

# **openCONFIGURATOR v1.4.0**

## **User Manual**

**v1.06**

**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 © 2014 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](mailto: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
EPSG	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

<b>Table of Figures .....</b>	<b>6</b>
<b>1. Preface.....</b>	<b>7</b>
1.1. Purpose.....	7
1.2. Intended audience and reading suggestions .....	7
1.3. Terminology used in this document.....	7
<b>2. Introduction.....</b>	<b>8</b>
2.1. What is openCONFIGURATOR?.....	8
2.2. openCONFIGURATOR features .....	8
<b>3. Install openCONFIGURATOR .....</b>	<b>9</b>
3.1. Operating environment .....	9
3.2. Default installation path.....	9
3.3. Linux .....	10
3.3.1. Package name.....	10
3.3.2. Pre-requisites.....	10
3.3.3. Installation types .....	11
3.4. Windows .....	11
<b>4. Using openCONFIGURATOR.....</b>	<b>12</b>
4.1. File menu .....	12
4.1.1. Open project.....	12
4.1.2. Save project.....	13
4.1.3. Save project as .....	13
4.1.4. Close project .....	13
4.2. Project menu .....	14
4.2.1. Build project.....	14
4.2.2. Clean project .....	14
4.2.3. Project settings .....	15
4.2.3.1. Save options .....	15
4.2.3.2. Auto generate options .....	16
4.3. View menu .....	17
4.4. Console window.....	18
4.5. Project wizard .....	19
4.5.1. Project wizard - settings.....	20
4.5.2. Project wizard – MN XDD .....	21
4.6. Add a CN .....	22
4.7. Object/sub-object properties .....	23
4.8. Process data objects .....	29
4.8.1. What is a channel? .....	29
4.8.2. Edit PDO properties .....	29
4.8.3. Mapping entry.....	32
4.8.4. PDO mapping vs access type.....	33
4.9. Delete a CN.....	34
<b>5. Output files .....</b>	<b>35</b>
<b>6. Uninstall.....</b>	<b>36</b>

---

6.1.	Linux .....	36
6.2.	Windows .....	36
<b>7.</b>	<b>Compile openCONFIGURATOR library from source .....</b>	<b>37</b>
7.1.	Pre-requisites.....	37
7.1.1.	Windows .....	37
7.1.2.	Linux.....	38
7.2.	Shared library - Compilation .....	39
<b>8.</b>	<b>Txt2Cdc.....</b>	<b>40</b>
<b>9.</b>	<b>References .....</b>	<b>41</b>
<b>10.</b>	<b>Support.....</b>	<b>42</b>
10.1.	Sourceforge forum .....	42
10.2.	Readme .....	42

## Table of Figures

Figure 1: File Menu .....	12
Figure 2: Project folder .....	13
Figure 3: Project menu .....	14
Figure 4: Project settings .....	15
Figure 5: View Menu .....	17
Figure 6: Console Window .....	18
Figure 7: Project wizard .....	19
Figure 8: Project wizard - settings .....	20
Figure 9: Project wizard - MN XDD .....	21
Figure 10: Add CN .....	22
Figure 11: Object properties .....	23
Figure 12: PDO properties .....	29
Figure 13: View of PDO mapping data .....	32
Figure 14: Delete a CN .....	34
Figure 15: Output files .....	35

# 1. Preface

## 1.1. Purpose

This document is intended for the users of openCONFIGURATOR version - 1.4.0.

## 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
	14.10
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>/Programfiles(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_1.4.0-1ubuntu12.04_i686.deb
12.04 - x64	openconfigurator_1.4.0-1ubuntu12.04_amd64.deb
14.04 - x86	openconfigurator_1.4.0-1ubuntu14.04_i686.deb
14.04 - x64	openconfigurator_1.4.0-1ubuntu14.04_amd64.deb
14.10 - x86	openconfigurator_1.4.0-1ubuntu14.10_i686.deb
14.10 - x64	openconfigurator_1.4.0-1ubuntu14.10_amd64.deb

Table 3: openCONFIGURATOR package names

#### 3.3.2. Pre-requisites

Ubuntu version	Dependencies
Common to all Ubuntu versions	libxml2, 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
14.10 - 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-1.4.0_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

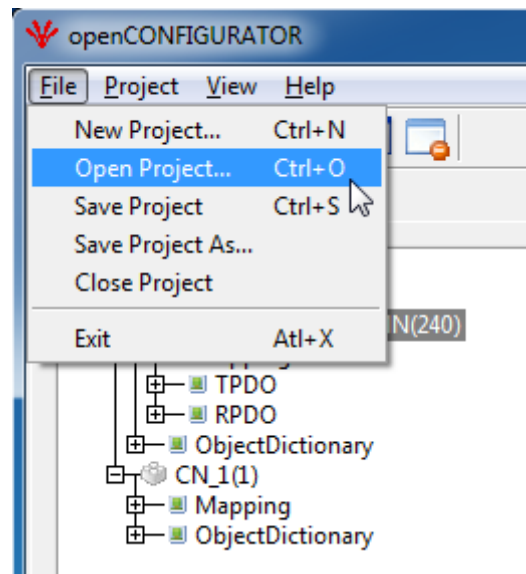


Figure 1: File Menu

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

#### **Important:**



Currently OpenCONFIGURATOR version 1.4.0 does not support projects created with earlier versions of openCONFIGURATOR.

### 4.1.2. 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.
- One XML file that contains the overall project settings and is found in the <project directory> folder.

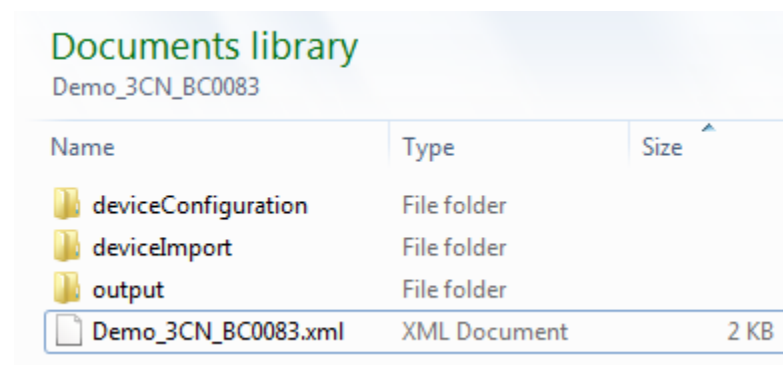


Figure 2: Project folder

### 4.1.3. 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.

### 4.1.4. Close project

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

## 4.2. Project menu

### 4.2.1. Build project

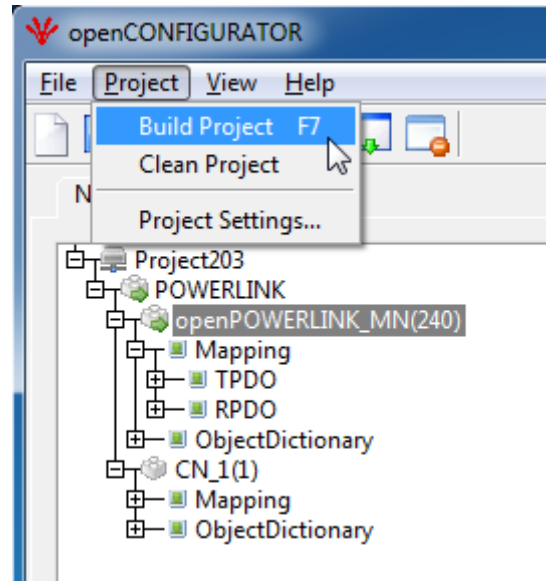


Figure 3: Project menu

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 3: 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 4: 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

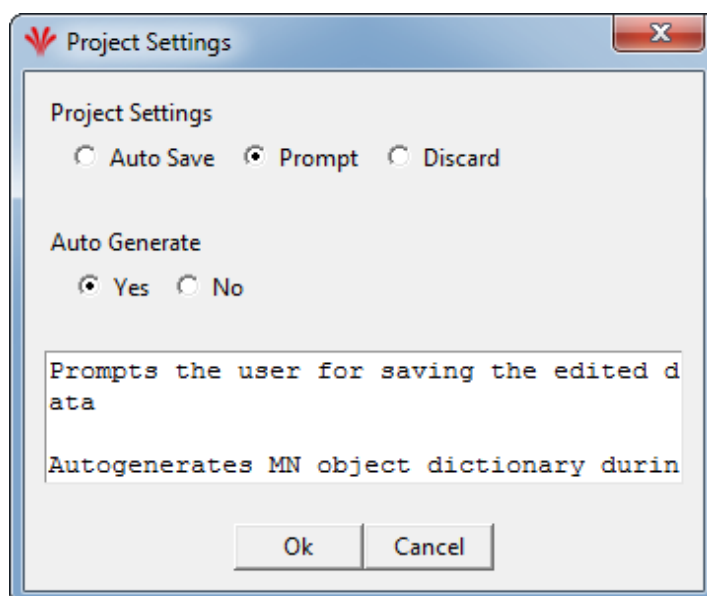


Figure 4: Project settings

#### 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 5: View Menu.

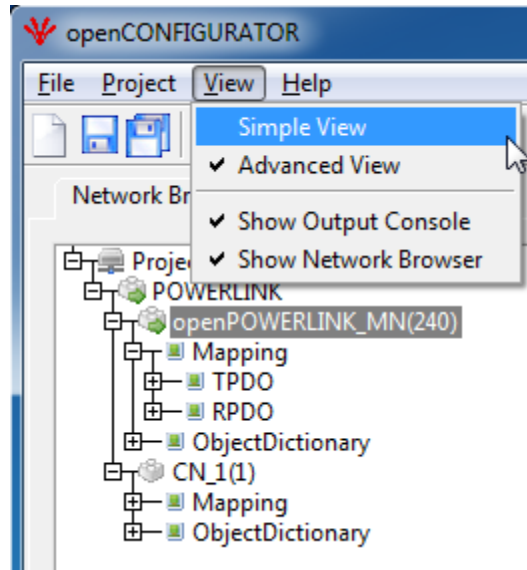


Figure 5: View Menu

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 6: Console Window.

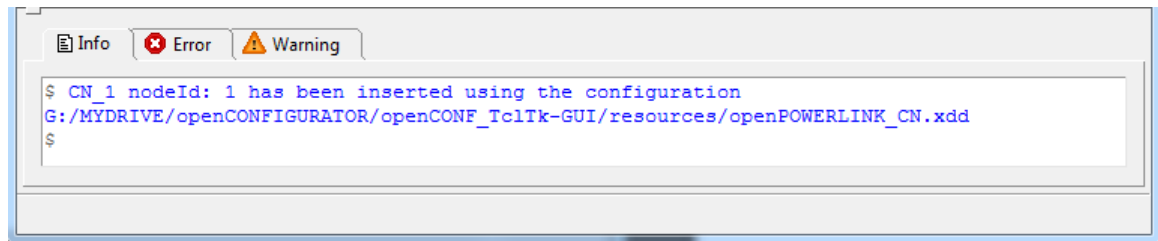


Figure 6: Console Window

## 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 7: 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.

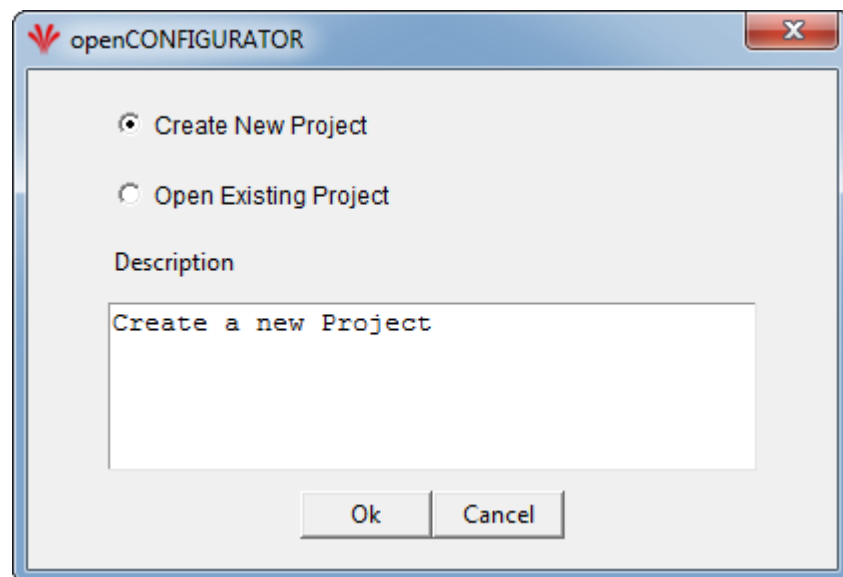


Figure 7: Project wizard

### 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 8: Project wizard - settings.

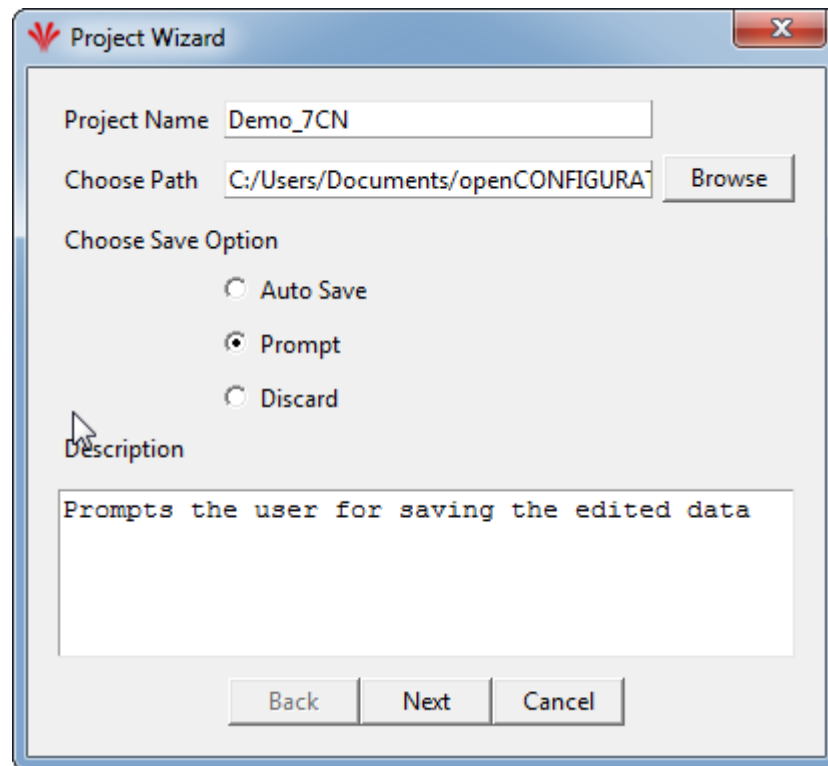


Figure 8: Project wizard - settings

- 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 9: Project wizard - MN XDD.

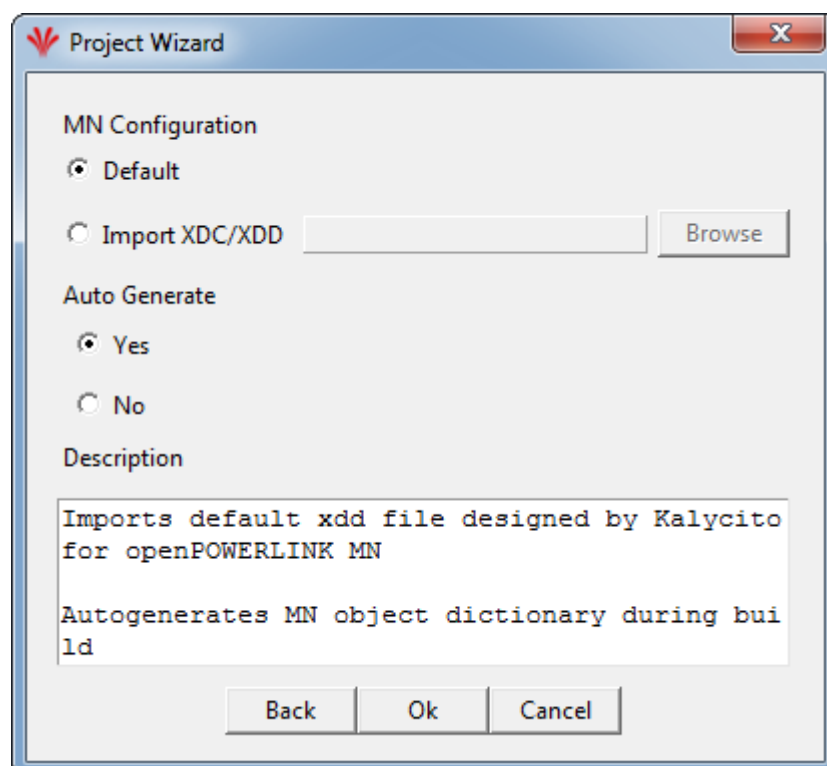


Figure 9: Project wizard - MN XDD

- 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 10: Add CN

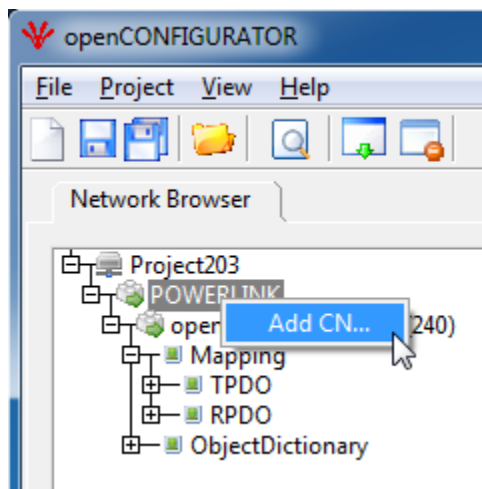


Figure 10: Add CN

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:

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 11: Object properties.

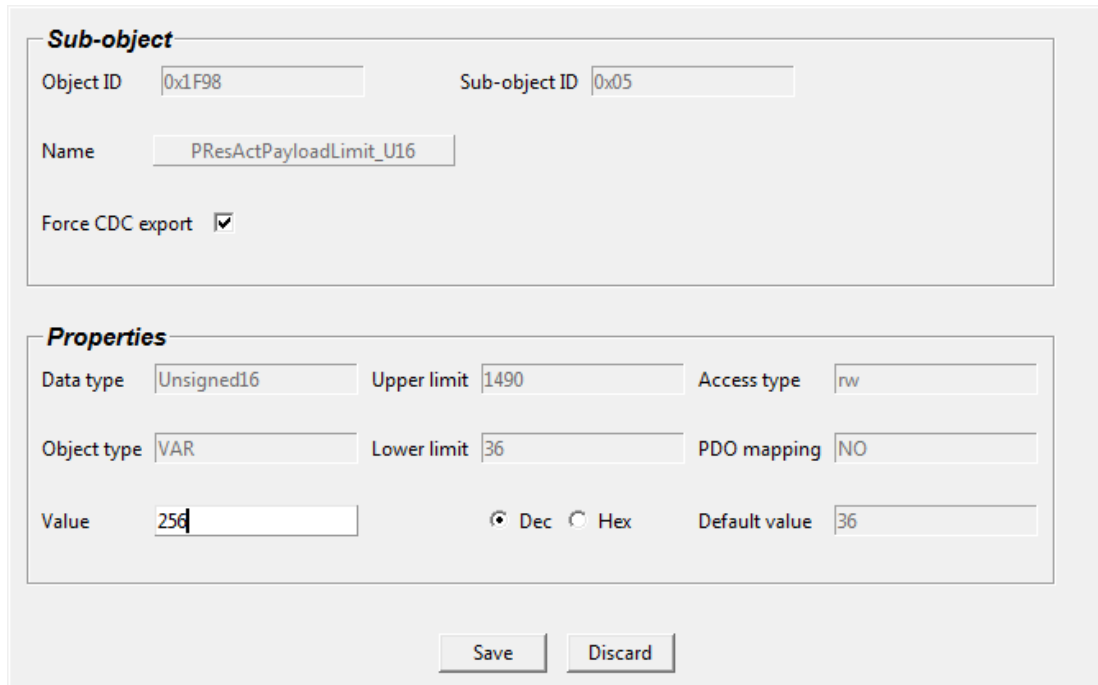


Figure 11: Object properties

- **Object ID**
  - The **Object ID** identifies the object that is currently active. This cannot be changed after creating an object.
  - 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. This cannot be changed after creating a sub-object.
  - 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 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. <b>Note:</b> 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. <b>Note:</b> 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 <sup>1</sup> padded to 8	Only supported via DOMAIN objects that uses parameter and varDeclaration. <sup>2</sup>
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 <sup>3</sup>	Not allowed
VISIBLE_STRING	0009	variable <sup>4</sup>	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

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

<sup>2</sup> Refer to the section 6.3 dataTypelist of EPSG XML Device Description – Implementation Guideline v1.0.0.

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

<sup>4</sup> 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 <sup>5</sup>	Not allowed
IP_ADDRESS	0402	32 <sup>6</sup>	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 <sup>7</sup>	Write only access.
rw / rws <sup>8</sup>	Read and write access.

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

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

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

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

<sup>8</sup> 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 <sup>9</sup>	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.

<sup>9</sup> 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.

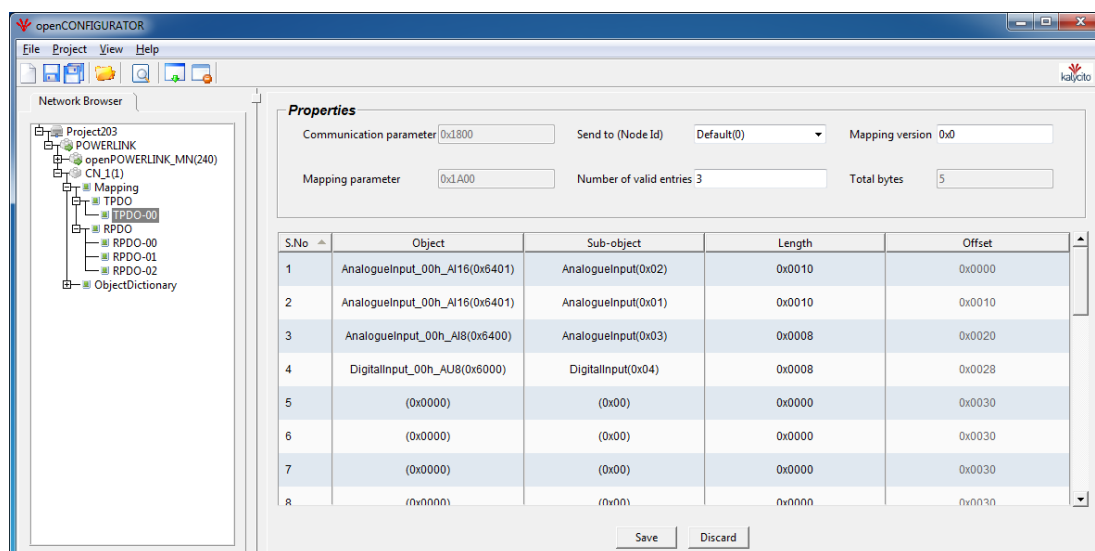
### 4.8.1. What is a channel?

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 12: PDO properties



S.No	Object	Sub-object	Length	Offset
1	AnalogueInput_00h_AI16(0x6401)	AnalogueInput(0x02)	0x0010	0x0000
2	AnalogueInput_00h_AI16(0x6401)	AnalogueInput(0x01)	0x0010	0x0010
3	AnalogueInput_00h_AI8(0x6400)	AnalogueInput(0x03)	0x0008	0x0020
4	DigitalInput_00h_AU8(0x6000)	DigitalInput(0x04)	0x0008	0x0028
5	(0x0000)	(0x00)	0x0000	0x0030
6	(0x0000)	(0x00)	0x0000	0x0030
7	(0x0000)	(0x00)	0x0000	0x0030
8	(0x0000)	(0x00)	0x0000	0x0030

Figure 12: 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 bytes<sup>10</sup>.

---

<sup>10</sup> 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 13: 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) <sup>11</sup>			
	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.

<sup>11</sup> 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 14: Delete a CN.

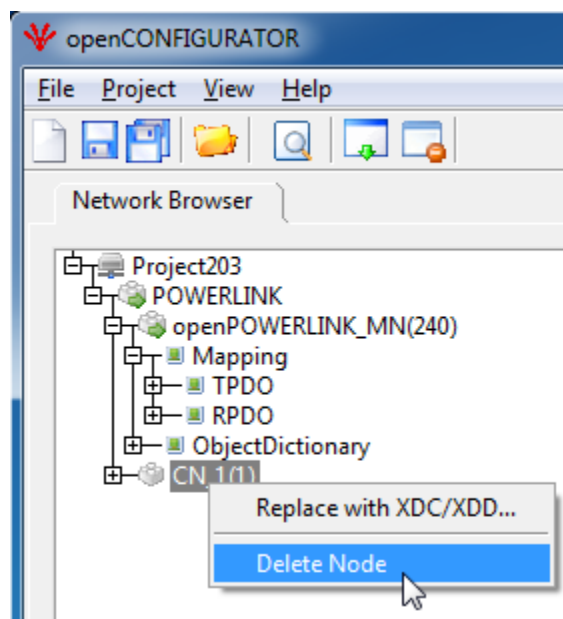


Figure 14: Delete a CN

### 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 15: 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	<a href="http://sourceforge.net/projects/openconf/">http://sourceforge.net/projects/openconf/</a>
Microsoft Visual C++	2010	<a href="http://microsoft.com">http://microsoft.com</a>
Boost libraries <sup>12</sup>	1.54.0	<a href="http://boost.org/">http://boost.org/</a>
CMake	2.8.8	<a href="http://www.cmake.org/">http://www.cmake.org/</a>
Active TCL - 32bit	8.6.0	<a href="http://www.activestate.com/activetcl/downloads">http://www.activestate.com/activetcl/downloads</a>
Swig	2.0.10	<a href="http://sourceforge.net/projects/swig/">http://sourceforge.net/projects/swig/</a>
Doxygen	1.7.6	<a href="http://www.doxygen.org/index.html">http://www.doxygen.org/index.html</a>

Table 18: Pre-requisites - Windows

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

### 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	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 <sup>13</sup>	1.54.0
	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

<sup>13</sup> 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.

## 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 Tcl/Tk wrapper 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. References

- EPSG Draft Standard 301 v1.1.0\_01 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 v1.3 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/>

## **10. Support**

### **10.1. Sourceforge forum**

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

### **10.2. Readme**

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