b>Tools &gt; IO Monitor</b> . - Check the input or output data.	<i>IO Monitor</i>	83
17	Disconnect	Depending of the FDT Container. For netDevice: - right click to the device icon of the Slave, - select <b>context menu entry Disconnect</b> .	<i>Connecting/Disconnecting Device</i>	84

Table 8: Getting Started - Configuration Steps (Slave DTM at the Master busline)

### 2.1.3 Setting Driver and Selecting Device

The following steps are needed to establish a connection from the CANopen Slave DTM to the CANopen Slave device:

Under **Settings** in the **Driver** pane:

1. Select one or several driver/s.
2. Configure the driver, if necessary.
- For netX Driver and communication via TCP/IP set the IP address of the device.

Under **Settings** in the **Device Assignment** pane:

3. Scan for the device/s (with or without firmware).
4. Select the device (with or without firmware) and apply the selection.

Under **Settings** in the **Firmware Download** pane, if not yet a firmware was loaded to the device:

5. Select and download the firmware.

Under **Settings** in the **Device Assignment** pane, if not yet a firmware was loaded to the device:

6. Scan for the device (with firmware) once more.
7. Select the device (with firmware) once more.



An overview of the descriptions for these steps you find in the section *Overview Settings* on page 24.

---

8. In the DTM interface dialog select the **OK** button, to apply the selection and to close the DTM interface dialog.
  9. Put a right-click on the CANopen Slave device icon.
  10. Select the **Connect** command from the context menu.
- The CANopen Slave device now is connected to the CANopen Slave DTM via an online connection. In the network view the device description at the device icon of the Slave is displayed with a green colored background.

## 2.1.4 Configuring Slave Parameters

The following steps are needed to configure the device parameters using the CANopen Slave DTM. This is valid for the two cases listed hereafter:

- Slave DTM at the Root-Line (Stand-Alone Slave)
- and Slave DTM at the Master busline.

Steps:

1. Define the object filters.
  - Select **Configuration > Object Dictionary** in the navigation area.
2. Configure the PDO.
  - Select **Configuration > Process Data Objects > PDO Properties** in the navigation area.
  - Configure the PDO to be used for the communication.
3. Configure the PDO Mapping.
  - Select **Configuration > Process Data Objects > PDO Mapping** in the navigation area.
  - Configure the list of the mappable or the list of the mapped objects each.



For more information refer to section *Object Dictionary* on page 52, *PDO Properties* on page 56 and *PDO Mapping* on page 59.

---

## 3 Settings

### 3.1 Overview Settings

#### Settings Dialog Panes

The table below gives an overview for the individual **Settings** dialog panes descriptions:

Section	Subsection	Page
<i>Driver</i>		25
	<i>Selecting the Driver</i>	25
	<i>Configuring the Driver</i>	26
	<i>cifX Device Driver</i>	27
	<i>netX Driver</i>	<b>28</b>
<i>Device Assignment</i>		35
	<i>Scanning for Devices</i>	36
	<i>Scanning for all Devices or for suitable only</i>	38
	<i>Selecting the Device (with or without firmware)</i>	39
	<i>Selecting the Device once more (with Firmware)</i>	40
<i>Firmware Download</i>		42

Table 9: Descriptions Settings

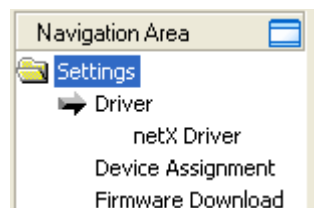


Figure 5: Navigation Area - Settings



**Note:** To edit the **Settings** dialog panes you need *User Rights* for "Maintenance".



Notice the descriptions in the section *Overview Configuration Steps* on page 18.

## 3.2 Driver

The **Driver** dialog pane displays the driver/s to be used for a CANopen Slave DTM to device communication connection.

Driver			
	Driver	Version	ID
<input type="checkbox"/>	CIFX Device Driver	1.0.3.1	{368BEC5B-0E92-4C0E-B4A9-64F62AE7AAFA}
<input type="checkbox"/>	netX Driver	1.1.1.0	{B54C8CC7-F333-4135-8405-6E12FC88EE62}

Figure 6: Driver Selection List

Parameter	Meaning
Driver	Name of the driver
Version	Version of the driver
ID	ID of the driver (driver identification)

Table 10: Driver Selection List Parameters

### 3.2.1 Selecting the Driver

To establish a connection you need to select a driver first.



**Note:** Not all the drivers displayed necessarily support the device. The used driver must be supported by the device or must be available for the device.

To select the driver/s to be used:

1. Select **Settings > Driver** in the navigation area.

➤ The available drivers are listed on the **Driver** dialog pane.

Driver			
	Driver	Version	ID
<input type="checkbox"/>	CIFX Device Driver	1.0.3.1	{368BEC5B-0E92-4C0E-B4A9-64F62AE7AAFA}
<input type="checkbox"/>	netX Driver	1.1.1.0	{B54C8CC7-F333-4135-8405-6E12FC88EE62}

Figure 7: Driver Selection List - recognized or installed Drivers

1. Check the checkbox for the driver/s in the selection list.

Driver			
	Driver	Version	ID
<input checked="" type="checkbox"/>	CIFX Device Driver	1.0.3.1	{368BEC5B-0E92-4C0E-B4A9-64F62AE7AAFA}
<input type="checkbox"/>	netX Driver	1.1.1.0	{B54C8CC7-F333-4135-8405-6E12FC88EE62}

Figure 8: Driver Selection List - Driver/s selected

### 3.2.2 Configuring the Driver

All drivers for which you can access to an individual driver dialog pane are displayed in the navigation area under the **Driver** entry. In the driver dialog panes you can configure the driver settings.

For some drivers no individual driver dialog panes exist, since no driver settings are required, e. g. for the **cifX Device Driver**.



**Note:** At the driver dialog panes under **Settings > Driver** the driver settings can be configured.

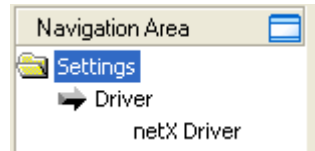


Figure 9: Navigation Area – Settings

To configure a driver:

1. Select **Settings > Driver > [Name of the assigned driver]**.
  - ↗ The corresponding driver dialog pane is displayed.
2. Configure the driver settings.
  - To get detailed information how to configure the single drivers, please refer to the corresponding description of the driver.



You can access to the single descriptions of the drivers as online help via the DTM user interface (**F1** key).

To access to the online help with the descriptions of the drivers:

- Select **Settings > Driver > [Name of the assigned driver]**.
- Press the **F1** key.

### 3.2.3 cifX Device Driver

In the CANopen Slave DTM for the **cifX Device Driver** no driver dialog pane is available, since for the **cifX Device Driver** no driver settings are required.

The **cifX Device Driver** is used, if the CANopen Slave DTM is installed in the same PC as the CANopen Slave device.



---

**Note:** To establish a connection from a DTM to a Slave device via the **cifX Device Driver**, the **cifX Device Driver** must be installed and the driver must have access to the Slave device.

---

### 3.2.4 netX Driver

The **netX Driver** is used to connect the DTM to the device via different connection types. The DTM communicates with the device via an USB connection, a serial (RS232) connection or a TCP/IP connection. The **netX Driver** establishes

- via the USB interface of the device and the USB port of the PC an USB connection to the device,
- via the RS232 interface of the device and the COM port of the PC a serial connection (RS232) to the device
- and via Ethernet a TCP/IP connection to the device.

To connect the DTM to the physical layer of the device the **netX Driver** software works in combination with the software components:

- “USB/COM connector” for the USB connection and for the serial connection (RS232) and
- “TCP/IP connector” for the Ethernet connection.

### 3.2.4.1 netX Driver - USB/RS232 Connection

The communication from the DTM to the device via an **USB/RS232 Connection** is used when the DTM is installed on a PC and between the PC and the device

- an USB connection
- or a serial connection (RS232) exists.

The DTM accesses to the device via the USB interface or via the RS232 interface of the device. This requires either to connect an USB port of the PC to the USB interface of the device using an USB cable or to connect a physical COM port of the PC to the RS232 interface of the device via a serial cable.

The **netX Driver / USB/RS232 Connection** supports all physical and virtual COM ports available at the PC.

Via the RS232 interface or USB interface of the device, the device is configured or diagnosis is performed.

### 3.2.4.2 Driver Parameters for netX Driver - USB/RS232 Connection

The settings of the driver parameters for the USB/RS232 connection are made via the **netX Driver / USB/RS232 Connection** configuration dialog.

➤ Open the **USB/RS232 Connection** dialog via navigation area **Settings > Driver > netX Driver**.

➤ The **USB/RS232 Connection** dialog is displayed:

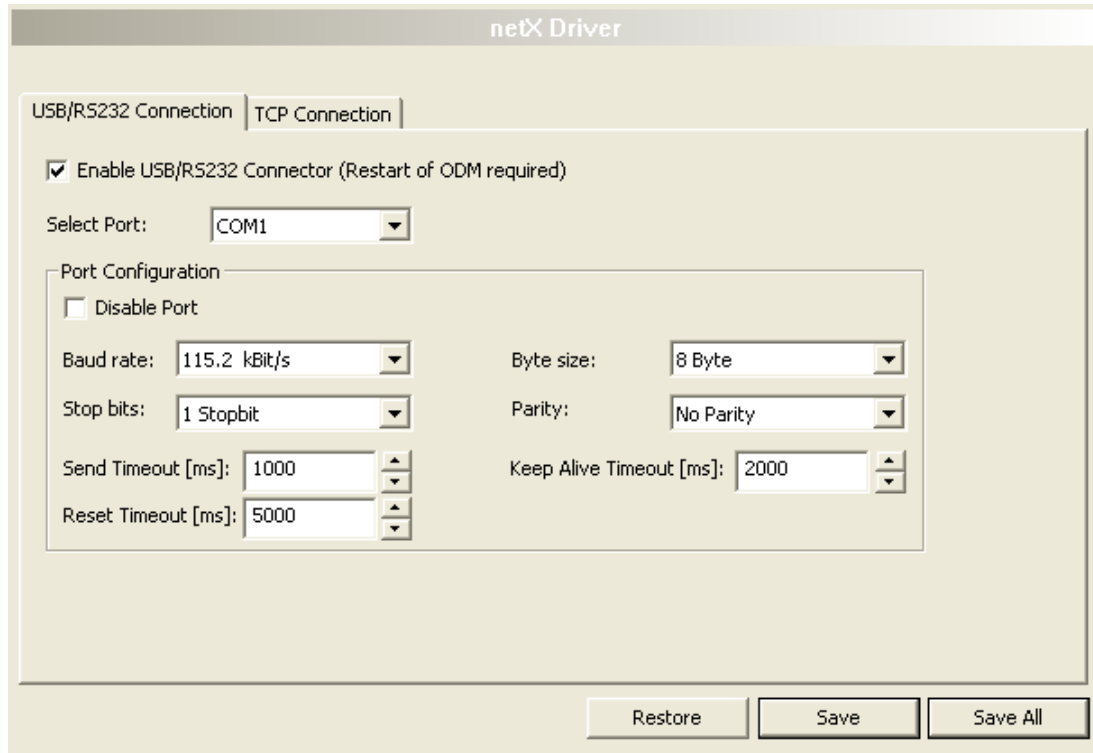



Figure 10: netX Driver > USB/RS232 Connection

Parameter	Meaning	Range of Value / Value
<b>Enable USB/RS232 Connector (Restart of ODM required)</b>	<p>checked: The netX Driver can communicate via the USB/RS232 interface.</p> <p>unchecked: The netX Driver can <u>not</u> communicate via the USB/RS232 interface.</p> <p>If the check mark for <b>Enable USB/RS232 Connector</b> is set or removed, then the ODM server must be restarted<sup>1</sup>, to make the new setting valid.</p> <p><sup>1</sup> Restart the ODM server via the <b>ODMV3 Tray Application</b>:</p> <p>- In the foot line click to  using the right mouse key.</p> <p>- In the context menu select <b>Service &gt; Start</b>.</p>	checked, unchecked; Default: unchecked
<b>Select Port</b>	Depending which COM ports (interfaces) are available on the PC; they will be listed under <b>Select Port</b> .	COM 1 to COM N
<b>Port Configuration</b>		
<b>Disable Port</b>	<p>checked: No connection.</p> <p>unchecked: The netX Driver tries to establish a connection using the configured USB/RS232 interface.</p>	checked, unchecked (Default)
<b>Baud rate</b>	<p>Transfer rate: number of bits per second.</p> <p>The device must support the baud rate.</p>	9.6, 19.2, 38.4, 57.6 or 115.2 [kBit/s]; Default (RS232): 115.2 [kBit/s]

Parameter	Meaning	Range of Value / Value
<b>Stop bits</b>	Number of stop bits sent after the transfer of the send data for synchronization purposes to the receiver.	Stop bit: 1, 1.5, 2; Default (RS232): 1
<b>Send Timeout [ms]</b>	Maximum time before the transfer of the transmission data is canceled, when the send process fails, for example, because of the transfer buffer is full.	100 ... 60.000 [ms]; Default (RS232 and USB): 1000 ms
<b>Reset Timeout [ms]</b>	Maximum time for a device reset, including the re-initialization of the physical interface used for the communication.	100 ... 60.000 [ms]; Default (RS232 and USB): 5000 ms
<b>Byte size</b>	Number of bits per byte by byte specification	7 Bit, 8 Bit; Default (RS232): 8 Bit
<b>Parity</b>	In the error detection in data transmission using parity bits, "parity" describes the number of bits occupied with 1 in the transmitted information word. No Parity: no parity bit Odd Parity: The parity is "odd ", if the number of bits occupied with 1 in the transmitted information word is odd. Even parity: The parity is "even", if the number of bits occupied with 1 in the transmitted information word is even. Mark Parity: if the parity bit is always 1, this is named mark-parity (it contains no information). Space Parity: if the parity bit always 0, this is named space-parity (it is an empty space).	No Parity, Odd Parity, Even Parity, Mark Parity, Space Parity; Default (RS232): No Parity
<b>Keep Alive Timeout [ms]</b>	The "Keep Alive" mechanism is used to monitor whether the connection to the device is active. Connection errors are detected using a periodic heartbeat mechanism. The heartbeat mechanism is initiated after the set time has elapsed, if the communication has failed.	100 ... 60.000 [ms]; Default (RS232 and USB): 2000 ms
<b>Restore</b>	Resets all settings in the configuration dialog to the default values.	
<b>Save</b>	Saving all settings made in the configuration dialog <b>netX Driver &gt; Save USB/RS232 Connection</b> , i. e. only for the selected connection type.	
<b>Save All</b>	Saving all settings made in the configuration dialog <b>netX Driver</b> , i. e. for all connection types.	

Table 11: Parameters netX Driver > USB/RS232 Connection

### 3.2.4.3 netX Driver - TCP/IP Connection

The communication from the DTM to the device via a **TCP/IP Connection** is used in the following two typical applications:

Application 1: The device has its own Ethernet interface. The DTM is installed on a PC and the TCP/IP connection is established from this PC to the stand-alone device. The IP address of the device is used.

Application 2: The device is installed in a remote PC. The DTM is installed on an additional PC and the TCP/IP connection is established from this PC to the remote PC. The IP address of the remote PC is used. For the TCP/IP connection is made, on the remote PC the cifX TCP/IP server must be started. The cifX TCP/IP server allows the remote access to the device via a TCP/IP connection.



**Note:** An exe file for the cifXTCP/IP server is provided on the product CD in the *Tools* directory.

---

Via the TCP/IP interface of the device or of the remote PC, the device is configured or diagnosis is performed.

### 3.2.4.4 Driver Parameters for netX Driver - TCP/IP Connection

The settings of the driver parameters for the TCP/IP connection are made via the **netX Driver / TCP Connection** configuration dialog.

- Open the **TCP Connection** dialog via navigation area **Settings > Driver > netX Driver**.
- The dialog **netX Driver** is displayed:
- Select **TCP Connection**.

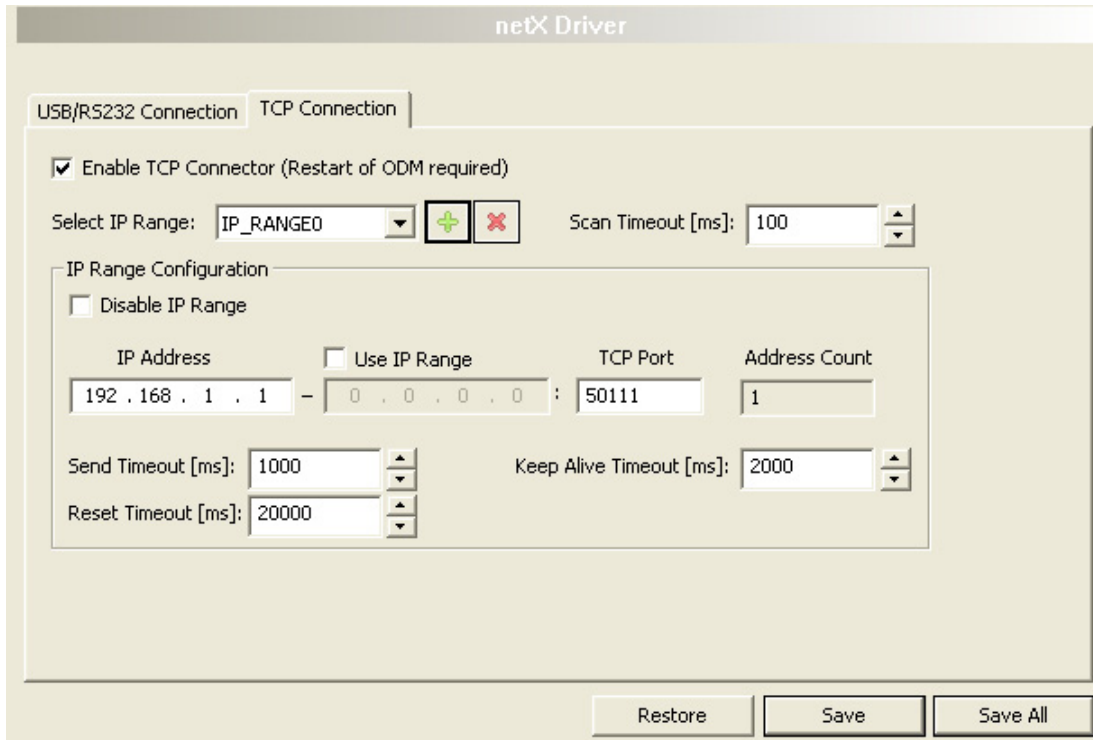





Figure 11: netX Driver > TCP Connection

Parameter	Meaning	Range of Value / Value
<b>Enable TCP Connector (Restart of ODM required)</b>	<p>checked: The netX Driver can communicate via the TCP/IP interface.</p> <p>unchecked: The netX Driver can <u>not</u> communicate via the TCP/IP interface.</p> <p>If the check mark for <b>Enable TCP Connector</b> is set or removed, then the ODM server must be restarted<sup>1</sup>, to make the new setting valid.</p> <p>_____</p> <p><sup>1</sup> Restart the ODM server via the <b>ODMV3 Tray Application</b>:</p> <p>-In the foot line click to  using the right mouse key.</p> <p>- In the context menu select <b>Service &gt; Start</b>.</p>	checked, unchecked; Default: unchecked
<b>Select IP Range</b>	<p>Via <b>Select IP Range</b> already created IP ranges can be selected.</p> <p>Via  an additional IP range can be added.</p> <p>Via  an IP range can be deleted.</p>	
<b>Scan Timeout [ms]</b>	With the scan timeout can be set, how long to wait for a response during a connection is established.	10 ... 10000 [ms]; Default: 100 ms

Parameter	Meaning	Range of Value / Value
<b>IP Range Configuration</b>		
<b>Disable IP Range</b>	checked: No connection. unchecked: The netX Driver tries to establish a connection using the configured TCP/IP interface.	checked, unchecked (Default)
<b>IP Address (left)</b>	Enter the IP address of the device, (if <b>Use IP Range</b> is not checked). Enter the start address of the IP scanning range, (if <b>Use IP Range</b> is checked).	valid IP address; Default: 192.168.1.1
<b>Use IP Range</b>	checked: An IP address range is used. unchecked: Only one IP address is used.	checked, unchecked; Default: unchecked
<b>IP Address (right)</b>	Enter the ending address of the IP scanning range, (only if <b>Use IP Range</b> is checked).	valid IP address; Default: 0.0.0.0
<b>Address Count</b>	Displays the scanning range address count, depending on the selected IP-start or IP-end address. (For this read the note given below.)	recommended: 10
<b>TCP Port</b>	Identifies the endpoint of a logical connection or addresses a specific endpoint on the device or PC.	0 - 65535; Default Hilscher device: 50111
<b>Send Timeout [ms]</b>	Maximum time before the transfer of the transmission data is canceled, when the send process fails, for example, because of the transfer buffer is full.	100 ... 60.000 [ms]; Default (TCP/IP): 1000 ms
<b>Reset Timeout [ms]</b>	Maximum time for a device reset, including the re-initialization of the physical interface used for the communication.	100 ... 60.000 [ms]; Default (TCP/IP): 2000 ms
<b>Keep Alive Timeout [ms]</b>	The "Keep Alive" mechanism is used to monitor whether the connection to the device is active. Connection errors are detected using a periodic heartbeat mechanism. The heartbeat mechanism is initiated after the set time has elapsed, if the communication has failed.	100 ... 60.000 [ms]; Default (TCP/IP): 2000 ms
<b>Restore</b>	Resets all settings in the configuration dialog to the default values.	
<b>Save</b>	Saving all settings made in the configuration dialog <b>netX Driver &gt; Save TCP/IP Connection</b> , i. e. only for the selected connection type.	
<b>Save All</b>	Saving all settings made in the configuration dialog <b>netX Driver</b> , i. e. for all connection types.	

Table 12: Parameters netX Driver &gt; TCP Connection



**Note:** Do not use large IP ranges in combination with a low scan timeout. Microsoft introduced in Windows® XP SP2 a limit of concurrent half-open outbound TCP/IP connections (connection attempts) to slow the spread of virus and malware from system to system. This limit makes it impossible to have more than 10 concurrent half-open outbound connections. Every further connection attempt is put in a queue and forced to wait. Due to this limitation a large IP range used in combination with a low scan timeout could prevent the connection establishment to a device.

### 3.3 Device Assignment

---



**Note:** In the **Device Assignment** you first must assign the CANopen Slave device to the CANopen Slave DTM by checking the check box. This is essential to establish an online connection from the CANopen Slave DTM to the CANopen Slave device later, as described in section *Connecting/Disconnecting Device* on page 84.

---

Therefore in the **Device Assignment** dialog pane you scan for the CANopen Slave device and select it.

If the device did not get a firmware or shall get a new firmware:

1. first you scan for the device/s (with or without firmware) and select it/them,
2. then you download a firmware to the device and
3. subsequently you scan for the device/s (with firmware) once more and select it/them again.

### 3.3.1 Scanning for Devices

1. Select **Settings > Device Assignment** in the navigation area.
- The dialog pane **Device Assignment** is displayed.

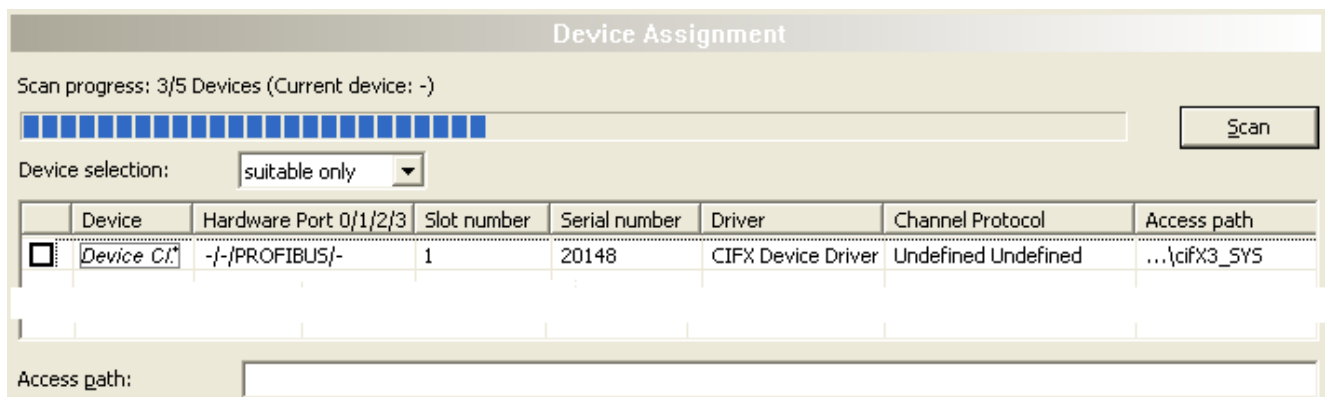


Figure 12: Device Assignment - detected Devices (\* The name of the device class is displayed.) – Example for a device without firmware

2. Select the list box entry **suitable only**.
3. Select the **Scan** button, to start the scanning process.
- Under **Device Selection** all devices are displayed, which can be connected to the CANopen Slave DTM via the preselected driver.



**Note:** For devices, which have been found via the **cifX Device Driver** in the column **Access path** the indication ...\\cifX[0toN]**\_SYS** is displayed. This is correct, as long as a device did not get a firmware. After the firmware download has been completed, in the column **Access path** the indication ...\\cifX[0toN]**\_Ch[0to3]** is displayed.

Parameter	Meaning	Range of Value / Value
Device selection	Selecting <b>suitable only</b> or <b>all</b> devices.	suitable only, all
Device	Device class of the CANopen Master Devices.	
Hardware Port 0/1/2/3	Shows, which hardware is assigned to which communication interface.	
Slot number	Shows the <b>Slot Number (Card ID)</b> preset at the cifX card via the <b>Rotary Switch Slot Number (Card ID)</b> . The indication <b>n/a</b> means, that no <b>Slot-Nummer (Karten-ID)</b> exists. This occurs, if the cifX card is not equipped with a <b>Rotary Switch Slot Number (Card ID)</b> or for cifX cards equipped with a <b>Rotary Switch Slot Number (Card ID)</b> , if the rotary switch is set to the value 0 (zero).	1 to 9, n/a
Serial number	Serial number of the device	
Driver	Name of the driver	
Channel Protocol	Shows, which firmware is loaded to which device channel. The data for the used channel consists of the protocol class and the communication class. a.) For devices without firmware: Undefined Undefined, b.) For devices with firmware: Protocol name corresponding to the used Firmware	
Access path (under Device selection last column on the right)	Depending on the used driver in the column <b>Access path</b> different data to the device are displayed. For the <b>cifX Device Driver</b> the following data are displayed: a.) For devices without firmware: ...\ <b>cifX[0toN]_SYS</b> , b.) For devices with firmware: ...\ <b>cifX[0toN]_Ch[0to3]</b> . <b>cifX[0toN]</b> = Board number 0 to N <b>Ch[0to3]</b> = Channel number 0 to 3	Depending on the device and on the driver: board or channel number, IP address or COM interface
Access path (at the lower side of the dialog pane)	If under <b>Device selection</b> the check box for a device is checked, under <b>Access path</b> (at the lower side of the dialog pane) the driver identification or depending on the used driver additional data to the device. For the <b>cifX Device Driver</b> the following data are displayed: a.) For devices without firmware: ...\ <b>cifX[0toN]_SYS</b> , b.) For devices with firmware: ...\ <b>cifX[0toN]_Ch[0to3]</b> . <b>cifX[0toN]</b> = Board number 0 to N <b>Ch[0to3]</b> = Channel number 0 to 3	driver identification (ID) depending on the device and on the driver: board or channel number, IP address or COM interface

Table 13: Parameters of the Device Assignment

### 3.3.1.1 Scanning for all Devices or for suitable only

#### all

1. In the **Device Assignment** pane select the list box entry **all**.
2. Select the **Scan** button, to start the scanning process.

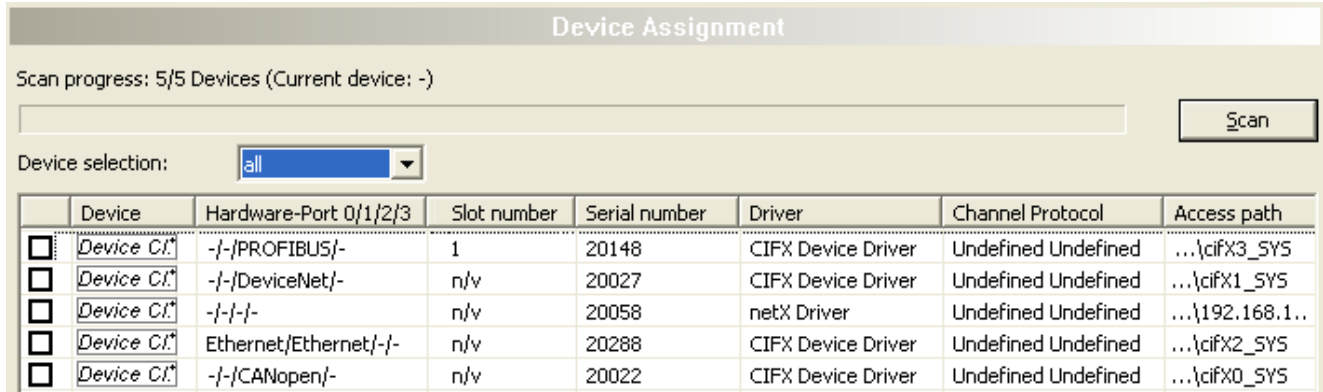


Figure 13: Device Assignment - detected Devices (\* The name of the device class is displayed.) Example for Devices without Firmware

- ↗ Under **Device Selection** all devices are displayed, which are attainable in the network and which can be connected to a single DTM each via the preselected drivers.



**Note:** During a subsequent firmware download in the selection window **Select Firmware File** all files from the selected folder are displayed, under **Files of Type** „All Files (\*.\*)“ is displayed and the check box **Validate the selected firmware file.** is unchecked.

#### suitable only

1. In the **Device Assignment** pane select the list box entry **suitable only**.
  2. Select the **Scan** button, to start the scanning process.
- ↗ Under **Device Selection** all devices are displayed, which can be connected to the CANopen Slave DTM via the preselected drivers.



**Note:** During a subsequent firmware download in the selection window **Select Firmware File** only firmware files from the selected folder are displayed, under **Files of Type** „Firmware Files (\*.nxm)“ or „Firmware Files (\*.nxf)“ is displayed and the check box **Validate the selected firmware file.** is checked.

### 3.3.2 Selecting the Device (with or without firmware)



**Note:** A connection with the CANopen Slave DTM can only be established with one CANopen Slave device.

To select the physical CANopen Slave device (with or without firmware):

1. In the **Device Assignment** pane in the selection list **Device selection** check the check box for the device.

	Device	Hardware Port 0/1/2/3	Slot number	Serial number	Driver	Channel Protocol	Access path
<input checked="" type="checkbox"/>	Device Cl*	-/-/PROFIBUS/-	1	20148	CIFX Device Driver	PROFIBUS Master	... \cifx3_SYS

Access path: {368BEC5B-0E92-4C0E-B4A9-64F62AE7AAFA}\cifx3\_SYS

Figure 14: Device Assignment - Selecting the Device (\* The name of the device class is displayed.) – Example for a device without firmware / one Device is selected

⇒ Under **Access path** (below in the dialog pane) the access path to the device, e. g. the driver identification, or depending on the used driver additional access data of the device are displayed.

2. Select the **Apply** button, to apply the selection.



**Note:** Before an online connection from the CANopen Slave DTM to the CANopen Slave device can be established, a firmware must be loaded to the device and the device must be selected once more.



For further information refer to section *Firmware Download* on page 42 or to section *Selecting the Device once more (with Firmware)* on page 40.

### 3.3.3 Selecting the Device once more (with Firmware)

To select the CANopen Slave device (with firmware) once more, proceed as described hereafter:

#### all

1. In the pane **Device Assignment** select the list box entry **all**.
2. Select the button **Scan**.
- ↗ Under **Device Selection** all devices are displayed, which are attainable in the network and which can be connected to a DTM via the preselected drivers.
3. In the **Device Assignment** pane in the selection list **Device selection** check the check box for the device.

Device Assignment

Scan progress: 5/5 Devices (Current device: -)

Device selection: **all**

	Device	Hardware-Port 0/1/2/3	Slot number	Serial number	Driver	Channel Protocol	Access path
<input checked="" type="checkbox"/>	Device Cl.*	-{/PROFIBUS/-	1	20148	CIFX Device Driver	PROFIBUS-DP Master	...\cifX3_Ch0
<input type="checkbox"/>	Device Cl.*	-{/DeviceNet/-	n/v	20027	CIFX Device Driver	DeviceNet Master	...\cifX1_Ch0
<input type="checkbox"/>	Device Cl.*	-/-/-	n/v	20058	netX Driver	Undefined Undefined	...\192.168....
<input type="checkbox"/>	Device Cl.*	Ethernet/Ethernet/-/-	n/v	20288	CIFX Device Driver	PROFINET IO Device	...\cifX2_Ch0
<input type="checkbox"/>	Device Cl.*	-{/CANopen/-	n/v	20022	CIFX Device Driver	Undefined Undefined	...\cifX0_SYS

Access path: {368BEC5B-0E92-4C0E-B4A9-64F62AE7AAFA}\cifX3\_Ch0

Figure 15: Device Assignment - Selecting the Device (\* The name of the device class is displayed.) – Example for Devices with and without Firmware / one Device is selected



**Note:** After the firmware download has been completed, for the devices which have been detected via the **cifX Device Driver** the following data are displayed:

- In the column **Channel Protocol**: the data for the firmware for the used channel
- In the column **Access path** or under **Access path** (below in the dialog pane): the data: ...cifX[0toN]\_Ch[0to3].  
cifX[0toN] = board number 0 to N  
Ch[0to3] = channel number 0 to 3

4. Select the **Apply** button, to apply the selection.
5. Or select the **OK** button, to apply the selection and to close the DTM interface dialog.
6. Connect the DTM to the device using the context menu (right mouse click).

Or:

### **suitable only**

1. In the pane **Device Assignment** select the list box entry **suitable only**.
2. Select the button **Scan**.
- ↗ Under **Device Selection** all devices are displayed, which can be connected to the CANopen Slave DTM via the preselected drivers.
3. In the **Device Assignment** pane in the selection list **Device selection** check the check box for the device.

	Device	Hardware Port 0/1/2/3	Slot number	Serial number	Driver	Channel Protocol	Access path
<input checked="" type="checkbox"/>	Device Cl.	-/PROFIBUS-	1	20148	CIFX Device Driver	PROFIBUS-DP Master	...\\cifX3_Ch0

Figure 16: Device Assignment - Selecting the Device (\* The name of the device class is displayed.) – Example for a device with firmware / one Device is selected



**Note:** After the firmware download has been completed, for the devices which have been detected via the **cifX Device Driver** the following data are displayed:

- In the column **Channel Protocol**: the data for the firmware for the used channel
- In the column **Access path** or under **Access path** (below in the dialog pane):  
the data: ...\\cifX[0toN]\_Ch[0to3].  
cifX[0toN] = board number 0 to N  
Ch[0to3] = channel number 0 to 3

4. Select the **Apply** button, to apply the selection.
5. Or select the **OK** button, to apply the selection and to close the DTM interface dialog.
6. Connect the DTM to the device using the context menu (right mouse click).



For further information how to establish an online connection from the CANopen Slave DTM to the CANopen Slave device, refer to section *Connecting/Disconnecting Device* on page 84.

## 3.4 Firmware Download

Using the **Firmware Download** dialog a firmware can be transferred to the device.



**CAUTION!** The firmware download

- Stops the bus communication,
- erases the configuration data base and
- overwrites the existing firmware in the device.

To complete the update and to make the device operable again, please re-download the configuration when this operation has finished.



**Note:** Prior to the firmware download, you must select the driver and the Slave device (with or without firmware).



For further information refer to section *Overview Settings* on page 24.

To load the firmware to the device:

1. In the navigation area select **Settings > Firmware Download**.  
 The dialog **Firmware-Download** pane is displayed.

Figure 17: Firmware Download

Parameter	Meaning
Name	Name of the Firmware
Version	Version and Build Version of the Firmware
Browse...	Button to select the firmware file.
Download	Button to download the firmware.

Table 14: Parameter Firmware Download

2. Select the button **Browse**.

 The selection window **Select Firmware File** is displayed.

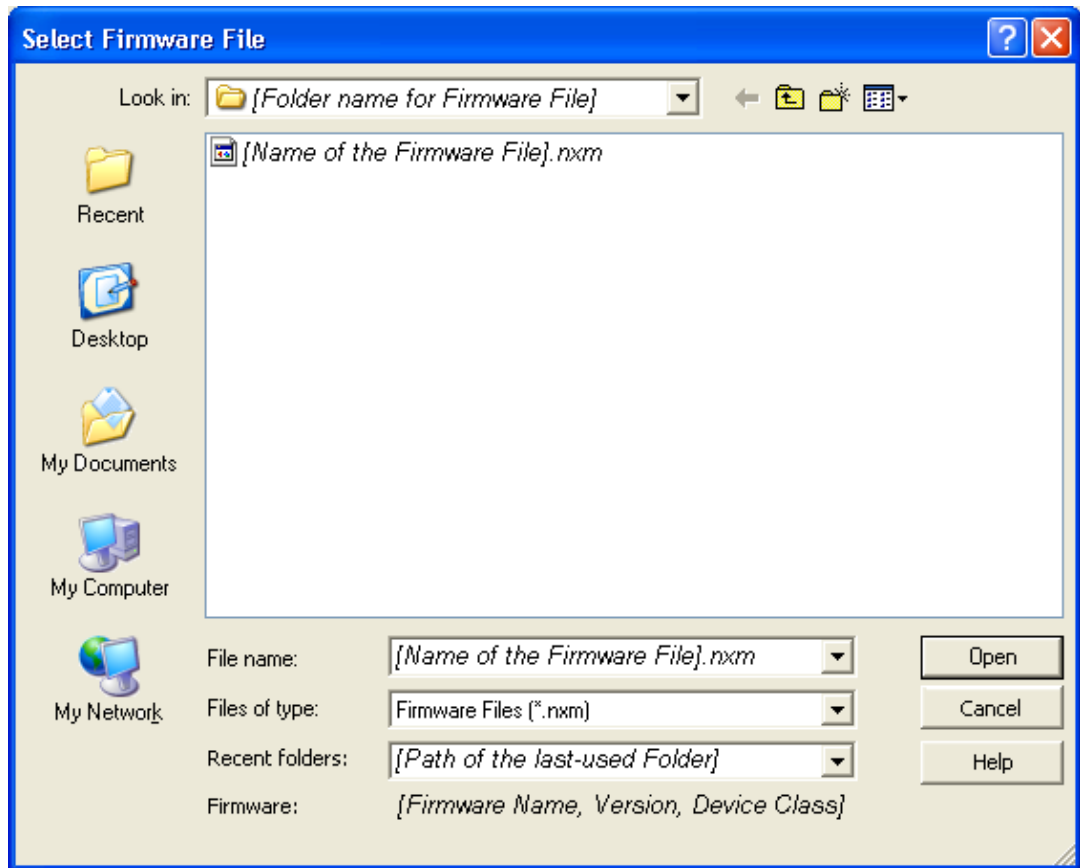


Figure 18: Selection Window Select Firmware File

Parameter	Meaning	Range of Value / Value
Files of Type	„All Files (*.*)“, if before in the <b>Device Assignment</b> pane the list box entry <b>all</b> has been selected. „Firmware Files (*.nxm)“ or Firmware Files (*.nxf), if before in the <b>Device Assignment</b> pane the list box entry <b>suitable only</b> has been selected.	All Files (*.*), Firmware Files (*.nxm), Firmware Files (*.nxf)
Recent folder	Path of the recently opened folder	
Firmware	As soon as the firmware file has been selected, under <b>Firmware</b> the name, the version and the build version as well as the device class for the selected firmware is displayed.	Name, Version, Build Version, Device Class for the selected firmware
Help	Button, to open the online help of the DTM.	

Table 15: Parameters Select Firmware File



Further descriptions to the selection window **Select Firmware File** are included in the context sensitive help (**F1** key) of the Microsoft Corporation.



**Note:** After in the **Device Assignment** pane the list box entry **all/suitable only** has been set, during a subsequent firmware download in the selection window **Select Firmware File** the following data are displayed or set:

(for list box entry →)	all	suitable only
In the selection window <b>Select Firmware File</b> :	all files from the selected folder	only firmware files from the selected folder
Under <b>Files of Type*</b> :	„All Files (*.*)“	„Firmware Files (*.nxm)“, „Firmware Files (*.nxf)“
Validation:	There is a restricted validation, if the selected firmware is applied for the download.	A validation is made, if the firmware file is suitable for the CANopen Slave DTM.

\*These settings in the selection window **Select Firmware File** can also be changed manually.

3. In the selection window mark the firmware file to be loaded using the mouse.
- ↗ In the selection window under **Firmware** the name and the version of the firmware are displayed.
4. In the selection window select the **Open** button.
- ↗ A validation is made, if the selected firmware file is suitable for the CANopen Slave DTM.
- ↗ In the dialog pane **Firmware Download** during the validation the button **Download** is grayed out.
- ↗ The selection window is closed.
5. In the dialog pane **Firmware Download** click to the **Download** button, to download the firmware.

Firmware is not valid for the selected device

- ↗ If a firmware file is selected, which is not valid for the assigned device, the request **Select Firmware File** is displayed.



Figure 19: Request Select Firmware File - Example

- Answer to the request with **No**.

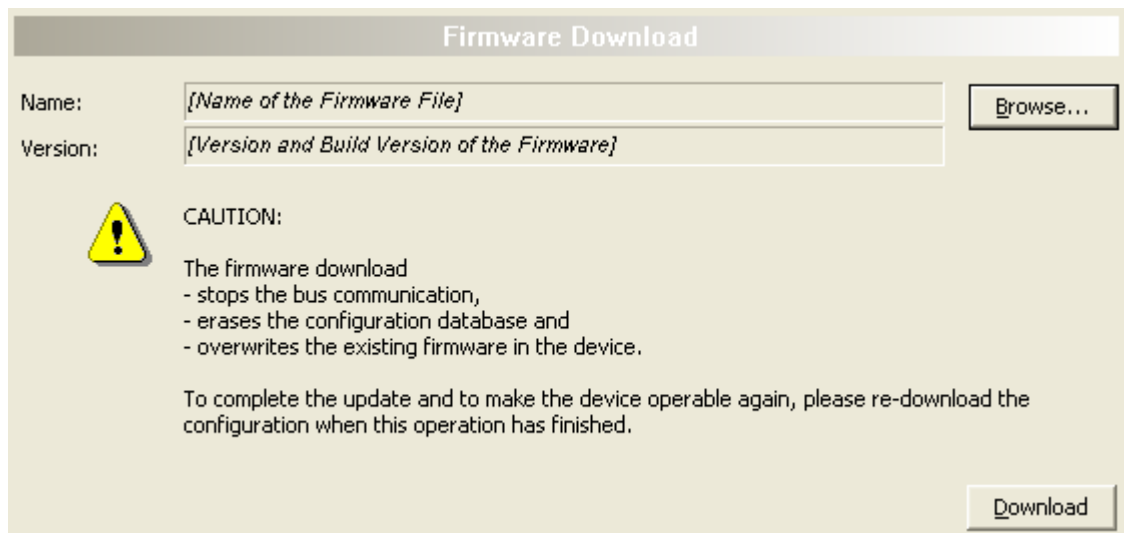


Figure 20: Firmware Download – Download

⇒ The request **Do you really want to download the firmware?** is displayed.

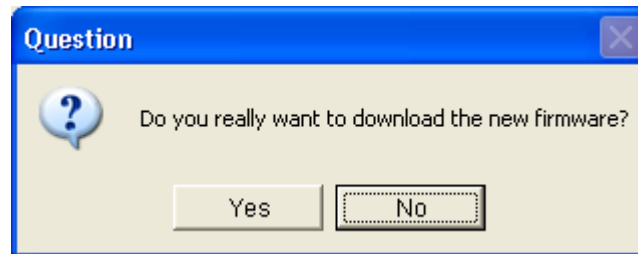


Figure 21: Firmware-Download – Question)

4. Select the **Yes** button.

- During the download a progress bar is displayed, in the status line a clock / green hook symbol is displayed and the button **Download** is grayed out.

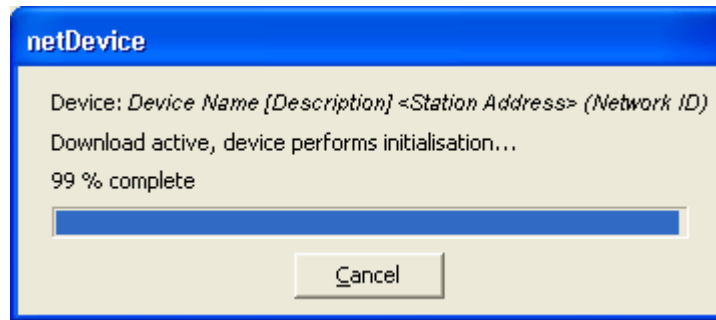


Figure 22: Clock Symbol and Hook Symbol green

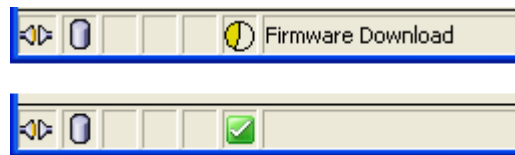


Figure 23: Firmware Download - Progress Bar

- If necessary, cancel the download via **Cancel**.

## 4 Configuration

### 4.1 Overview Configuration

#### Dialog Panes “Configuration”

The table below gives an overview for the **Configuration** dialog panes descriptions:

Section	Subsection	Page
<i>Configuration</i>	<i>Overview Configuration</i>	47
	<i>General Settings</i>	48
	<i>Special Function Objects</i>	50
	<i>Object Dictionary</i>	52
	<i>Process Data Objects</i>	56
	<i>PDO Properties</i>	56
	<i>PDO Mapping</i>	59
	<i>Address Table</i>	61

Table 16: Dialog Panes Configuration

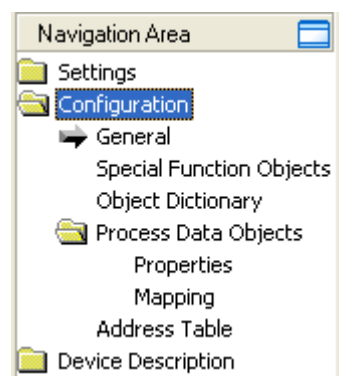


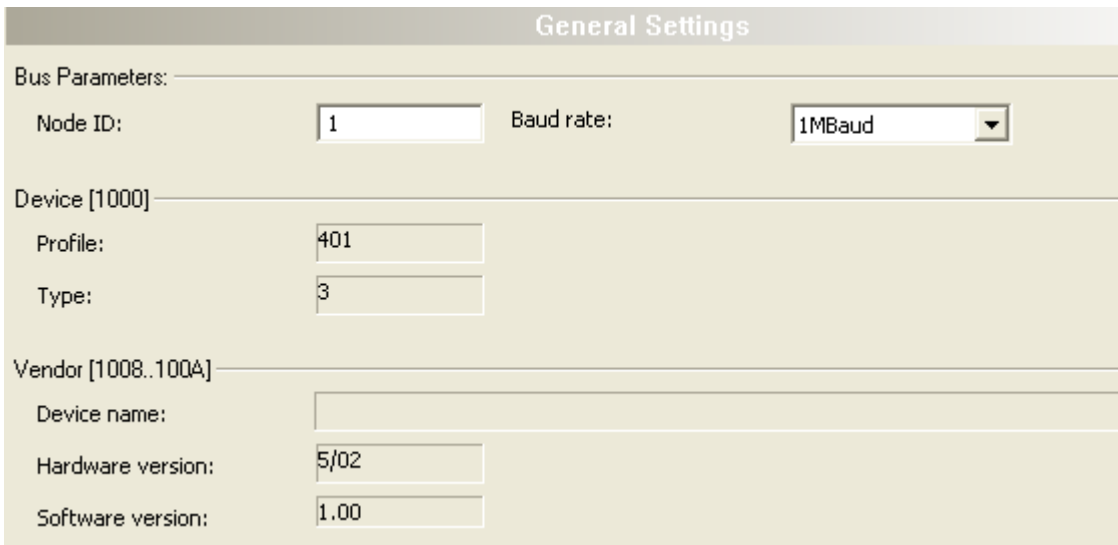
Figure 24: Navigation Area - Configuration



Notice the descriptions in the section *Overview Configuration Steps* on page 18.

## 4.2 General Settings

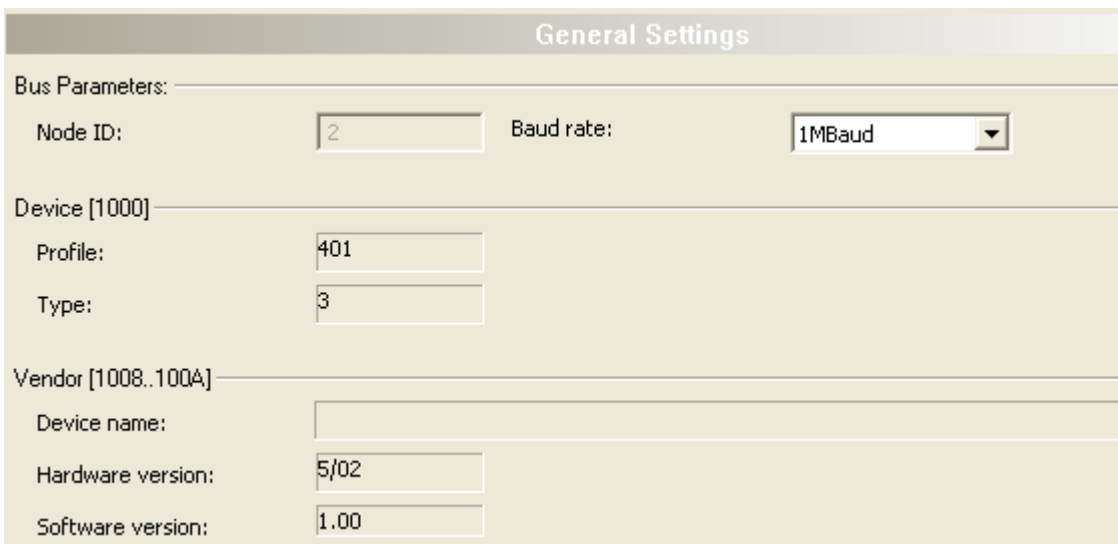
The dialog **General Settings** displays EDS file data:



The screenshot shows a dialog box titled "General Settings" with a light beige background. It is divided into three sections: "Bus Parameters:", "Device [1000]", and "Vendor [1008..100A]".

- Bus Parameters:** Node ID is set to "1" in a text box. Baud rate is set to "1MBaud" in a dropdown menu.
- Device [1000]:** Profile is set to "401" in a text box. Type is set to "3" in a text box.
- Vendor [1008..100A]:** Device name is an empty text box. Hardware version is set to "5/02" in a text box. Software version is set to "1.00" in a text box.

Figure 25: General Settings (Slave DTM at the Root-Line (Stand-Alone Slave))



The screenshot shows a dialog box titled "General Settings" with a light beige background. It is divided into three sections: "Bus Parameters:", "Device [1000]", and "Vendor [1008..100A]".

- Bus Parameters:** Node ID is set to "2" in a text box. Baud rate is set to "1MBaud" in a dropdown menu.
- Device [1000]:** Profile is set to "401" in a text box. Type is set to "3" in a text box.
- Vendor [1008..100A]:** Device name is an empty text box. Hardware version is set to "5/02" in a text box. Software version is set to "1.00" in a text box.

Figure 26: General Settings (Slave DTM at the Master busline)

Parameter	Meaning	Range of Values / Value
<b>Node ID</b>	<p>The <b>Node ID (address)</b> is required to address the device at the bus and must be unique within the CANopen network. Therefore it is not allowed to use this number twice in the same network and must match with the set Node address of the device. Otherwise it is not possible for the Master to build up a communication to this device.</p> <p>Slave DTM at the Root-Line (Stand-Alone Slave): The Node ID is set in the Slave DTM.</p> <p>Slave DTM at the Master busline): The Node ID is set in the Master DTM.</p>	1 ... 127
<b>Baudrate</b>	Baud rate of CANopen connection	1 Mbaud, 800 Kbaud, 500 Kbaud, 250 Kbaud, 125 Kbaud, 100 Kbaud, 50 Kbaud, 20 Kbaud, 10 Kbaud, Default: 1 MBaud
<b>Device Profile and Device Type</b>	<p>Because of the information of the <b>Device Profile</b> and the <b>Device Type</b> during start of communication, the Master can read out the Object 1000H from the Node and compare it with these data.</p> <p>Each CANopen Node has a mandatory Object 1000H, which must be present in the object directory. This object is named <b>Device Type</b>. The <b>Device Type</b> also includes the information about the <b>Device Profile</b>.</p> <p>The Master reads out the Object 1000H from the Node when starting up the CANopen bus and compares the entries, which are made in the two available fields <b>Device Profile</b> and <b>Device Type</b>. If the <b>Device Profile</b> and the <b>Device Type</b> do not match, the Master reports a parameterization error and does not establish a process data transfer to the Node. The verification can be also deactivated.</p>	Value read out from the EDS file
<b>Device Name, Hard and Software Version</b>	Displayed manufacturer data read out from the EDS file.	Value read out from the EDS file

Table 17: General Settings

### 4.3 Special Function Objects

The dialogue **Special Function Objects** displays data of the synchronization, time stamp and emergency telegram, which are adjusted in the Master DTM and can not be edited here. These parameters can be changed via the CANopen Master DTM.

Figure 27: Special Function Objects - Synchronization Telegram

Parameter	Meaning
<b>Synchroni- zation Message</b>	A PDO in CANopen can be configured in Event Driven mode or Cyclic Transmission. Both kinds of transmission types can be synchronized to a special <b>synchronization message</b> which is sent by the master in defined time intervals.
<b>SYNC COB- ID [1005]</b>	The <b>SYNC COB-ID</b> is assigned by the master and cannot be edited here. It can be changed only by the CANopen Master DTM.  The <b>SYNC COB ID</b> specifies the Identifier of the synchronization telegram. If the communication cycle period is not equal to zero, the transmission of the SYNC telegram is activated.  If the field <b>Device generates SYNC message</b> is marked, the device produces the synchronisation message.  If the field is <b>29-bit</b> is marked, the 29-bit CAN ID of the CAN ID extended frame is valid.
<b>Communi- cation Cycle Period [1006]</b>	The <b>Communication Cycle Period</b> is assigned by the Master and cannot be edited here. It can be changed only by the CANopen Master DTM.  The <b>Communication Cycle Period</b> specifies the time for the interval for the transmission the SYNC telegram.
<b>Synchronous Window Length [1007]</b>	The <b>Synchronous Window Length</b> is assigned by the master and cannot be edited here. It can be changed only by the CANopen Master DTM.  The <b>Synchronous Window Length</b> specifies the length of the time window for synchronous PDO (process data objects).

Table 18: Special Function Objects - Synchronization Telegram

TimeStamp Message

TIME COB ID [1012]:   Device consumes TIME message  
 Device produces TIME message  
 29-bit

Figure 28: Special Function Objects - Time Stamp Message

Parameter	Meaning
<b>Time Stamp Message</b>	For transmission of time data.
<b>TIME COB-ID [1012]</b>	<p>The <b>TIME COB ID</b> is assigned by the Master and cannot be edited here. It can be changed only by the CANopen Master DTM.</p> <p>The <b>TIME COB ID</b> specifies the COB ID of the time stamp object.</p> <p>If the field <b>Device consumes TIME message</b> is marked, the device consumes the time stamp message.</p> <p>If the field <b>Device produces TIME message</b> is marked, the device produces the time stamp message.</p> <p>If the field is <b>29-bit</b> is marked, the 29-bit CAN ID of the CAN ID extended frame is valid.</p>

Table 19: Special Function Objects - Time Stamp Message

Emergency Message

EMCY COB ID [1014]:   EMCY exists  
 29-bit

Figure 29: Special Function Objects - Emergency Message

Parameter	Meaning
<b>Emergency-Telegram</b>	Emergency telegrams are sent by the Node when a node internal event occurs.
<b>EMCY COB-ID [1014]</b>	<p>If <b>EMCY COB ID</b> is activated, the CANopen Master can receive EMCY (Emergency) telegrams.</p> <p>The <b>EMCY COB ID</b> is assigned by the Master and cannot be edited here. It can be changed only by the CANopen Master DTM. The <b>EMCY COB ID</b> specifies the COB ID of the Emergency telegram.</p> <p>If the field is <b>29-bit</b> is marked, the 29-bit CAN ID of the CAN ID extended frame is valid.</p>

Table 20: Special Function Objects - Emergency Message

## 4.4 Object Dictionary

The dialog **Object Dictionary** represents the object dictionary of the device. The display shows data read out from the EDS file.

By means of filters in the table **Object Configuration** lists with configured and/or not configured objects can be displayed. A search function allows searching for a special object within the lists.

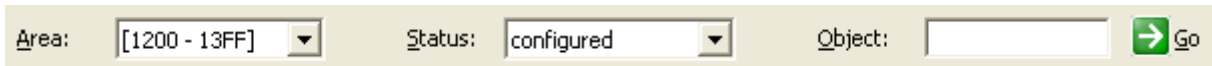


Figure 30: Object Dictionary - Filter Object Configuration

Parameter	Meaning	Range of Values / Value
<b>Area</b>	Via <b>Area</b> a filtered object dictionary area can be selected, which is displayed in the table <b>Object Configuration</b> . If <b>All</b> is selected, in the table <b>Object Configuration</b> all objects are displayed, which are defined in the EDS file.	<b>All</b> 0x1000 ... 0x11FF 0x1200 ... 0x 13FF 0x1400 ... 0x1FFF 0x2000 ... 0x5FFF 0x6000 ... 0x9FFF 0xA000 ... 0xFFFF
<b>Status</b>	Via <b>Status</b> it is possible to specify whether in the table <b>Object Configuration</b> all objects, only the configured objects or only the not configured objects of the selected range are to be displayed. Only the objects configured are relevant for data exchange.	<b>All</b> configured not configured
<b>Object</b>	In the searching field <b>Object</b> the object index and/or the object index and subindex of a certain object can be entered. If the arrow button -> <b>Go</b> is clicked, the searched object (if available) is displayed in the upper line of the table <b>Object Configuration</b> . To enter an object index with subindex a dot is used. Example: 1400.01	Min: 0000 Max: FFFF

Table 21: Object Dictionary - Filter Object Configuration

The objects read out from the EDS file are displayed in the table **Object Configuration**. For better readability for objects with subindex a heading (object index without subindex) is displayed.

Object configuration:

Configure	Index.Subindex	Name	Access
<input type="checkbox"/>	0x1000	MKr Device Type	RO
<input type="checkbox"/>	0x1001	MKr Error Register	RO
<input type="checkbox"/>	0x100B	Node-ID	RO
<input checked="" type="checkbox"/>	0x100C	Guard Time	RW
<input checked="" type="checkbox"/>	0x100D	Life Time Factor	RW
<input checked="" type="checkbox"/>	0x100E	COB-ID Guarding-Protocol	RW
<input checked="" type="checkbox"/>	0x1014	COB-ID Emergency Message	RW
<b>0x1016</b>		<b>Consumer Heartbeat Time</b>	
<input type="checkbox"/>	0x1016.00	Number of entries	RO
<input checked="" type="checkbox"/>	0x1016.01	MKr Consumer Heartbeat Time 1	RW
<input checked="" type="checkbox"/>	0x1016.02	MKr Consumer Heartbeat Time 2	RW
<input checked="" type="checkbox"/>	0x1016.03	Consumer Heartbeat Time 3	RW
<input checked="" type="checkbox"/>	0x1016.04	Consumer Heartbeat Time 4	RW
<input checked="" type="checkbox"/>	0x1016.05	Consumer Heartbeat Time 5	RW
<input checked="" type="checkbox"/>	0x1016.06	Consumer Heartbeat Time 6	RW
<input checked="" type="checkbox"/>	0x1016.07	Consumer Heartbeat Time 7	RW
<input checked="" type="checkbox"/>	0x1016.08	Consumer Heartbeat Time 8	RW

Figure 31: Object Dictionary - Object Configuration

The table **Object Configuration** the following parameter data are provided.




Parameter	Meaning	Range of Values / Value	
<b>Configure</b>	The Objects activated in the configuration are checked. The Objects which are not configured are unchecked.	configured (checked), not configured (unchecked)	
	Symbol / Checkbox	Description	
		The objects marked with a key symbol can not be enabled or disabled for the configuration in the <b>Object Directory</b> dialog, but they can be added or removed from the configuration elsewhere in the user interface.	
	 <input checked="" type="checkbox"/>	Objects activated in the configuration	configured (checkbox locked)
	 <input type="checkbox"/>	Objects which are <u>not</u> activated in the configuration	<u>not</u> configured (checkbox locked)
	<input checked="" type="checkbox"/>	Objects activated in the configuration	configured
	<input type="checkbox"/>	Objects which are <u>not</u> activated in the configuration	<u>not</u> configured
<b>Index.Sub-index</b>	All objects are addressed in the object index and the corresponding subindex, which are defined by the EDS file.	Object index 0x1000 ... 0xFFFF Sub index 0x00 ... 0xFF	
<b>Name</b>	Symbolic name of the object, which is defined by the EDS file.	From EDS file	
<b>Access</b>	Gives the access right of the object, which is defined by the EDS file.	RO = read only (read) RW = read, write (read, write) WO = write only (write) CONST = constant	

Table 22: Object Dictionary - Object Configuration

If in the table **Object Configuration** a line is clicked by the cursor, the selected object, the current value, the default value, the data type, the minimum and maximum value are displayed in the fields below the table. By **Display mode** data display can be chosen in decimal or in hexadecimal mode.

Selected object:	01000 MKr Device Type		
Display mode:	Decimal	Data type:	UNSIGNED32
Current value:	0	Min:	0
Default:	0	Max:	4294967295

Figure 32: Object Dictionary - Data selected Object

Parameter	Meaning	Range of Value
<b>Selected Object</b>	In the display field <b>Selected Object</b> the object index, the subindex and the name of the selected object are indicated. These data are defined by the EDS file.	
<b>Display Mode</b>	By selection of the <b>Display Mode</b> decimal and/or hexadecimal from the list field the values are displayed in decimal and/or hexadecimal mode.	<u>Hexadecimal</u> Decimal
<b>Current Value</b>	In the input field <b>Current Value</b> a value can be assigned to the selected object.	
<b>Default, Data Type, Min/Max</b>	In the display fields <b>Default</b> , <b>Data Type</b> and/or <b>Min/Max</b> the default value defined in the EDS file, the data type and/or the minimum and maximum value for the object is indicated. The values <b>Min.</b> and <b>Max.</b> are displayed in decimal mode by default.	

Table 23: Object Dictionary - Data selected Object

## 4.5 Process Data Objects

### 4.5.1 PDO Properties

In the dialog **PDO Properties** the transmit and the receive PDOs are displayed. By the list field **PDO Type** the display can be changed from transmit PDOs (TPDO) to receive PDOs (RPDO) and vice versa.



Figure 33: Process Data Objects > PDO Properties - PDO Type

The PDOs supported by the node are read out of the EDS file and displayed in the table **PDO Configuration**. The PDOs to be used for the communication can be specified i.e. configured in this window.

PDO configuration:

Configure	Index	PDO name
<input checked="" type="checkbox"/>	1400	RxPDO1 parameter
<input checked="" type="checkbox"/>	1401	RxPDO2 parameter
<input checked="" type="checkbox"/>	1402	RxPDO3 parameter
<input checked="" type="checkbox"/>	1403	RxPDO4 parameter
<input type="checkbox"/>	1404	RxPDO5 parameter
<input type="checkbox"/>	1405	RxPDO6 parameter
<input type="checkbox"/>	1406	RxPDO7 parameter
<input type="checkbox"/>	1407	RxPDO8 parameter
<input type="checkbox"/>	1408	RxPDO9 parameter
<input type="checkbox"/>	1409	RxPDO10 parameter
<input type="checkbox"/>	140A	RxPDO11 parameter
<input type="checkbox"/>	140B	RxPDO12 parameter
<input type="checkbox"/>	140C	RxPDO13 parameter
<input type="checkbox"/>	140D	RxPDO14 parameter
<input type="checkbox"/>	140E	RxPDO15 parameter
<input type="checkbox"/>	...	...

Figure 34: Process Data Objects > PDO Properties - PDO Configuration

Parameter	Meaning	Range of Values / Value												
<b>PDO Type</b>	Filter function for the table <b>PDO Configuration</b> as - Transmit PDOs - <b>TPDO</b> = Transmit PDO or as - Receive PDOs - <b>RPDO</b> = Receive PDO.  Transmit PDOs are sent by the node and are input data of the Master.  Receive PDOs are output data of the Master and are received from the node.	TPDO RPDO												
<b>Configure</b>	By activating/configuring of a PDO the PDO is used for the communication. The corresponding parameter values are part of the master configuration. During initialization the master transfers these parameters automatically into the node (default behavior).  <i><b>Note:</b> The transmission of the parameters during the initialization phase can also be deactivated and/or become deactivated. I.e., the node uses parameters, which can be different from the parameters set here.</i>	configured (checked), <u>not</u> configured (unchecked)												
	<table border="1"> <thead> <tr> <th>Checkbox</th> <th>Beschreibung</th> <th></th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>PDOs activated in the configuration The PDO is used for the communication.</td> <td>configured (checked) (checkbox locked)</td> </tr> <tr> <td><input checked="" type="checkbox"/></td> <td>PDOs activated in the configuration The PDO is used for the communication.</td> <td>configured (checked)</td> </tr> <tr> <td><input type="checkbox"/></td> <td>PDOs which are <u>not</u> activated in the configuration The PDO is <u>not</u> used for the communication.</td> <td><u>not</u> configured (unchecked)</td> </tr> </tbody> </table>	Checkbox	Beschreibung		<input type="checkbox"/>	PDOs activated in the configuration The PDO is used for the communication.	configured (checked) (checkbox locked)	<input checked="" type="checkbox"/>	PDOs activated in the configuration The PDO is used for the communication.	configured (checked)	<input type="checkbox"/>	PDOs which are <u>not</u> activated in the configuration The PDO is <u>not</u> used for the communication.	<u>not</u> configured (unchecked)	
	Checkbox	Beschreibung												
	<input type="checkbox"/>	PDOs activated in the configuration The PDO is used for the communication.	configured (checked) (checkbox locked)											
<input checked="" type="checkbox"/>	PDOs activated in the configuration The PDO is used for the communication.	configured (checked)												
<input type="checkbox"/>	PDOs which are <u>not</u> activated in the configuration The PDO is <u>not</u> used for the communication.	<u>not</u> configured (unchecked)												
<b>Index</b>	Object Index of the Process data object (PDO)	0x1400 ... 0x15FF 0x1800 ... 0x19FF												
<b>PDO Name</b>	Here RxPDO name and/or TxPDO name is indicated. These are defined in the EDS file.													

Table 24: Process Data Objects > PDO Properties - PDO Configuration (examine)

Each process data object (PDO) has characteristics. These are displayed below the table.

Figure 35: Process Data Objects > PDO Properties - Data selected Object

Parameter	Meaning	Range of Values / Value
<b>COB-ID</b>	In this field the CAN telegram identifier is displayed.	0 ... 2047
<b>Trans- mission Type</b>	For the transmit and/or receive PDOs different transmission types are possible. For a PDO in CANopen event driven, synchronous or asynchronous transmission can be configured. Transmission types can be synchronized to the synchronization message SYNC for example, which is sent by the master in defined time intervals.  Synchronous means that the transmission of the PDO is related to the SYNC message.  Asynchronous means that the transmission of the PDO is not related to the SYNC message and can be done at any time.  <b>Note:</b> The support of the different transmission types is manufacturer and device dependent. For CANopen the support of individual and/or all transmission types is not required. Whether a device supports the desired transmission type, must be reread and/or examined in the technical manual of the used device, if necessary.	0 ... 255  synchronous acyclic (0) synchronous cyclic (1-240) synchronous RTR (252) asynchronous RTR (253) Event driven, profile specific (254) Event driven, manufacturer specific (255)
<b>Trans- mission Factor</b>	For synchronous TPDOs for the transmission type <b>synchronous cyclic (1-240)</b> another factor is to be set, to which SYNC telegram the data transmission refers.  A transmission factor of 1 means that the message will be transferred with each SYNC telegram. A transmission factor of n means that the message will be transferred with each n-th SYNC telegram. Asynchronous TPDOs are not transferred in a temporal correlation with a SYNC.	
<b>Inhibit Time</b>	The <b>Inhibit Timer</b> (if supported) describes the time interval, which at least must be waited between the transmissions of two equal telegrams. Thus a too frequent transmission of the same telegram is suppressed.	
<b>Event Timer</b>	The <b>Event Timer</b> (if supported) is possible only for TPDO transmission types 254 and 255.  The expiration of the timer is used in the node as event, in order to send the TPDO. Manufacturer and/or device-specifically also an application event can activate the sending of the TPDOs and reset the <b>Event Timer</b> .	TPDO 254, 255

Table 25: Process Data Objects > PDO Properties - Data selected Object

### 4.5.2 PDO Mapping

The dialog **PDO Mapping** permits to map the contents of a PDO.

By use of the **Filter PDO type** field in the table **Mappable Objects** the receive PDOs (RPDO) or alternatively the transmit PDOs (TPDO) can be displayed.

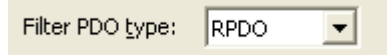


Figure 36: Process Data Objects > PDO Mapping - Filter PDO Type

In the list field **Object Dictionary Area** an object dictionary area can be preselected.



Figure 37: Process Data Objects > PDO Mapping - Object Dictionary Area

The mappable objects appropriate to the preselection are listed in the table **Mappable Objects**.

Index.Subindex	Parameter	Length	Access
<b>02200</b>	<b>Digital Byte Outputs</b>		
02200.01	Output Byte 1	8	WO
02200.02	Output Byte 2	8	WO
02200.03	Output Byte 3	8	WO
02200.04	Output Byte 4	8	WO
02200.05	Output Byte 5	8	WO
02200.06	Output Byte 6	8	WO
02200.07	Output Byte 7	8	WO

Figure 38: Process Data Objects > PDO Mapping - Mappable Objects

Parameter	Meaning	Range of Value
<b>Object Dictionary Area</b>	Object dictionary filter range.	<b>All</b> 0x1000 ... 0x11FF 0x1200 ... 0x 13FF 0x1400 ... 0x1FFF 0x2000 ... 0x5FFF 0x6000 ... 0x9FFF 0xA000 ... 0xFFFF
<b>Mappable Objects</b>	List of the mappable objects.	from EDS file
<b>Index. Subindex</b>	All objects are addressed in the object index and if necessary in the corresponding subindexes, which are defined by the EDS file.	0x1000 ... 0xFFFF as well as 0 ... 0xFF
<b>Parameter</b>	Name of the object from the EDS file.	from EDS file
<b>Length</b>	The length of the PDOs in bytes.	
<b>Access</b>	Gives the access rights of the process data objects, which are defined by the EDS file.	rw = read, write

Table 26: Process Data Objects > PDO Mapping - Mappable Objects

In the list field **PDO Contents Mapping for** the PDO is selected, the PDO contents to be displayed.



Figure 39: Process Data Objects > PDO Mapping - Filter Mapped Objects

The mapped objects appropriate to the preselection are listed in the table **Mapped Objects**.

Mapped objects:

Index	Subindex	Parameter	Length
02200	01	Output Byte 1	8
02200	02	Output Byte 2	8
02200	03	Output Byte 3	8
02200	04	Output Byte 4	8
02200	05	Output Byte 5	8
02200	06	Output Byte 6	8
02200	07	Output Byte 7	8
02200	08	Output Byte 8	8

Figure 40: Process Data Objects > PDO Mapping - Mapped Objects

Parameter	Meaning	Range of Value
<b>PDO Contents Mapping for</b>	The PDO list field <b>PDO Contents Mapping for</b> contains all configured PDOs of the pane PDO Properties. The objects responsible for data exchange (max. 8 byte/PDO) are assigned to the PDOs.	
<b>Filter PDO Type</b>	The mapping is proceeded separately for the RPDOs and/or the TPDOs.	RPDO TPDO
<b>Mapped Objects</b>	The table <b>Mapped Objects</b> contains only configured objects and always corresponds in the PDO list field <b>PDO Contents Mapping for</b> selected PDOs.	
<b>Index. Subindex</b>	All objects are addressed in the object index and if necessary in the corresponding subindexes, which are defined by the EDS file.	0x1000 ... 0xFFFF as well as 0 ... 0xFF
<b>Parameter</b>	Name of the parameter from the EDS file.	from EDS file
<b>Length</b>	The Length specifies the length of the PDOs.	

Table 27: Process Data Objects > PDO Mapping - Mapped Objects

To change the object sequence in the table **Mapped Objects** more easily, shifting buttons are available: move completely above, above, down and completely down.

## 4.6 Address Table

The **Address Table** dialog pane shows a list of all addresses used in the process data image. The displayed addresses refer to the used CANopen.

To configure the address data:

- Select **Configuration > Address Table** in the navigation area.

**Address Table**

Display mode: Hexadecimal

Inputs:

PDO Name	PDO Idx	COB-ID	Obj.Idx Subidx	Obj. Name	Type	Length	Address
RxPDO_01	0x01400		0x02200.01	1 Byte Out (1)	UNSIGNED8	0x0008	0x0000
RxPDO_01	0x01400		0x02200.02	1 Byte Out (2)	UNSIGNED8	0x0008	0x0001
RxPDO_01	0x01400		0x02200.03	1 Byte Out (3)	UNSIGNED8	0x0008	0x0002
RxPDO_01	0x01400		0x02200.04	1 Byte Out (4)	UNSIGNED8	0x0008	0x0003
RxPDO_01	0x01400		0x02200.05	1 Byte Out (5)	UNSIGNED8	0x0008	0x0004

Outputs:

PDO Name	PDO Idx	COB-ID	Obj.Idx Subidx	Obj. Name	Type	Length	Address
TxPDO_01	0x01800		0x02000.01	1 Byte In (1)	UNSIGNED8	0x0008	0x0000
TxPDO_01	0x01800		0x02000.02	1 Byte In (2)	UNSIGNED8	0x0008	0x0001
TxPDO_01	0x01800		0x02000.03	1 Byte In (3)	UNSIGNED8	0x0008	0x0002
TxPDO_01	0x01800		0x02000.04	1 Byte In (4)	UNSIGNED8	0x0008	0x0003
TxPDO_01	0x01800		0x02000.05	1 Byte In (5)	UNSIGNED8	0x0008	0x0004

Figure 41: Configuration > Address Table

### 4.6.1 Description of the Paramters

Parameter	Meaning
PDO-Name	Here RxPDO name and/or TxPDO name is indicated. These are defined in the EDS file.
PDO-Idx	Object Index of the Process data object (PDO)
COB-ID	COB-ID of the CAN telegram
Obj. Idx Subidx	All objects are addressed in the object index and if necessary in the corresponding subindexes, which are defined by the EDS file.
Obj. Name	Object Name
Type	Input data type or output data type
Length	Input data length or output data length of the PDOs in bytes
Address	Output data offset address or input data offset address

Table 28: Address Table Pane Parameters - Inputs / Outputs

## 4.6.2 Display Mode

### Display Mode

- Use the **Display Mode** drop-down list to select data display mode decimal or hexadecimal.

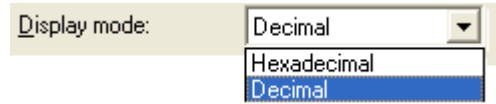


Figure 42: Configuration > Address Table - Display Mode

### Sort Addresses

- To sort the address data by **Name of Station** or by **Address**, click on the respective column header.

## 5 Device Description

### 5.1 About Device Description

#### Dialog Panes “Device Description”

The table below gives an overview for the **Device Description** dialog panes descriptions:

Section	Subsection	Page
<i>Device Description</i>	<i>Device</i>	64
	<i>EDS</i>	64

Table 29: Descriptions of the Dialog Panes Device Description

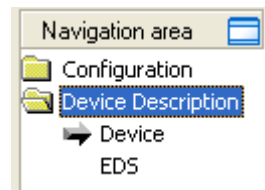


Figure 43: Navigation Area - Device Description

## 5.2 Device

The **Device Info** dialog contains manufacturer information about the device, which is defined in the EDS file. The following information is indicated:

Parameter	Meaning
Vendor name	Vendor name of the device
Product name	Name of the device
Ident. number	Identification number of the device
Revision	Hardware reference

Table 30: General Device Information

## 5.3 EDS

The **EDS Viewer** shows the content of the EDS file in a text view.

Under **Filename** the file directory path and the file name of the displayed EDS file is displayed. **Find what** offers a search feature to search for text contents within the text of the EDS file.

In the EDS Viewer window on the left side, the line number is displayed for simple overview, the further entries show the EDS file in text format.

Parameter	Meaning
Filename	File directory path and the file name of the displayed EDS file.
Find what	Search feature to search for text contents within the text of the EDS file.
Match case	Search option
Match whole word	Search option

Table 31: Device Description – EDS Viewer

## 6 Diagnosis

### 6.1 Overview Diagnosis

The dialog **Diagnosis** serves to diagnose the device behavior and communication errors. For diagnosis the device must reside in online state.

The **Extended Diagnosis** helps to find communication and configuration errors, when default diagnosis fails.



For further information to the **Extended Diagnosis** refer to section *Overview Extended Diagnosis* on page 70.

#### **Dialog Panes “Diagnosis”**

The table below gives an overview for the individual **Diagnosis** dialog panes descriptions:

Section	Page
<i>General Diagnosis</i>	66
<i>Firmware Diagnosis</i>	69

Table 32: Descriptions of the Diagnosis Panes



**Note:** Accessing the **Diagnosis** panes of the CANopen Slave DTM requires an online connection from the CANopen Slave DTM to the CANopen Slave device.



For further information, refer to section *Connecting/Disconnecting Device* on page 84.

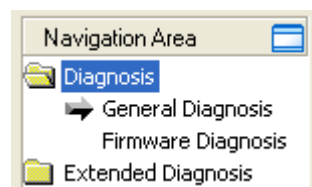


Figure 44: Navigation Area - Diagnosis

## 6.2 General Diagnosis

Information regarding the Device State and other general diagnosis parameters are displayed in the **General Diagnosis** dialog.

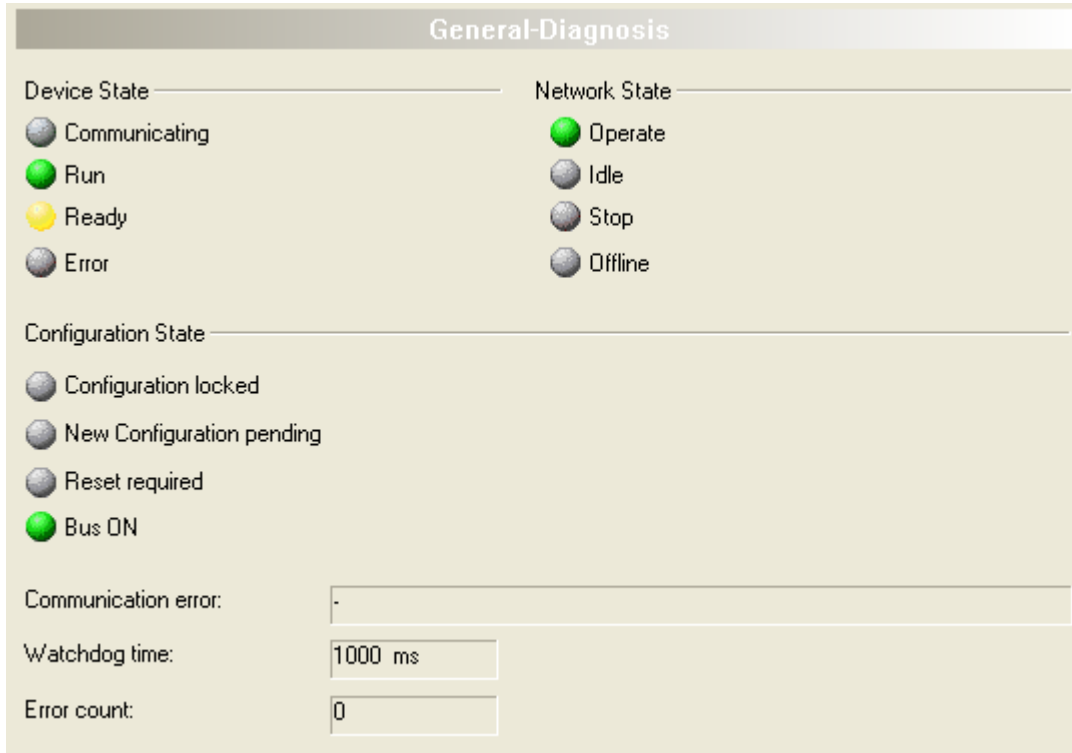


Figure 45: General Diagnosis

























LED	Meaning	Color	State
<b>Device State</b>			
<b>Communicating</b>	Shows whether the CANopen device executes the network communication.	 (green)	COMMUNICATING
		 (gray)	Not COMMUNICATING
<b>Run</b>	Shows whether the CANopen device has been configured correctly.	 (green)	Configuration OK
		 (gray)	Configuration not OK
<b>Ready</b>	Shows whether the CANopen device has been started correctly. The CANopen device waits for a configuration.	 (yellow)	Device READY
		 (gray)	Device not READY
<b>Error</b>	Shows whether the CANopen device records a device status error. For further information about the error characteristics and the number of counted errors refer to the extended diagnosis.	 (red)	ERROR
		 (gray)	No ERROR
<b>Network State</b>			
<b>Operate</b>	Shows whether the CANopen device is in data exchange. In a cyclic data exchange the input data or the output data of the CANopen Slave are transmitted to the CANopen Master.	 (green)	In OPERATION state
		 (gray)	Not in OPERATION state
<b>Idle</b>	Shows whether the CANopen device is in idle state.	 (yellow)	In IDLE state
		 (gray)	Not in IDLE state
<b>Stop</b>	Shows whether the CANopen device is in Stop state: There is no cyclic data exchange at the CANopen network. The CANopen device was stopped by the application program or it changed to the Stop state because of a bus error.	 (red)	In STOP state
		 (gray)	Not in STOP state
<b>Offline</b>	The CANopen Slave is offline as long as it does not have a valid configuration.	 (yellow)	In OFFLINE state
		 (gray)	Not in OFFLINE state
<b>Configuration State</b>			
<b>Configuration locked</b>	Shows whether the CANopen device configuration is locked, to avoid the configuration data are typed over.	 (yellow)	Configuration LOCKED
		 (gray)	Configuration not LOCKED
<b>New Configuration pending</b>	Shows whether a new CANopen device configuration is available.	 (yellow)	New Configuration pending
		 (gray)	No new Configuration pending
<b>Reset required</b>	Shows whether a firmware reset is required as a new CANopen device configuration has been loaded into the device.	 (yellow)	RESET required
		 (gray)	No RESET required
<b>Bus ON</b>	Shows whether the bus communication was started or stopped. I. e., whether the device is active on the bus or no bus communication to the device is possible and no response telegrams are sent.	 (green)	Bus ON
		 (gray)	Bus OFF

Table 33: Indication General Diagnosis

Parameter	Meaning
Communication Error	Shows the message text of the communication error. If the cause of the current error is resolved, „ – “ is displayed.
Watchdog time	Shows the watchdog time in ms.
Error Count	This field holds the total number of errors detected since power-up, respectively after reset. The protocol stack counts all sorts of errors in this field no matter if they were network related or caused internally.

Table 34: Parameter General Diagnosis

### 6.3 Firmware Diagnosis

In the dialog **Firmware Diagnosis** the actual task information of the firmware is displayed.

Under **Firmware** or **Version** the name of the firmware and version (including the date) are indicated.

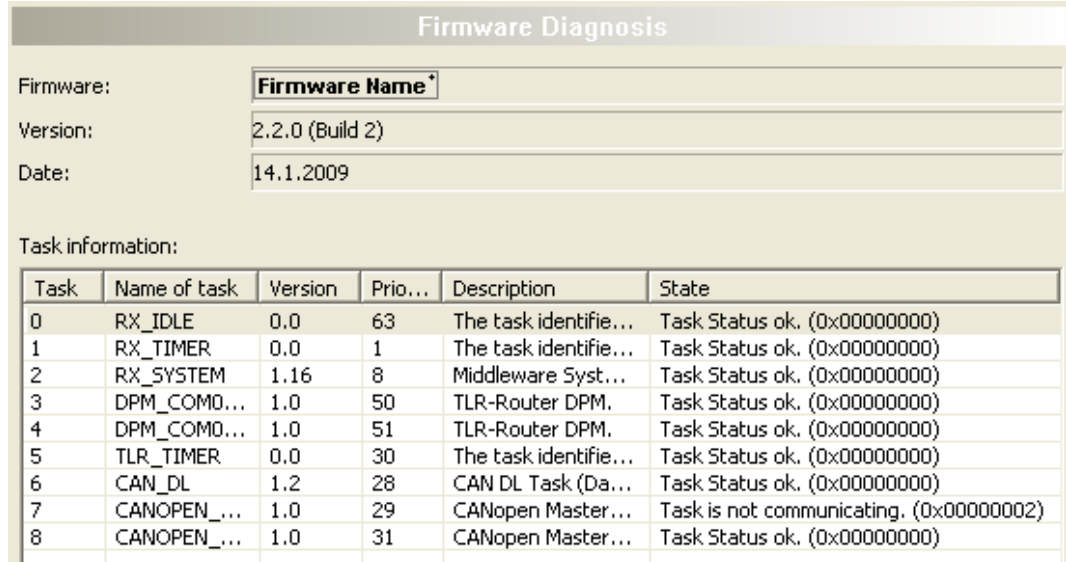


Figure 46: Firmware Diagnosis (\*The name of the Firmware is displayed.)

#### Task Information:

The table **Task Information** is listing the task information of the single firmware tasks.

Column	Meaning
Task	Task number
Name of task	Name of the task
Version	Version of the task
Prio	Priority of the task
Description	Description of the task
Status	Status of the task

Table 35: Description Table Task Information

# 7 Extended Diagnosis

## 7.1 Overview Extended Diagnosis

The **Extended Diagnosis** of the CANopen Slave DTM helps to find communication and configuration errors. Therefore it contains a list of diagnosis structures as online counter, states and parameters.

### Dialog Panes “Extended Diagnosis”

The table below gives an overview for the **Extended Diagnosis** dialog panes descriptions:

Navigationbereich CANopen-Slave	Ordnername im Navigationbereich	Dialogfenster	Handbuchseite	
	<i>RX_SYSTEM</i>	<i>Task Information</i>	71	
		<i>IniBatch Status</i>	72	
	<i>DPM_COMO_SMBX</i>	<i>Task Information</i>	71	
	<i>DPM_COMO_RMBX</i>	<i>Task Information</i>	71	
	<i>CANopen_DL</i>	<i>Task Information</i>	71	
	<i>CANOPEN_SLAVE</i>	<i>Task Information</i>	71	
		<i>Common Diagnostic</i>	73	
		<i>Commands</i>	74	
	<i>CANOPEN_APS</i>	<i>Task Information</i>	71	
		<i>Slave configuration</i>	76	
		<i>Commands</i>	77	

Table 36: Descriptions of the Dialog Panes Extended Diagnosis



**Note:** Accessing the **Extended Diagnosis** dialog panes of the CANopen Slave DTM requires an online connection from the CANopen Slave DTM to the CANopen Slave device.



For further information, refer to section *Connecting/Disconnecting Device* on page 84.

## 7.2 Task Information

Task Information	
Task states	
Name	Value
Identifier	
Major version	
Minor version	<i>[The displayed values depend from the corresponding task]</i>
Maximum Packet size	
Default Que	
Unique identifier	
Init result	

Figure 47: Extended Diagnosis > [Folder Name] > Task Information Example Display

Name	Description
Identifier	Identification number of the task
Major version	Task version, contains incompatible changes
Minor version	Task version, contains compatible changes
Maximum packet size	Maximum packet size, which the task sends
Default Queue	Queue handle, which is accessible via DPM by mailbox.
UUID	Unique user ID, 16 Byte indicator used for task identification and its affiliation e. g. to a stack (therein different identification data are coded in).
Init result	Error Code, 0= no Error  The description of the error codes can be found in this manual or in the corresponding software reference manuals.

Table 37: Extended Diagnosis > [Folder Name] > Task Information

### 7.3 IniBatch Status

IniBatch-Status	
Task-Status	
Name	Value
Communication Channel	0
Current State	Error
IniBatch Result	No DBM file
OpenDbm Result	24975
SendPacket Result	0
Confirmation Result	0
Last Packet Number	0
Last Packet Command	0
Last Packet Length	0
Last Packet Destination	0

Figure 48: Extended Diagnosis > [Folder Name] > IniBatch Status Example Display

Name	Description
Communication Channel	Number of the communication channel used by the device.
Current State	0 =Idle; 1 =IniBatch packets in progress; 2 =Retrying to send last packet; 3 = Error
IniBatch Result	0 = Ok; 1 = No DBM file; 2 = No Packet table; 3 = No data set available; 4 = Data set is shorter than packet length; 5 = Packet Buffer is shorter than Packet length; 6 = Invalid packet destination; 7 = Logical queue not defined 8 = Send packet failed; 9 = Too many retries; 10 = Error in confirmation packet status
OpenDbm Result	Error when opening the IniBatch data base Under "OpenDbm Result" the error code is typed in, when "IniBatch Result" == "No DBM file" (1) is.
SendPacket Result	Error when sending a packet Under "SendPacket Result" the error code is typed in, when "IniBatch Result" == "send packet failed" (8) is.
Confirmation Result	Confirmation error when sending packets Under "Confirmation Result" the packet specific error code from the ulSta is typed in, when "IniBatch Result " == "Error in confirmation packet status" (10) is.
Last Packet Number	Value depends by the communication system.
Last Packet Command	Value depends by the communication system.
Last Packet Length	Value depends by the communication system.
Last Packet Destination	Value depends by the communication system.

Table 38: Extended Diagnosis > [Folder Name] > IniBatch Status

The task status "Confirmation Result" is bus specific. The other task statuses are rcx-related error codes.

### 7.3.1 CANOPEN\_SLAVE

#### 7.3.1.1 Common Diagnostic

Common diagnostic	
Task states	
Name	Value
Last received COB-ID	0
CAN telegrams sent	0
CAN telegrams received	0
Number of detected CAN errors	0
Baudrate	1MBaud

Figure 49: Extended Diagnosis > CANOPEN\_SLAVE > Common Diagnostic

Name	Description										
Last received COB-ID	Last received CAN-Message Header-ID										
CAN telegrams sent	Number of sent CAN-Messages										
CAN telegrams received	Number of received CAN-Messages										
Number of detected CAN errors	Number of detected CAN errors										
Baud rate	Baud rate of CANopen connection										
	Available Baud Rate:										
	<table border="0"> <tr> <td>1 MBaud</td> <td>100 KBaud</td> </tr> <tr> <td>800 KBaud</td> <td>50 KBaud</td> </tr> <tr> <td>500 KBaud</td> <td>20 KBaud</td> </tr> <tr> <td>250 KBaud</td> <td>10 KBaud</td> </tr> <tr> <td>125 KBaud</td> <td></td> </tr> </table>	1 MBaud	100 KBaud	800 KBaud	50 KBaud	500 KBaud	20 KBaud	250 KBaud	10 KBaud	125 KBaud	
1 MBaud	100 KBaud										
800 KBaud	50 KBaud										
500 KBaud	20 KBaud										
250 KBaud	10 KBaud										
125 KBaud											

Table 39: Extended Diagnosis > CANOPEN\_SLAVE > Common Diagnostic

### 7.3.1.2 Commands

Commands	
Task states	
Name	Value
Register req.	1
Register cnf.	1
Start/Stop req.	19
Start/Stop cnf.	19
Busparam req.	0
Busparam cnf.	0
Get buffer req.	1
Get buffer cnf.	1
State change ind.	1
State change res.	1
Set watchdog fail req.	0
Set watchdog fail cnf.	0
Data exch. req.	0
Data exch. cnf.	0
Send emergency req.	0
Send emergency cnf.	0
NMT command req.	0
NMT command cnf.	0
CAN_DL stop req.	0
CAN_DL stop cnf. pos.	0
CAN_DL stop cnf. neg.	0
CAN_DL register req.	0
CAN_DL register cnf. pos.	0
CAN_DL register cnf. neg.	0
CAN_DL set param req.	0
CAN_DL set param cnf. pos.	0
CAN_DL set param cnf. neg.	0
CAN_DL start req.	0
CAN_DL start cnf. pos.	0
CAN_DL start cnf. neg.	0
CAN_DL event ind.	0
CAN_DL event res.	0
CAN_DL register cnf. pos.	0
CAN_DL send data cnf. pos.	0
CAN_DL send data cnf. neg.	0
CAN_DL enable id req.	0
CAN_DL enable id cnf. pos.	0
CAN_DL enable id cnf. neg.	0
CAN_DL event ack. req.	0
CAN_DL event ack. cnf. pos.	0
CAN_DL event ack. cnf. neg.	0
CAN_DL recv data ind.	0
CAN_DL recv data res.	0
Unknown req./cnf.	0
Cyclic ind.	5504497
Get packet failed	0
Send packet failed	0

Figure 50: Extended Diagnosis > CANOPEN\_SLAVE > Commands

Name	Description
[Service]	Diagnosis counter of the encapsulation layer. Indicates the services processed. (The services of the single packets are described in the API manual.)
Get packet failed	Number of errors during requesting a packet
Send packet failed	Number of errors during sending a packet

Table 40: Extended Diagnosis > CANOPEN\_SLAVE > Commands

## 7.3.2 CANOPEN\_APS

### 7.3.2.1 Slave configuration

Slave configuration	
Task states	
Name	Value
Flags	0
Database found	no
Warmstart configuration	no
Initialization state	Complete
Initialization result	0x00000000

Figure 51: Extended Diagnosis > CANOPEN\_APS > Slave Configuration

Name	Description
Flags	Bit0 set: Configuration data base found not set: No configuration data base found  Bit1 set: Configuration by packets not set: No packets for configuration
Database found	Yes: Configuration data base found No: No configuration data base found
Warmstart configuration	Yes: Configuration by packets No: No packets for configuration
Initialization state	0 = Idle; 1 = Send initialize request; 2 = Wait for initialize confirmation; 3 = Send register request; 4 = Wait for register confirmation; 5 = Send get buffer request; 6 = Wait for get buffer confirmation; 7 = Send bus parameter request; 8 = Wait for bus parameter confirmation; 9 = Complete; 10 = Failed
Initialization result	Error code of the initialisation, 0 = no error

Table 41: Extended Diagnosis > CANOPEN\_APS > Slave Configuration

### 7.3.2.2 Commands

Commands	
Task states	
Name	Value
Register req.	1
Register cnf.	1
Start/Stop req.	20
Start/Stop cnf.	20
Init req.	1
Init cnf.	1
Busparam req.	0
Busparam cnf.	0
Get buffer req.	1
Get buffer cnf.	1
State change ind.	1
State change res.	1
Set watchdog fail req.	0
Set watchdog fail cnf.	0
Config pck. routed	0
Command pck. routed	0
Unknown req./cnf.	0
Cyclic ind.	5590838
Get packet failed	0
Send packet failed	0

Figure 52: Extended Diagnosis > CANOPEN\_APS > Commands

Name	Description
[Service]	Diagnosis counter of the encapsulation layer. Indicates the services processed. (The services of the single packets are described in the API manual.)
Get packet failed	Number of errors during requesting a packet
Send packet failed	Number of errors during sending a packet

Table 42: Extended Diagnosis > CANOPEN\_APS > Commands

## 8 Tools

### 8.1 Overview Tools

Under **Tools** the Packet Monitor and the IO Monitor are provided for test and diagnosis purposes.

#### **Tools Panes**

The table below gives an overview for the individual **Tools** dialog panes descriptions:

Section	Page
<i>Packet Monitor</i>	79
<i>IO Monitor</i>	83

Table 43: Descriptions of the Diagnosis Panes



**Note:** Accessing the **Tools** panes of the CANopen Slave DTM requires an online connection from the CANopen Slave DTM to the CANopen Slave device.



For further information, refer to section *Connecting/Disconnecting Device* on page 84.

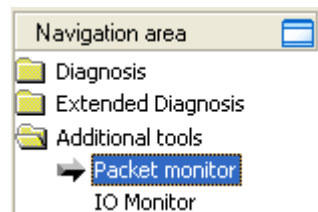


Figure 53: Navigation Area - Tools

## 8.2 Packet Monitor

The **Packet Monitor** serves for test and diagnosis purposes.

Data packets, i. e. messages are self-contained blocks of defined data length. The packets are used to communicate with the firmware and they are exchanged between the application (configuration software) and the firmware in the device. Packets can be sent once or cyclically to the connected device controlled by the user and packets received can be displayed.

Data packets comprise from a **Packet Header** and the **Send Data** or from a **Packet Header** and the **Receive Data**. The packet header can be evaluated by the receiver of the packet and contain the sender and receiver address, the data length, an ID number, status and error messages and the command or response code. The minimum packet size amounts 40 Byte for the packet header. The sending and receiving data are added.



For further information to the packet description refer to the *Protocol API Manual*.

- Open the **Packet Monitor** via **Tools > Packet Monitor**.

Figure 54: Packet Monitor

**Display Mode** switches the representation of the send and reception data between decimal and hexadecimal.

- Select **Reset Counter** to reset the packet counter.

## 8.2.1 Sending Packet

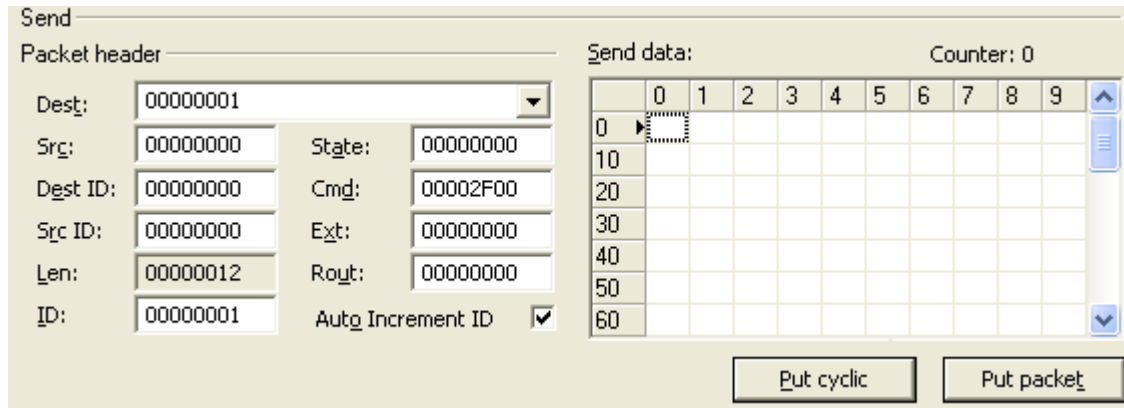


Figure 55: Send > Packet Header and Send Data

### Packet Header

Under **Send > Packet Header** the elements of the packet header of the sending packet are displayed, which is transmitted from the application (configuration software) to the device. The packet header of the sending packets contain the elements described in the following table.

Element		Description
<b>Dest</b>	Destination Queue Handle	Contains the identifier of the receiver for the packet (destination task queue of the firmware).
<b>Src</b>	Source Queue Handle	Contains the identifier of the sender of the packet (sending task).
<b>Dest ID</b>	Destination Queue Reference	Contains an identifier for the receiver of unsolicited sent packets from the firmware to the application (configuration software).
<b>Src ID</b>	Source Queue Reference	Contains an identifier of the sender.
<b>Len</b>	Packet Data Length (in Bytes)	Length of the send respectively receive data.
<b>ID</b>	Packet Identification As Unique Number	Identifies identical data packets among each other.
<b>State</b>	Status / Error Code	Transmits status or error codes to the packet sender.
<b>Cmd</b>	Command / Response Code	Command or respond code.
<b>Ext</b>	Extension	Field for extensions (reserved).
<b>Rout</b>	Routing Information	Internal value of the firmware.

Table 44: Descriptions Packet Header

- Under **Dest** select the receiver (*destination task queue*).
- Under **Cmd** select the command identification (*Request*).

**Auto Increment ID** is an increment for the identifier of the data packets and increments the ID by 1 for each newly sent packet.

### Send Data

- Under **Send > Send data** enter the send data of the packet, which shall be transmitted from the application (configuration software) to the mailbox of the device. The meaning of the transmitted data depends on the command or response code.

### Sending Packets once or cyclic

- To send packet once, select **Put packet**.
- To send packet cyclic, select **Put cyclic**.

## 8.2.2 Receiving Packet

The screenshot shows the 'Receive' interface with the following fields and data:

Packet header		Receive data:	Counter: 0
Dest:	00000001	0	00 00 00 00 04
Src:	00000000	1	00 00 01 00 01
State:	00000000	2	
Dest ID:	00000000	3	
Cmd:	00002F01	4	
Src ID:	00000000	5	
Ext:	00000000	6	
Len:	00000012	7	
Rout:	00000000	8	
ID:	0000003E	9	

Figure 56: Packet Header and Receive Data

### Packet Header

Under **Receive > Packet Header** the elements of the packet header of the receiving packet are displayed, which are transmitted back from the device to the application (configuration software). The packet header of the receiving packets contain the elements described in the following table.

Element	Description	
<b>Dest</b>	Destination Queue Handle	Contains the identifier of the receiver for the packet (destination task queue of the firmware).
<b>Src</b>	Source Queue Handle	Contains the identifier of the sender of the packet (sending task).
<b>Dest ID</b>	Destination Queue Reference	Contains an identifier for the receiver of unsolicited sent packets from the firmware to the application (configuration software).
<b>Src ID</b>	Source Queue Reference	Contains an identifier of the sender.
<b>Len</b>	Packet Data Length (in Bytes)	Length of the send respectively receive data.
<b>ID</b>	Packet Identification As Unique Number	Identifies identical data packets among each other.
<b>State</b>	Status / Error Code	Transmits status or error codes to the packet sender.
<b>Cmd</b>	Command / Response Code	Command or respond code.
<b>Ext</b>	Extension	Field for extensions (reserved).
<b>Rout</b>	Routing Information	Internal value of the firmware.

Table 45: Descriptions Packet Header

**Receive Data**

Under **Receive > Receive Data** the receiving data of the packet, which is transmitted back from the device to the application (configuration software) are displayed.

## 8.3 IO Monitor

The **IO Monitor** serves for test and diagnosis purposes. It provides to view data of the process data image and to change output data easily. The display is always in a Byte manner.



**Note:** Only change and write output data, if you know that no plant disturbances are caused by this. All output data written by the IO Monitor are transmitted at the bus and have effect on subordinate drives, IO etc.

**IO Monitor**

Columns: 10      Display mode: Decimal

Input data

Offset: 0      Go

	0	1	2	3	4	5	6	7	8	9
0	227	207	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
60	n	n	n	n	n	n	n	n	n	n

16:04:08: COM-flag not set      Clear

Output data

Offset: 0      Go

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
60	n	n	n	n	n	n	n	n	n	n

Update

Figure 57: IO Monitor

**Columns** switches the number of columns.

**Display Mode** switches the representation of the input and output data between decimal and hexadecimal.

**Offset / Go** moves the indication of the data to the entered offset value.

**Clear** clears displayed error messages.

- Enter the output value and select the button **Update**.
- Always the data of the process image are displayed, also when these Bytes have not been reserved by the configuration.

## 9 Online Functions

### 9.1 Connecting/Disconnecting Device



---

**Note:** Several CANopen Slave DTM functions e. g. **Diagnosis** or for the configuration download in the FDT Framework require an online connection from the CANopen Slave DTM to the CANopen Slave device.

---

#### Connecting Device

The following steps are needed to establish a connection from the CANopen Slave DTM to a CANopen Slave device:

Under **Settings** in the **Driver** pane:

1. Select one or several driver/s.
2. Configure the driver, if necessary.

Under **Settings** in the **Device Assignment** pane:

3. Scan for the device/s (with or without firmware).
4. Select the device (with or without firmware) and apply the selection.

Under **Settings** in the **Firmware Download** pane, if not yet a firmware was loaded to the device:

5. Select and download the firmware.

Under **Settings** in the **Device Assignment** pane, if not yet a firmware was loaded to the device:

6. Scan for the device (with firmware) once more.
7. Select the device (with firmware) once more.



---

An overview of the descriptions for these steps you find in the section *Overview Settings* on page 24.

---

8. In the DTM interface dialog select the **OK** button, to apply the selection and to close the DTM interface dialog.
  9. Put a right-click on the CANopen Slave device icon.
  10. Select the **Connect** command from the context menu.
- ↗ The CANopen Slave device now is connected to the CANopen Slave DTM via an online connection. In the network view the device description at the device icon of the Slave is displayed with a green colored background.

### **Disconnecting Device**

To disconnect an online connection from the CANopen Slave device to the CANopen Slave DTM take the following steps:

1. In the DTM interface dialog select the **OK** button, to close the DTM interface dialog.
  2. Right-click on the CANopen Slave device icon.
  3. Select the **Disconnect** command from the context menu.
- ⇒ Now the CANopen Slave device is disconnected from the CANopen Slave DTM.

## 9.2 Download Configuration



---

**Note:** To download configuration parameter data to the CANopen Slave device an online connection from the CANopen Slave DTM to the CANopen Slave device is required.

---



---

For further information, refer to section *Connecting/Disconnecting Device* on page 84.

---

To transfer the configuration with the corresponding parameter data to the CANopen Slave device you download the data to it using the frame application of the configuration software.

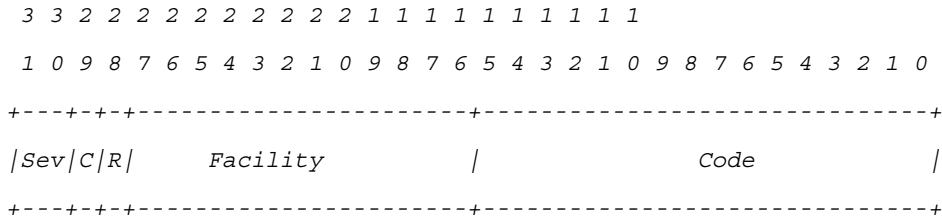
# 10 Error Codes

## 10.1 Error Code Definition

For COM based application, like the ODM Server and ODM drivers, a common error definition is used, similar to the Microsoft Windows® HRESULT definition.

Error Code Structure:

COM Errors are HRESULTs, which are 32 bit values using the following layout:



where

- Sev* - is the severity code:
  - 00 - Success
  - 01 - Informational
  - 10 - Warning
  - 11 - Error
- C* - is the Customer code flag
- R* - is a reserved bit
- Facility* - is the facility code
- Code* - is the facility's status code

In this common error definition, several error code regions are already reserved by Windows® itself, the ODM and some other modules.

## 10.2 Overview Error Codes

Overview Error Codes	Range
General Hardware Errors RCX Operating System	<i>RCX General Task</i> : 0xC02B0001 to 0xC02B4D52
	<i>RCX Common Status &amp; Errors Codes</i> : 0x00000000 to 0xC002000C
	<i>RCX Status &amp; Error Codes</i> : 0x00000000 to 0xC0000008
ODM Server	<i>General ODM Error Codes</i> : 0x8004C700 to 0x8004C761
	<i>General ODM Driver Error Codes</i> : 0x8004C7A0 to 0x8004C7C2
ODM Drivers	<i>cifX Driver Specific ODM Error</i> : 0x8004C001 to 0x8004C0A4
cifX Device Driver and netX Driver	<i>Generic Error</i> : 0x800A0001 bis 0x800A0017
	<i>Generic Driver Error</i> : 0x800B0001 bis 0x800B0042
	<i>Generic Device Error</i> : 0x800C0010 bis 0x800C0041
netX Driver	<i>CIFX API Transport</i> : 0x800D0001 bis 0x800D0013
	<i>CIFX API Transport Header State Error</i> : 0x800E0001 bis 0x800E000B
DBM	<i>ODM Error Codes</i> : 0xC004C810 to 0xC004C878

Table 46: Overview Error Codes and Ranges



The fieldbus specific error codes are described in the manuals of the corresponding protocol tasks.

## 10.3 General Hardware Error Codes

### 10.3.1 RCX General Task Errors

Error Code (Definition)	Value	Description
RCX_E_QUE_UNKNOWN	0xC02B0001	Unknown Queue
RCX_E_QUE_INDEX_UNKNOWN	0xC02B0002	Unknown Queue Index
RCX_E_TASK_UNKNOWN	0xC02B0003	Unknown Task
RCX_E_TASK_INDEX_UNKNOWN	0xC02B0004	Unknown Task Index
RCX_E_TASK_HANDLE_INVALID	0xC02B0005	Invalid Task Handle
RCX_E_TASK_INFO_IDX_UNKNOWN	0xC02B0006	Unknown Index
RCX_E_FILE_XFR_TYPE_INVALID	0xC02B0007	Invalid Transfer Type
RCX_E_FILE_REQUEST_INCORRECT	0xC02B0008	Invalid File Request
RCX_E_TASK_INVALID	0xC02B000E	Invalid Task
RCX_E_SEC_FAILED	0xC02B001D	Security EEPROM Access Failed
RCX_E_EEPROM_DISABLED	0xC02B001E	EEPROM Disabled
RCX_E_INVALID_EXT	0xC02B001F	Invalid Extension
RCX_E_SIZE_OUT_OF_RANGE	0xC02B0020	Block Size Out Of Range
RCX_E_INVALID_CHANNEL	0xC02B0021	Invalid Channel
RCX_E_INVALID_FILE_LEN	0xC02B0022	Invalid File Length
RCX_E_INVALID_CHAR_FOUND	0xC02B0023	Invalid Character Found
RCX_E_PACKET_OUT_OF_SEQ	0xC02B0024	Packet Out Of Sequence
RCX_E_SEC_NOT_ALLOWED	0xC02B0025	Not Allowed In Current State
RCX_E_SEC_INVALID_ZONE	0xC02B0026	Security EEPROM Invalid Zone
RCX_E_SEC_EEPROM_NOT_AVAIL	0xC02B0028	Security EEPROM Eeprom Not Available
RCX_E_SEC_INVALID_CHECKSUM	0xC02B0029	Security EEPROM Invalid Checksum
RCX_E_SEC_ZONE_NOT_WRITEABLE	0xC02B002A	Security EEPROM Zone Not Writeable
RCX_E_SEC_READ_FAILED	0xC02B002B	Security EEPROM Read Failed
RCX_E_SEC_WRITE_FAILED	0xC02B002C	Security EEPROM Write Failed
RCX_E_SEC_ACCESS_DENIED	0xC02B002D	Security EEPROM Access Denied
RCX_E_SEC_EEPROM_EMULATED	0xC02B002E	Security EEPROM Emulated
RCX_E_INVALID_BLOCK	0xC02B0038	Invalid Block
RCX_E_INVALID_STRUCT_NUMBER	0xC02B0039	Invalid Structure Number
RCX_E_INVALID_CHECKSUM	0xC02B4352	Invalid Checksum
RCX_E_CONFIG_LOCKED	0xC02B4B54	Configuration Locked
RCX_E_SEC_ZONE_NOT_READABLE	0xC02B4D52	Security EEPROM Zone Not Readable

Table 47: RCX General Task Errors

### 10.3.2 RCX Common Status & Errors Codes

Error Code (Definition)	Value	Description
RCX_S_OK	0x00000000	Success, Status Okay
RCX_E_FAIL	0xC0000001	Fail
RCX_E_UNEXPECTED	0xC0000002	Unexpected
RCX_E_OUTOFMEMORY	0xC0000003	Out Of Memory
RCX_E_UNKNOWN_COMMAND	0xC0000004	Unknown Command
RCX_E_UNKNOWN_DESTINATION	0xC0000005	Unknown Destination
RCX_E_UNKNOWN_DESTINATION_ID	0xC0000006	Unknown Destination ID
RCX_E_INVALID_PACKET_LEN	0xC0000007	Invalid Packet Length
RCX_E_INVALID_EXTENSION	0xC0000008	Invalid Extension
RCX_E_INVALID_PARAMETER	0xC0000009	Invalid Parameter
RCX_E_WATCHDOG_TIMEOUT	0xC000000C	Watchdog Timeout
RCX_E_INVALID_LIST_TYPE	0xC000000D	Invalid List Type
RCX_E_UNKNOWN_HANDLE	0xC000000E	Unknown Handle
RCX_E_PACKET_OUT_OF_SEQ	0xC000000F	Out Of Sequence
RCX_E_PACKET_OUT_OF_MEMORY	0xC0000010	Out Of Memory
RCX_E_QUE_PACKETDONE	0xC0000011	Queue Packet Done
RCX_E_QUE_SENDPACKET	0xC0000012	Queue Send Packet
RCX_E_POOL_PACKET_GET	0xC0000013	Pool Packet Get
RCX_E_POOL_GET_LOAD	0xC0000015	Pool Get Load
RCX_E_REQUEST_RUNNING	0xC000001A	Request Already Running
RCX_E_INIT_FAULT	0xC0000100	Initialization Fault
RCX_E_DATABASE_ACCESS_FAILED	0xC0000101	Database Access Failed
RCX_E_NOT_CONFIGURED	0xC0000119	Not Configured
RCX_E_CONFIGURATION_FAULT	0xC0000120	Configuration Fault
RCX_E_INCONSISTENT_DATA_SET	0xC0000121	Inconsistent Data Set
RCX_E_DATA_SET_MISMATCH	0xC0000122	Data Set Mismatch
RCX_E_INSUFFICIENT_LICENSE	0xC0000123	Insufficient License
RCX_E_PARAMETER_ERROR	0xC0000124	Parameter Error
RCX_E_INVALID_NETWORK_ADDRESS	0xC0000125	Invalid Network Address
RCX_E_NO_SECURITY_MEMORY	0xC0000126	No Security Memory
RCX_E_NETWORK_FAULT	0xC0000140	Network Fault
RCX_E_CONNECTION_CLOSED	0xC0000141	Connection Closed
RCX_E_CONNECTION_TIMEOUT	0xC0000142	Connection Timeout
RCX_E_LONELY_NETWORK	0xC0000143	Lonely Network
RCX_E_DUPLICATE_NODE	0xC0000144	Duplicate Node
RCX_E_CABLE_DISCONNECT	0xC0000145	Cable Disconnected
RCX_E_BUS_OFF	0xC0000180	Network Node Bus Off
RCX_E_CONFIG_LOCKED	0xC0000181	Configuration Locked
RCX_E_APPLICATION_NOT_READY	0xC0000182	Application Not Ready
RCX_E_TIMER_APPL_PACKET_SENT	0xC002000C	Timer App Packet Sent

Table 48: RCX Common Status & Errors Codes

### 10.3.3 RCX Status & Error Codes

Error Code (Definition)	Value	Description
RCX_S_OK	0x00000000	SUCCESS, STATUS OKAY
RCX_S_QUE_UNKNOWN	0xC02B0001	UNKNOWN QUEUE
RCX_S_QUE_INDEX_UNKNOWN	0xC02B0002	UNKNOWN QUEUE INDEX
RCX_S_TASK_UNKNOWN	0xC02B0003	UNKNOWN TASK
RCX_S_TASK_INDEX_UNKNOWN	0xC02B0004	UNKNOWN TASK INDEX
RCX_S_TASK_HANDLE_INVALID	0xC02B0005	INVALID TASK HANDLE
RCX_S_TASK_INFO_IDX_UNKNOWN	0xC02B0006	UNKNOWN INDEX
RCX_S_FILE_XFR_TYPE_INVALID	0xC02B0007	INVALID TRANSFER TYPE
RCX_S_FILE_REQUEST_INCORRECT	0xC02B0008	INVALID FILE REQUEST
RCX_S_UNKNOWN_DESTINATION	0xC0000005	UNKNOWN DESTINATION
RCX_S_UNKNOWN_DESTINATION_ID	0xC0000006	UNKNOWN DESTINATION ID
RCX_S_INVALID_LENGTH	0xC0000007	INVALID LENGTH
RCX_S_UNKNOWN_COMMAND	0xC0000004	UNKNOWN COMMAND
RCX_S_INVALID_EXTENSION	0xC0000008	INVALID EXTENSION

Table 49: RCX Status & Error Codes

#### 10.3.3.1 RCX Status & Error Codes Slave State

Error Code (Definition)	Value	Description
RCX_SLAVE_STATE_UNDEFINED	0x00000000	UNDEFINED
RCX_SLAVE_STATE_OK	0x00000001	OK
RCX_SLAVE_STATE_FAILED	0x00000002	FAILED (at least one slave)

Table 50: RCX Status & Error Codes Slave State

## 10.4 ODM Error Codes

### 10.4.1 General ODM Error Codes

Error Code (Definition)	Value	Description
CODM3_E_INTERNALERROR	0x8004C700	Internal ODM Error
ODM3_E_DESCRIPTION_NOTFOUND	0x8004C701	Description not found in ODM database
CODM3_E_WRITEREGISTRY	0x8004C710	Error writing to the registry
CODM3_E_BAD_REGULAR_EXPRESSION	0x8004C711	Invalid regular expression
CODM3_E_COMCATEGORIE_MANAGER_FAILED	0x8004C712	Component Category Manager could not be instantiated
CODM3_E_COMCATEGORIE_ENUMERATION_FAILED	0x8004C713	Driver could not be enumerated by the Category Manager
CODM3_E_CREATE_LOCAL_BUFFER	0x8004C714	Error creating local buffers
CODM3_E_UNKNOWNHANDLE	0x8004C715	Unknown handle
CODM3_E_QUEUE_LIMIT_REACHED	0x8004C717	Queue size limit for connection reached
CODM3_E_DATASIZE_ZERO	0x8004C718	Zero data length passed
CODM3_E_INVALID_DATA	0x8004C719	Invalid data content
CODM3_E_INVALID_MODE	0x8004C71A	Invalid mode
CODM3_E_DATABASE_READ	0x8004C71B	Error reading database
CODM3_E_CREATE_DEVICE_THREAD	0x8004C750	Error creating device thread
CODM3_E_CREATE_DEVICE_THREAD_STOP_EVENT	0x8004C751	Error creating device thread stop event
CODM3_E_CLIENT_NOT_REGISTERED	0x8004C752	Client is not registered at the ODM
CODM3_E_NO_MORE_CLIENTS	0x8004C753	Maximum number of clients reached
CODM3_E_MAX_CLIENT_CONNECTIONS_REACHED	0x8004C754	Maximum number of client connections reached
CODM3_E_ENTRY_NOT_FOUND	0x8004C755	Driver/device not found
CODM3_E_DRIVER_NOT_FOUND	0x8004C757	The requested driver is unknown to the ODM
CODM3_E_DEVICE_ALREADY_LOCKED	0x8004C758	Device is locked by another process
CODM3_E_DEVICE_UNLOCKED_FAILED	0x8004C759	Device could not be unlocked, lock was set by another process
CODM3_E_DEVICE_LOCK_NECESSARY	0x8004C75A	Operation requires a device lock to be set
CODM3_E_DEVICE_SUBSCRIPTIONLIMIT	0x8004C75B	Maximum number of servers registered for this device reached
CODM3_E_DEVICE_NOTSUBSCRIBED	0x8004C75C	Process is not registered as a server on this device
CODM3_E_DEVICE_NO_MESSAGE	0x8004C75D	No message available
CODM3_E_TRANSFERTIMEOUT	0x8004C760	Message transfer timeout
CODM3_E_MESSAGE_INSERVICE	0x8004C761	Message in service

Table 51: ODM Error Codes - General ODM Error Codes

## 10.4.2 General ODM Driver Error Codes

Error Code (Definition)	Value	Description
CODM3_E_DRV_OPEN_DEVICE	0x8004C7A0	Packet type unsupported by driver
CODM3_E_DRV_INVALID_IDENTIFIER	0x8004C7A1	Invalid device identifier
CODM3_E_DRV_DEVICE_PARAMETERS_MISMATCH	0x8004C7A3	Parameters differ from requested device
CODM3_E_DRV_BROWSE_NO_DEVICES	0x8004C7A4	No devices found
CODM3_E_DRV_CREATE_DEVICE_INST	0x8004C7A5	Device instance could not be created
CODM3_E_DRV_DEVICE_NOMORE_TX	0x8004C7A6	Device connection limit reached
CODM3_E_DRV_DEVICE_DUPLICATE_TX	0x8004C7A7	Duplicate transmitter ID
CODM3_E_DRV_DEVICE_NOT_CONFIGURED	0x8004C7A8	Device is not configured
CODM3_E_DRV_DEVICE_COMMUNICATION	0x8004C7A9	Device communication error
CODM3_E_DRV_DEVICE_NO_MESSAGE	0x8004C7AA	No message available
CODM3_E_DRV_DEVICE_NOT_READY	0x8004C7AB	Device not ready
CODM3_E_DRV_INVALIDCONFIGURATION	0x8004C7AC	Invalid driver configuration
CODM3_E_DRV_DLINVALIDMODE	0x8004C7C0	Invalid download mode
CODM3_E_DRV_DLINPROGRESS	0x8004C7C1	Download is active
CODM3_E_DRV_ULINPROGRESS	0x8004C7C2	Upload is active

Table 52: ODM Error Codes - General ODM Driver Error Codes

### 10.4.3 cifX Driver Specific ODM Error Codes

cifX Driver Specific ODM Error Codes		
Error Code (Definition)	Value	Description
DRV_E_BOARD_NOT_INITIALIZED	0x8004C001	DRIVER Board not initialized
DRV_E_INIT_STATE_ERROR	0x8004C002	DRIVER Error in internal init state
DRV_E_READ_STATE_ERROR	0x8004C003	DRIVER Error in internal read state
DRV_E_CMD_ACTIVE	0x8004C004	DRIVER Command on this channel is active
DRV_E_PARAMETER_UNKNOWN	0x8004C005	DRIVER Unknown parameter in function
DRV_E_WRONG_DRIVER_VERSION	0x8004C006	DRIVER Version is incompatible with DLL
DRV_E_PCI_SET_CONFIG_MODE	0x8004C007	DRIVER Error during PCI set configuration mode
DRV_E_PCI_READ_DPM_LENGTH	0x8004C008	DRIVER Could not read PCI dual port memory length
DRV_E_PCI_SET_RUN_MODE	0x8004C009	DRIVER Error during PCI set run mode
DRV_E_DEV_DPM_ACCESS_ERROR	0x8004C00A	DEVICE Dual port ram not accessible(board not found)
DRV_E_DEV_NOT_READY	0x8004C00B	DEVICE Not ready (ready flag failed)
DRV_E_DEV_NOT_RUNNING	0x8004C00C	DEVICE Not running (running flag failed)
DRV_E_DEV_WATCHDOG_FAILED	0x8004C00D	DEVICE Watchdog test failed
DRV_E_DEV_OS_VERSION_ERROR	0x8004C00E	DEVICE Signals wrong OS version
DRV_E_DEV_SYSERR	0x8004C00F	DEVICE Error in dual port flags
DRV_E_DEV_MAILBOX_FULL	0x8004C010	DEVICE Send mailbox is full
DRV_E_DEV_PUT_TIMEOUT	0x8004C011	DEVICE PutMessage timeout
DRV_E_DEV_GET_TIMEOUT	0x8004C012	DEVICE GetMessage timeout
DRV_E_DEV_GET_NO_MESSAGE	0x8004C013	DEVICE No message available
DRV_E_DEV_RESET_TIMEOUT	0x8004C014	DEVICE RESET command timeout
DRV_E_DEV_NO_COM_FLAG	0x8004C015	DEVICE COM-flag not set. Check if Bus is running
DRV_E_DEV_EXCHANGE_FAILED	0x8004C016	DEVICE I/O data exchange failed
DRV_E_DEV_EXCHANGE_TIMEOUT	0x8004C017	DEVICE I/O data exchange timeout
DRV_E_DEV_COM_MODE_UNKNOWN	0x8004C018	DEVICE I/O data mode unknown
DRV_E_DEV_FUNCTION_FAILED	0x8004C019	DEVICE Function call failed
DRV_E_DEV_DPMSIZE_MISMATCH	0x8004C01A	DEVICE DPM size differs from configuration
DRV_E_DEV_STATE_MODE_UNKNOWN	0x8004C01B	DEVICE State mode unknown
DRV_E_DEV_HW_PORT_IS_USED	0x8004C01C	DEVICE Output port already in use
DRV_E_USR_OPEN_ERROR	0x8004C01E	USER Driver not opened (device driver not loaded)
DRV_E_USR_INIT_DRV_ERROR	0x8004C01F	USER Can't connect to device
DRV_E_USR_NOT_INITIALIZED	0x8004C020	USER Board not initialized (DevInitBoard not called)
DRV_E_USR_COMM_ERR	0x8004C021	USER IOCTL function failed
DRV_E_USR_DEV_NUMBER_INVALID	0x8004C022	USER Parameter DeviceNumber invalid
DRV_E_USR_INFO_AREA_INVALID	0x8004C023	USER Parameter InfoArea unknown
DRV_E_USR_NUMBER_INVALID	0x8004C024	USER Parameter Number invalid
DRV_E_USR_MODE_INVALID	0x8004C025	USER Parameter Mode invalid
DRV_E_USR_MSG_BUF_NULL_PTR	0x8004C026	USER NULL pointer assignment
DRV_E_USR_MSG_BUF_TOO_SHORT	0x8004C027	USER Message buffer too small

<b>cifX Driver Specific ODM Error Codes</b>		
<b>Error Code (Definition)</b>	<b>Value</b>	<b>Description</b>
DRV_E_USR_SIZE_INVALID	0x8004C028	USER Parameter Size invalid
DRV_E_USR_SIZE_ZERO	0x8004C02A	USER Parameter Size with zero length
DRV_E_USR_SIZE_TOO_LONG	0x8004C02B	USER Parameter Size too long
DRV_E_USR_DEV_PTR_NULL	0x8004C02C	USER Device address null pointer
DRV_E_USR_BUF_PTR_NULL	0x8004C02D	USER Pointer to buffer is a null pointer
DRV_E_USR_SENDSIZE_TOO_LONG	0x8004C02E	USER Parameter SendSize too large
DRV_E_USR_RECVSIZE_TOO_LONG	0x8004C02F	USER Parameter ReceiveSize too large
DRV_E_USR_SENDBUF_PTR_NULL	0x8004C030	USER Pointer to send buffer is a null pointer
DRV_E_USR_RECVBUF_PTR_NULL	0x8004C031	USER Pointer to receive buffer is a null pointer
DRV_E_DMA_INSUFF_MEM	0x8004C032	DMA Memory allocation error
DRV_E_DMA_TIMEOUT_CH4	0x8004C033	DMA Read I/O timeout
DRV_E_DMA_TIMEOUT_CH5	0x8004C034	DMA Write I/O timeout
DRV_E_DMA_TIMEOUT_CH6	0x8004C035	DMA PCI transfer timeout
DRV_E_DMA_TIMEOUT_CH7	0x8004C036	DMA Download timeout
DRV_E_DMA_DB_DOWN_FAIL	0x8004C037	DMA Database download failed
DRV_E_DMA_FW_DOWN_FAIL	0x8004C038	DMA Firmware download failed
DRV_E_CLEAR_DB_FAIL	0x8004C039	DMA Clear database on the device failed
DRV_E_DEV_NO_VIRTUAL_MEM	0x8004C03C	DMA USER Virtual memory not available
DRV_E_DEV_UNMAP_VIRTUAL_MEM	0x8004C03D	DMA USER Unmap virtual memory failed
DRV_E_GENERAL_ERROR	0x8004C046	DRIVER General error
DRV_E_DMA_ERROR	0x8004C047	DRIVER General DMA error
DRV_E_WDG_IO_ERROR	0x8004C048	DRIVER I/O WatchDog failed
DRV_E_WDG_DEV_ERROR	0x8004C049	DRIVER Device Watchdog failed
DRV_E_USR_DRIVER_UNKNOWN	0x8004C050	USER Driver unknown
DRV_E_USR_DEVICE_NAME_INVALID	0x8004C051	USER Device name invalid
DRV_E_USR_DEVICE_NAME_UNKNOWN	0x8004C052	USER Device name unknown
DRV_E_USR_DEVICE_FUNC_NOTIMPL	0x8004C053	USER Device function not implemented
DRV_E_USR_FILE_OPEN_FAILED	0x8004C064	USER File could not be opened
DRV_E_USR_FILE_SIZE_ZERO	0x8004C065	USER File size zero
DRV_E_USR_FILE_NO_MEMORY	0x8004C066	USER Not enough memory to load file
DRV_E_USR_FILE_READ_FAILED	0x8004C067	USER File read failed
DRV_E_USR_INVALID_FILETYPE	0x8004C068	USER File type invalid
DRV_E_USR_FILENAME_INVALID	0x8004C069	USER Invalid filename
DRV_E_FW_FILE_OPEN_FAILED	0x8004C06E	USER Firmware file could not be opened
DRV_E_FW_FILE_SIZE_ZERO	0x8004C06F	USER Not enough memory to load firmware file
DRV_E_FW_FILE_NO_MEMORY	0x8004C070	USER Not enough memory to load firmware file
DRV_E_FW_FILE_READ_FAILED	0x8004C071	USER Firmware file read failed
DRV_E_FW_INVALID_FILETYPE	0x8004C072	USER Firmware file type invalid
DRV_E_FW_FILENAME_INVALID	0x8004C073	USER Firmware file name not valid
DRV_E_FW_DOWNLOAD_ERROR	0x8004C074	USER Firmware file download error
DRV_E_FW_FILENAME_NOT_FOUND	0x8004C075	USER Firmware file not found in the internal table
DRV_E_FW_BOOTLOADER_ACTIVE	0x8004C076	USER Firmware file BOOTLOADER active

<b>cifX Driver Specific ODM Error Codes</b>		
<b>Error Code (Definition)</b>	<b>Value</b>	<b>Description</b>
DRV_E_FW_NO_FILE_PATH	0x8004C077	USER Firmware file no file path
DRV_E_CF_FILE_OPEN_FAILED	0x8004C078	USER Configuration file could not be opened
DRV_E_CF_FILE_SIZE_ZERO	0x8004C079	USER Configuration file size zero
DRV_E_CF_FILE_NO_MEMORY	0x8004C07A	USER Not enough memory to load configuration file
DRV_E_CF_FILE_READ_FAILED	0x8004C07B	USER Configuration file read failed
DRV_E_CF_INVALID_FILETYPE	0x8004C07C	USER Configuration file type invalid
DRV_E_CF_FILENAME_INVALID	0x8004C07D	USER Configuration file name not valid
DRV_E_CF_DOWNLOAD_ERROR	0x8004C07E	USER Configuration file download error
DRV_E_CF_FILE_NO_SEGMENT	0x8004C07F	USER No flash segment in the configuration file
DRV_E_CF_DIFFERS_FROM_DBM	0x8004C080	USER Configuration file differs from database
DRV_E_DBM_SIZE_ZERO	0x8004C083	USER Database size zero
DRV_E_DBM_NO_MEMORY	0x8004C084	USER Not enough memory to upload database
DRV_E_DBM_READ_FAILED	0x8004C085	USER Database read failed
DRV_E_DBM_NO_FLASH_SEGMENT	0x8004C086	USER Database segment unknown
DEV_E_CF_INVALID_DESCRIPTOR_VERSION	0x8004C096	CONFIG Version of the descriptor table invalid
DEV_E_CF_INVALID_INPUT_OFFSET	0x8004C097	CONFIG Input offset is invalid
DEV_E_CF_NO_INPUT_SIZE	0x8004C098	CONFIG Input size is 0
DEV_E_CF_MISMATCH_INPUT_SIZE	0x8004C099	CONFIG Input size does not match configuration
DEV_E_CF_INVALID_OUTPUT_OFFSET	0x8004C09A	CONFIG Invalid output offset
DEV_E_CF_NO_OUTPUT_SIZE	0x8004C09B	CONFIG Output size is 0
DEV_E_CF_MISMATCH_OUTPUT_SIZE	0x8004C09C	CONFIG Output size does not match configuration
DEV_E_CF_STN_NOT_CONFIGURED	0x8004C09D	CONFIG Station not configured
DEV_E_CF_CANNOT_GET_STN_CONFIG	0x8004C09E	CONFIG Cannot get the Station configuration
DEV_E_CF_MODULE_DEF_MISSING	0x8004C09F	CONFIG Module definition is missing
DEV_E_CF_MISMATCH_EMPTY_SLOT	0x8004C0A0	CONFIG Empty slot mismatch
DEV_E_CF_MISMATCH_INPUT_OFFSET	0x8004C0A1	CONFIG Input offset mismatch
DEV_E_CF_MISMATCH_OUTPUT_OFFSET	0x8004C0A2	CONFIG Output offset mismatch
DEV_E_CF_MISMATCH_DATA_TYPE	0x8004C0A3	CONFIG Data type mismatch
DEV_E_CF_MODULE_DEF_MISSING_NO_SI	0x8004C0A4	CONFIG Module definition is missing,(no Slot/Idx)

Table 53: cifX Driver Specific ODM Error Codes

## 10.5 Error Codes cifX Device Driver and netX Driver

### 10.5.1 Generic Error Codes

Error Code (Definition)	Value	Description
CIFX_INVALID_POINTER	0x800A0001L	Invalid pointer (NULL) passed to driver
CIFX_INVALID_BOARD	0x800A0002L	No board with the given nameindex available
CIFX_INVALID_CHANNEL	0x800A0003L	No channel with the given index available
CIFX_INVALID_HANDLE	0x800A0004L	Invalid handle passed to driver
CIFX_INVALID_PARAMETER	0x800A0005L	Invalid parameter
CIFX_INVALID_COMMAND	0x800A0006L	Invalid command
CIFX_INVALID_BUFFER_SIZE	0x800A0007L	Invalid buffer size
CIFX_INVALID_ACCESS_SIZE	0x800A0008L	Invalid access size
CIFX_FUNCTION_FAILED	0x800A0009L	Function failed
CIFX_FILE_OPEN_FAILED	0x800A000AL	File could not be opened
CIFX_FILE_SIZE_ZERO	0x800A000BL	File size is zero
CIFX_FILE_LOAD_INSUFF_MEM	0x800A000CL	Insufficient memory to load file
CIFX_FILE_CHECKSUM_ERROR	0x800A000DL	File checksum compare failed
CIFX_FILE_READ_ERROR	0x800A000EL	Error reading from file
CIFX_FILE_TYPE_INVALID	0x800A000FL	Invalid file type
CIFX_FILE_NAME_INVALID	0x800A0010L	Invalid file name
CIFX_FUNCTION_NOT_AVAILABLE	0x800A0011L	Driver function not available
CIFX_BUFFER_TOO_SHORT	0x800A0012L	Given buffer is too short
CIFX_MEMORY_MAPPING_FAILED	0x800A0013L	Failed to map the memory
CIFX_NO_MORE_ENTRIES	0x800A0014L	No more entries available
CIFX_CALLBACK_MODE_UNKNOWN	0x800A0015L	Unkown callback handling mode
CIFX_CALLBACK_CREATE_EVENT_FAILED	0x800A0016L	Failed to create callback events
CIFX_CALLBACK_CREATE_RECV_BUFFER	0x800A0017L	Failed to create callback receive buffer

Table 54: Generic Error Codes

## 10.5.2 Generic Driver Error Codes

Error Code (Definition)	Value	Description
CIFX_DRV_NOT_INITIALIZED	0x800B0001L	Driver not initialized
CIFX_DRV_INIT_STATE_ERROR	0x800B0002L	Driver init state error
CIFX_DRV_READ_STATE_ERROR	0x800B0003L	Driver read state error
CIFX_DRV_CMD_ACTIVE	0x800B0004L	Command is active on device
CIFX_DRV_DOWNLOAD_FAILED	0x800B0005L	General error during download
CIFX_DRV_WRONG_DRIVER_VERSION	0x800B0006L	Wrong driver version
CIFX_DRV_DRIVER_NOT_LOADED	0x800B0030L	CIFx driver is not running
CIFX_DRV_INIT_ERROR	0x800B0031L	Failed to initialize the device
CIFX_DRV_CHANNEL_NOT_INITIALIZED	0x800B0032L	Channel not initialized (xOpenChannel not called)
CIFX_DRV_IO_CONTROL_FAILED	0x800B0033L	IOControl call failed
CIFX_DRV_NOT_OPENED(	0x800B0034L	Driver was not opened
CIFX_DRV_DOWNLOAD_STORAGE_UNKN OWN	0x800B0040L	Unknown download storage type (RAMFLASH based) found
CIFX_DRV_DOWNLOAD_FW_WRONG_CHA NNEL	0x800B0041L	Channel number for a firmware download not supported
CIFX_DRV_DOWNLOAD_MODULE_NO_BAS EOS	0x800B0042L	Modules are not allowed without a Base OS firmware

Table 55: Generic Driver Error Codes

### 10.5.3 Generic Device Error Codes

Error Code (Definition)	Value	Description
CIFX_DEV_DPM_ACCESS_ERROR	0x800C0010L	Dual port memory not accessible (board not found)
CIFX_DEV_NOT_READY	0x800C0011L	Device not ready (ready flag failed)
CIFX_DEV_NOT_RUNNING	0x800C0012L	Device not running (running flag failed)
CIFX_DEV_WATCHDOG_FAILED	0x800C0013L	Watchdog test failed
CIFX_DEV_SYSERR	0x800C0015L	Error in handshake flags
CIFX_DEV_MAILBOX_FULL	0x800C0016L	Send mailbox is full
CIFX_DEV_PUT_TIMEOUT	0x800C0017L	Send packet timeout
CIFX_DEV_GET_TIMEOUT	0x800C0018L	Receive packet timeout
CIFX_DEV_GET_NO_PACKET	0x800C0019L	No packet available
CIFX_DEV_MAILBOX_TOO_SHORT	0x800C001AL	Mailbox too short
CIFX_DEV_RESET_TIMEOUT	0x800C0020L	Reset command timeout
CIFX_DEV_NO_COM_FLAG	0x800C0021L	COM-flag not set
CIFX_DEV_EXCHANGE_FAILED	0x800C0022L	IO data exchange failed
CIFX_DEV_EXCHANGE_TIMEOUT	0x800C0023L	IO data exchange timeout
CIFX_DEV_COM_MODE_UNKNOWN	0x800C0024L	Unknown IO exchange mode
CIFX_DEV_FUNCTION_FAILED	0x800C0025L	Device function failed
CIFX_DEV_DPMSIZE_MISMATCH	0x800C0026L	DPM size differs from configuration
CIFX_DEV_STATE_MODE_UNKNOWN	0x800C0027L	Unknown state mode
CIFX_DEV_HW_PORT_IS_USED	0x800C0028L	Device is still accessed
CIFX_DEV_CONFIG_LOCK_TIMEOUT	0x800C0029L	Configuration locking timeout
CIFX_DEV_CONFIG_UNLOCK_TIMEOUT	0x800C002AL	Configuration unlocking timeout
CIFX_DEV_HOST_STATE_SET_TIMEOUT	0x800C002BL	Set HOST state timeout
CIFX_DEV_HOST_STATE_CLEAR_TIMEOUT	0x800C002CL	Clear HOST state timeout
CIFX_DEV_INITIALIZATION_TIMEOUT	0x800C002DL	Timeout during channel initialization
CIFX_DEV_BUS_STATE_ON_TIMEOUT	0x800C002EL	Set Bus ON Timeout
CIFX_DEV_BUS_STATE_OFF_TIMEOUT	0x800C002FL	Set Bus OFF Timeout
CIFX_DEV_MODULE_ALREADY_RUNNING	0x800C0040L	Module already running
CIFX_DEV_MODULE_ALREADY_EXISTS	0x800C0041L	Module already exists

Table 56: Generic Device Error Codes

## 10.6 Error Codes netX Driver

### 10.6.1 CIFX API Transport Error Codes

Error Code (Definition)	Value	Description
CIFX_TRANSPORT_SEND_TIMEOUT	0x800D0001L	Time out while sending data
CIFX_TRANSPORT_RECV_TIMEOUT	0x800D0002L	Time out waiting for incoming data
CIFX_TRANSPORT_CONNECT	0x800D0003L	Unable to communicate to the device no answer
CIFX_TRANSPORT_ABORTED	0x800D0004L	Transfer has been aborted due to keep alive timeout or interface detachment
CIFX_CONNECTOR_FUNCTIONS_READ_ERROR	0x800D0010L	Error reading the connector functions from the DLL
CIFX_CONNECTOR_IDENTIFIER_TOO_LONG	0x800D0011L	Connector delivers an identifier longer than 6 characters
CIFX_CONNECTOR_IDENTIFIER_EMPTY	0x800D0012L	Connector delivers an empty identifier
CIFX_CONNECTOR_DUPLICATE_IDENTIFIER	0x800D0013L	Connector identifier already used

Table 57: CIFX API Transport Error Codes

### 10.6.2 CIFX API Transport Header State Error Codes

Error Code (Definition)	Value	Description
CIFX_TRANSPORT_ERROR_UNKNOWN	0x800E0001L	Unknown error code in transport header
CIFX_TRANSPORT_CHECKSUM_ERROR	0x800E0002L	CRC16 checksum failed
CIFX_TRANSPORT_LENGTH_INCOMPLETE	0x800E0003L	Transaction with incomplete length detected
CIFX_TRANSPORT_DATA_TYPE_UNKNOWN	0x800E0004L	Device does not support requested data type
CIFX_TRANSPORT_DEVICE_UNKNOWN	0x800E0005L	Device not available unknown
CIFX_TRANSPORT_CHANNEL_UNKNOWN	0x800E0006L	Channel not available unknown
CIFX_TRANSPORT_SEQUENCE	0x800E0007L	Sequence error detected
CIFX_TRANSPORT_BUFFER_OVERFLOW	0x800E0008L	Buffer overflow detected
CIFX_TRANSPORT_RESOURCE	0x800E0009L	Device signals out of resources
CIFX_TRANSPORT_KEEPALIVE	0x800E000AL	Device connection monitoring error (Keep alive)
CIFX_TRANSPORT_DATA_TOO_SHORT	0x800E000BL	Received transaction data too short

Table 58: CIFX API Transport Header State Error Codes

## 10.7 ODM Error Codes DBM V4

ODM Error Codes DBM V4		
Error Code (Definition)	Value	Description
CDBM_E_MD5_INVALID	0XC004C810	Checksum invalid
CDBM_E_INTERNALERROR	0XC004C811	Internal Error
CDBM_W_WRITEREGISTRY	0X8004C812	Error writing to the registry
CDBM_E_UNEXPECTED_VALUE_IN_OLD_HEADER_FORMAT	0XC004C813	Error in a file containing the old DBM Header format.
CDBM_E_CHECKSUM_INVALID	0XC004C814	The Checksum of the old Header is invalid
CDBM_E_DB_ALREADY_LOADED_FORMAT	0XC004C815	A database is already loaded
CDBM_E_NO_VALID_TRANSACTION	0XC004C816	No valid transaction handle given
CDBM_E_STD_STRUCT_ERROR	0XC004C817	An error occurred during validation of data
CDBM_E_UNSUPPORTED_DATA_TYPE_FORMAT	0XC004C818	Unsupported DataType
CDBM_W_CLASS_DELETED_FORMAT	0X8004C819 (Warning)	Using an Object which is marked as deleted
CDBM_W_CLIENT_DISCONNECTED	0X8004C81A (Warning)	A Client has already an outstanding connection to a Table. The connection is now destroyed.
CDBM_E_STRUCTURE_DEFINITION_INVALID	0XC004C81B	A structure definition of an Element in a Table is invalid
CDBM_E_NO_DATA_AVAILABLE	0XC004C81C	No data available for this operation
CDBM_E_NO_VALID_STRUCTURE	0XC004C81D	No valid structure available for this operation
CDBM_E_NO_TOGGLE_STRING_FOUND	0XC004C81E	No Toggle string found for this number
CDBM_E_ELEMENT_OUT_OF_RANGE	0XC004C81F	An element wasn't found in the Record of a Table
CDBM_E_ELEMENT_NOT_IN_TABLE	0XC004C820	The element is not part of the Table
CDBM_E_CANNOT_CONVERT_INTO_CLIENT_TYPE	0XC004C821	The data can't be converted into the Client type
CDBM_E_TRANSACTION_ALREADY_OPEN	0XC004C822	A transaction is already open. Please close this one first before opening a new one.
CDBM_I_OLD_WITHOUT_HEADER	0X4004C823 (Informational)	Use of an old DBM file Format without Header
CDBM_E_HR_FROM	0XC004C824	An HRESULT was received from a Subroutine
CDBM_E_PARAMETER	0XC004C825	A Parameter is invalid
CDBM_E_NOTIMPL	0XC004C826	Method is currently not implemented
CDBM_E_OUTOFMEMORY	0XC004C827	Out of memory
CDBM_E_NO_OPEN_TRANSACTION	0XC004C828	No transaction open
CDBM_E_NO_CONTENTS	0XC004C829	No contents available
CDBM_REC_NO_NOT_FOUND	0XC004C82A	Record not found
CDBM_STRUCTURE_ELEMENT_NOT_FOUND	0XC004C82B	Element of the Structure not found
CDBM_E_NO_MORE_RECORDS_IN_TABTYPE	0XC004C82C	Table type 3 can contain only one record
CDBM_E_WRITE	0XC004C82D	The data in the VARIANT must be given in a SafeArray
CDBM_E_WRITE_NO_PARRAY	0XC004C82E	The VARIANT contains no valid [parray] element

<b>ODM Error Codes DBM V4</b>		
<b>Error Code (Definition)</b>	<b>Value</b>	<b>Description</b>
CDBM_E_WRITE_CANT_ACCESS_DATA	0XC004C82F	Unable to access SafeArray Data in the VARIANT
CDBM_E_WRITE_DATA	0XC004C830	To write the data of this Element it must be given as a BSTR, or as an Array of VT_UI1/VT_I1
CDBM_E_WRITE_BSTR_E1	0XC004C831	The BSTR string must have an even length.
CDBM_E_WRITE_BSTR_E2	0XC004C832	The BSTR string must contain only hex digits (0..9 and a/A..f/F).
CDBM_E_WRITE_CANT_INTERPRET_ARRAY	0XC004C833	Unable to interpret data in the SafeArray.
CDBM_E_WRITE_VT_ERROR	0XC004C834	Data type in the SafeArray is not VT_UI1 or VT_I1.
CDBM_E_WRITE_LENGTH	0XC004C835	Data length is invalid for write operation of this type.
CDBM_WRITE_ELEMENT	0XC004C836	Element not found in the Record of the Table
CDBM_MIN_MAX_ERROR	0XC004C837	Can't write data because of min underflow or max overflow
CDBM_TABLE_EXIST	0XC004C838	Table already exist in the database
CDBM_MIN_MAX_INVALID	0XC004C839	The Min value is greater than the Max Value
CDBM_DEF_MIN_MAX_INVALID	0XC004C83A	The Default Value is not in the range between the Min value and the Max Value
CDBM_CANT_CHANGE_STRUCTURE_WHILE_RECORDS_EXIST	0XC004C83B	It's not allowed to change the structure while Records exist in the Table
CDBM_NEW_STRUCT_NEEDS_TYPE	0XC004C83C	In a newly added structure the data type must be set also
CDBM_VALUE_ERROR	0XC004C83D	Range error while validating a value
CDBM_DATATYPE_UNSUPPORTED_IN_RCS	0XC004C83E	The data type is unsupported in the RCS file format
CDBM_I_COUNT_OF_TABLES_EXCEEDS_RCS_RANGE	0X4004C83F (Informational)	The count of Tables exceeds the RCS range of Tables. This can cause problems if the file is downloaded to RCS Systems
CDBM_I_COUNT_OF_TABLES_EXCEEDS_OLDDBM_RANGE	0X4004C840 (Informational)	The count of Tables exceeds the DBM32.DLL range of Tables. This can cause problems if the file is used with older Tools using the DBM32.DLL
CDBM_UNSUPPORTED_DATATYPE_IN_RCS_MODE	0XC004C841	The Data type is not compatible with the old database format
CDBM_WRITE_UNSTRUCTURED_1	0XC004C842	The data of an unstructured record can only be written with the 'Write' Method not with 'WriteElement'.
CDBM_READ_UNSTRUCTURED_1	0XC004C843	The data of an unstructured record can only be read with the 'Read' Method not with 'ReadElement'
CDBM_WRITE_DATA_LENGTH_INVALID	0XC004C844	The given data length doesn't correspond with the expected data length.
CDBM_UNKNOWN_VIEW_MODE	0XC004C845	The View Mode is unknown.
CDBM_E_DIAG_TABLE	0XC004C846	It doesn't make much sense to add or delete records from a diagnostic table because those changes are never saved.
CDBM_E_ADR_STRING_ERROR	0XC004C847	The given Address string doesn't fit the required format of this type where all address bytes must be in the range between 0 and FF

<b>ODM Error Codes DBM V4</b>		
<b>Error Code (Definition)</b>	<b>Value</b>	<b>Description</b>
CDBM_ERROR_FROM_VAR_CHANGE_TYPE	0XC004C848	Function VariantChangeType return an error when trying to convert the Parameter
CDBM_E_MINERROR	0XC004C849	Error while comparing the Value with the lower range
CDBM_E_MAXERROR	0XC004C84A	Error while comparing the Value with the upper range
CDBM_E_RANGE_ERROR	0XC004C84B	Value out of Range
CDBM_E_TABLE_TYPE1	0XC004C84C	Table type 1 doesn't have a unique record length over all records
CDBM_E_TABLE_TYPE3_ADDREC	0XC004C84D	Table type 3 doesn't allow to insert more than one Record
CDBM_E_TABTYPE1	0XC004C84E	It's not allowed to insert more Records than structure definitions in Table Type 1
CDBM_E_TOGGLE_NOT_FOUND	0XC004C84F	Could not find the string for this value in the list of valid toggle strings
CDBM_E_TOGGLE_VALUE_IS_EMPTY_STRING	0XC004C850	The toggle string for this value is empty.
CDBM_VARIANT2BYTEARRAY_ERROR	0XC004C851	Error during conversion of Variant to byte array
CDBM_E_SET_ELEM_PROP_DEPENDENCY	0XC004C852	The Toggle Type needs also the additional string and the additional number entries in the Method
CDBM_E_TABTYPE1_REC_DOESNT_CORRESPOND_WITH_ELEMENT	0XC004C853	When reading the records of Table type 1 elementwise the record number must correspond with the element number
CDBM_TABTYPE1_NO_DATA_FOUND_FOR_RECORD	0XC004C854	When reading the records of Table type 1 and structure definitions are present it's assumed that for each structure element a corresponding record must exist
CDBM_E_TABTYPE1_WRITE_ELEMENT_NE_RECORD	0XC004C855	When writing the records of Table type 1 elementwise and structure definitions are present it's only allowed to write the corresponding element number in each record
CDBM_E_TABTYPE1_WRITE_ELEMENT_NOT_FOUND	0XC004C856	When writing the records of Table type 1 with an array and structure definitions are present it's assumed that a corresponding element number of this record exist
CDBM_I_TABLE_NAME_EXCEEDS_RCS_RANGE	0X4004C857 (Informational)	The Table name exceeds the maximum length of RCS compatible Table names
CDBM_W_CUT_STRING	0X8004C858 (Warning)	The string exceeds the maximum length and will be limited to the maximum length
CDBM_I_STRING_TOO_SHORT	0X4004C859 (Informational)	The string is below the minimum length. The minimum length will be reduced.
CDBM_I_STRING_TOO_LONG	0X4004C85A (Informational)	The string is exceeding the maximum. The maximum length will be extended.
CDBM_E_STRING_TOO_SHORT	0XC004C85B (Error)	The string is below the minimum length.
CDBM_E_STRING_TOO_LONG	0XC004C85C (Error)	The string is exceeding the maximum length
CDBM_E_WRONG_TYPE_FOR_WRITE	0XC004C85D	Writing on the Element type with the given Data type is not implemented
CDBM_E_NO_APPEND_IN_STRUCTURED_RECORDS	0XC004C85E	Method IDbmRecord::AppendData is not allowed for structured records

<b>ODM Error Codes DBM V4</b>		
<b>Error Code (Definition)</b>	<b>Value</b>	<b>Description</b>
CDBM_E_DATA_UNAVAILABLE	0XC004C85F	No data available
CDBM_E_CANT_CONVERT_INT	0XC004C860	Unable to convert the value into the Element type
CDBM_E_DBM_FILE_OVERFLOW	0XC004C861	You try to write a RCS like database which needs too much bytes
CDBM_E_PW_ERROR	0XC004C862	Password not correct
CDBM_E_FILELENGTH_CORRUPT	0XC004C863	The file length doesn't correspond to the length given in the Header.
CDBM_E_STRUCT_TYPE	0XC004C864	Error in the file.
CDBM_E_MD5SUM_INVALID	0XC004C865	MD5 sum invalid
CDBM_E_STRUCT_LENGTH	0XC004C866	Error in the expected and given structure length at a specific offset in the file.
CDBM_E_APPEND	0XC004C867	Append of data is only allowed if the Record contains only one data field and the field type will support this
CDBM_APPEND_NOT_SUPPORTED	0XC004C868	Append of Data not supported by this filed type
CDBM_DATA_TYPE_APPEND_ERROR	0XC004C869	Can't append Data of this type.
CDBM_E_UNSTRUCTURED_TABLE_DOESNT_SUPPORT_LENGTH	0XC004C86A	A Table without structure information doesn't support a record length
CDBM_E_DISABLED_WHILE_TRANSACTION_IS_OPEN	0XC004C86B	The Method is disabled while a transaction is open. Please close this one first and call the Method again.
CDBM_E_UNABLE_TO_CALL_READ_ON_LINKED_LIST	0XC004C86C	The Method is disabled on a LinkedList type. Please use the IRecordCollection on this type.
CDBM_E_ELEMENT_HAS_NO_SUBSTRUCTURE	0XC004C86D	An Element from a Table has no substructure
CDBM_STRUCT_ERROR_FROM_VAR_CHANGE_TYPE	0XC004C86E	Error from calling VariantChangeType
CDBM_E_FOREIGNKEY_DEF	0XC004C86F	The definition of a FOREIGNKEY must contain the name of the related Table in the description and this Table must exist at this time
CDBM_E_FOREIGNKEY_REF_TAB	0XC004C870	The description of a FOREIGNKEY must refer to a Table of type 'eDbmTableTypeLinkedList'
CDBM_E_KEY	0XC004C871	To create a Record Collection with a KEY it's necessary to have the data type KEY at the first position in all Records of the searched Table
CDBM_E_KEY_TABLE_TYPE	0XC004C872	This Method needs a Table of type 'eDbmTableTypeLinkedList'
CDBM_DATATYPE_NOT_IMPLEMENTED	0XC004C873	This data type is currently not implemented
CDBM_INSERT_POS_NOT_FOUND	0XC004C874	The position of the Record where the new one should be inserted wasn't found
CDBM_E_INSERT_REC_QI	0XC004C875	Error during insertion of a Record
CDBM_E_TAB_PROP	0XC004C876	Invalid Property in Table
CDBM_E_KEY_NOT_FOUND	0XC004C877	The KEY wasn't found in the Table
CDBM_E_KEY_INVALID	0XC004C878	The KEY is invalid for this operation

Table 59: ODM Error Codes DBM V4

# 11 Appendix

## 11.1 COB-ID (Predefined Connection Set)

COB ID stands for communication object identifier. This is the 11 bit covering telegram identifier of a CAN telegram. Thereby the upper 4 bits (bit 11 to 8) are the function identifier and the lower 7 bits (bit 7 to bits 0) the bus address of the node.

Broadcast Objects:

Object	Function Code	COB ID hex	COB ID dec	Index in the Object Dictionary
NMT	0000	00H	0	-
SYNC	0001	80H	128	1005H, 1006H, 1007H
TIME STAMP	0010	100H	256	1012H, 1013H

Table 60: COB ID - Broadcast Objects

Peer-to-Peer Objects:

Objects	Function Code	COB ID hex	COB ID dec	Index in the Object Dictionary
Emergency	0001	81H-FFH	129-255	1014H, 1015H
PDO 1 (tx)	0011	181H-1FFH	385-511	1800H (1A00H)
PDO 1 (rx)	0100	201H-27FH	513-639	1400H (1600H)
PDO 2 (tx)	0101	281H-2FFH	641-767	1801H (1A01H)
PDO 2 (rx)	0110	301H-37FH	769-895	1401H (1601H)
PDO 3 (tx)	0111	381H-3FFH	897-1023	1802H (1A02H)
PDO 3 (rx)	1000	401H-47FH	1025-1151	1402H (1602H)
PDO 4 (tx)	1001	481H-4FFH	1153-1279	1803H (1A03H)
PDO 4 (rx)	1010	501H-57FH	1281-1407	1403H (1603H)
SDO (tx)	1011	581H-5FFH	1409-1535	1200H
SDO (rx)	1100	601H-67FH	1537-1663	1200H
NMT Error Control	1110	701H-77FH	1793-1919	1016H, 1017H

Table 61: COB ID - Peer-to-Peer Objects

## 11.2 User Rights

User-rights are set within the FDT-container. Depending on the level the configuration is accessible by the user or read-only.

To access the **Settings, Configuration, Device Description** and **Diagnosis** panes of the CANopen Slave DTM you do not need special user rights. Also all users can select the decimal or hexadecimal Display mode or sort table entries.



**Note:** To edit, set or configure the parameters of the **Settings** and **Configuration** panes, you need user rights for *Maintenance*, for *Planning Engineer* or for *Administrator*.

The **Device Description** panes do not contain any editable elements. The indicated values in are only for information purposes.

The following tables give an overview of the user right groups and which user rights you need to configure the single parameters.

### 11.2.1 Settings

	Observer	Operator	Maintenance	Planning Engineer	Administrator
<i>Driver</i>	D	D	X	X	X
<i>Selecting the Driver</i>	-	-	X	X	X
<i>Configuring the Driver</i>	-	-	X	X	X
<i>Device Assignment</i>	D	D	X	X	X
<i>Scanning for Devices</i>	-	-	X	X	X
<i>Selecting the Device (with or without firmware)</i>	-	-	X	X	X
<i>Selecting the Device once more (with Firmware)</i>	-	-	X	X	X
<i>Firmware Download</i>	D	D	X	X	X

Table 62: Settings ( D = Displaying, X = Editing, Configuring)

### 11.2.2 Configuration

	Observer	Operator	Maintenance	Planning Engineer	Administrator
Overview Configuration	D	D	X	X	X
General Settings	D	D	X	X	X
Special Function Objects	D (X)	D (X)	X	X	X
Object Dictionary	D (X)	D (X)	X	X	X
Process Data Objects	D (X)	D (X)	X	X	X
PDO Properties	D (X)	D (X)	X	X	X
PDO Mapping	D	D	X	X	X
Address Table	D	D	X	X	X

Table 63: Configuration (D = Displaying, X = Editing, Configuring)

## 11.3 References

- [1] Device Type Manager (DTM) Style Guide, Version 1.0 ; FDT-JIG - Order No. <0001-0008-000>

## 12 Lists

### 12.1 List of Figures

Figure 1: Dialog Structure of the CANopen Slave DTM	13
Figure 2: Navigation Area	14
Figure 3: Status Bar – Status Fields 1 to 6	17
Figure 4: Status Bar Display Examples	17
Figure 5: Navigation Area - Settings	24
Figure 6: Driver Selection List	25
Figure 7: Driver Selection List - recognized or installed Drivers	25
Figure 8: Driver Selection List - Driver/s selected	25
Figure 9: Navigation Area – Settings	26
Figure 10: netX Driver > USB/RS232 Connection	30
Figure 11: netX Driver > TCP Connection	33
Figure 12: Device Assignment - detected Devices (* The name of the device class is displayed.) – Example for a device without firmware	36
Figure 13: Device Assignment - detected Devices (* The name of the device class is displayed.) Example for Devices without Firmware	38
Figure 14: Device Assignment - Selecting the Device (* The name of the device class is displayed.) – Example for a device without firmware / one Device is selected	39
Figure 15: Device Assignment - Selecting the Device (* The name of the device class is displayed.) – Example for Devices with and without Firmware / one Device is selected	40
Figure 16: Device Assignment - Selecting the Device (* The name of the device class is displayed.) – Example for a device with firmware / one Device is selected	41
Figure 17: Firmware Download	42
Figure 18: Selection Window Select Firmware File	43
Figure 19: Request Select Firmware File - Example	44
Figure 20: Firmware Download – Download	45
Figure 21: Firmware-Download – Question)	45
Figure 22: Clock Symbol and Hook Symbol green	46
Figure 23: Firmware Download - Progress Bar	46
Figure 24: Navigation Area - Configuration	47
Figure 25: General Settings (Slave DTM at the Root-Line (Stand-Alone Slave))	48
Figure 26: General Settings (Slave DTM at the Master busline)	48
Figure 27: Special Function Objects - Synchronization Telegram	50
Figure 28: Special Function Objects - Time Stamp Message	51
Figure 29: Special Function Objects - Emergency Message	51
Figure 30: Object Dictionary - Filter Object Configuration	52
Figure 31: Object Dictionary - Object Configuration	53
Figure 32: Object Dictionary - Data selected Object	55
Figure 33: Process Data Objects > PDO Properties - PDO Type	56
Figure 34: Process Data Objects > PDO Properties - PDO Configuration	56
Figure 35: Process Data Objects > PDO Properties - Data selected Object	58
Figure 36: Process Data Objects > PDO Mapping - Filter PDO Type	59
Figure 37: Process Data Objects > PDO Mapping - Object Dictionary Area	59
Figure 38: Process Data Objects > PDO Mapping - Mappable Objects	59
Figure 39: Process Data Objects > PDO Mapping - Filter Mapped Objects	60
Figure 40: Process Data Objects > PDO Mapping - Mapped Objects	60
Figure 41: Configuration > Address Table	61
Figure 42: Configuration > Address Table - Display Mode	62
Figure 43: Navigation Area - Device Description	63
Figure 44: Navigation Area - Diagnosis	65

---

Figure 45: General Diagnosis	66
Figure 46: Firmware Diagnosis (*The name of the Firmware is displayed.)	69
Figure 47: Extended Diagnosis > [Folder Name] > Task Information Example Display	71
Figure 48: Extended Diagnosis > [Folder Name] > IniBatch Status Example Display	72
Figure 49: Extended Diagnosis > CANOPEN_SLAVE > Common Diagnostic	73
Figure 50: Extended Diagnosis > CANOPEN_SLAVE > Commands	74
Figure 51: Extended Diagnosis > CANOPEN_APS > Slave Configuration	76
Figure 52: Extended Diagnosis > CANOPEN_APS > Commands	77
Figure 53: Navigation Area - Tools	78
Figure 54: Packet Monitor	79
Figure 55: Send > Packet Header and Send Data	80
Figure 56: Packet Header and Receive Data	81
Figure 57: IO Monitor	83

## 12.2 List of Tables

Table 1: Descriptions Dialog Panes	6
Table 2: General Device Information	14
Table 3: Overview Dialog Panes	15
Table 4: OK, Cancel, Apply and Help	16
Table 5: Selecting, inserting, deleting Table Line	16
Table 6: Status Bar Icons [1]	17
Table 7: Getting Started - Configuration Steps (Slave DTM at the Root-Line (Stand-Alone Slave))	19
Table 8: Getting Started - Configuration Steps (Slave DTM at the Master busline)	21
Table 9: Descriptions Settings	24
Table 10: Driver Selection List Parameters	25
Table 11: Parameters netX Driver > USB/RS232 Connection	31
Table 12: Parameters netX Driver > TCP Connection	34
Table 13: Parameters of the Device Assignment	37
Table 14: Parameter Firmware Download	42
Table 15: Parameters Select Firmware File	43
Table 16: Dialog Panes Configuration	47
Table 17: General Settings	49
Table 18: Special Function Objects - Synchronization Telegram	50
Table 19: Special Function Objects - Time Stamp Message	51
Table 20: Special Function Objects - Emergency Message	51
Table 21: Object Dictionary - Filter Object Configuration	52
Table 22: Object Dictionary - Object Configuration	54
Table 23: Object Dictionary - Data selected Object	55
Table 24: Process Data Objects > PDO Properties - PDO Configuration (examine)	57
Table 25: Process Data Objects > PDO Properties - Data selected Object	58
Table 26: Process Data Objects > PDO Mapping - Mappable Objects	59
Table 27: Process Data Objects > PDO Mapping - Mapped Objects	60
Table 28: Address Table Pane Parameters - Inputs / Outputs	61
Table 29: Descriptions of the Dialog Panes Device Description	63
Table 30: General Device Information	64
Table 31: Device Description – EDS Viewer	64
Table 32: Descriptions of the Diagnosis Panes	65
Table 33: Indication General Diagnosis	67
Table 34: Parameter General Diagnosis	68
Table 35: Description Table Task Information	69
Table 36: Descriptions of the Dialog Panes Extended Diagnosis	70
Table 37: Extended Diagnosis > [Folder Name] > Task Information	71
Table 38: Extended Diagnosis > [Folder Name] > IniBatch Status	72
Table 39: Extended Diagnosis > CANOPEN_SLAVE > Common Diagnostic	73
Table 40: Extended Diagnosis > CANOPEN_SLAVE > Commands	75
Table 41: Extended Diagnosis > CANOPEN_APS > Slave Configuration	76
Table 42: Extended Diagnosis > CANOPEN_APS > Commands	77
Table 43: Descriptions of the Diagnosis Panes	78
Table 44: Descriptions Packet Header	80
Table 45: Descriptions Packet Header	81
Table 46: Overview Error Codes and Ranges	88
Table 47: RCX General Task Errors	89
Table 48: RCX Common Status & Errors Codes	90
Table 49: RCX Status & Error Codes	91
Table 50: RCX Status & Error Codes Slave State	91
Table 51: ODM Error Codes - General ODM Error Codes	92

---

Table 52: ODM Error Codes - General ODM Driver Error Codess	93
Table 53: cifX Driver Specific ODM Error Codes	96
Table 54: Generic Error Codes	97
Table 55: Generic Driver Error Codes	98
Table 56: Generic Device Error Codes	99
Table 57: CIFX API Transport Error Codes	100
Table 58: CIFX API Transport Header State Error Codes	100
Table 59: ODM Error Codes DBM V4	104
Table 60: COB ID - Broadcast Objects	105
Table 61: COB ID - Peer-to-Peer Objects	105
Table 62: Settings ( D = Displaying, X = Editing, Configuring)	106
Table 63: Configuration (D = Displaying, X = Editing, Configuring)	106

## 13 Glossary

### COB-ID

Communication Object Identifier. Identifier of the CAN telegram.

### DTM

Device Type Manager

The Device Type Manager (DTM) is a software module with graphical user interface for the configuration and/or for diagnosis of devices.

### EDS

An Electronic Data Sheet (EDS) provides information necessary to access and alter the configurable parameters of a device. An Electronic Data Sheet (EDS) is an external file that contains information about configurable attributes for the device, including object addresses of each parameter. The application objects in a device represent the destination addresses for configuration data. These addresses are encoded in the EDS.

### FDT

Field Device Tool

FDT specifies an interface, in order to be able to use DTM (Device Type Manager) in different applications of different manufacturers.

### Master

Master devices initiate the data exchange at the bus.

### Node ID

The Node ID is the network address of the device. The network address serves to distinguish itself from other devices on the network. Therefore a unique address must be assigned to each device.

### Slave

Slave devices are configured by the Master and perform then the communication.

## 13.1 Contacts

### Headquarters

#### Germany

Hilscher Gesellschaft für  
Systemautomation mbH  
Rheinstrasse 15  
65795 Hattersheim  
Phone: +49 (0) 6190 9907-0  
Fax: +49 (0) 6190 9907-50  
E-Mail: [info@hilscher.com](mailto:info@hilscher.com)

#### Support

Phone: +49 (0) 6190 9907-99  
E-Mail: [de.support@hilscher.com](mailto:de.support@hilscher.com)

### Subsidiaries

#### China

Hilscher Systemautomation (Shanghai) Co. Ltd.  
200010 Shanghai  
Phone: +86 (0) 21-6355-5161  
E-Mail: [info@hilscher.cn](mailto:info@hilscher.cn)

#### Support

Phone: +86 (0) 21-6355-5161  
E-Mail: [cn.support@hilscher.com](mailto:cn.support@hilscher.com)

#### France

Hilscher France S.a.r.l.  
69500 Bron  
Phone: +33 (0) 4 72 37 98 40  
E-Mail: [info@hilscher.fr](mailto:info@hilscher.fr)

#### Support

Phone: +33 (0) 4 72 37 98 40  
E-Mail: [fr.support@hilscher.com](mailto:fr.support@hilscher.com)

#### India

Hilscher India Pvt. Ltd.  
New Delhi - 110 025  
Phone: +91 11 40515640  
E-Mail: [info@hilscher.in](mailto:info@hilscher.in)

#### Italy

Hilscher Italia srl  
20090 Vimodrone (MI)  
Phone: +39 02 25007068  
E-Mail: [info@hilscher.it](mailto:info@hilscher.it)

#### Support

Phone: +39 02 25007068  
E-Mail: [it.support@hilscher.com](mailto:it.support@hilscher.com)

#### Japan

Hilscher Japan KK  
Tokyo, 160-0022  
Phone: +81 (0) 3-5362-0521  
E-Mail: [info@hilscher.jp](mailto:info@hilscher.jp)

#### Support

Phone: +81 (0) 3-5362-0521  
E-Mail: [jp.support@hilscher.com](mailto:jp.support@hilscher.com)

#### Korea

Hilscher Korea Inc.  
Suwon, 443-810  
Phone: +82-31-204-6190  
E-Mail: [info@hilscher.kr](mailto:info@hilscher.kr)

#### Switzerland

Hilscher Swiss GmbH  
4500 Solothurn  
Phone: +41 (0) 32 623 6633  
E-Mail: [info@hilscher.ch](mailto:info@hilscher.ch)

#### Support

Phone: +49 (0) 6190 9907-99  
E-Mail: [ch.support@hilscher.com](mailto:ch.support@hilscher.com)

#### USA

Hilscher North America, Inc.  
Lisle, IL 60532  
Phone: +1 630-505-5301  
E-Mail: [info@hilscher.us](mailto:info@hilscher.us)

#### Support

Phone: +1 630-505-5301  
E-Mail: [us.support@hilscher.com](mailto:us.support@hilscher.com)