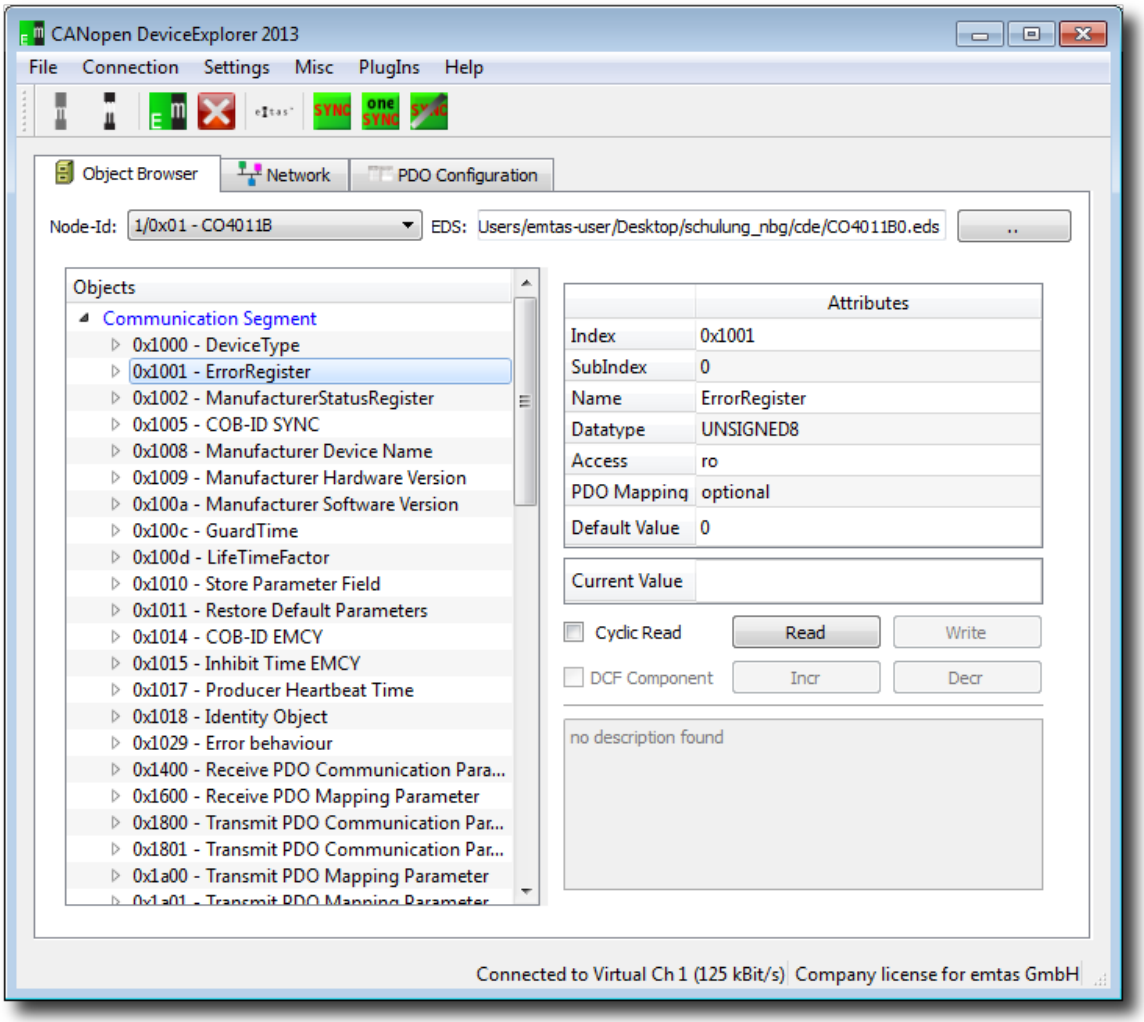
e m t a s ™

User Manual
CANopen DeviceExplorer



Version history

Version	Changes	Date	Editor
V1.1.5	Additional PlugIns added	2013/06/26	ged
V1.3	Additional Features added	2013/09/20	ged
V2.0	Additional PlugIns and Features added	2014/02/20	ged

Disclaimer

The programs and documentations supplied by emtas GmbH are created with due diligence, checked and tested on several applications. Nevertheless, emtas GmbH can not take over guarantee and liability that the program and the documentation are error-free respective are suitable to serve the special purpose. For consequential damages, which emerged on the strength of use the program, every legal responsibility or liability is excluded.

emtas GmbH has the right to modify the products described or their documentation at any time without prior warning, as long as these changes are made for reasons of reliability or technical improvement.

Copyright

Fritz-Haber-Str. 9
D-06217 Merseburg
Germany
Tel. +49 3461/79416-0
Fax. +49 3461/79416-10
service@emtas.de
http://www.emtas.de

Table of Contents

1 Introduction

Thank your for using the CANopen DeviceExplorer. The CANopen DeviceExplorer is a versatile tool to test, configure and monitor CANopen devices. The following manual explains the installation and usage of the program.

2 Installation

2.1 Windows

To install the tool on Windows start the setup setup_canopen_deviceexplorer.exe and follow the instructions of the setup. The setup creates a shortcut to start the program. At the first connection to CAN the program requires a license file, which is copied to the program and enables the licensed features.

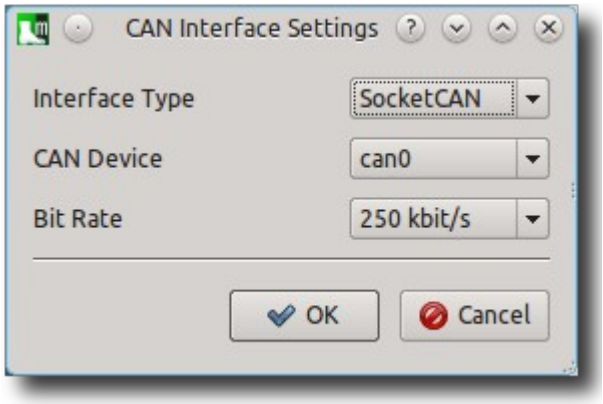
It is possible to use the tool without license file and without CAN interface hardware in a Demo mode. In this mode the tool receives all CAN messages it has sent by itself and additionally there is a virtual CANopen node running on Node-Id 32.

2.2 Linux

To install the tool in Linux just unzip the ZIP file setup_canopen_deviceexplorer.zip into a directory. To start the program run the script CANopenDeviceExplorer.sh in this directory. At the first connection to CAN the program requires a license file, which is copied to the program and enables the licensed features.

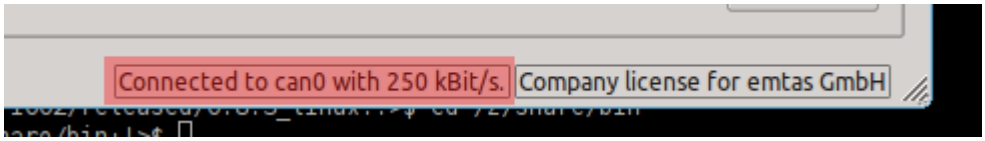
3 First steps

The first step at the very first start of the tool is the configuration of the CAN interface. Open CAN interface settings at the menu entry “Connection → CAN Interface Settings”. The following mask appears.

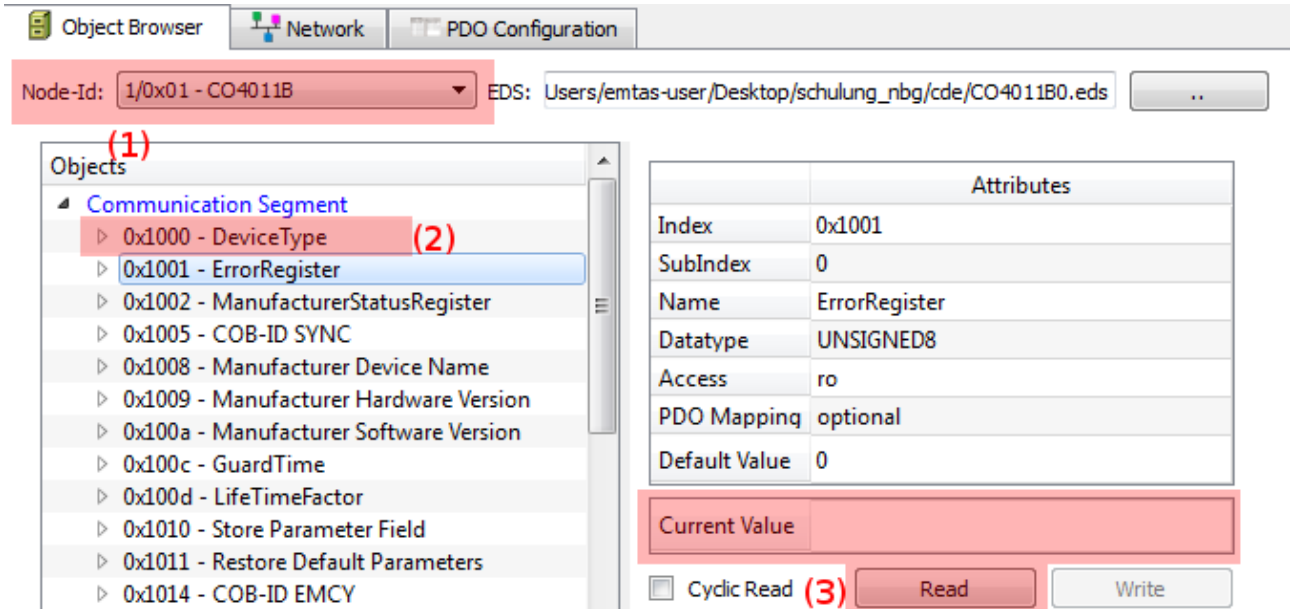


Choose the type of the CAN interface, the name of the CAN device and the bit rate in the CANopen network and confirm the settings with OK.

Connect now the CANopen DeviceExplorer with the CAN interface via “Connection → Connect”. In the status bar you can see now “Connected to “ with the name of the CAN device and the current bitrate.



To test the functionality of the CAN connection, select the “Object Browser” tab. After that configure the node-Id of the device(1) and click in the object tree at the object 0x1000(2) and press “Read”(3) afterwards. Now the object 0x1000[1] is read and the result is displayed at “Current Value”.



In parallel the CAN messages can be watched in CAN view. CAN View can be activated via “PlugIns → CAN View”.

If a valid value has been read, this indicates that the node Id is correct and the CAN connection works. If you do not receive a response, but a SDO timeout, please check the node Id and the CAN bit rate. Unless switched off, the tool saves all settings when quiting the tool, so at next start you can start with the same settings.

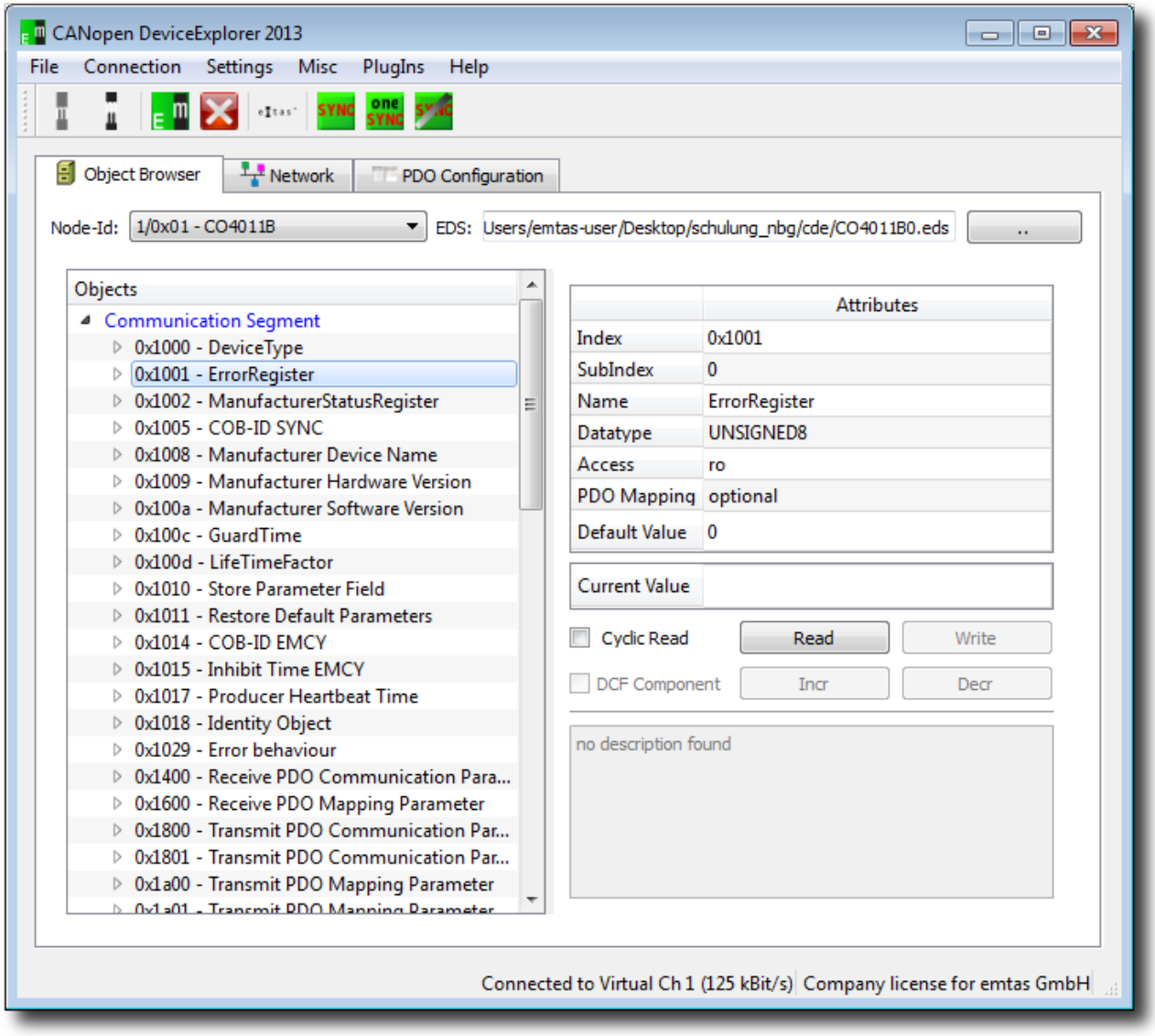
4 Program components

4.1 Object Browser

Using the object browser all objects of a CANopen device can be read or written

1 Device Type … available at each CANopen device

– depending on access rights. An EDS file, which contains the information about the CANopen objects, can be imported for each device in the network. If the global option “Read objects automatically at selection” is active, the objects are read by SDO immediately when they are selected in the object tree, otherwise after pressing the 'Read' button. If no EDS file is available or if any other objects shall be accessed, it is possible to specify the index, subindex or data type directly in the 'Attributes' table. After reading a value it is displayed in the 'Current Value' field in decimal and hexadecimal notation. When writing a value the last value from the 'Current Value' field is transmitted.



Meaning of the buttons

- **Read**

  Read the value of the currently selected object

- **Write**

  Write the value of the currently selected object

- **Incr**

  Increment the value of the currently selected object by 1.

- **Decr**

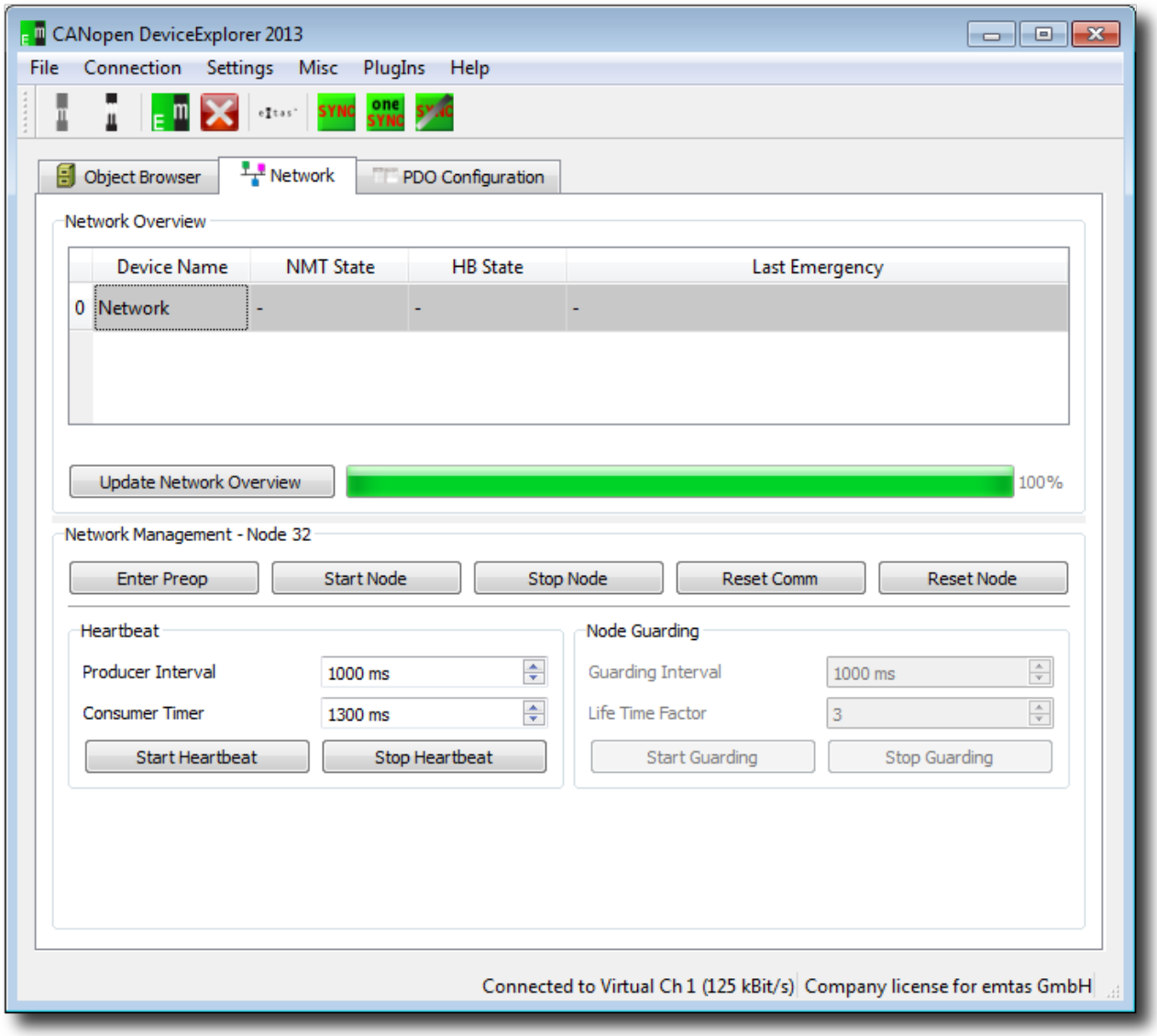
  Decrement the value of the currently selected object by 1.

- **Cyclic Read**

  Cyclic read of the object (interval: 1s) and display of the value in the Current Value field

4.2 Network Overview

The Network tab shows all CANopen devices in the network with their NMT states and their last received Emergency messages. Additionally, NMT master commands to start or stop distinct devices or for the complete network can be sent. The tab also provides masks to configure Heartbeat and Node Guarding and there monitoring by the tool.



- **Network Overview**

  The table lists all device in the network. If the devices send Heartbeat messages these are detected automatically by the CANopen DeviceExplorer. Alternatively, using “Update Network Overview” scans for devices in the CANopen network. The search is done by SDO read access to object 0x1000 of all nodes.

  The columns of the table have the following meanings:

  - Device Name – Name of the device – read from EDS file
  - NMT State – current NMT state
  - HB State - State of Heartbeat or NodeGuarding monitoring
  - Last Emergency – last Emergency message sent by the node

  A right click in the table opens a pop-up menu with access to the

following actions: Start a node, Reset communication, Enter Pre-operational and load an EDS file.

- **Network Management**

  The CANopen NMT master commands Start, Enter Pre-Operational, Stop, Reset Communication and Reset Node can be send to the node which is selected in network overview. If the complete network is selected, the commands are sent to all nodes (id 0).

- **Heartbeat**

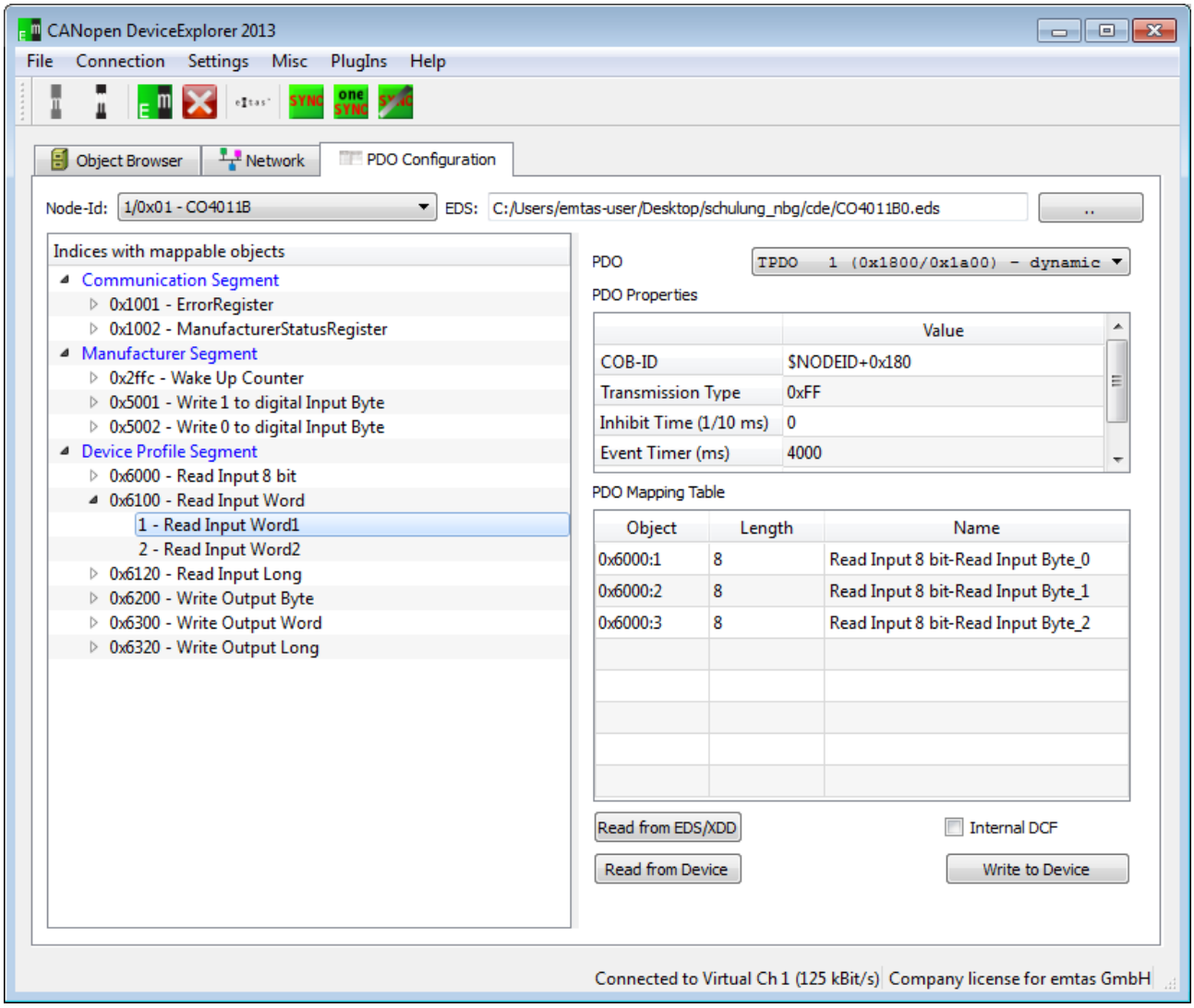
  Configuration of the Heartbeat producer interval and the Heartbeat consumer time for the selected node. After the start of the heartbeat monitoring errors(loss of heartbeat messages) are displaed in the column “HB State”. The value for the producer interval is send by SDO to the device. The configuration mask is only active, if the device supports Heartbeat and if the object 0x1017 exists in the EDS file.

- **Node Guarding**

  Configuration of Node Guarding for the selected node. After the start of the Node Guarding monitoring errors(Loss of messages, Guarding Lost or Connection Lost) are shown in the column “HB state”. The values for the guarding interval and the life time factor are written by SDO to the device. The Node Guarding configuration is only active if the device supports Node Guarding and if it is indicated in the EDS file of the device.

4.3 PDO Configuration

The PDO Configuration tab can be used to configure PDOs with static or dynamic mapping. The mapping of dynamic changeable PDOs can be configured by Drag&Drop.



According to the object browser in the top row the node Id and the EDS file of the device can be configured. The object tree on the left side only contains objects that can be mapped into PDOs. On the right side of the tab there is a PDO with its properties and its mapping table.

- **Read from EDS**

  Read the properties and the mapping of the PDO from the device description file.

- **Read from Device**

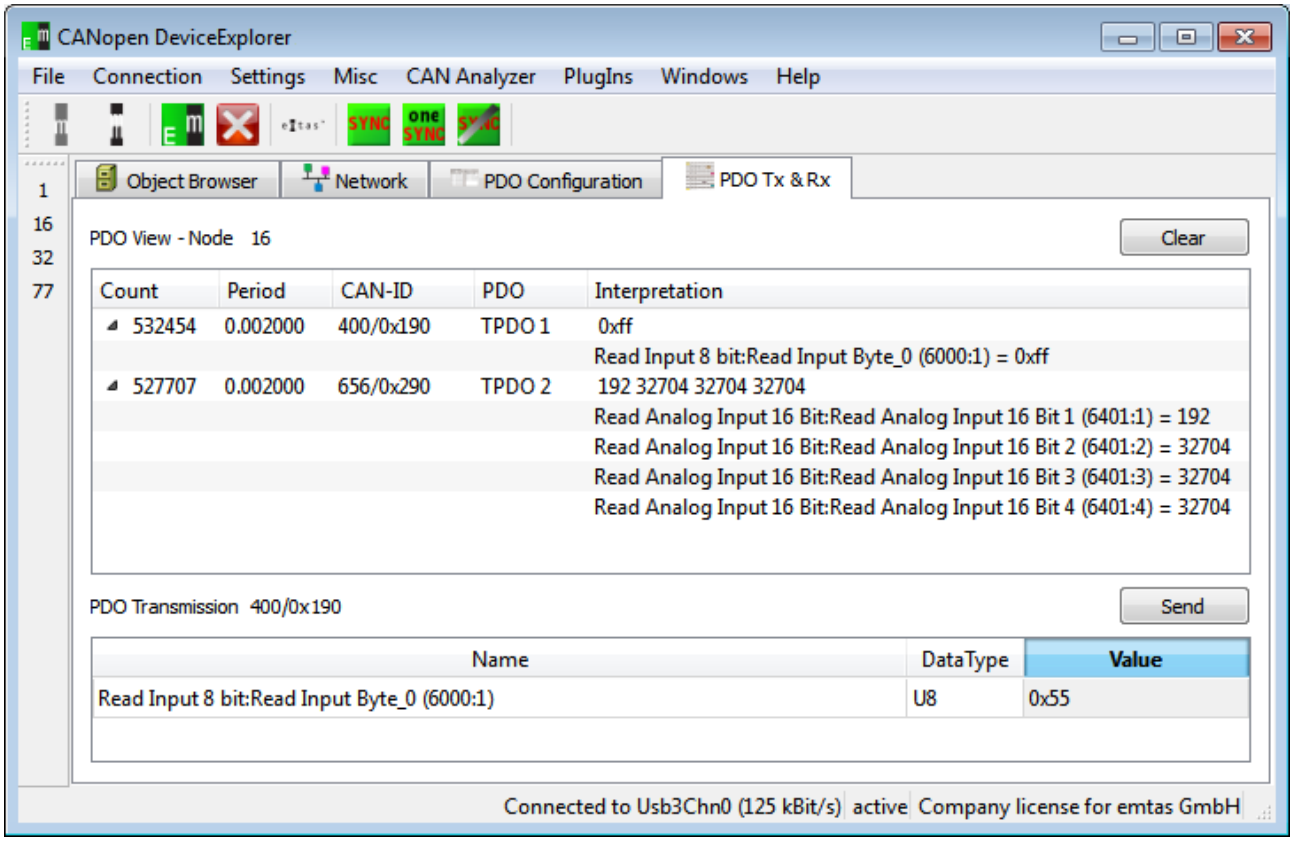
  Read the properties and the mapping of the PDO from the device by SDO.

- **Write to Device**

  Download of properties and mapping table to the device by SDO. Specialities of the CANopen protocol are considered, like e.g. that a PDO has to be deactivated before it can be configured.

4.4 PDO Tx & Rx

The PDO Tx&Rx Tab provides the functionality do send and receive PDOs of the active CANopen node. In the upper part of the tab the received PDOs are displayed with the interpreted values and in the lower part PDOs of this node can be sent.



The interpretation is done according to the current PDO mapping of the device.

5 Menu

The menu provides access to various functions and settings of the CANopen DeviceExplorer.

File

- **Load EDS**

  Load an EDS file for the selected node.

- **Device Configuration**
  - **Load DCF**

    Load a DCF file with saved parameter values for the active node.
  - **Save DCF**

    Save a DCF file with current parameter values of the active node
  - **Save Concise DCF**

Save the current parameter values for the active node in a binary DCF file.

- **Projects**
  - **New**

    Create a new empty project
  - **Open**

    Open an existing project
  - **Save**

    Save the current project using the current name. With new projects the tool asks for a name of the project file.
  - **Recent Projects**

    List of 10 recently used projects.
- **Quit**

  Quit the application.

Connection

- **CAN Interface Settings**

  Dialog to configure the CAN interface and the bit rate
- **Connect**

  Connect to CAN using the configured interface
- **Disconnect**

  Disconnect from CAN

Settings

- **Options**

  Open the options dialog to configure various settings of the program.
- **Save**

  Save the current settings. If “Settings → Options → Save settings automatically at exit” is enable, the settings are automatically saved when the program is quit.
- **Export Settings**

  Export of the current settings into a configuration file. It can be used to store various settings of different use cases.
- **Import Settings**

Import of the settings from a configuration file.

- **Update Licence File**

  Dialog to select a new license file. The content of the license file can be viewed and it can be imported.

- **Check for Updates**

  Query the web server for updates of the tool. Beside the IP address no additional data is transmitted.

Misc

- **EDS Viewer**

  Open the EDS file in the text viewer.

- **HTML Project Documentation**

  Creates an HTML project documentation. It contains the names and paths to EDS files of all projects and the current values of all write-able resp. configurable objects of the devices.

- **Update Device Configuration**

  Open the dialog to Send/Read/Store/Restore of certain objects of the CANopen devices.

- **Update Network Configuration**

  Open the dialog to send the device configurations to multiple devices in the network.

CAN Analyzer

This menu contains program components to display and interpret CAN messages.

- **CAN View**
  The CAN View shows received and transmitted CAN messages. To send a CAN message the Transmit table below can be used. The values for CAN-IDs, DLC and data can be specified as decimal values or as hexadecimal values with leading 0x.

  'Type' specifies the format of the CAN message:

dB or empty	CAN base data message (“standard” CAN message)
rB	CAN base RTR message
dE	CAN extended data message
rE	CAN extended RTR message

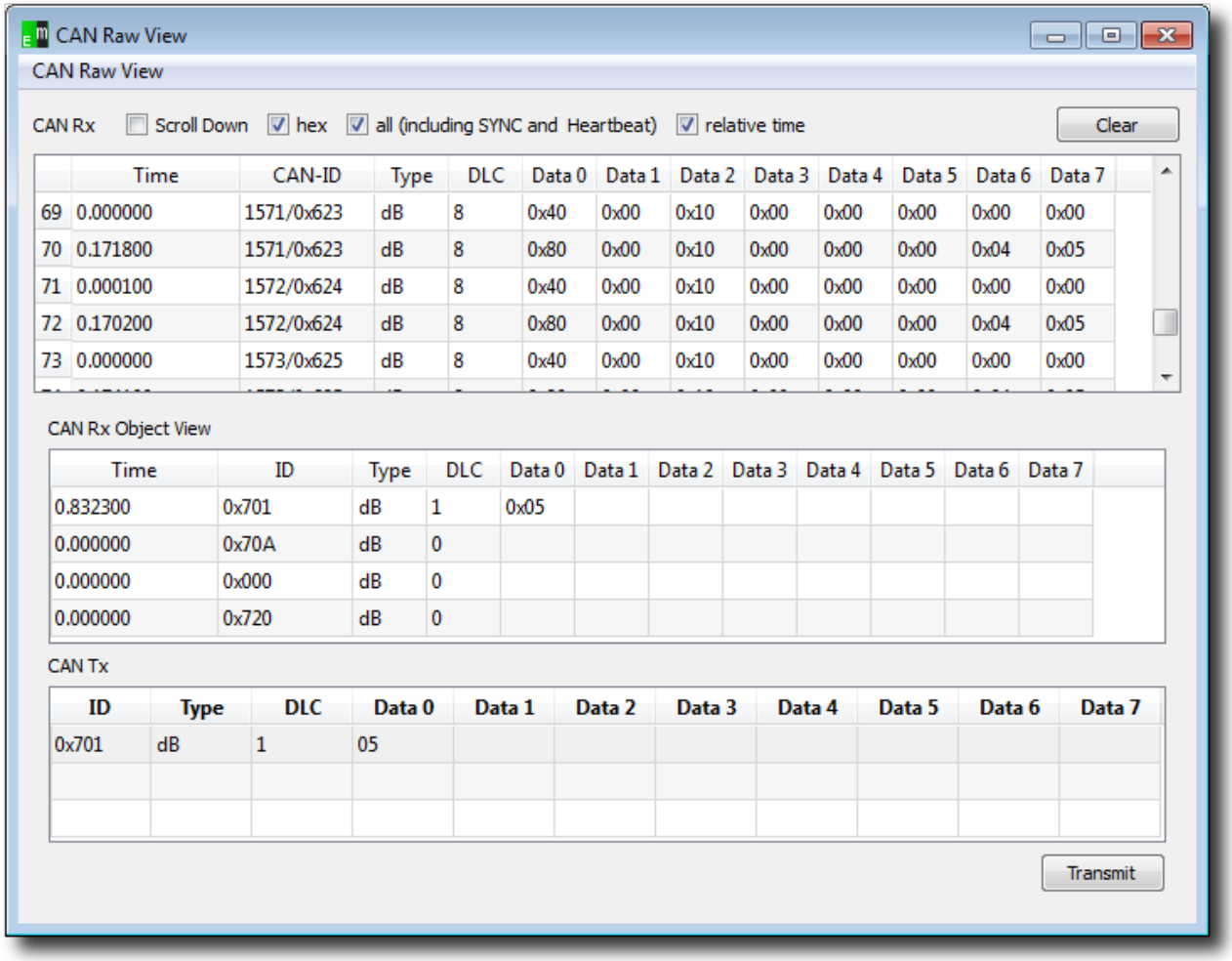
  Up to 16 transmit messages can be defined. The selected one is sent by

clicking on the Transmit button. Cyclic messages can be sent automatically by the tool, if the value in the column 'Interval(ms)' is larger than 0.

In the menu of the CAN View windows a filter for distinct CAN IDs can be defined. A list of CAN IDs can be defined, which either can be displayed or ignored. The list can be a list of single values like (100,200,0x400,0x500) or a range (0x100-0x200) or a combination of both e.g. 1,2,0x300-0x400,720.

The filter type defines the behavior of the filter:

- PASS – only the CAN IDs in the filter list are displayed
- REJCECT – the CAN IDs in the filter list are ignored, all other CAN IDs are displayed



The time stamp of the CAN message can be absolute or relative values and the accuracy depends on the used CAN interfaces and the operation system. For most CAN interfaces no TX time stamp is available.

The recorded CAN messages can be exported into text files by CAN View

→ Export CAN-Logging.

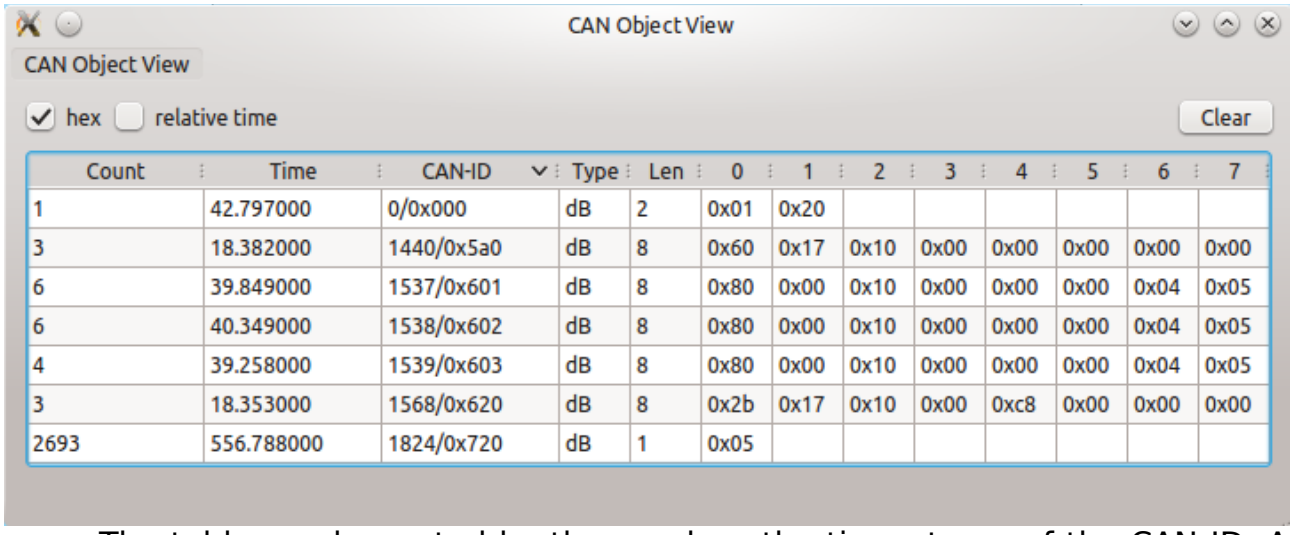
The format of the save text is explained below:

```
3.653302 0x5a0/1440 (8): 43 18 10 03 00 00 00 00
time stamp
           CAN-ID
                    DLC
                         Data in hexadecimal notation
```

The CAN View PlugIn is part of the standard scope of delivery of the CANopen DeviceExplorer.

- **CAN Object View**

  The CAN Object View shows all received CAN messages in the so called 'Object View'. That means all received CAN IDs are shown in a table with the last data and the number of receptions.

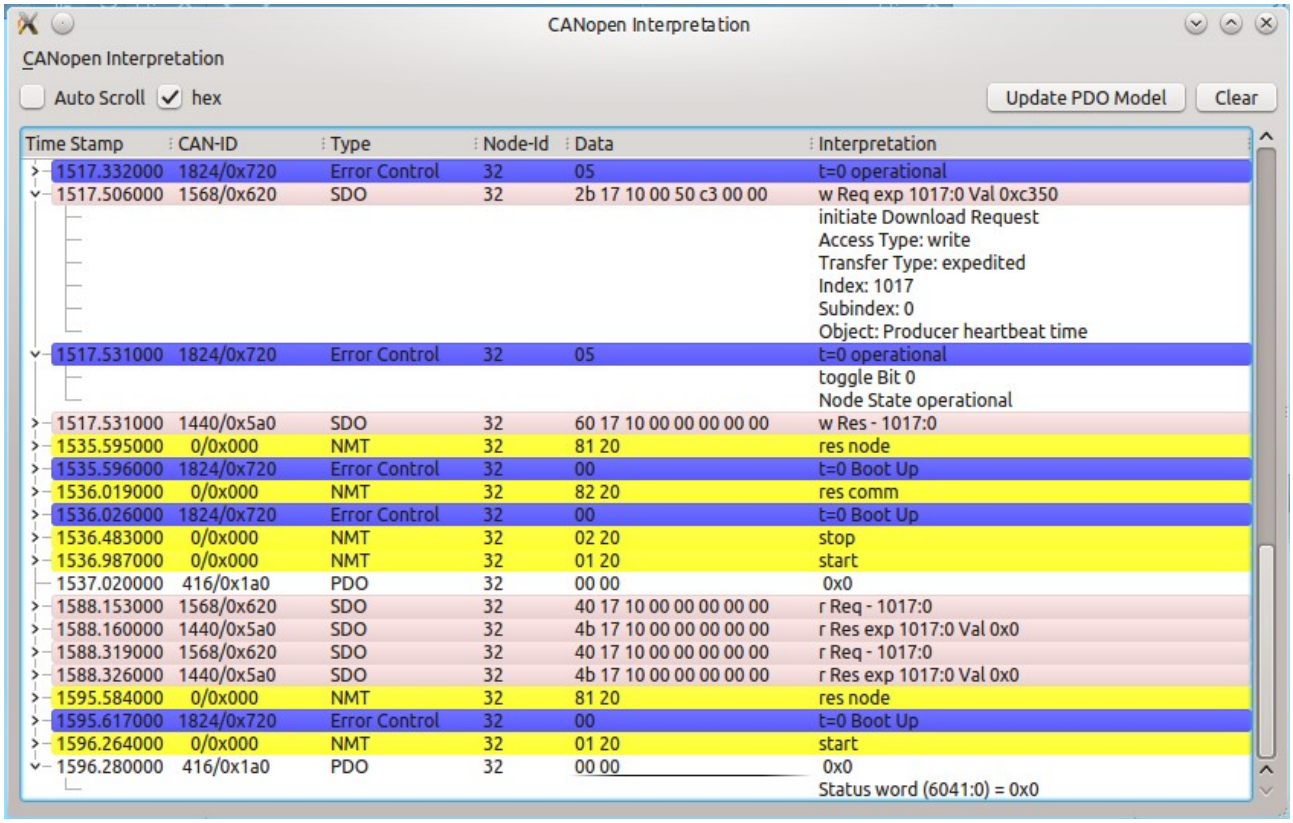


Count	Time	CAN-ID	Type	Len	0	1	2	3	4	5	6	7
1	42.797000	0/0x000	dB	2	0x01	0x20						
3	18.382000	1440/0x5a0	dB	8	0x60	0x17	0x10	0x00	0x00	0x00	0x00	0x00
6	39.849000	1537/0x601	dB	8	0x80	0x00	0x10	0x00	0x00	0x00	0x04	0x05
6	40.349000	1538/0x602	dB	8	0x80	0x00	0x10	0x00	0x00	0x00	0x04	0x05
4	39.258000	1539/0x603	dB	8	0x80	0x00	0x10	0x00	0x00	0x00	0x04	0x05
3	18.353000	1568/0x620	dB	8	0x2b	0x17	0x10	0x00	0xc8	0x00	0x00	0x00
2693	556.788000	1824/0x720	dB	1	0x05							

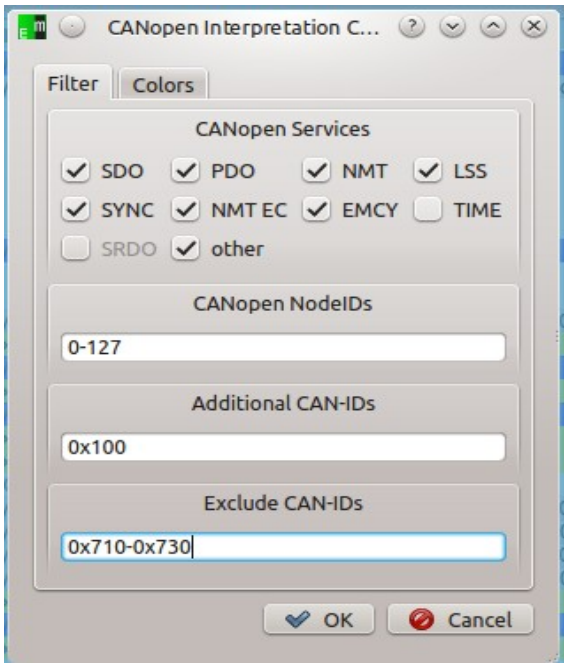
The table can be sorted by the number, the time stamp of the CAN ID. A filter can be configured in the same way as in the CAN View PlugIn. The CAN Object View is included in the standard scope of delivery of the CANopen DeviceExplorer.

- **CANopen Interpretation (optional)**

  The CANopen Interpreter interprets all received CAN messages according to the CANopen protocol and displays the CANopen service of the message, the source or target node ID and a service-specific interpretation in a human-readable format. Additionally the CANopen Interpreter can interpret the content of PDO messages according to the PDO mapping as defined in EDS or DCF files.



A filtering of the CAN message is possible by various criteria:



- CANopen Service – only the enabled services are displayed
- CANopen NodeID – only the selected node IDs are displayed. A definition of ranges like e.g (1,2-30,40) is possible.
- Additional CAN-IDs: CAN IDs, which are rejected by the previous filters can be enabled again.
- Exclude CAN-IDs: CAN IDs, which have passed the previous filters can be filtered out selectively. A definition of ranges like e.g. 100,0x710-0x730 is possible.

Interpreted CANopen messages can be exported as text files and raw CAN loggings can be imported and interpreted as well.

- **CANopen Object View (optional)**

  The CANopen ObjectView combines the CANopen interpretation with an

object view of the CAN messages. All received CAN-IDs are shown with the last interpreted values.

- **EnergyBus Object View (optional)**

  The EnergyBus ObjectView combines the EnergyBus interpretation with an object view of the CAN messages. All received CAN-IDs are shown with the last interpreted values.

- **User Interpretation**

  The user interpretation allows to interpret the CAN message according to the configuration of the user.

- **File Logger**

  The File Logger is able to record CAN messages directly into log files according to certain trigger conditions.

PlugIns

Menu to activate various extensions of the CANopen DeviceExplorer. The availability of the PlugIns depends on the license.

Help

- **Help F1**

  Show the relevant chapter of the manual for the active component.

- **Manual**

  Show the complete manual as PDF file.

- **About**

  Show about dialog including license information.

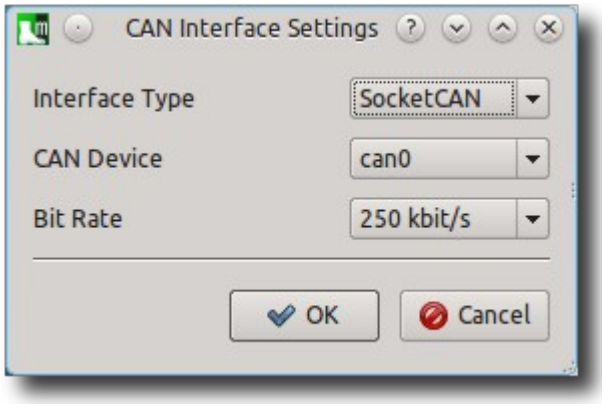
- **About Qt**

  Information about the Qt framework and license information about the used Qt components.

6 Settings

6.1 CAN Settings

The CAN interface can be configured using the CAN settings dialog.



- **Interface Type**

  Selection of the CAN interface. Currently different CAN interface manufacturers are supported on Windows and on Linux.

- **CAN Device**

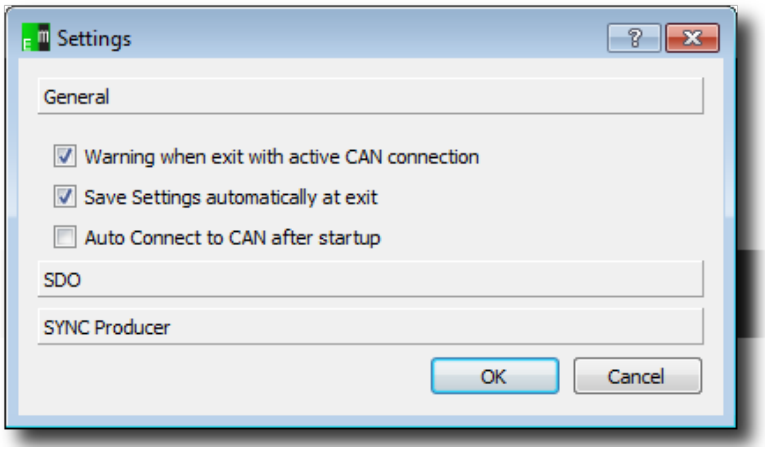
  Depending on the Interface type the available CAN devices are listed here to choose from.

- **Bit Rate**

  Configuration of the bit rate of the CANopen DeviceExplorer. The configured bit rate must match with the bit rate of the CAN network.

6.2 Program Settings

The option dialog provides access to various settings of the CANopen DeviceExplorer.



General Settings

- Warning when exit with active CAN connection
- Save Settings automatically at exit

- Auto Connect to CAN after startup

SDO – Settings

- SDO Timeout for normal SDO access (ms)
- Read objects automatically at selection

SYNC Producer

- COB-ID of SYNC producer
- Interval in milliseconds
- Use Sync Counter
- Max Sync Counter Value

Scripting

- Autostart Script at connect

7 PlugIns

7.1 SDO Object Monitor

The SDO Object Monitor reads objects of various nodes in the CANopen network cyclically and displays the received values or error codes. Objects can be dragged from the object browser into the SDO Object Monitor.

7.2 Data Plotter

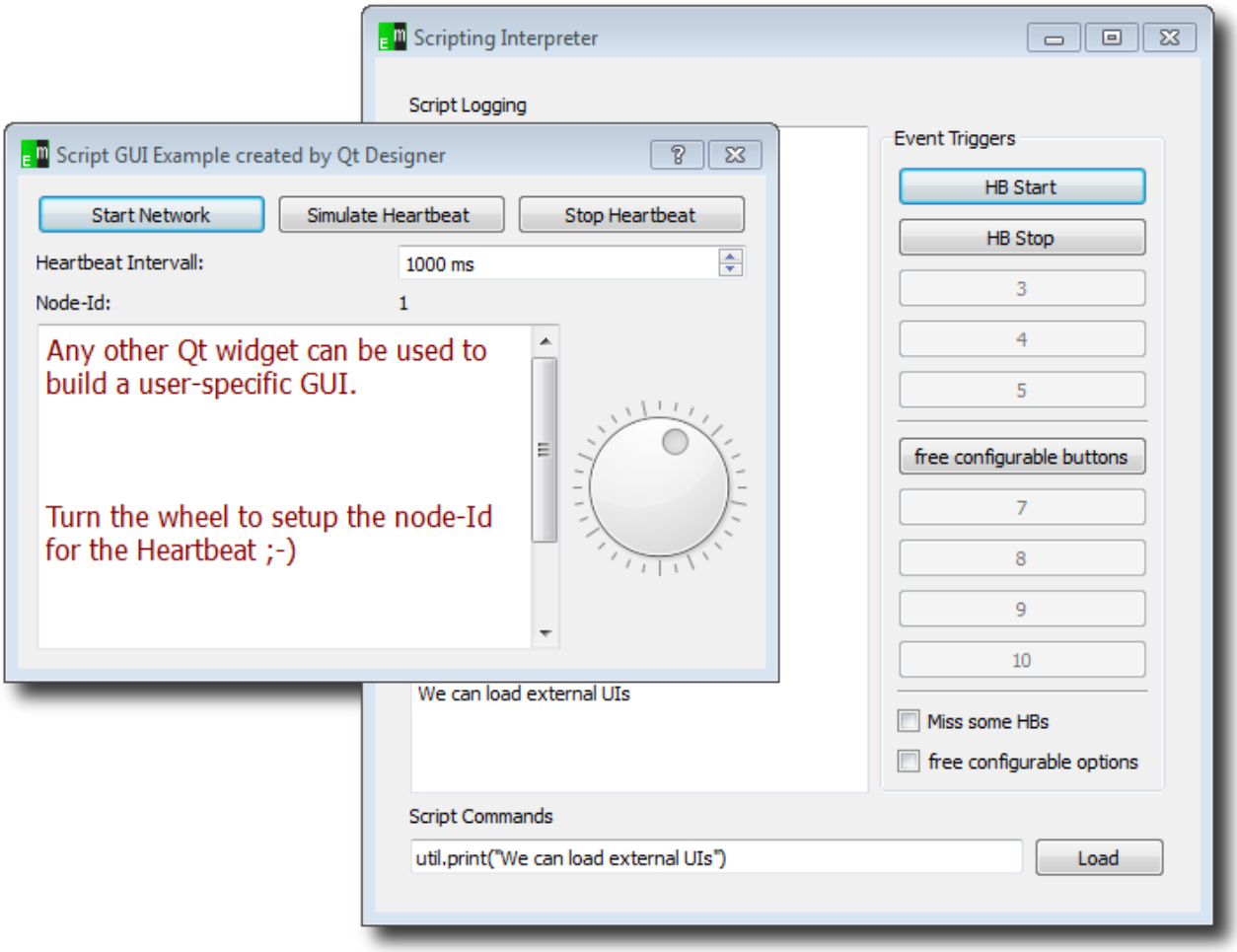
The Data Plotter can visualize the values of an object which is read by the SDO object monitor. Right click on an object in the SDO Object Monitor to add it into the Data Plotter.

7.3 EnergyBus Interpretation (optional)

The optional EnergyBus Interpretation displays the state of EnergyBus devices in an EnergyBus (CiA-454) network.

7.4 CAN/CANopen Scripting (optional)

The Scripting Interpreter provides the possibility to run QtScript(JavaScript) programs with special extensions for CANopen.



Even own user interfaces can be generated by the QtDesigner and used in scripts.

All additional CANopen-specific commands are explained in the separate document "CANopen Scripting Interpreter – API Reference" (cde_script_api.pdf).

The following example shows the usage of a QtScript program:

```
// print something to console and set device to operational
util.print("Test of simple device");
nmt.preopNetwork();
nmt.startNode(32);
i = 0;
util.print("We are in " + util.pwd());

// set node id for SDO access
sdo.setNodeId(32);

// loop over objects 0x4000 to 0x04010
for (object = 0x4000; object < 0x4005; object ++) {
    str = "Test object ";
    str = str + object;
    util.print(str);
```

```
    // write value to object 0x4000..
    result = sdo.write(object, 0x0, UNSIGNED32, i);
    if (result == "SDO_OK") {
        util.print(" Write OK");
    } else {
        util.print(" Write NOT OK");
    }
    // wait a bit to allow device update its internal values
    util.msleep(10);

    // read from 0x4100.. and expect same value
    result = sdo.read(object+0x100, 0x0, 0x07);
    if (result == i) {
        util.print("Read OK");
    } else {
        util.print(" Read NOT OK");
    }
    i++;
}

// user defined function that can be called from Scripting tab
function urk(count) {
    for (i = 0; i < count; i++) {
        nmt.startNetwork();
        nmt.stopNetwork();
        var dlc = 4;
        var canId = 0x100 + i;
        can.sendBaseFrame(canId, dlc, 1 , 2 , 3 ,4 , 0, 0, 0 ,0);
    }
}
// call user defined function
urk(4);
// setup cyclic timer every 2 seconds
timerId = util.every(2000, “urk(10)“ );
```

This example is installed with the CDE as example1.js.

Script commands can be entered into the command line at the bottom of the window. Complete script files can be loaded as well if a path to a script file is specified in the command line. The command line stores it history and using the cursor buttons up and down older commands can be selected again. The command line history is stored when the program is closed.

7.5 LSS Master (optional)

The LSS Master PlugIn provides Layer Setting Services (LSS) according to CiA-305. All LSS commands can be send in the expert view, so that device developers can test their LSS implementations. The Basic View provides an automatic node ID assignment for CANopen devices.

7.6 Process Data Linker (optional)

The Process Data Linker is a powerful extension to define connection between process data of different devices. CANopen-PLCs according to CiA 405 are supported and besides DCF files a PLC variable definition according to IEC 61131 is generated. The Process Data Linker is explained in the section Process Data Linker in detail.

8 Process Data Linker (optional)

Using the Process Data Linker connections between process data of different CANopen devices can be established automatically. After the definition of the connections the Process Data Linker automatically generates a configuration of all PDOs in the CANopen network, which is called PDO Linking.

8.1 Overview

In order to use the Process Data Linker PlugIn all EDS files of the device have to be loaded in the object browser and assigned to the individual node-IDs. An active CAN connection to the devices is only required if the devices shall be configured directly by the CANopen DeviceExplorer, otherwise no CAN connection is necessary to use the Process Data Linker.

If a valid license for the Process Data Linker is available, it can be activate via PlugIns → Process Data Linker. It will open a window with its own menu and 4 tabs.

8.2 Menu of Process Data Linker

Object Linker

- **Generate Linking**

  Generation of PDO configuration based on settings in object matrix and linking overview. After that, the configured PDOs are shown in the “Configured PDOs” tab.

- **Generate Linking & DCF Files**

  Generation of PDO configuration based on settings in object matrix and linking overview. After that, the configured PDOs are shown in the “Configured PDOs” tab. Additionally, the DCF file for each node are generated.

- **Download Configuration**

  Open dialog to configure all connected CANopen devices directly.

- **Close**

  Close Process Data Linker. All settings will remain valid until the CANopen DeviceExplorer is closed.

Settings

- **PLC Configuration**

  Configuration of network variables of CANopen-PLCs, if there are any in the project.

- **Linking Configuration**

  Configuration of various settings that configure the linking algorithm.

File Viewer

- **HTML Project Documentation**

  Open browser to show HTML project documentation.

- **DCF Files**

  Show generated DCF files in text viewer.

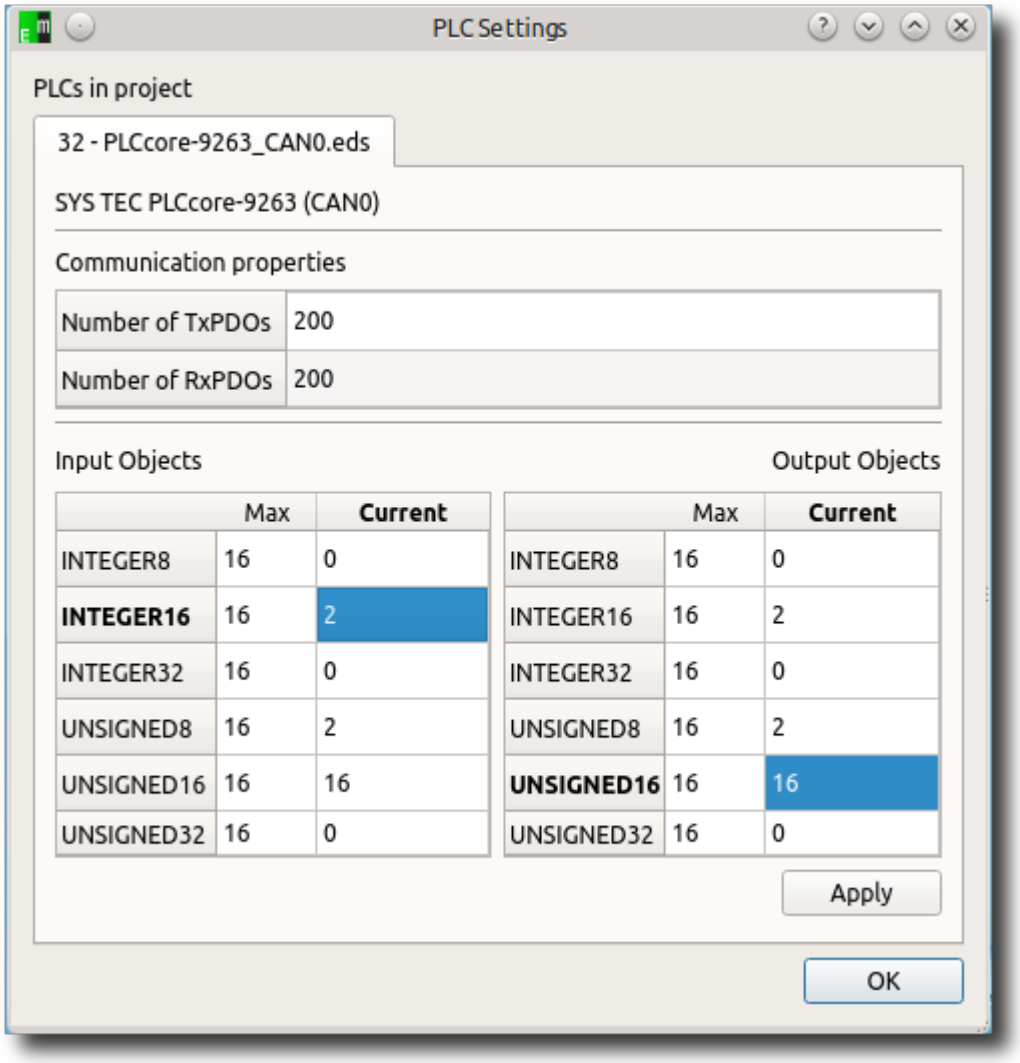
- **PLC Variable Definition**

  Show generated IEC61131 variable definitions in text viewer.

8.3 Usage of CANopen PLCs according to CiA 405

CANopen PLCs according to CiA 405 use so called network variables that are stored in CANopen object dictionary starting at index 0xA000. These network variables are created dynamically and can be renamed according to the application. Open the Dialog “PLC Configuration” to create network variables. Within this dialog the number of network variables for each data type and direction can be specified. The Process Data Linker also supports multiple PLCs in one CANopen network.

Please take into consideration that the definition of Input and Output follows the CiA specifications 405 and 302-4. Thanks means that variables that are received by the PLC are called outputs and variables which are modified and sent by the PLC are called Inputs.



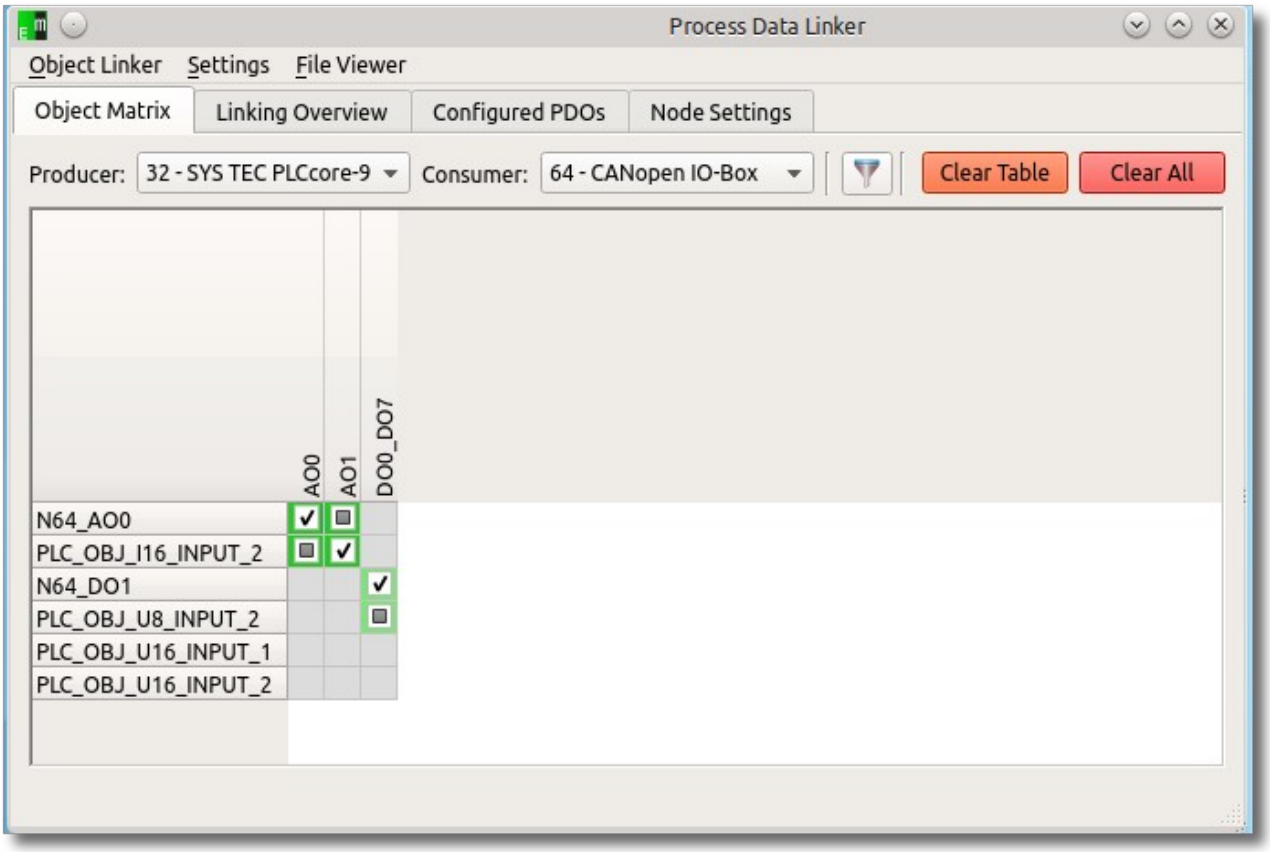
Configuration of number of network variables per PLCs

If no PLC is present in the network, this configuration can be ignored.

8.4 Object Matrix

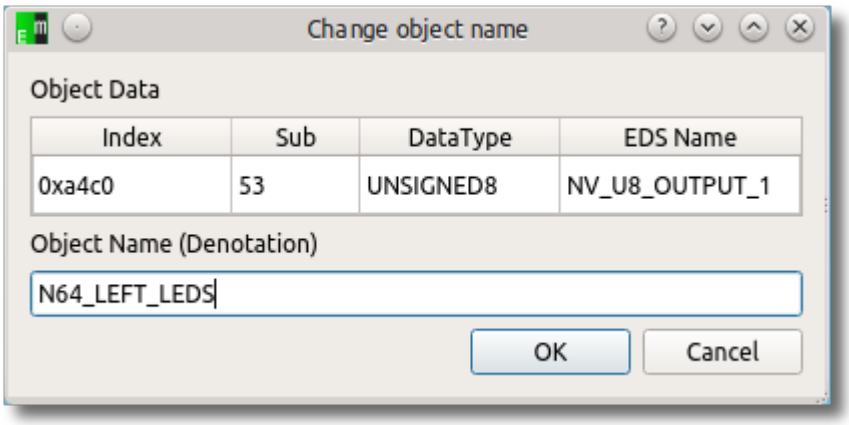
All communication connections are defined in the object matrix as communication pairs between a producer(source, sender) and a consumer(drain, receiver). To configure it, a pair of producer and consumer has to be selected. After that the matrix shows all linkable objects of both devices for the given direction.

If the intersection is white with a green border ☐ a connection between both objects is possible. If the fields are gray no connection is possible. An active connection is marked by a ☑ and an intersection which is partly filled with light gray ▣ indicates that the object at the consumer is already linked with another object. Anyway this connection can be changed by a single click.



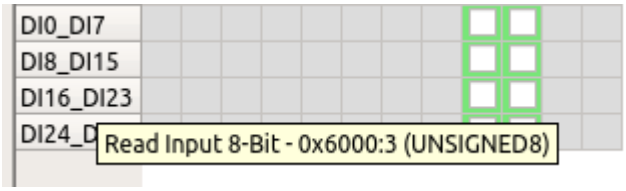
Object Matrix with connected objects

The button “Clear All” deletes all connections in a network and the button “Clear Table” only deletes the connections between the current producer-consumer-pair. A filtering by data types is possible via the filter button  or by a text filter next to it. In addition to that objects can be renamed. This new name will be used in all following masks and it will be written as denotation into the DCF file.

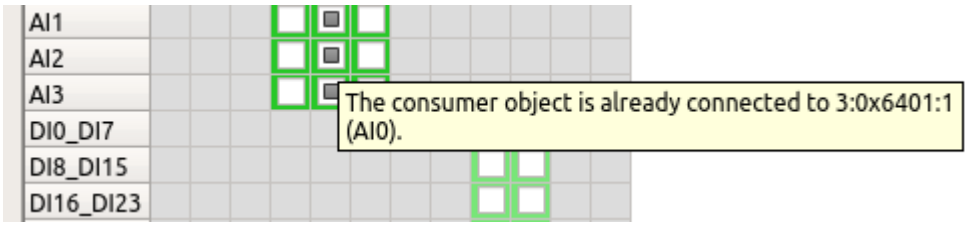


Dialog to rename an object

Additional information are shown as tool tip over an object or an intersection.



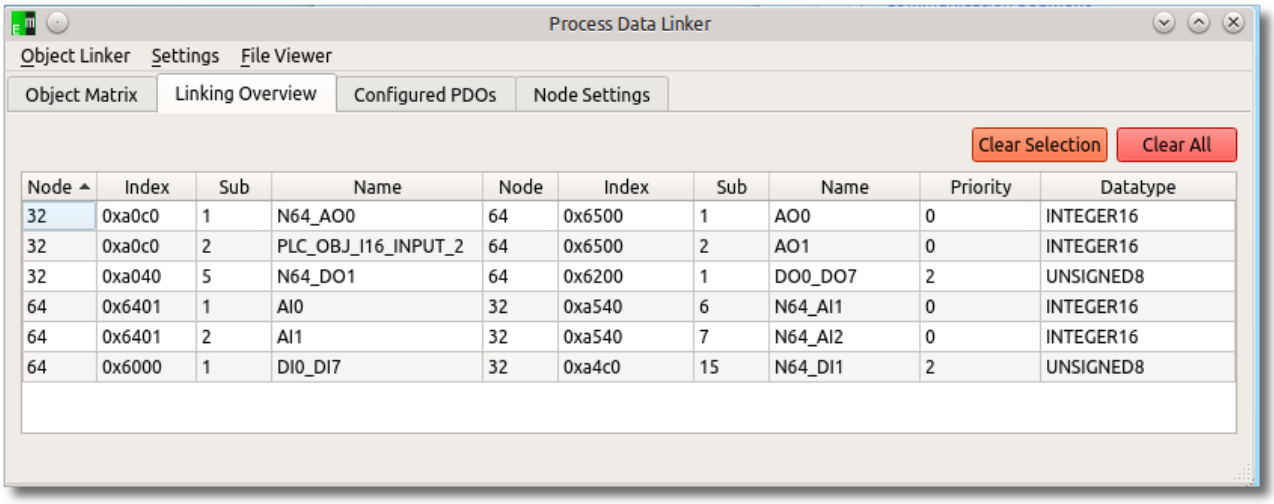
Tooltip at an object



Tooltip at an intersection

8.5 Linking Overview

The linking overview shows all connections in a table. The names of the objects and the priorities of the connections can be configured here as well.



Node	Index	Sub	Name	Node	Index	Sub	Name	Priority	Datatype
32	0xa0c0	1	N64_AO0	64	0x6500	1	AO0	0	INTEGER16
32	0xa0c0	2	PLC_OBJ_I16_INPUT_2	64	0x6500	2	AO1	0	INTEGER16
32	0xa040	5	N64_DO1	64	0x6200	1	DO0_DO7	2	UNSIGNED8
64	0x6401	1	AI0	32	0xa540	6	N64_AI1	0	INTEGER16
64	0x6401	2	AI1	32	0xa540	7	N64_AI2	0	INTEGER16
64	0x6000	1	DI0_DI7	32	0xa4c0	15	N64_DI1	2	UNSIGNED8

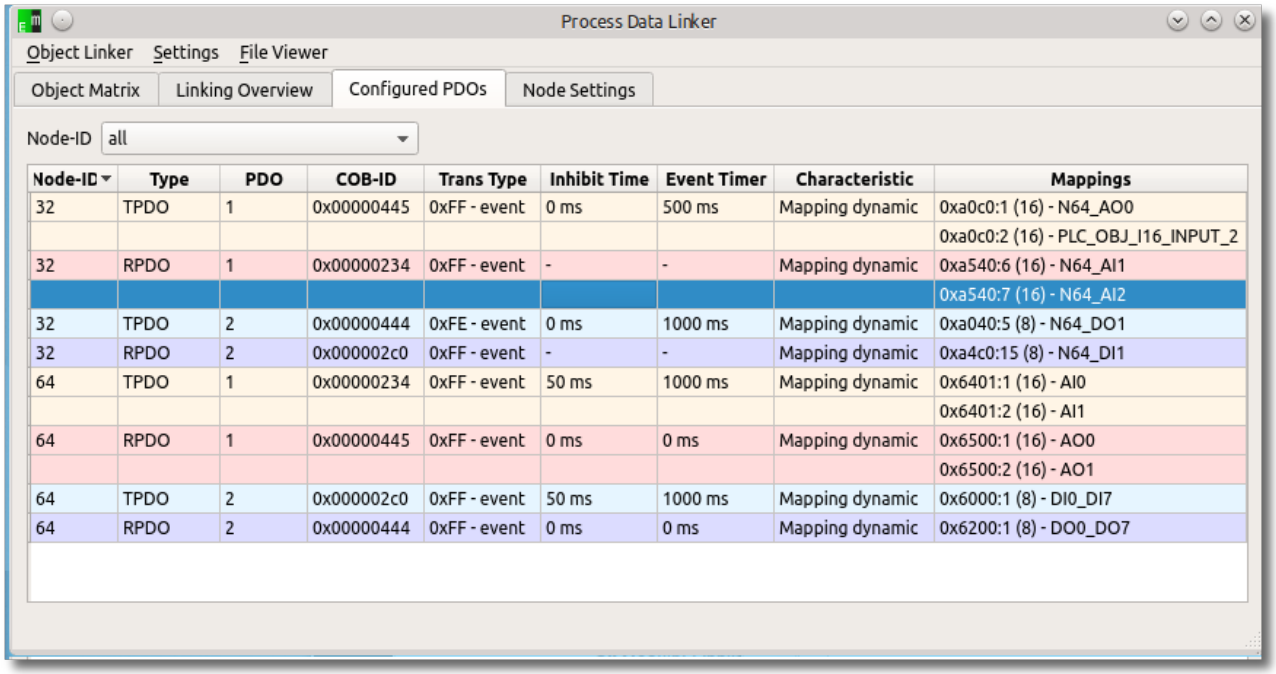
Table view for all connected objects

The priorities are only relevant for connections if both nodes support dynamic mapping. 0 is the highest priority and 255 the lowest. The objects are sorted into the Transmit-PDOs of the producer according to the priority. Only objects with the same priority are put into the same PDO. This mechanism can also be used to separate digital and analog data.

8.6 Configured PDOs

This tab show the configured PDOs as calculated by the Process Data Linker.

Specific properties of the PDOs can be configured additionally. These are the COB-ID, the transmission type, the inhibit time and the event timer.



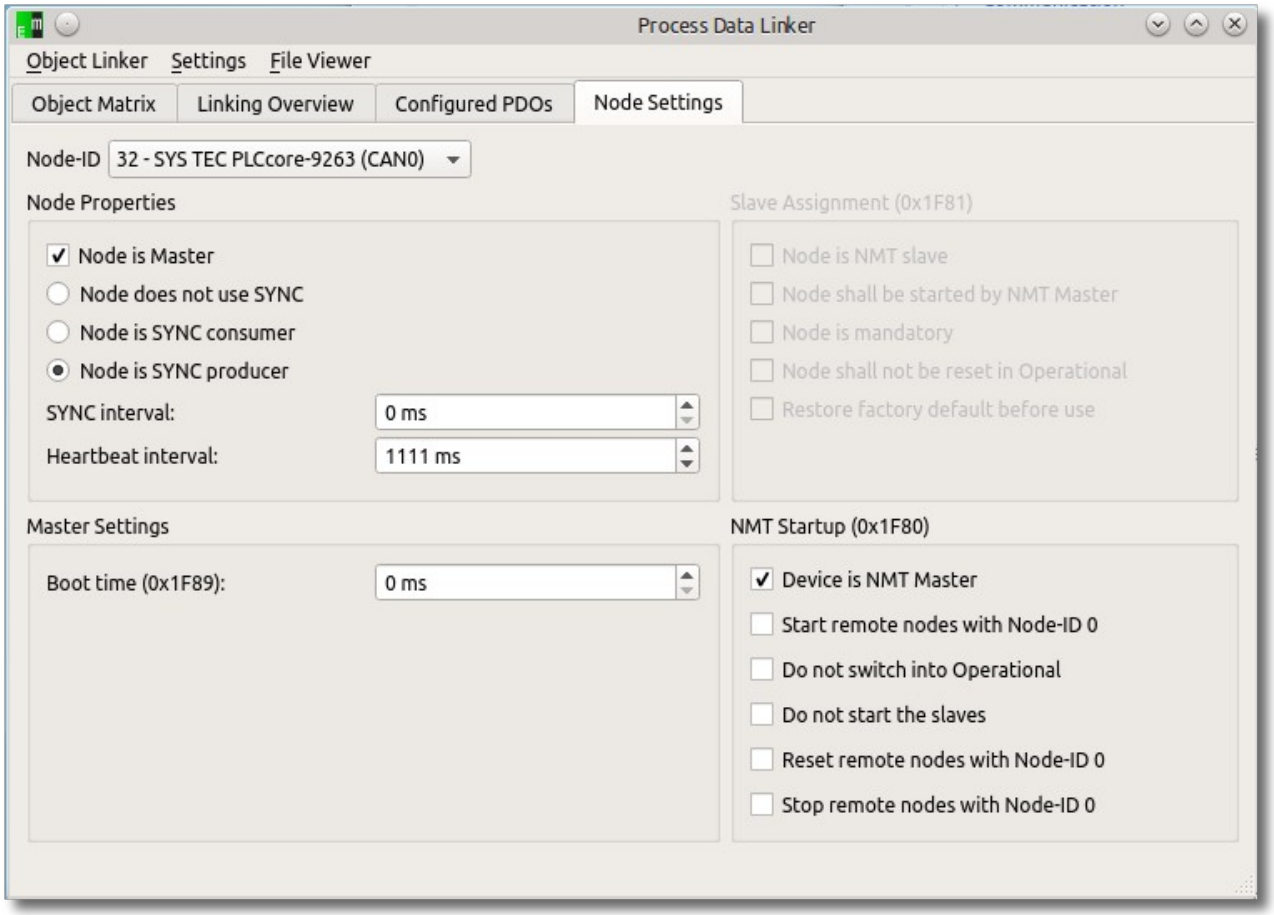
Node-ID	Type	PDO	COB-ID	Trans Type	Inhibit Time	Event Timer	Characteristic	Mappings
32	TPDO	1	0x00000445	0xFF - event	0 ms	500 ms	Mapping dynamic	0xa0c0:1 (16) - N64_AO0
								0xa0c0:2 (16) - PLC_OBJ_I16_INPUT_2
32	RPDO	1	0x00000234	0xFF - event	-	-	Mapping dynamic	0xa540:6 (16) - N64_AI1
								0xa540:7 (16) - N64_AI2
32	TPDO	2	0x00000444	0xFE - event	0 ms	1000 ms	Mapping dynamic	0xa040:5 (8) - N64_DO1
32	RPDO	2	0x000002c0	0xFF - event	-	-	Mapping dynamic	0xa4c0:15 (8) - N64_DI1
64	TPDO	1	0x00000234	0xFF - event	50 ms	1000 ms	Mapping dynamic	0x6401:1 (16) - AI0
								0x6401:2 (16) - AI1
64	RPDO	1	0x00000445	0xFF - event	0 ms	0 ms	Mapping dynamic	0x6500:1 (16) - AO0
								0x6500:2 (16) - AO1
64	TPDO	2	0x000002c0	0xFF - event	50 ms	1000 ms	Mapping dynamic	0x6000:1 (8) - DI0_DI7
64	RPDO	2	0x00000444	0xFF - event	0 ms	0 ms	Mapping dynamic	0x6200:1 (8) - DO0_DO7

PDO overview

8.7 Node Settings

The node properties box allows the configuration of SYNC and heartbeat settings of a device.

All additional nodes settings are only relevant, if a CANopen Manager is present in the network and if its network configuration objects (0x1F80, 0x1F81, 0x1F89) shall be configured. In this case a node has to be assigned ad master. In the fields “Slave Assignment” and “NMT Startup” the objects of the CANopen manager can be configured according to CiA 302.



Knoten- und Mastereinstellungen

8.8 Step-by-Step Guide

1) At first all EDS files for all nodes have to be loaded in the object browser. In this mask also additional objects like polarity can be configured manually. The values which are written and read there are stored in the DCF file as well.

2) The Process Data Linker PlugIn can be activated via PlugIns → Process Data Linker. It consists of 4 tabs. The 1st tab is the object matrix which is used to configure connections between producers and consumer.

3)The Linking Overview Tab shows a connections in a table. Names and priorities of the connections can be changed there as well. Connections with the same priority are put into the same PDO if possible (dynamic mapping)

If all connections are defined, the PDO configuration can be generated by “Generate Linking”. The result is shown in the “Configured PDO”-Tab.

4) DCF files can be generated via “Object Linker” → “Generate Linking & DCF Files.” The tool asks for a path for the project file and stores the generated files there as well. If a CANopen-PLC is used, the DCF files can be imported later on into the programming system of the PLC.

9 Support & Contact

Our support team supports you at any CANopen DeviceExplorer related question or problem. Please contact us by email (service@emtas.de) or by phone +49 3461/794160. If a CANopen device does not react as expected, a logging of the CAN communication is useful for the analysis. Please send us your current CAN logging by email, ideally also before you contact us by phone.