

openCONFIGURATOR

User Manual

1.4.1

Prepared by
Kalycito Infotech Pvt. Ltd.,
India

License

The application and the source code for openCONFIGURATOR are released under BSD license by Kalycito Infotech Private Limited. Please refer to the header section of the source files for the applicable license and the corresponding terms and conditions.

In this manual are descriptions for copyrighted products that are not explicitly indicated as such. The absence of the Trademark (TM) and copyright (©) symbols does not imply that a product is not protected. Additionally, registered patents and trademarks are similarly not expressly indicated in this manual.

The information in this document has been carefully checked and is believed to be entirely reliable. However, Kalycito Infotech Private Limited assumes no responsibility for any inaccuracies. Kalycito Infotech Private Limited neither gives any guarantee nor accepts any liability whatsoever for consequential damages resulting from the use of this manual or its associated product. Kalycito Infotech Private Limited reserves the right to alter the information contained herein without prior notification and accepts no responsibility for any damages which might result.

Additionally, Kalycito Infotech Private Limited offers no guarantee nor accepts any liability for damages arising from the improper usage or improper installation of the hardware or software. Kalycito Infotech Private Limited further reserves the right to alter the layout and/or design of the hardware or software without prior notification and accepts no liability for doing so.

Copyright © 2015 Kalycito Infotech Private Limited. Rights - including those of translation, reprint, broadcast, photo-mechanical or similar reproduction and storage or processing in computer systems, in whole or in part - are reserved. No reproduction may occur without the express written consent from Kalycito Infotech Private Limited.

Registered Office

Kalycito Infotech Private Limited,
E-LAB, Science and Technology Entrepreneurial Park I,
PSG College of Technology, Avinashi Road,
Coimbatore – 641004, Tamil Nadu, INDIA
Phone: 00 91 422 4518454
Email: info@kalycito.com

Abbreviations

API	Application Process Interface
CAN	Controller Area Network
CDC	Concise Device Configuration
CiA	CAN in Automation
CN	POWERLINK Controlled Node (slave)
DLL	Dynamic Link Library
EPL	Ethernet POWERLINK
EPSCG	Ethernet POWERLINK Standardization Group
GUI	Graphical User Interface
ID	Identifier
IEC	International Electro-technical Commission
MN	POWERLINK Managing node
MNOBD	Object Dictionary of the Managing Node
NMT	Network Management
PDO	Process Data Objects
PReq	Poll Request (POWERLINK frame type)
Pres	Poll Response (POWERLINK frame type)
RPDO	Receive Process Data Object
SWIG	Simplified Wrapper and Interface Generator
TCL	Tool Command Language
TPDO	Transmit Process Data Object
XAP	Extend Application Process variables
XDC	XML Device Configuration file
XDD	XML Device Description file
XML	Extensible Markup Language

Table of Contents

Table of Figures	7
1. Preface	8
1.1. Purpose	8
1.2. Intended audience and reading suggestions	8
1.3. Terminology used in this document	8
2. Introduction	9
2.1. What is openCONFIGURATOR?	9
2.2. openCONFIGURATOR features.....	9
3. Install openCONFIGURATOR.....	10
3.1. Operating environment.....	10
3.2. Default installation path	10
3.3. Linux	11
3.3.1. Package name	11
3.3.2. Pre-requisites	11
3.3.3. Installation types	12
3.4. Windows.....	12
4. Using openCONFIGURATOR.....	13
4.1. File menu.....	13
4.1.1. Open project	13
4.1.2. Upgrade pre v1.4.0 openCONFIGURATOR projects	14
4.1.2.1. Upgrade benefits	14
4.1.2.2. Old project backup	15
4.1.2.3. Upgrade process	16
4.1.3. Save project	18
4.1.4. Save project as.....	18
4.1.5. Close project.....	18
4.2. Project menu.....	19

4.2.1. Build project	19
4.2.2. Clean project	19
4.2.3. Project settings	20
4.2.3.1. Save options	20
4.2.3.2. Auto generate options.....	21
4.3. View menu.....	22
4.4. Console window	23
4.5. Project wizard.....	24
4.5.1. Project wizard - settings	25
4.5.2. Project wizard – MN XDD.....	26
4.6. Add a CN.....	27
4.7. Object/sub-object properties	28
4.8. Process data objects.....	34
4.8.1. What is a channel?.....	34
4.8.2. Edit PDO properties	34
4.8.3. Mapping entry	37
4.8.4. PDO mapping vs access type	38
4.9. Delete a CN	39
5. Output files.....	40
6. Uninstall	41
6.1. Linux	41
6.2. Windows.....	41
7. Compile openCONFIGURATOR library from source	42
7.1. Pre-requisites	42
7.1.1. Windows.....	42
7.1.2. Linux	43
7.2. Shared library - Compilation.....	44
8. Txt2Cdc.....	45

9. Project upgrade – Console application.....	46
10. References	47
11. Support	48
11.1. Sourceforge forum.....	48
11.2. Readme	48

Table of Figures

Figure 1: File Menu	13
Figure 2: openCONFIGURATOR project – before upgrade.....	15
Figure 3: openCONFIGURATOR project - after upgrade	15
Figure 4: Project converter window	16
Figure 5: Project folder	18
Figure 6: Project menu	19
Figure 7: Project settings	20
Figure 8: View Menu.....	22
Figure 9: Console Window.....	23
Figure 10: Project wizard	24
Figure 11: Project wizard - settings.....	25
Figure 12: Project wizard - MN XDD.....	26
Figure 13: Add CN	27
Figure 14: Object properties.....	28
Figure 15: PDO properties	34
Figure 16: View of PDO mapping data.....	37
Figure 17: Delete a CN.....	39
Figure 18: Output files	40
Figure 19: Project upgrade app - sample project.....	46

1. Preface

1.1. Purpose

This document is intended for the users of openCONFIGURATOR version - 1.4.1.

1.2. Intended audience and reading suggestions

A common knowledge of POWERLINK and/or CANopen technology is assumed throughout this document.

1.3. Terminology used in this document

To make reading the docs easier, the names of all the screens and Menus from openCONFIGURATOR are marked up in a different font. The **Build Project** for instance.

A menu choice is indicated with an arrow. **View → Advanced view** means: select **Advanced view** from the openCONFIGURATOR **View** menu.

User Interface Buttons are indicated like this: Press **Ok** to continue.



Important:

Important notes are indicated with this icon.



Warning:

Very important warnings are indicated with this icon. If such warnings are ignored, it might lead to data corruption or unpredictable behavior in the application.

2. Introduction

2.1. What is openCONFIGURATOR?

openCONFIGURATOR is an open-source configuration tool for easy setup, configuration and maintenance of any POWERLINK network. It ideally complements openPOWERLINK, the open source POWERLINK protocol stack for master and slave.

The application and the source code for the openCONFIGURATOR are licensed under BSD license. Please refer to the License section for more information.

2.2. openCONFIGURATOR features

Here is a short list of features:

- Generates a generic POWERLINK stack configuration (CDC format).
- Generates the network variables in multiple formats.




ProcessImage variables		Description
	XAP.h	ANSI C header file that describes the process image for C projects
	ProcessImage.cs	C# class that describes the process image for .NET projects
	XAP.xml	XML file that describes the process image generally

Table 1: List of ProcessImage variable formats

- Computes the PDO mapping of the MN automatically based on the mapping of the CNs.
- The Process Image variables of the MN are as per the “CiA 302-4 CANopen additional application layer functions – Part4: Network variables and process image” specification.
- Support for Multiplexing.
- Support for PRes chaining.
- Support Cross Traffic communication.
- Dynamic PDO mapping.
- Implements Communication profile EPSG 301 1.2.0.

3. Install openCONFIGURATOR

3.1. Operating environment

openCONFIGURATOR is designed to work on both 32 bit and 64 bit versions of the following platforms:

Operating system	Version
Ubuntu	14.04 LTS
	12.04 LTS
Windows	Xp - Sp3
	Vista - Sp2
	7 - Sp1
	8 and 8.1

Table 2: List of supported operating systems

3.2. Default installation path

Windows:

<WindowsDrive>/Program Files(x86)/openCONFIGURATOR-<version>

Linux:

/opt/openCONFIGURATOR/

3.3. Linux

All supported versions of Ubuntu requires Debian package to run openCONFIGURATOR. Download the latest Debian package (32 or 64 bit version - depending on the Linux version) of openCONFIGURATOR from <http://sourceforge.net/projects/openconf/>

3.3.1. Package name

All openCONFIGURATOR packages follow the nomenclature of `openconfigurator_<openconf_version-build>ubuntu<ubuntu_version>_<arch>.deb`

Ubuntu version	Package name
12.04 - x86	openconfigurator_1.4.1_ubuntu12.04_i686.deb
12.04 - x64	openconfigurator_1.4.1_ubuntu12.04_amd64.deb
14.04 - x86	openconfigurator_1.4.1_ubuntu14.04_i686.deb
14.04 - x64	openconfigurator_1.4.1_ubuntu14.04_amd64.deb

Table 3: openCONFIGURATOR package names

3.3.2. Pre-requisites

Ubuntu version	Dependencies
Common to all Ubuntu versions	libxml2, libxml2-utils, tcllib, tklib, tcl-thread, libboost-filesystem1.54.0, libboost-system1.54.0, libboost-date-time1.54.0, libboost-log1.54.0, libboost-thread1.54.0
12.04 - x86 and x64	tcl8.5, tk8.5
14.04 - x86 and x64	tcl8.6, tk8.6

Table 4: Linux - pre-requisites

3.3.3. Installation types

Graphical install:

- Double click on the downloaded .deb package to install using the graphical package manager.

Terminal install:

- `sudo apt-get update`
- `sudo dpkg -i openconfigurator-1.4.1_ubuntu14.04_amd64.deb`

Offline install:

- To install offline, download and install all the listed dependency packages for the respective platform and proceed to openCONFIGURATOR installation.

3.4. Windows

Download the latest 32 bit version of openCONFIGURATOR from

<http://sourceforge.net/projects/openconf/>

openCONFIGURATOR is available with an easy to use installer. Double click on the installer file and follow the instructions. The installer will take care of the rest.



Important:

Administrator privileges are required to install openCONFIGURATOR.

In case of any problems encountered during or after installation of openCONFIGURATOR, please post in the project support forum at <http://sourceforge.net/p/openconf/discussion/help/>.

4. Using openCONFIGURATOR

4.1. File menu

The user can create a new project by selecting **File → New Project** or by using the keyboard shortcut **CTRL + N**.

4.1.1. Open project

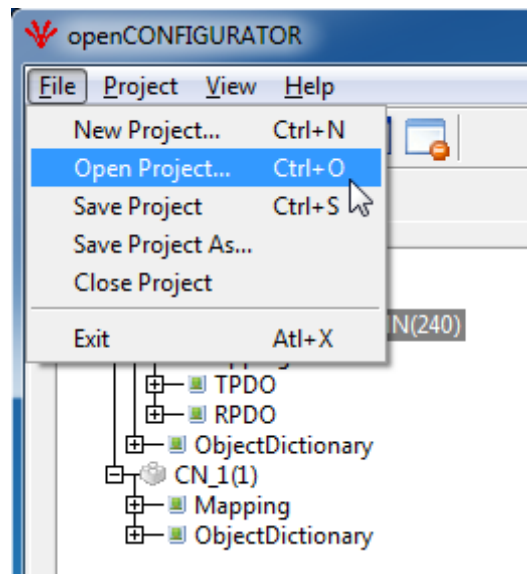


Figure 1: File Menu

The user can open the already created projects by selecting **File → Open Project** or by using the keyboard shortcut **CTRL + O**.

Important:



It is highly recommended to upgrade projects created with openCONFIGURATOR versions pre 1.4.x. Please refer to section 4.1.2 for greater details in the upgrade process.

4.1.2. Upgrade pre v1.4.0 openCONFIGURATOR projects

As openCONFIGURATOR version 1.4.0 and higher cannot deal with old projects created prior to this version an upgrade is highly recommended. The project files no longer supported have an '.oct' file extension.

4.1.2.1. Upgrade benefits

- In pre 1.4.x projects the node configuration files are only available in the '.octx' format which is not compliant to the XDD specification. The upgrade process will convert these files to compliant configuration files if the user provides the base XDD files.
- The upgrade process will also convert the project file to the new schema validated format which is easier to use and maintain.
- The upgrade will preserve the project node configuration as long as the user specifies the original XDDs used to create the nodes.

4.1.2.2. Old project backup

The old projects will be moved to <ProjectName>_<old_version> in the same directory without any modification.

For eg. Referring to the Figure 2: openCONFIGURATOR project – before upgrade and Figure 3: openCONFIGURATOR project - after upgrade, if user upgrades the project named “BR_X20_Evalpacks” which was originally created with version 1.2.0, then after conversion the project will be moved to a new folder named “BR_X20_Evalpacks_1.2.0”.

```

C: .
├── BR_X20_Evalpacks
│   ├── BR_X20_Evalpacks.oct
│   ├── octx
│   │   ├── 1.octx
│   │   ├── 2.octx
│   │   ├── 240.octx
│   │   └── 3.octx
│   └── scripts
│       └── Transfer.bat

```

Figure 2: openCONFIGURATOR project – before upgrade

```

C: .
├── BR_X20_Evalpacks
│   ├── BR_X20_Evalpacks.xml
│   ├── deviceConfiguration
│   │   ├── EvalPackA_1.xdc
│   │   ├── EvalPackB_2.xdc
│   │   ├── EvalPackC_3.xdc
│   │   └── openPOWERLINK_MN.xdc
│   ├── deviceImport
│   │   ├── EvalPackA_1.xdc
│   │   ├── EvalPackB_2.xdc
│   │   ├── EvalPackC_3.xdc
│   │   └── openPOWERLINK_MN.xdd
│   └── output
├── BR_X20_Evalpacks_1.2.0
│   ├── BR_X20_Evalpacks.oct
│   ├── octx
│   │   ├── 1.octx
│   │   ├── 2.octx
│   │   ├── 240.octx
│   │   └── 3.octx
│   └── scripts
│       └── Transfer.bat

```

Figure 3: openCONFIGURATOR project - after upgrade

4.1.2.3. Upgrade process

Please follow the steps below to upgrade pre 1.4.x openCONFIGURATOR projects to the new format.

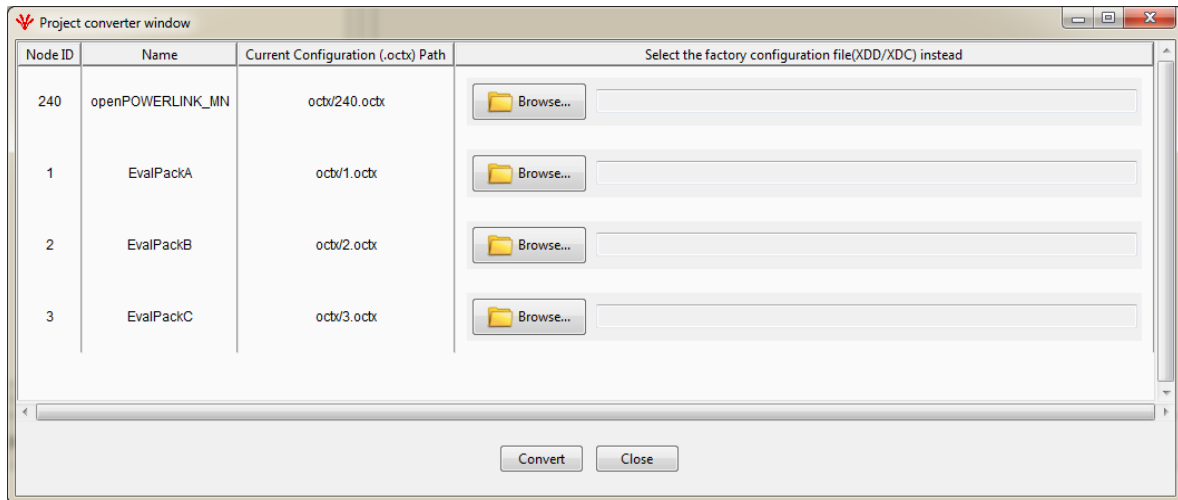


Figure 4: Project converter window

- 1) Once the old project is opened from openCONFIGURATOR 1.4.0 or later, a ‘Project converter window’ will open as shown in Figure 4: Project converter window.
- 2) It is highly recommended to provide the default device configuration file (XDD/XDC) that was used for each node in the older project. To do so, click ‘Browse’ and set the path of the XDD/XDC file for each node.
- 3) In case the default configuration file is not provided, openCONFIGURATOR 1.4.0 or later, shall convert the “.octx” files available in the older project to “.xdc” format with the following changes in order to comply with the standard XDC format:
 - a. The values in the “actualValue” field for each Object and Sub-Object will be copied from the “.octx” to the XDC file.
 - b. Missing elements and fields in the “.octx” are included but the values are assumed as default.

Important:



XDD/XDC names for each node must be unique or they will be overwritten in the deviceImport folder.

- 4) Once the device configuration file have been chosen, click **‘Convert’** button to upgrade the old projects or click **‘Close’** to cancel the upgrade process.
- 5) On successful completion, the upgraded project will be opened automatically with openCONFIGURATOR. The older project will be moved to `<ProjectName>_<old_version>` in the same directory without any modification.
- 6) A debug log for the upgrade process will be created in the location `<project directory>`.

Important:



The log file will be created regardless of what is defined in the `boost_log_settings.ini` present in the openCONFIGURATOR installation directory.

If you face any errors and need support, please post on the help forum at <http://sourceforge.net/p/openconf/discussion/help/>


4.1.3. Save project

The project can be saved by selecting **File → Save Project** or by using the keyboard shortcut **CTRL+ S** or by clicking on **Save** icon.

Save project will save the following files in the <project directory>/deviceConfiguration folder.

- One XDC file for each one of the nodes present in the project.
 - The name of the file will be same as <ImportedXDDName_NodeID> of the node.
 - ♦ Where <ImportedXDDName> is the name of the XDD imported.
 - ♦ NodeID is the ID of the node.
- A copy of XDD/XDCs used in the project are stored in <project directory>/deviceImport folder.

Important:



XDD/XDC names must be unique or they will be overwritten in the deviceImport folder.

- One XML file that contains the overall project settings and is found in the <project directory> folder.

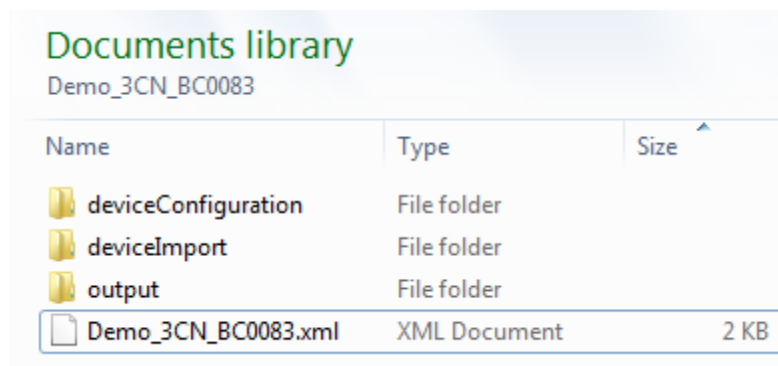


Figure 5: Project folder

4.1.4. Save project as

The user can also save a copy of the project by selecting **File → Save Project As**. The application will switch to the newly created copy.

4.1.5. Close project

Close the project by selecting **File → Close Project**.

4.2. Project menu

4.2.1. Build project

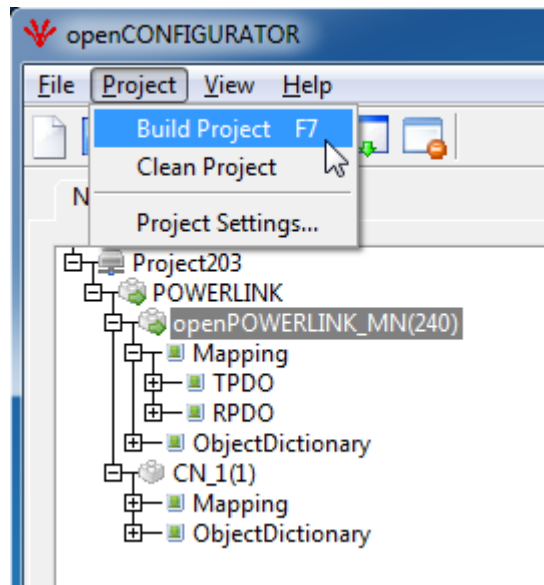


Figure 6: Project menu

The user can build the project by selecting **Project** → **Build Project** or by using the function key **F7** or by clicking on **Build Project** icon as shown in Figure 6: Project menu.

4.2.2. Clean project

The user can remove the output files (Eg: mnobd.cdc, xap.h) from the project by selecting **Project** → **Clean Project**.

4.2.3. Project settings

The user at any time can change the settings of the project by selecting **Project** → **Project Settings** option as shown in Figure 7: Project settings.

4.2.3.1. Save options

Options	Description
Auto Save	Saves the configuration automatically without prompting the user
Prompt (Default)	Prompt the user with the option to save before exiting from the project
Discard	Requires the user to manually save the configuration by clicking save button

Table 5: Save options

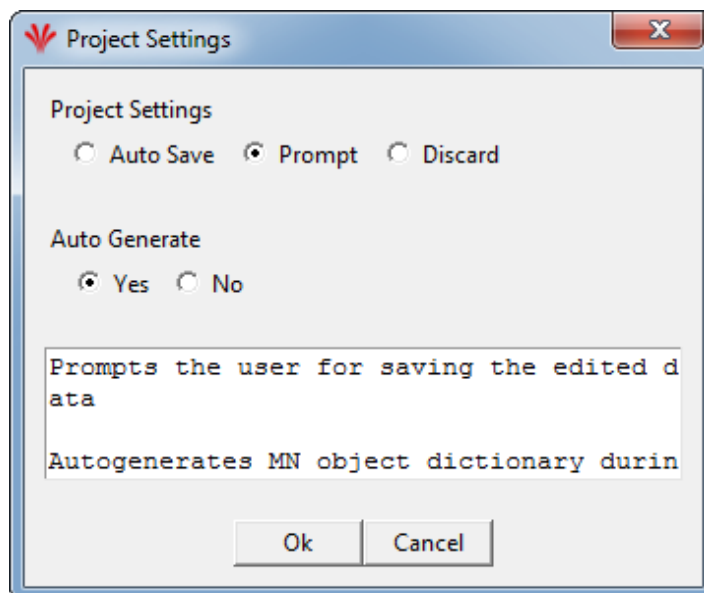


Figure 7: Project settings

4.2.3.2. Auto generate options

Options	Description
Yes (Default)	Available CN configurations will be used to auto generate MN configuration
No	The user will have to manually generate/update the MN configuration

Table 6: Auto generate MN OBD options

If 'Auto Generate' is set to 'Yes', the following items are generated:

- PDO mapping for the MN
- PRes and PReq payload length values
- Offset for the CN if the type of station is 'Chained station'
- Process Image Variables
- Node Assignment configuration of the MN.

If 'Auto Generate' is set to 'No':

- The user has to manually create/update the mapping and relevant changes in the object dictionary of the MN.
- The changes made to the project are reflected in the CDC regardless of correctness of the configuration data.

4.3. View menu

The user can switch between the ‘Simple View’ and ‘Advanced View’ options of the tree browser as shown in Figure 8: View Menu.

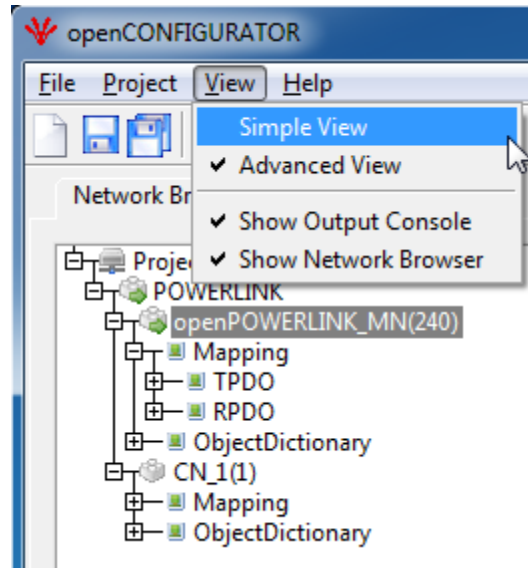


Figure 8: View Menu

In ‘Simple View’, the attributes listed below are visible in the tree browser:

- All node configurations added to the POWERLINK Network.
- A Mapping tree node for each node which is split into TPDO and RPDO that simplifies the mapping activity.

In ‘Advanced View’, the attributes listed below can be viewed by expanding the corresponding entries in the tree browser:

- All node configurations added to the POWERLINK Network.
- A Mapping tree node for each node which is split into TPDO and RPDO that simplifies the mapping activity.
- An **ObjectDictionary** tree node which lists all the objects present in the node configuration (XDD or XDC).

4.4. Console window

The user can view the status messages, warnings and error messages in the console window as shown in Figure 9: Console Window.

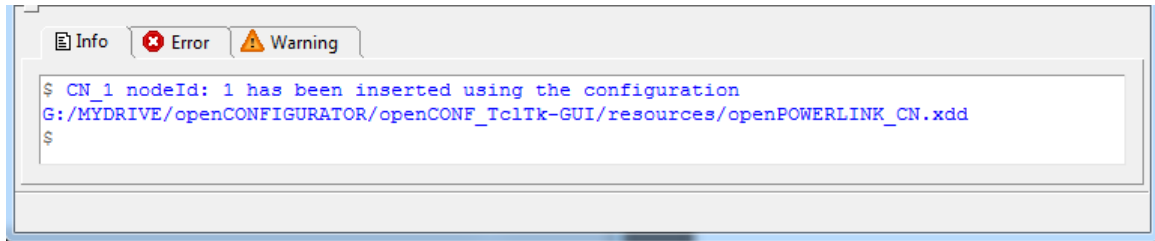


Figure 9: Console Window

4.5. Project wizard

The project wizard can be used to create a new project or open an existing one:

- When the user launches openCONFIGURATOR, it will ask either to **Create New Project** or **Open Existing Project** as shown in Figure 10: Project wizard.
- Alternatively, the user can also create a new project or open an existing one through the menu bar by using the **File → New Project** or **File → Open Project** options.

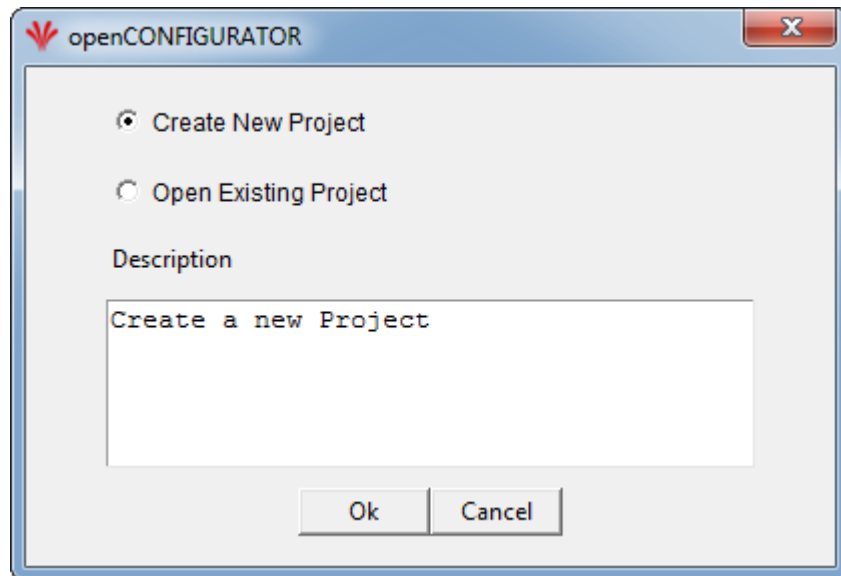


Figure 10: Project wizard

4.5.1. Project wizard - settings

The options given below are to be entered/selected as applicable in the 'Project Wizard' dialog box before clicking *Next* button as shown in Figure 11: Project wizard - settings.

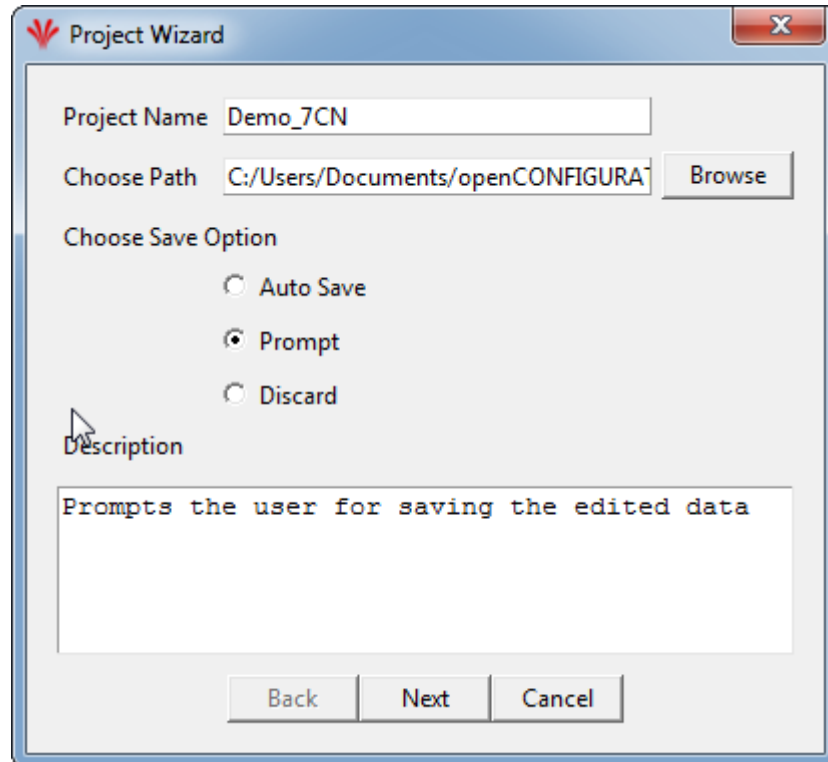


Figure 11: Project wizard - settings

- Project Name
 - The maximum limit is 32 characters.
 - Special characters and spaces are not allowed.
- Choose Path
 - The user can select the location for placing the projects by clicking the *Browse* button next to the 'Choose Path' field.
 - The default path is the <Documents>/openCONFIGURATOR_Projects.
- Choose Save option
 - Refer to the section Table 5: Save options

4.5.2. Project wizard – MN XDD

The options given below are to be entered/selected as applicable in the 'Project Wizard' dialog box before clicking *Ok* button as shown in Figure 12: Project wizard - MN XDD.

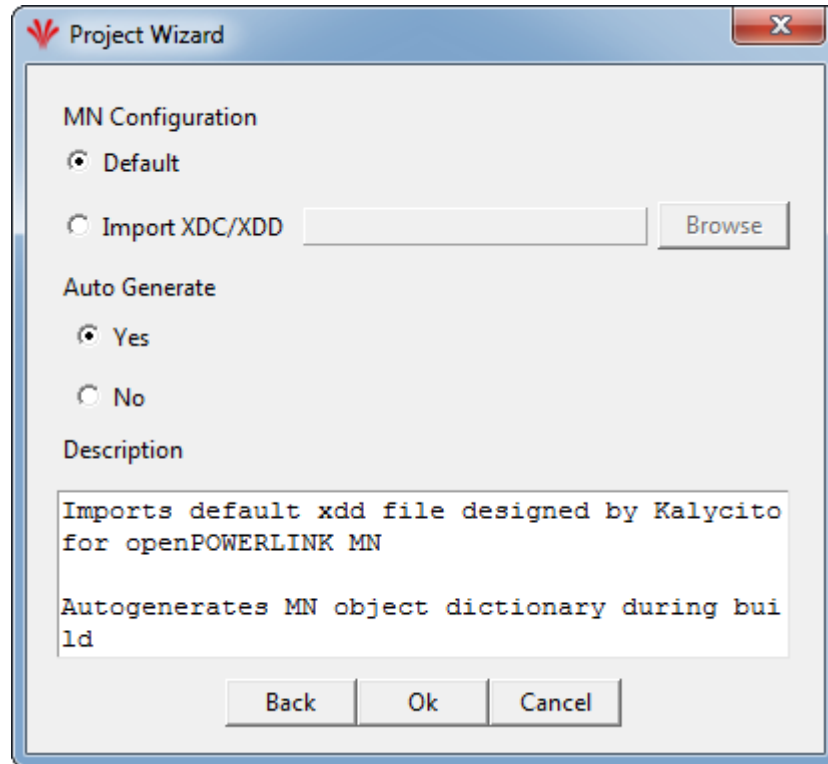


Figure 12: Project wizard - MN XDD

- MN Configuration

Options	Description
Default	Default MN XDD will be available with the installation package.
Import XDD/XDC	User defined MN configuration

Table 7: MN configuration

- Auto Generate
 - Please refer to the section Table 6: Auto generate MN OBD options.

4.6. Add a CN

A CN can be added by right clicking on the POWERLINK and selecting Add CN option from the sub menu as shown in Figure 13: Add CN.

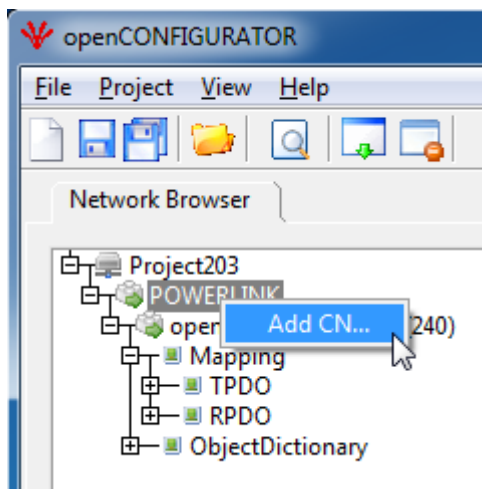


Figure 13: Add CN

After clicking on ‘Add CN’ a pop-up will appear where the user can enter a CN configuration by referring to the Table 8: CN configuration options and can select the XDD/XDC files for that CN.

New CN configuration		Description	Range
Name		Name of the Node	1-32 chars
Node ID (decimal value)		Node Id of the Node	1-239
CN configuration	Default	Default CN XDD which will be available with the installation package.	-
	Import XDD/XDC	User defined configuration for the CN	-

Table 8: CN configuration options

Important:



XDD/XDC names for each node must be unique or they will be overwritten in the deviceImport folder.

Important:



Please validate XDDs with the XDD-Check tool. It is a free utility available at the EPSG website: <http://ethernet-powerlink.org>

4.7. Object/sub-object properties

The properties that are available for any object/sub-object are shown in Figure 14: Object properties.

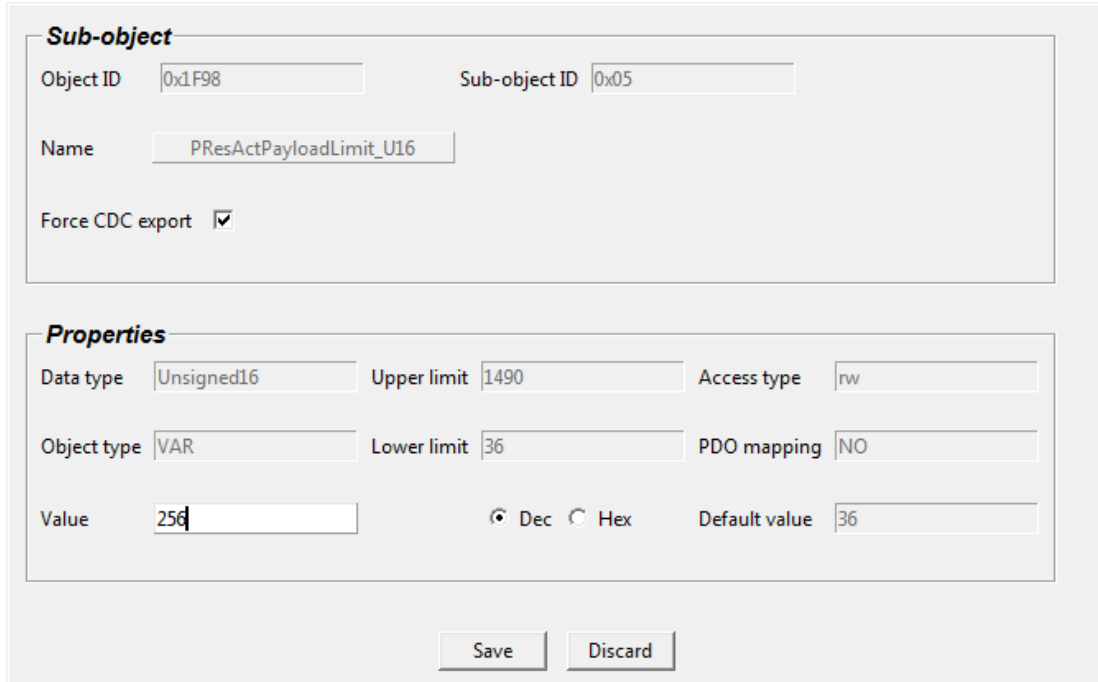


Figure 14: Object properties

- **Object ID**
 - The **Object ID** identifies the object that is currently active.
 - A 16 bit identifier of the object (given in hex).
 - Eg: 0x1F81, 0x25F4, 0x6201, 0xA480.
- **Sub-object ID**
 - This field will only be visible if the currently active node is a sub-object.
 - The **Sub-object ID** identifies the sub-object that is currently active.
 - An 8 bit identifier of the sub-object (given in hex). Eg: 0x00, 0xFE.
- **Name**
 - Name provides a textual description of the function of that particular object/sub-object.
 - Name shall be in accordance to IEC 61121-3 standards.
 - Name shall have a maximum limit of 32 characters.
 - Eg: NMT_FeatureFlags_U32, DigitalInput_U8, etc.
- **Force CDC export**
 - Forces the presence of the actual value in the CDC regardless of the default value.

- **Object type**

- This attribute determines the type of object.
- It is given as a decimal value with the following meaning:

Object type	Code	Description
VAR	7	An object with a single value.
ARRAY	8	A multiple data field object where each field is a simple variable of the same data type. Eg: array of unsigned64 etc. Note: Sub-object 0x00 is always of Unsigned8 and therefore not part of ARRAY data.
RECORD	9	A multiple data field object where the data fields may be any combinations of simple variables. Note: Sub-object 0x00 is always of Unsigned8 and therefore not part of RECORD data.
DEFSTRUCT	6	Defines a record type.

Table 9: List of Object types

- **Data type**

- This attribute determines the data type of the object.
- It is usually given as a reference to the data type definition area of the [Object Dictionary] and therefore a 16 bit unsigned integer.
- The following table provides the list of supported data types, its data size and whether it is allowed to be mapped to a PDO object.

Data type	ID (hex)	Data size (bits)	Allowed for PDO mapping
BIT, Boolean	0001	1 ¹ padded to 8	Only supported via DOMAIN objects that uses parameter and varDeclaration. ²
INTEGER8	0002	8	Yes
INTEGER16	0003	16	Yes
INTEGER32	0004	32	Yes
INTEGER64	0015	64	Yes
UNSIGNED8	0005	8	Yes
UNSIGNED16	0006	16	Yes
UNSIGNED32	0007	32	Yes
UNSIGNED64	001B	64	Yes
OCTET_STRING	000A	variable ³	Not allowed
VISIBLE_STRING	0009	variable ⁴	Not allowed
Domain	000F	variable	Yes (via parameter list)
INTEGER24	0010	24	Not allowed
INTEGER40	0012	40	Not allowed
INTEGER48	0013	48	Not allowed
INTEGER56	0014	56	Not allowed

¹ Refer to the section 6.1.4.2 Boolean of EPSG Communication Profile Specification DS-301 v1.2.0.

² Refer to the section 6.3 dataTypeList of EPSG XML Device Description – Implementation Guideline v1.0.0.

³ Refer to the section 6.1.6.1 Octet String of EPSG Communication Profile Specification DS-301 v1.2.0.

⁴ Refer to the section 6.1.6.2 Visible String of EPSG Communication Profile Specification DS-301 v1.2.0.

UNSIGNED24	0016	24	Not allowed
UNSIGNED40	0018	40	Not allowed
UNSIGNED48	0019	48	Not allowed
UNSIGNED56	001A	56	Not allowed
REAL32	0008	32	Not allowed
REAL64	0011	64	Not allowed
MAC_ADDRESS	0401	48 ⁵	Not allowed
IP_ADDRESS	0402	32 ⁶	Not allowed
Unicode_String	000B	16	Not allowed
Time_of_Day	000C	48	Not allowed
Time_Diff	000D	48	Not allowed

Table 10: List of data types - Object/sub-object

- **Limit**
 - This property indicates the range (high and low limits) for the value in the respective object.
 - It depends on the data type specified for the object.
- **Access type**
 - This attribute determines the access from the network side to the object.

Access type	Description
Const	Read only access, value is constant.
ro	Read only access.
wo / wos ⁷	Write only access.
rw / rws ⁸	Read and write access.

Table 11: List of access types for an object/sub-object

⁵ Refer to the section 6.1.4.8 MAC Address of EPSG Communication Profile Specification DS-301 v1.2.0.

⁶ Refer to the section 6.1.4.9 IP address of EPSG Communication Profile Specification DS-301 v1.2.0.

⁷ Refer to the section 6.2.1 Access Type of EPSG Communication Profile Specification DS-301 v1.2.0.

⁸ Refer to the section 6.2.1 Access Type of EPSG Communication Profile Specification DS-301 v1.2.0.

- **PDO mapping**

- This attribute defines whether an object can be mapped to a PDO or not.

PDO mapping	Description
NO	Objects cannot be mapped to a PDO.
DEFAULT ⁹	The object is mapped by default to a PDO. (The mapping must be specified in an according mapping parameter!).
OPTIONAL	Objects may be mapped into any PDO. See AccessType.
TPDO	Objects shall be mapped only to a Transmit PDO.
RPDO	Objects shall be mapped only to a Receive PDO.

Table 12: List of PDO mapping types

Important:



The mapping configuration also depends on the access type of the object. For example: an object with access type read-only can only be mapped to a TPDO, even if PDO mapping specifies it as optional.



Important: Any object shall always be accessible via SDO.

- **Value**

- The desired value that overrides the default value.

- **Default value**

- The default value determines the value of the object when it is not configured.

- **Dec - Hex radio buttons**

- The user can toggle between decimal or hexadecimal view of the value by choosing from these radio buttons.

⁹ Refer to the section 6.2.1 PDO mapping of EPSG Communication Profile Specification DS-301 v1.2.0.

The user can edit only the following properties for the object/sub-object:

- Name
- Force CDC export option
- Actual value

Important:



The user cannot add/delete any object/sub-object and edit certain properties of object/sub-object through openCONFIGURATOR. The user is not expected to change the XDD/XDC as it should be provided by the device manufacturer.

4.8. Process data objects

Process data objects are used for isochronous data exchange between POWERLINK nodes. The Objects (0x14xx, 0x16xx and 0x18xx, 0x1Axx) are collectively known as PDO mapping objects. PDO objects will be presented in a tabular structure and the user can select the values listed in the drop-down boxes.

4.8.1. What is a channel?

The PDO_T(R)xCommParam & PDO_T(R)xMappParam pair is collectively known as a TPDO(RPDO) channel respectively.

The user can map data from an Object/sub-object to a PDO object and is known as PDO mapping. The PDO Mapping determines the payload of a PDO frame in a POWERLINK network.

4.8.2. Edit PDO properties

The following are the properties of the PDO, which is shown in Figure 15: PDO properties

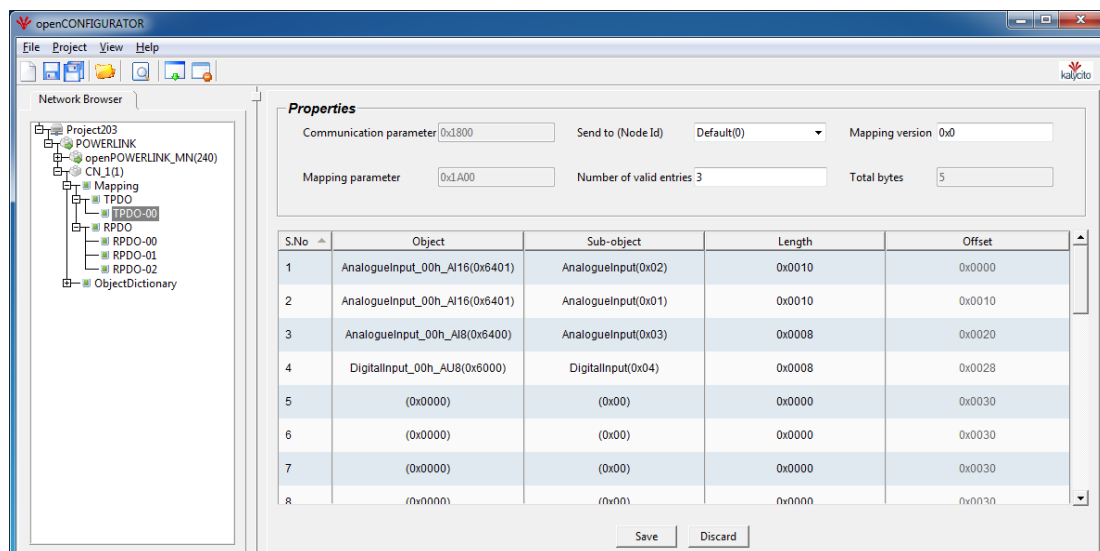


Figure 15: PDO properties

- Communication parameter - (0x14XX for RPDO and 0x18XX for TPDO)
 - An object which describes the communication attributes of the PDO channel.
- Mapping parameter - (0x16XX for RPDO and 0x1AXX for TPDO)
 - An object which describes the mapping of the objects contained in PDO payload to object dictionary entries.
- Receive from/Send to Node ID
 - The Node ID which is stored in sub-object(0x01) of the communication parameter

object.

- It ranges from 00h to F9h.
- On a CN, the node ID is interpreted in the following way:

Node ID	Interpretation
00h	RPDO: Read data from the PReq frame
	TPDO: Send data to the network (PRes frame)
01-F9h	RPDO: Read data from the PRes frame of node with ID ranges from 01-F9h
	TPDO: Not allowed on a CN

Table 13: PDO properties - CN settings

Important:



To read data from a PRes of the MN (e.g. on a CN using PRes Chaining), set the Node ID to F0h (240) instead of 00h.

- On an MN, the node ID is interpreted in the following way:

Node ID	Interpretation
00h	RPDO: PDO is not used
	TPDO: Send data to the network (PRes frame)
01-F9h	RPDO: Read data from the PRes frame of node with ID ranges from 01-F9h
	TPDO: Send data in the PReq frame to node with ID ranges from 01-F9h

Table 14: PDO properties - configurations

Important:



To send data to a CN using PRes Chaining, set the node ID to 00h

- Mapping version
 - Version of the PDO mapping implementation.
- Number of valid entries
 - Number of objects that shall be mapped to the PDO.
 - Ranges from 00h to FEh.
- Total bytes
 - Total size in bytes of the payload data that is configured to communicate in the PDO.
 - Ranges from 36 Bytes up to 1490 bytes¹⁰.

¹⁰ Refer to the section PDO Mapping of <http://sourceforge.net/p/openpowerlink/wiki/PDO%20Mapping/>

4.8.3. Mapping entry

Each row in the PDO table describes the mapping of one object from the object dictionary to the according RPDO or TPDO.

The following table describes about each in more detail:

Column	Description	Allowed range	
Object	Object ID of the object to be mapped	0x1000 - 0x9FFF	Object/sub-object that passes the mapping criteria will be listed.
Sub-object	Sub-object ID of the object to be mapped	0x00 - 0xFE	
Length	Length of the mapped object (Bit count)	Depends on the data type of the object/sub-object	
Offset	Offset related to the start of the PDO payload (Bit count)	Cumulative sum of the payload length	
Reserved	Reserved for future use.	Defaults to 00h. Hidden in the table.	

Table 15: PDO properties configuration

The individual entries of mapping are interpreted according to the following sample illustration (all values in hex):

View from the PDO table	Object	Sub-object	Length(bit)	Offset(bit)	Reserved
	ABCD	EF	GHIJ	KLMN	OP
View from the XDD / CDC / Actual-Value	Length(bit)	Offset(bit)	Reserved	Sub-object	Object
	GHIJ	KLMN	OP	EF	ABCD
View from the Network	Object	Sub-object	Reserved	Offset(bit)	Length(bit)
	CDAB	EF	OP	MNKL	IJGH

Figure 16: View of PDO mapping data

4.8.4. PDO mapping vs access type

For an object to be mapped to a PDO, the following conditions should be met:

PDO mapping of an object	Access type for a CN object							
	If object mapped to a RPDO (0x1600 – 0x16FF)				If object mapped to a TPDO (0x1A00) ¹¹			
	Const	Ro	Wo/ Wos	Rw/ Rws	Const	Ro	Wo/ Wos	Rw/ Rws
No								
Default			✓	✓		✓		✓
Optional			✓	✓		✓		✓
TPDO						✓		✓
RPDO			✓	✓				

Table 16: PDO mapping vs Access type - compatibility table

- **Example:**

- An Object (0x6000) with PDOmapping="TPDO" and AccessType="Ro" can be mapped only to a TPDO but not to an RPDO.
- An Object (0x6200) with PDOmapping="Optional" and AccessType="Rw" can be mapped to both TPDO and RPDO.



Important:

Also refer to the list of data types allowed for PDO mapping.

¹¹ A CN can have only one TPDO.

4.9. Delete a CN

A CN can be deleted by right clicking on the node and select **Delete Node** option from the sub-menu. The sub-menu will appear as shown in Figure 17: Delete a CN.

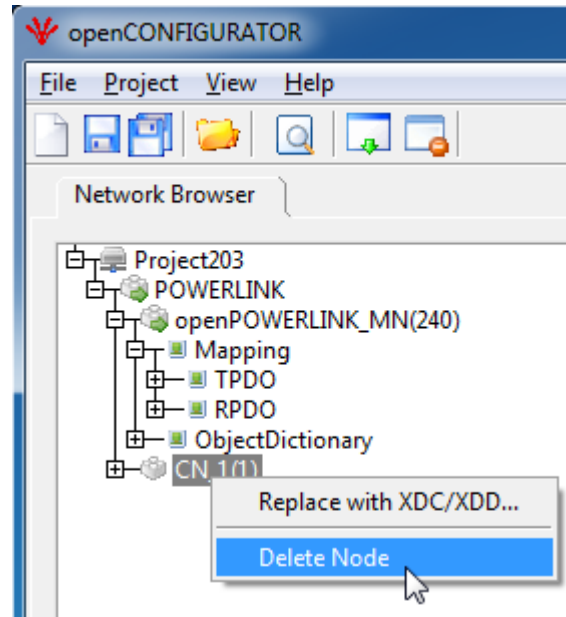


Figure 17: Delete a CN

Warning:



If the user wishes to delete the node, the objects and their sub-objects will also be deleted. The user cannot undo the operation.

Important:



Instead of deleting and again adding a new CN in its place, the user can replace the configuration files of the existing CN by choosing the **Replace with XDC/XDD** option in the sub-menu. This will update the configuration of the existing CN with the new CN configuration.

5. Output files

The following files will be created after a successful build of the project. These files will be present in <Project location >/<Project Name>/output directory.






Name	Type	Size
 mnobd.cdc	CDC File	1 KB
 mnobd.txt	Text Document	1 KB
 ProcessImage.cs	CS File	1 KB
 xap.h	H File	1 KB
 xap.xml	XML Document	1 KB

Figure 18: Output files






Output files		Description
	mnobd.cdc	Concise Device Configuration - binary file used with the openPOWERLINK stack
	mnobd.txt	Text version of the binary CDC file
	XAP.h	ANSI C header file that describes the process image for C projects
	ProcessImage.cs	C# class that describes the process image for .NET projects
	XAP.xml	XML file that describes the process image generally

Table 17: openCONFIGURATOR output files

6. Uninstall

6.1. Linux

Uninstall openCONFIGURATOR using either one of the following methods:

- Remove openconfigurator from Ubuntu software center or using Synaptic package manager.

or

- Run this command
 - *sudo apt-get remove openconfigurator*

6.2. Windows

- Uninstall openCONFIGURATOR by navigating to the **Control Panel → Programs and Features** and choosing the openCONFIGURATOR package.
- Alternatively one can uninstall openCONFIGURATOR from the windows start menu.

7. Compile openCONFIGURATOR library from source

7.1. Pre-requisites

7.1.1. Windows

Install the pre-requisites to compile openCONFIGURATOR in windows systems.

Package	Minimum version	Download URL
openCONFIGURATOR source	1.4.0	http://sourceforge.net/projects/openconf/
Microsoft Visual C++	2010	http://microsoft.com
Boost libraries ¹²	1.54.0 ¹³	http://boost.org/
CMake	2.8.8	http://www.cmake.org/
Active TCL - 32bit	8.6.0	http://www.activestate.com/activetcl/downloads
Swig	2.0.10	http://sourceforge.net/projects/swig/
Doxygen	1.7.6	http://www.doxygen.org/index.html

Table 18: Pre-requisites - Windows

¹² See Readme.txt in <openCONFIGURATOR Library Source>\openCONFIGURATOR to compile and install boost libraries

¹³ The Boost version 1.54.0 is fixed for all supported platforms.

7.1.2. Linux

The user can install all the pre-requisites using the package manager in Ubuntu or through terminal as 'sudo apt-get install PACKAGE*' where PACKAGE refers to any of the packages listed below:

Ubuntu version	Dependencies	
	Package Name	Minimum version
Common	build-essential	-
	libxml2, libxml2-dev, libxml2-utils	2.7.8
	tcllib	1.14
	tklib	0.5
	swig2.0	2.0.11
	cmake-gui	2.8.8
	g++	4.6.3
	libboost1.54-all-dev ¹⁴	1.54.0 ¹⁵
	doxygen (optional)	1.7.6
12.04 - x86 12.04 - x64	tcl8.5, tk8.5, tcl8.5-dev, tk8.5-dev	8.5.0
14.04 - x86 14.04 - x64	tcl8.6, tk8.6, tcl8.6-dev, tk8.6-dev	8.6.0

Table 19: Pre-requisites - Linux

¹⁴ Install boost libraries manually if the package is not available via apt-get. Please refer <http://www.boost.org> for manual installation of boost libraries.

¹⁵ The Boost version 1.54.0 is fixed for all supported platforms.

7.2. Shared library - Compilation

- Unzip the openCONFIGURATOR_Library source package downloaded.
- Move to '<openCONFIGURATOR_Library>'.
 - Run CMake and choose openCONFIGURATOR root directory as source directory and a custom build directory.
 - Click on **Configure** .
 - Select the preferred C/C++ compiler. Tested using
 - Visual Studio 10
 - Visual Studio 11
 - Unix Makefiles.
 - Click **Finish** to complete the configuration process.
 - On successful selection of the compiler, the text 'Configuring done' will be visible in the console window.
 - After the CMAKE configuration additional build options are available:
 - OPEN_CONFIGURATOR_TCL_WRAPPER (adds openCONFIGURATOR Tcl/Tk wrapper target to build)
 - OPEN_CONFIGURATOR_PROJECT_UPGRADE_LIBRARY (adds Project upgrade library to the target)
 - OPEN_CONFIGURATOR_PROJECT_UPGRADE_WRAPPER_LIBRARY (adds Project Upgrade Tcl/Tk wrapper target to build)
 - OPEN_CONFIGURATOR_PROJECT_UPGRADE_CONSOLE_APP (adds a console application to upgrade pre1.4.0 projects target to build)
 - OPEN_CONFIGURATOR_DOCUMENTATION (adds documentation target to build)
 - OPEN_CONFIGURATOR_UNIT_TESTS (adds unit test target to build).
 - Click **Generate** to generate the toolchain files in the chosen custom build directory.
 - Build the library and specified targets with the toolchain in (Debug/Release) mode.
 - The binaries will be built inside the chosen binary path.
 - Copy all the files present in the bin directory into the openCONFIGURATOR installed path. (Refer to section 3.2 Default installation path).

8. Txt2Cdc

The users who want to edit and generate their own CDC can edit the `mnobd.txt` and generate the CDC with a utility (`txt2cdc.exe/txt2cdc`).

Steps to be followed:

- Open terminal/command prompt
- Move to the directory where `mnobd.txt` resides, (a sample is given below)
cd openCONFIGURATOR_Projects/Project1/output
- Convert the edited text file to CDC using the command given below:
<openCONFIGURATOR installed directory>/txt2cdc mnobd.txt mnobd.cdc

9. Project upgrade – Console application

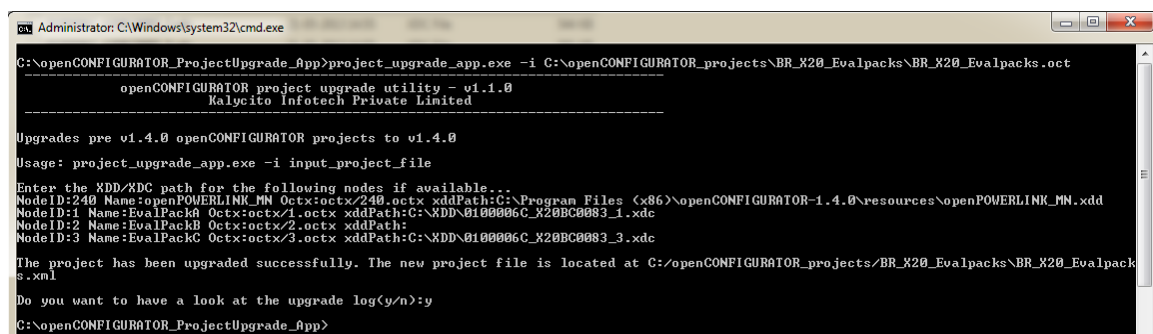
The “project_upgrade_app” is a console application to upgrade pre v1.4.0 openCONFIGURATOR projects to the new version. This application prompts the user to provide the file path of the default device configuration for each node. It is highly recommended to provide the default device configuration file (XDD/XDC) that was used for each node in the older project.

Usage:

```
project_upgrade_app -i <Input project file>.oct
```

Eg.: project_upgrade_app -i BR_X20_Evalpacks.oct

A sample conversion is shown in the Figure 19: Project upgrade app - sample project



```
Administrator: C:\Windows\system32\cmd.exe
C:\openCONFIGURATOR_ProjectUpgrade_App>project_upgrade_app.exe -i C:\openCONFIGURATOR_projects\BR_X20_Evalpacks\BR_X20_Evalpacks.oct
openCONFIGURATOR project upgrade utility - v1.1.0
Kalycito Infotech Private Limited
-----
Upgrades pre v1.4.0 openCONFIGURATOR projects to v1.4.0
Usage: project_upgrade_app.exe -i input_project_file
Enter the XDD/XDC path for the following nodes if available...
NodeID:240 Name:openPOWERLINK_MN Octx:octx/240.octx xddPath:C:\Program Files (x86)\openCONFIGURATOR-1.4.0\resources\openPOWERLINK_MN.xdd
NodeID:1 Name:EvalPackA Octx:octx/1.octx xddPath:C:\XDD\0100006C_X20BC0083_1.xdc
NodeID:2 Name:EvalPackB Octx:octx/2.octx xddPath:
NodeID:3 Name:EvalPackC Octx:octx/3.octx xddPath:C:\XDD\0100006C_X20BC0083_3.xdc
The project has been upgraded successfully. The new project file is located at C:\openCONFIGURATOR_projects\BR_X20_Evalpacks\BR_X20_Evalpacks.xml
Do you want to have a look at the upgrade log(y/n):y
C:\openCONFIGURATOR_ProjectUpgrade_App>
```

Figure 19: Project upgrade app - sample project

Important:



XDD/XDC file names for each node must be unique or they will be overwritten in the deviceImport folder.

10. References

- EPSG Draft Standard 301 v1.2.0 available at <http://www.ethernet-powerlink.org>
- XML Device Description Implementation Guidelines v1.0.0 available at <http://www.ethernetpowerlink.org>
- openCONFIGURATOR High level design document v1.3 available at <http://www.sourceforge.net/projects/openconf>
- openCONFIGURATOR User quick start guide v1.4.1 available at <http://www.sourceforge.net/projects/openconf>
- openPOWERLINK wiki pages available at <http://sourceforge.net/p/openpowerlink/wiki/>
- Complete openPOWERLINK guide is available at <http://openpowerlink.sourceforge.net/>

11. Support

11.1. Sourceforge forum

For more information on using openCONFIGURATOR, please post on the help forum at <http://sourceforge.net/p/openconf/discussion/help/>

11.2. Readme

The Readme.txt present in the openCONFIGURATOR installation directory lists the feature additions, bug fixes and known issues for that version.