User Manual

For
openCONFIGURATOR
Version 1.3.0

Prepared By
Kalycito Infotech Pvt Ltd.,
India

Identifier	User_Manual_openCONF	**Version**	1.05
Prepared By	Ramakrishnan P	**Date**	06-Sep-2013
Approved By	Vinod PA	**Confidentiality**	Public Domain Document

Document Revision History

Version	Date	Modified By	Remarks
0.01	31-Mar-2009	Powerlink Team – Kalycito	Initial Draft
0.02	09-Apr-2009	Powerlink Team – Kalycito	Incorporates information for Vista
1.00	16-Apr-2009	Powerlink Team – Kalycito	Incorporates Known bugs and their work around and compilation - how to
1.01	08-May-2009	Powerlink Team – Kalycito	Updated known bugs and their updated work around(s). New Notes to the user are added.
1.02	22-May-2009	Powerlink Team – Kalycito	Added installation procedure for RPM and Debian based systems and few cosmetic changes
1.03	25-Apr-2013	Ramakrishnan P	Major changes to the document architecture, Updated with the features and bug fixes till v1.3rc1
1.04	05-Jul-2013	Ramakrishnan P	Updated with the features and bug fixes for the v1.3rc2. Added several new sections.
1.05	06-Sep-2013	Ramakrishnan P	Added CMake compilation steps.

License

The application and the source code for the openCONFIGURATOR is licensed under BSD license by Kalycito Infotech Private Limited. Please refer to the file's header 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.

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

Sales Office:

Kalycito Infotech Private Limited,

479-B1, Avinashi road, Peelamedu,
Coimbatore – 641004, Tamil Nadu, INDIA.
Phone 00 91 422 2565644
Email: sales@kalycito.com

Table of Contents

Figures

Table Index

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	Managing node's Object Dictionary
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

1. Introduction

1.1. Purpose

This document is intended for the users of openCONFIGURATOR version - 1.3.0.

1.2. Intended audience and reading suggestions

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

1.3. Operating environment

openCONFIGURATOR is designed to work on the following operating environments,

- Various Linux distributions
- Windows XP
- Windows Vista
- Windows 7

2. Key Features

- Generates a generic POWERLINK stack configuration (CDC format)
- Generates the network variables in multiple formats
  - xap.xml - XML file that describes the process image generally
  - 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
- 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.
- Computes the MN PDO mapping automatically
- Support for Multiplexing
- Support for PRes chaining
- Support Cross Traffic communication
- Dynamic PDO mapping
- Implements Communication profile EPSG 301 1.1.0
- Main package is available under Kalycito BSD license.

3. Setup - openCONFIGURATOR

3.1. Linux

Download the latest 32 or 64 bit version of openCONFIGURATOR from http://sourceforge.net/projects/openconf/

- Un-tar the openCONFIGURATOR.tar.gz file
- Open a terminal window and change path to the extracted directory
- To check and install the required packages, run the following command
  - *sudo ./configure*
- If configuration succeeds, 'Makefile' will be created
- To install openCONFIGURATOR, run the following command
  - *sudo make install*
- After install has completed, openCONFIGURATOR can be launched in two ways
  - From the terminal window, run the following command
    - *openCONFIGURATOR*
  - From the GUI
    - Go to Applications > Programming
    - Click on 'openCONFIGURATOR'

3.2. Windows

Download the latest 32 or 64 bit version of openCONFIGURATOR from http://sourceforge.net/projects/openconf/

- For Windows (XP, Vista and 7), please install Active TCL x86 version 8.5 from http://www.activestate.com/activetcl/downloads
- Unzip the openCONFIGURATOR.zip file and double click on the openCONFIGURATOR_Setup.exe file

Note: For Windows 7 and Vista the setup should be run as Administrator [right click on the setup file and click on 'Run as Administrator']

- Now the Installer Dialog will open as shown in the Figure 1: Installer - License page.
- Read through the License and if you agree, click on **'I Agree'** button to proceed with the installation.

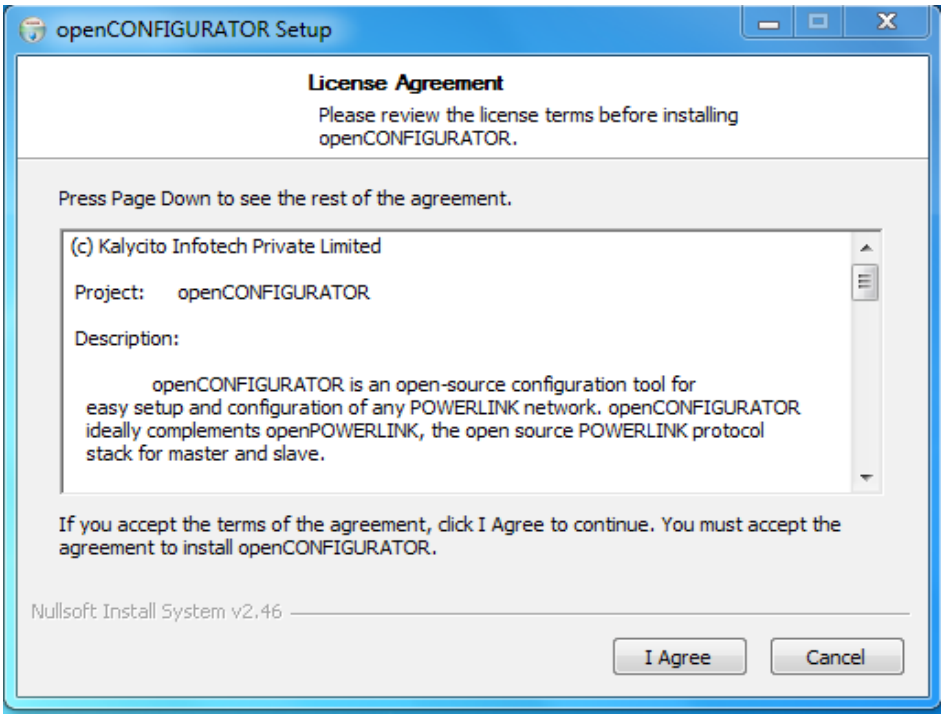

Figure 1: Installer - License page

- Click on **'Next'** button to continue with the installation.

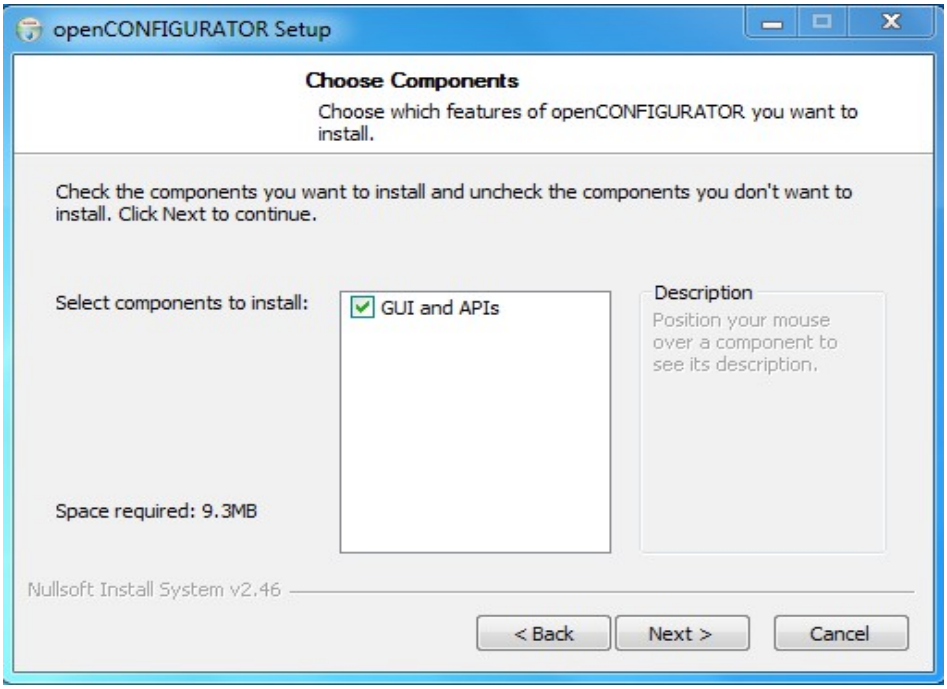

Figure 2: Installer - Components Page

- Select the directory where the application should be installed and click on **'Next'** button.

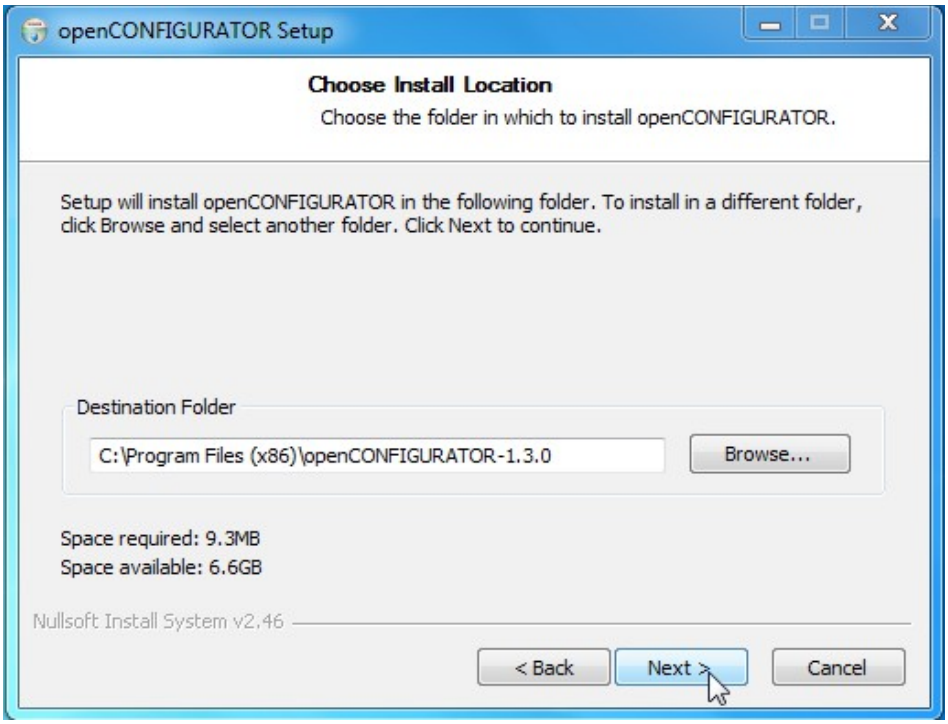

Figure 3: Installer - Install Path

- Select **'Do not create shortcuts'** check box if you do not wish to create start menu entry and click on **'Install'** button to complete the installation.

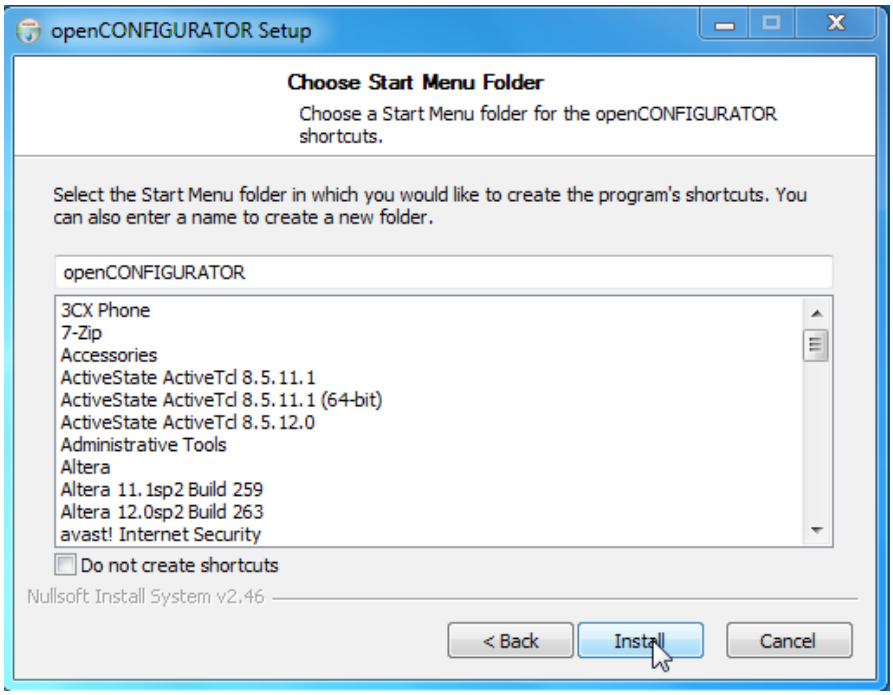

Figure 4: Installer - Start Menu

- For launching openCONFIGURATOR
  - Go to Start Menu > All Programs > openCONFIGURATOR
  - Click on openCONFIGURATOR

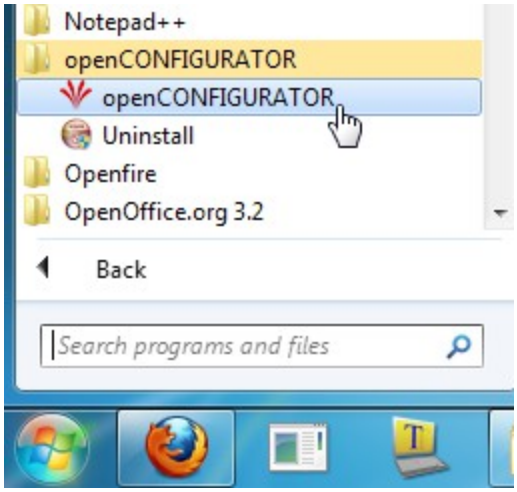

Figure 5: Windows - Launch

Note: If you have chosen not to create shortcuts, you will not find the start menu entry. So you can launch the application by double clicking on the openCONFIGURATOR.exe in the installation directory that you've set during the installation.

4. Using openCONFIGURATOR

4.1. File Menu

4.1.1. New Project

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

4.1.2. Open Project

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

Note: It is highly recommended to maintain the projects in the latest version of openCONFIGURATOR.

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 the **'Save'** icon.

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

- One octx file for each CN is present in the project. The name of the file is the 'NodeId' of the CN.
- One octx file exists for the MN. The name of the file is '240' (NodeId of MN).
- One oct file contains the overall project settings and is found in the <project directory> folder.

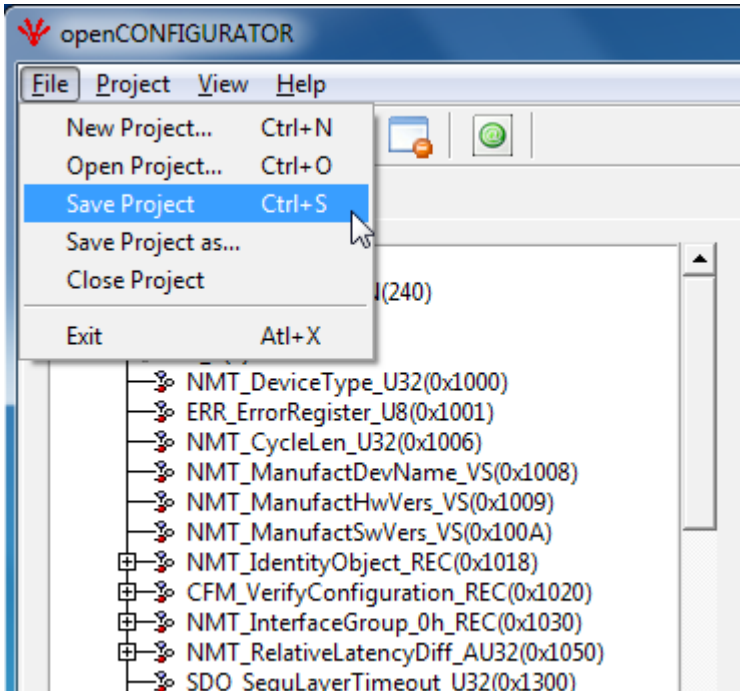

Figure 6: Save Project Menu

4.1.4. Save Project As

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

4.1.5. Close Project

By selecting this option the openCONFIGURATOR will close the project. This can be done by selecting **"File > Close Project**" as shown in the Figure 7: Close Project.

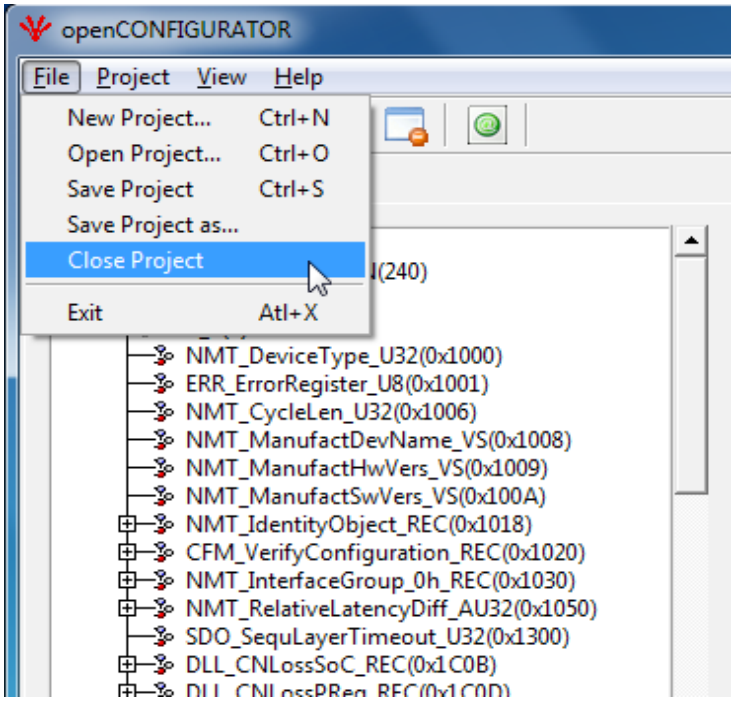

Figure 7: Close Project

4.2. Project Menu

4.2.1. Build Project

The user can build the project by selecting **'Project > Build Project'** or by using the function key **'F7'** or by clicking on the **'Build Project'** button as shown in the Figure 8: Build Project Menu.

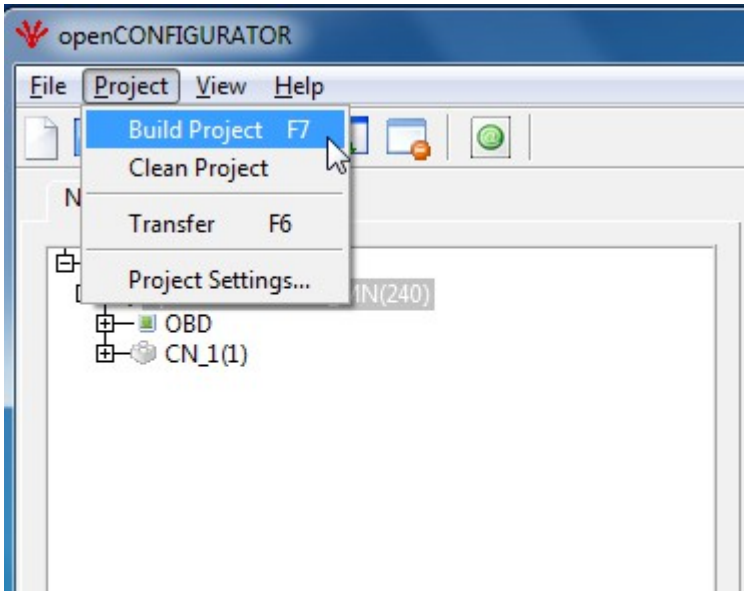

Figure 8: Build Project Menu

4.2.2. Clean Project

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

4.2.3. Transfer

The output files generated by openCONFIGURATOR can be directly copied to a user specified path by following the steps given below:

- Set the destination path in the **Transfer.bat(Windows)/Transfer.sh(Linux)** file present in the installation directory.
- Copy the output files (Eg: mnobd.cdc, xap.h) from the project by selecting the **'Project > Transfer'** or by using the Function key '**F6**'.

4.2.4. Project Settings

The user can at any time change the settings for the project by selecting the **'Project > Project Settings'** option.

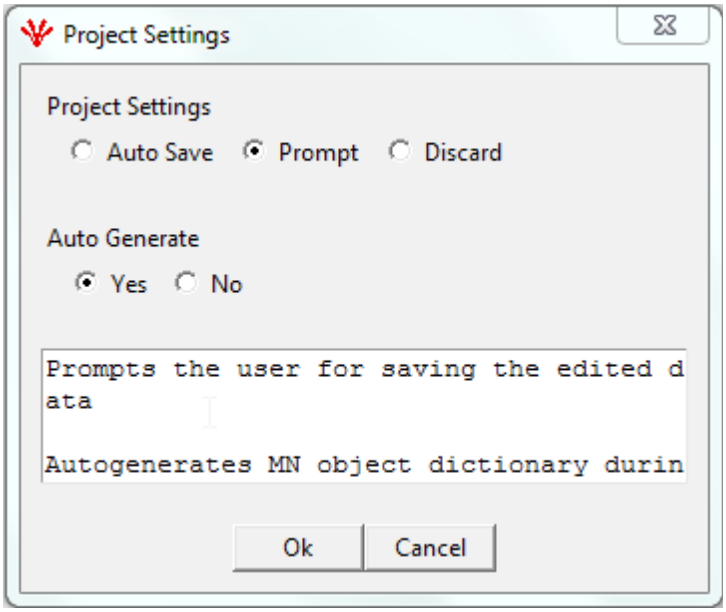

Figure 9: Project Settings Window

4.3. View Menu

The user can switch between the **'Simple View'** and **'Advanced View'** options of the tree browser.

In **'Simple View'**, the attributes listed below are visible in the tree browser:

- CN Name
- Node ID

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

- Index of Node
- SubIndex of an Index
- PDO of a Node

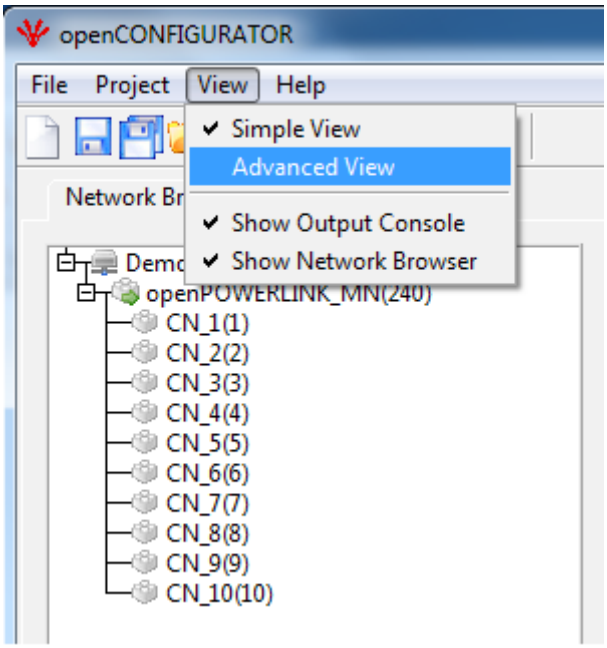

Figure 10: View menu

4.4. Console window

The user can view the status messages, warnings and error messages during build in the console window.

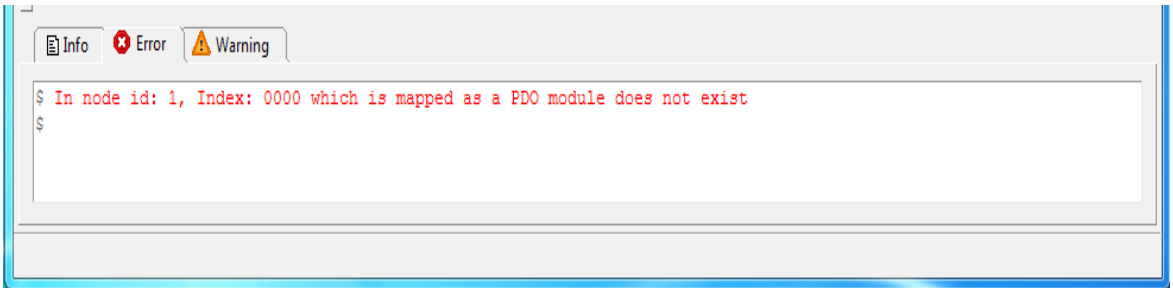

Figure 11: Console window

4.5. Project wizard

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

- When the user launches openCONFIGURATOR, it will ask either to **'Create New Project'** or **'Open Existing Project'** as shown in the Figure 12: Create New Project.
- 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.

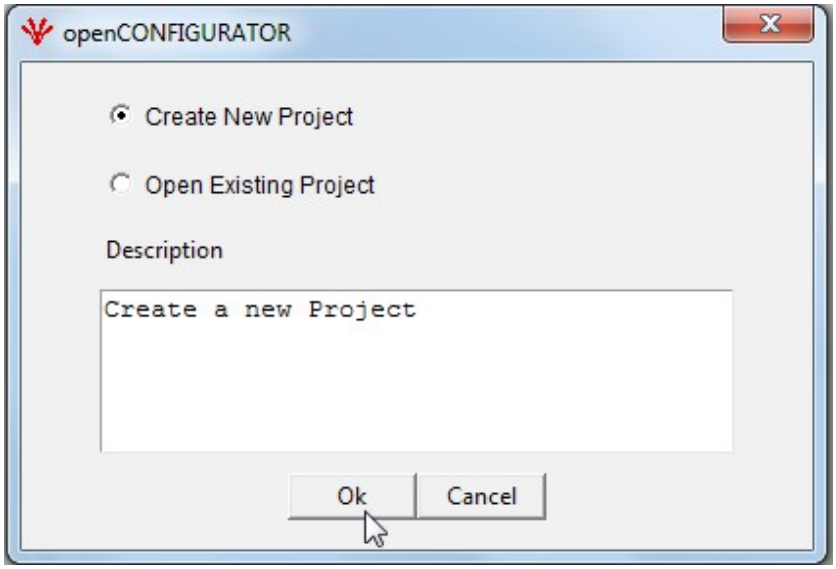

Figure 12: Create New Project

4.5.1. Project Wizard - Name

The options given below are to be entered/selected as applicable in the **'Project Wizard'** dialog box before clicking **'Next'**:

- **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.

Note: The default path is the user's home directory/openCONFIGURATOR_Projects.

- **Choose Save Option**

Save option	Description
Auto Save	Saves the configuration automatically without prompting the user
Prompt	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 1: Save options

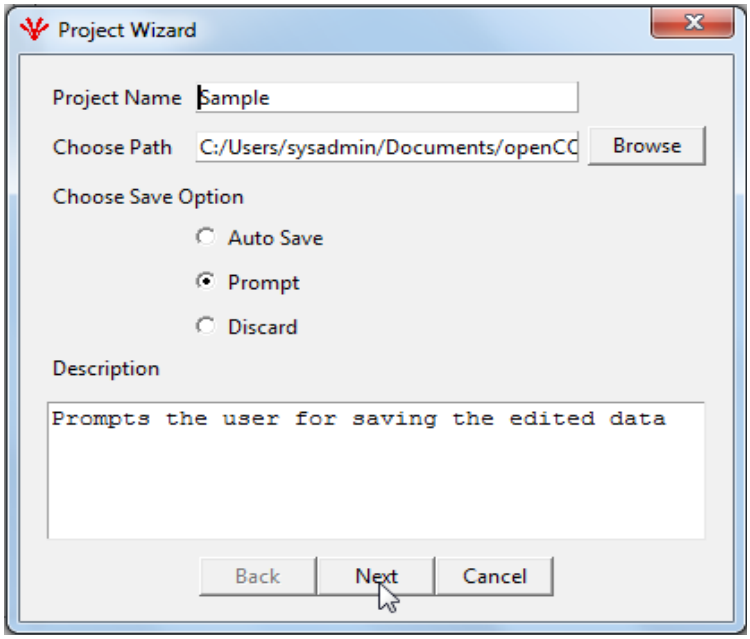

Figure 13: Project Wizard - Name

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'**. They appear as shown in the Figure 14: Project Wizard – MN XDD:

- **MN Configuration**

Configuration Option	Description
Default	Default MN XDD which will be available with the installation package
Import XDD/XDC	User defined MN configuration

Table 2: MN Configuration

- **Auto Generate**

Auto Generate Option	Description
Yes	Available CN configurations will be used to auto generate MN configuration
No	The user will have to manually generate/update the MN configuration

Table 3: Auto Generate Option

- 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 station is chained
  - Process Image Variables
- If **'Auto Generate'** is set to **'No'**,:
  - PDO mapping of MN will not be generated automatically.
  - The user has to manually create/update the mapping and relevant changes in the MN's object dictionary.
  - The changes made to the project are reflected in the CDC regardless of the correctness of the configuration data.

Note: The user can change the 'Auto Generate Mode' at any time from the 'Project Settings' window by clicking on the **Project > Project Settings.** Refer to the Figure 9: Project Settings Window

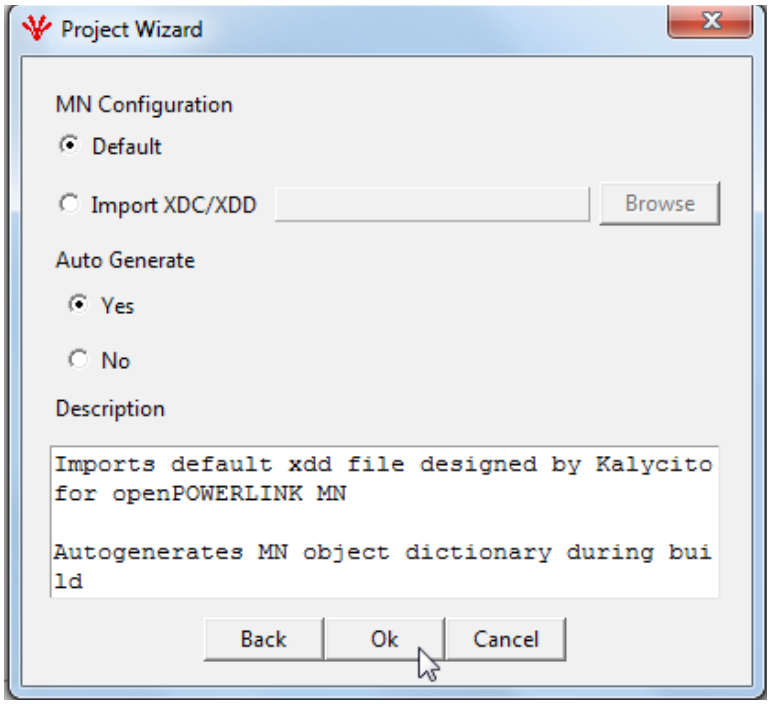

Figure 14: Project Wizard – MN XDD

- An alternate way to do an "Auto Generate" is to right click on the MN and select 'Auto Generate' as shown in the Figure 15: Auto Generate MNOBD.

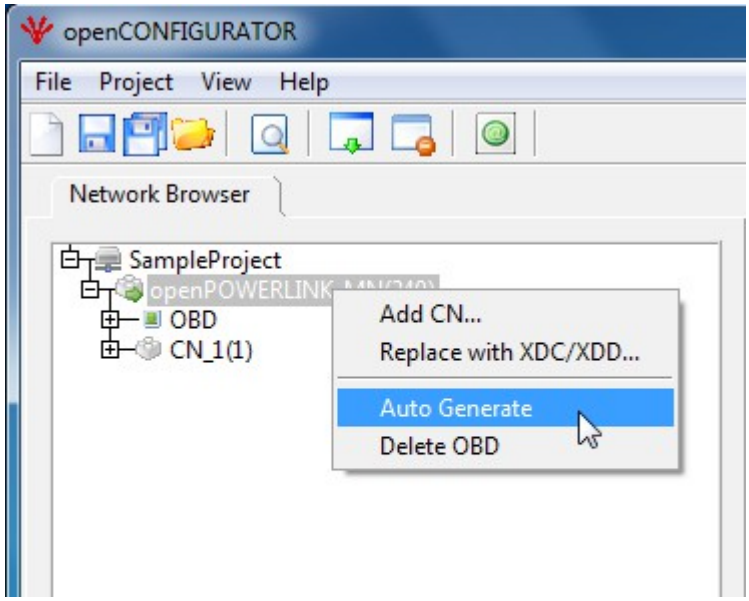

Figure 15: Auto Generate MNOBD

4.6. Adding a CN

A CN can be added by right clicking on the MN and selecting 'Add CN' option from the sub menu that appears as shown in the Figure 16: Add CN Menu.

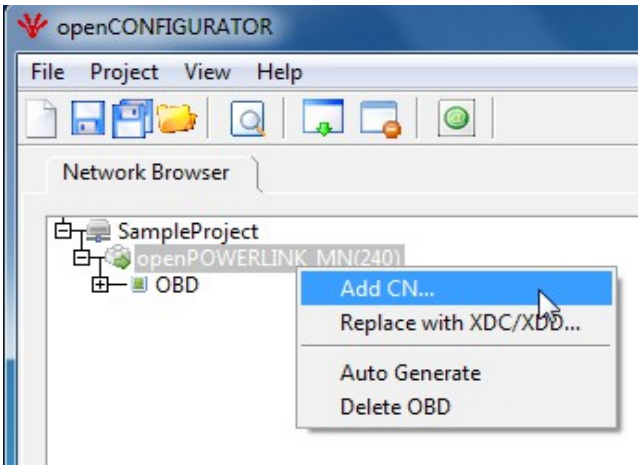

Figure 16: Add CN Menu

After clicking on 'Add CN' a pop-up will appear where the user can enter a CN configuration by referring to the Table 4: New CN configuration and can select the XDD/XDC files for that CN as shown in the Figure 17: Add CN Window.

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 4: New CN configuration

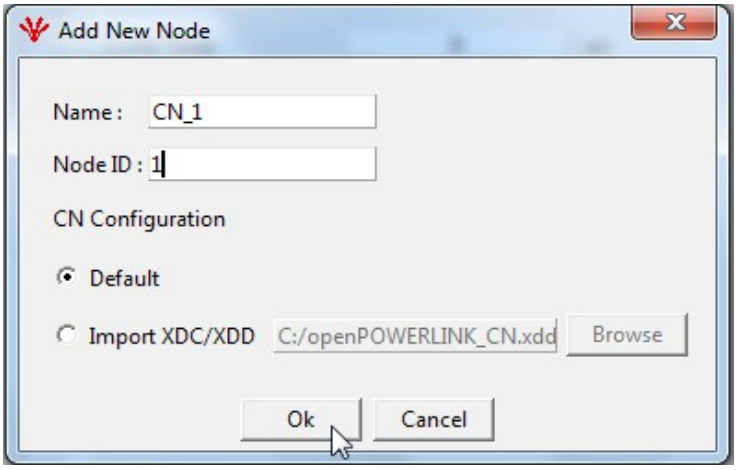

Figure 17: Add CN Window

Note: Please validate your XDD with the XDD-Check tool (a free utility available at the EPSG homepage: http://www.ethernet-powerlink.org)

4.7. Adding an Index

The Index can be added for an MN/CN by right clicking on node of which the Index has to be added and clicking '**Add Index'** option as shown in the Figure 18: Add Index Menu.

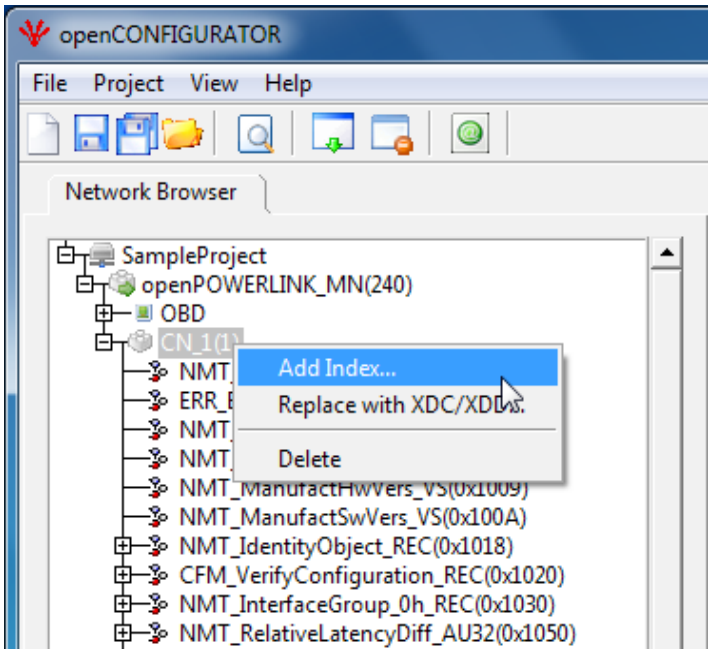

Figure 18: Add Index Menu

After selecting the **'Add Index'** option, a pop-up with a text box will appear as shown in Figure 19: Add Index Window. The user can add an index by referring to the Table 5: Index configuration.

Name	Range (hex)	Applies to	Adds with default properties	Properties Editable in	
				MN	CN
Communication Profile Area Objects	1000 - 1FFF	MN and CN	Yes	Partial	Partial
Manufacture Specific Profile Area Objects	2000 - 5FFF	CN	No	Yes	Yes
Standardized Device Profile Area Objects	6000 - 9FFF	CN	No	Yes	Yes
Standardized Interface Profile Area Objects	A000 - BFFF	MN	No	No	Yes
Reserved for further use	C000 - FFFF	Reserved	No	Yes	Yes

Table 5: Index configuration

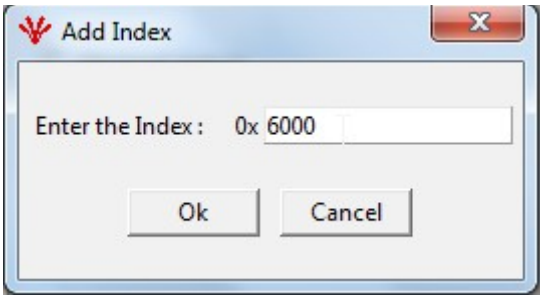

Figure 19: Add Index Window

The user can enter the Index Id and click **'Ok'**. The Index will be added in the node as shown in the Figure 20: Index Added - Tree.

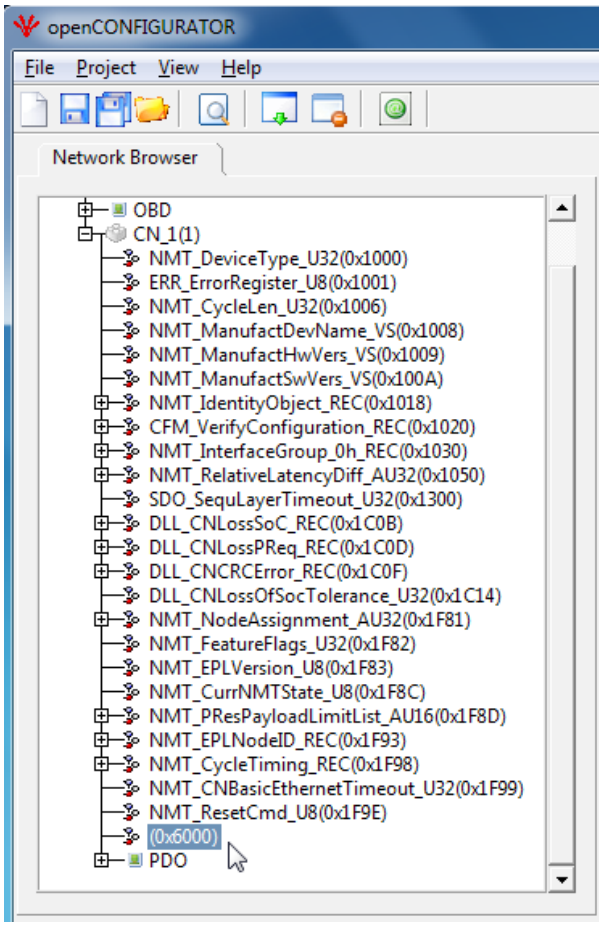

Figure 20: Index Added - Tree

See "Editing Object / SubObject Properties" on how to add properties for the Index.

Note: Index Id's (0x14xx, 0x16xx, 0x18xx, 0x1Axx) will be added under the PDO section. The user can also add those indexes by right clicking on PDO.

4.8. Adding a SubIndex

To add a SubIndex, right click on the respective Index and select **'Add SubIndex'** from the sub-menu that appears. Refer to Table 6: SubIndex configuration for more details.

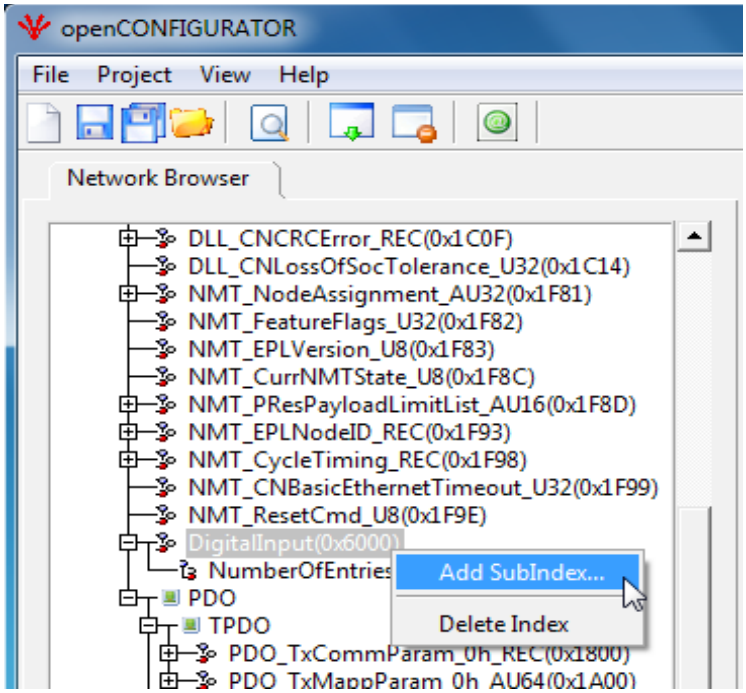

Figure 21: Add SubIndex Menu

Name	Index Range	SubIndex Range	Default Properties	Properties Editable in	
				MN	CN
Communication Profile Area Objects	0x1000 - 0x1FFF	0x00 - 0xFE*	Yes	Partial	Partial
Other Objects	0x2000 - 0xFFFF	0x00 - 0xFE	No	Yes	Yes

Table 6: SubIndex configuration

Note: The Objects with ObjectType Var and DefType cannot have SubIndexes.

* Refer to the EPSG_Specification

4.9. Editing Object / SubObject Properties

The user can edit the properties for the Objects/SubObjects by referring to the Figure 22: Edit an Object.

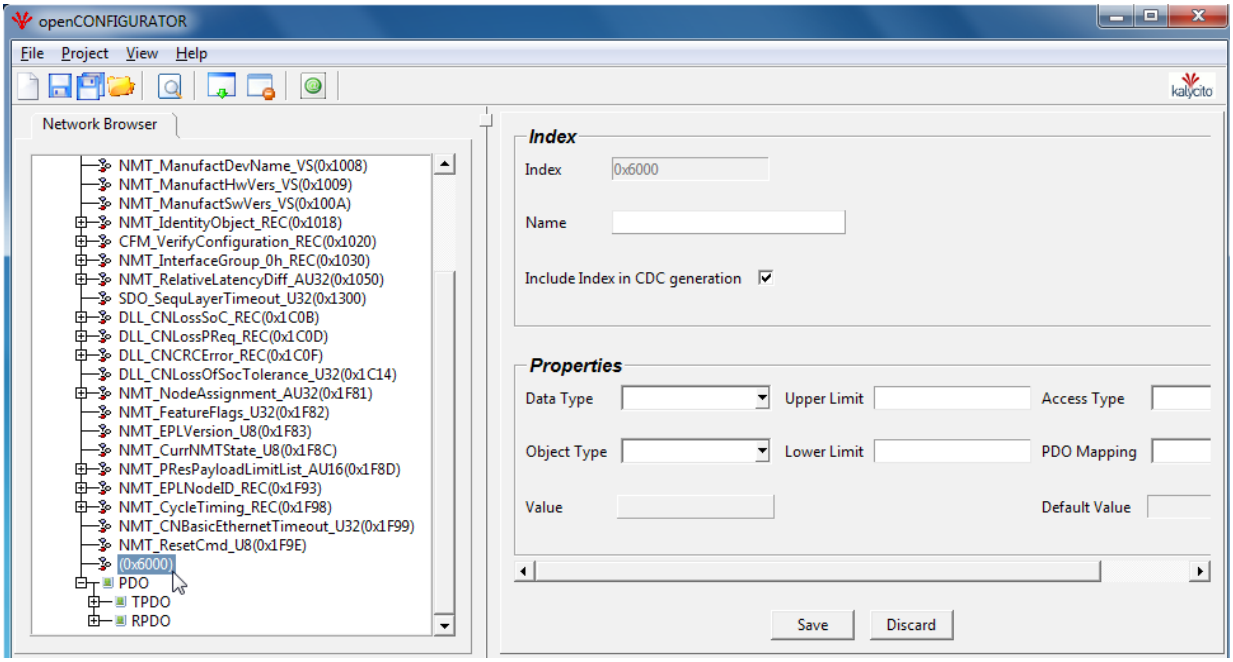

Figure 22: Edit an Object

- **Index**
  - The Index is the Id for the Object/Subobject which will be added. This cannot be changed after creating an Index/SubIndex.
  - Index shall be declared as hexadecimal value.
  - Eg: 0x1F81, 0x25F4, 0x6201, 0xA480 – for Objects and 0x00, 0xFE – for SubObjects.
- **Name**
  - Name provides a textual description of the function of that particular 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.,
- **Include In CDC generation**
  - Include in CDC check determines the inclusion of the actual value of the Index/SubIndex in the CDC.

Note: To include a Sub-Index in the CDC generation, its parent Index should also be included in the CDC generation.

- **Object Type**

Object Type	Comments
VAR	An object with a single value such as Unsigned16, Integer32, OctetString, MAC address, etc.,
ARRAY	A multiple data field object where each field is a simple variable of the same basic datatype. Eg: array of unsigned64 etc., **Note:** SubIndex 0x00 is always of Unsigned8 and therefore not part of ARRAY data.
RECORD	A multiple data field object where the data fields may be any combinations of simple variables. **Note:** SubIndex 0x00 is always of Unsigned8 and therefore not part of RECORD data.

Table 7: Object type definitions

Note: Object of type DEFTYPE and DEFSTRUCT cannot be created via the application.

- **Data Type**
  - The following table provides the list of supported datatypes, its data size and whether it is allowed to be mapped to a PDO object.

Data Type	Data Size(Bits)	Allowed for PDO mapping
BIT, BOOLEAN	1*	Not supported
INTEGER8, UNSIGNED8	8	Yes
INTEGER16, UNSIGNED16	16	Yes
INTEGER24, UNSIGNED24	24	Not supported
INTEGER32, UNSIGNED32	32	Yes
INTEGER40, UNSIGNED40	40	Not supported
INTEGER48, UNSIGNED48	48	Yes
INTEGER56, UNSIGNED56	56	Not supported
INTEGER64, UNSIGNED64	64	Yes
REAL32	32	Not supported
REAL64	64	Not supported
MAC_ADDRESS	48*	Not supported
IP_ADDRESS	32*	Not supported
OCTET_STRING (32Characters max)	128*	Not supported

Table 8: Datatype definitions

- Refer to section 6.1.4 of the EPSG_Specification

- **PDO Mapping**
  - Indicates whether an entry may be mapped to a PDO.

PDO Mapping	Description
NO	Objects cannot be mapped to a PDO
DEFAULT *	Objects is a part of the default mapping
OPTIONAL	Objects may be mapped into a PDO
TPDO	Objects shall be mapped to a Transmit PDO
RPDO	Objects shall be mapped to a Receive PDO

Table 9: PDO mapping list

* Refer to section 6.2.1 of the EPSG_Specification

- **Access Type**
  - Defines the access rights for a particular object.

Access Type	Description
Const	Read only access, value is constant
RO	Read only access
WO	Write only access
RW	Read and write access

Table 10: Access type list

- Default Value
  - The default value that is present in the node for the respective object.
- Value (Actual)
  - The desired value that overrides the default value.
- Limit
  - This property indicates the range (high and low limits) for the value in the respective object.
  - It depends on the datatype specified for the object.
- Dec/Hex radio button
  - The user can toggle between decimal or hexadecimal view of the value by choosing from these radio buttons.

4.10. 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 dropdown boxes.

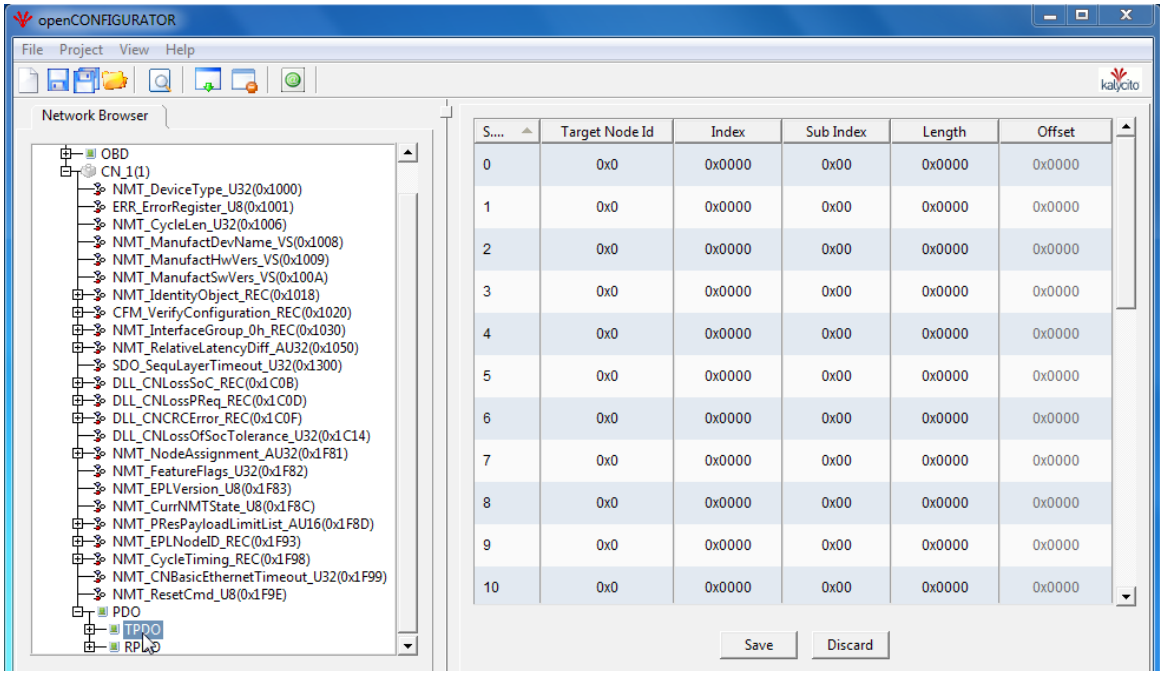

Figure 23: A sample PDO mapping table

4.10.1. Editing PDO objects

Column	Description	Allowed Range	
Target Node Id	Node Id of the PDO target	0x00	Broadcast Node Id
		0x01 – 0xEF	Available CN Node Id
		0xF0	MN Node Id
Index	Index of the object to be mapped	0x1000 - 0x9FFF	Index/SubIndex that passes the mapping criteria will be listed. Refer to the Table 12: PDO Mapping vs AccessType
Sub Index	Sub-Index of the object to be mapped	0x00*, 0x01 - 0xFE	
Length	Length of the mapped object (Bit count)	Depends on the DataType of the Index / SubIndex object	
Offset	Offset related to the start of the PDO payload (Bit count)	Cumulative sum of the payload length	

Table 11: PDO table properties

4.10.2. PDO mapping vs AccessType

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

		If object mapped to a RPDO (1600 – 16FF)				If object mapped to a TPDO (1A00)*			
		Access Type property for an object							
		Const	Ro	Wo	Rw	Const	Ro	Wo	Rw
PDO Mapping property of an Object	No								
	Default			✓	✓		✓		✓
	Optional			✓	✓		✓		✓
	TPDO						✓		✓
	RPDO			✓	✓				

Table 12: PDO Mapping vs AccessType

* A CN can have only one TPDO.

- **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.

Note: Also refer to the list of datatypes allowed for PDO mapping in the Table 8: Datatype definitions

4.11. Delete SubIndex

The subIndex of an Index of a particular node can be deleted by clicking on the node, then right clicking on the SubIndex which has to be deleted in the expanded node tree, and clicking on the 'Delete SubIndex' option in the menu that appears as shown in the Figure 24: Delete SubIndex.

Note: The SubIndex/ID 'NumberOfEntries'/0x00 cannot be deleted.

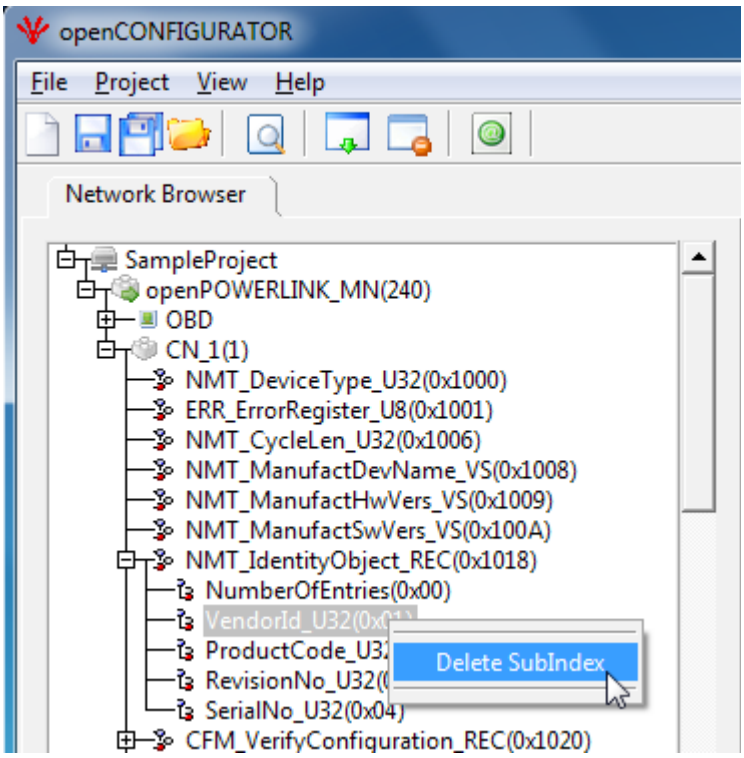

Figure 24: Delete SubIndex

4.12. Delete Index

The index of a particular node can be deleted by clicking on the node, then right clicking on the Index which should be deleted in the expanded node tree, and clicking on the 'Delete Index' option in the menu that appears as shown in the Figure 25: Delete an Index.

Warning: If the user wishes to delete the index, the sub-indexes present under the index will also be deleted. The user cannot undo the operation.

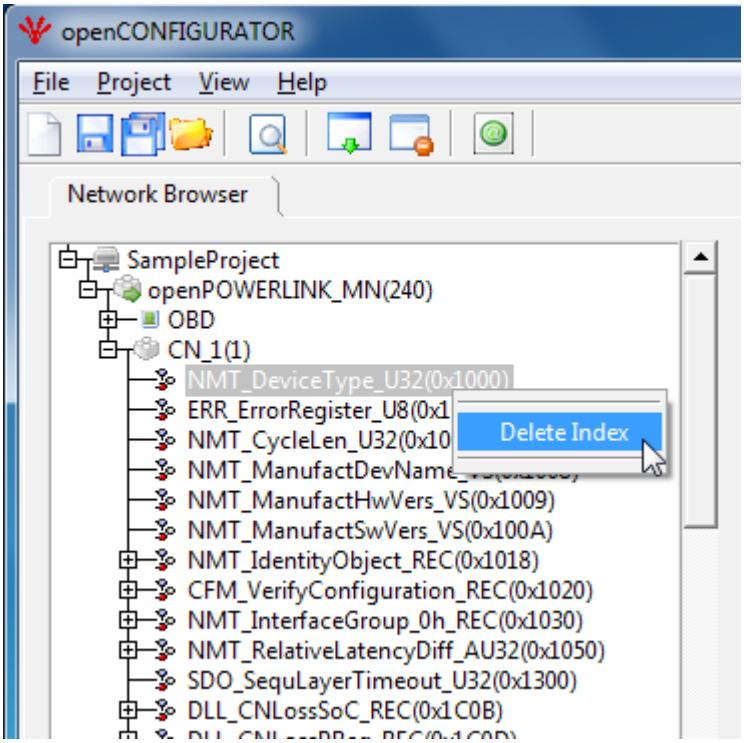

Figure 25: Delete an Index

4.13. Delete a CN

A CN can be deleted by right clicking on the node. A menu will appear as shown in the Figure 26: Delete a CN.

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

Tip: Instead of deleting and creating a CN, the user can also replace the configuration files by choosing the **"Replace with XDC/ XDD"** option in the submenu. This will update the node's configuration with the new configuration.

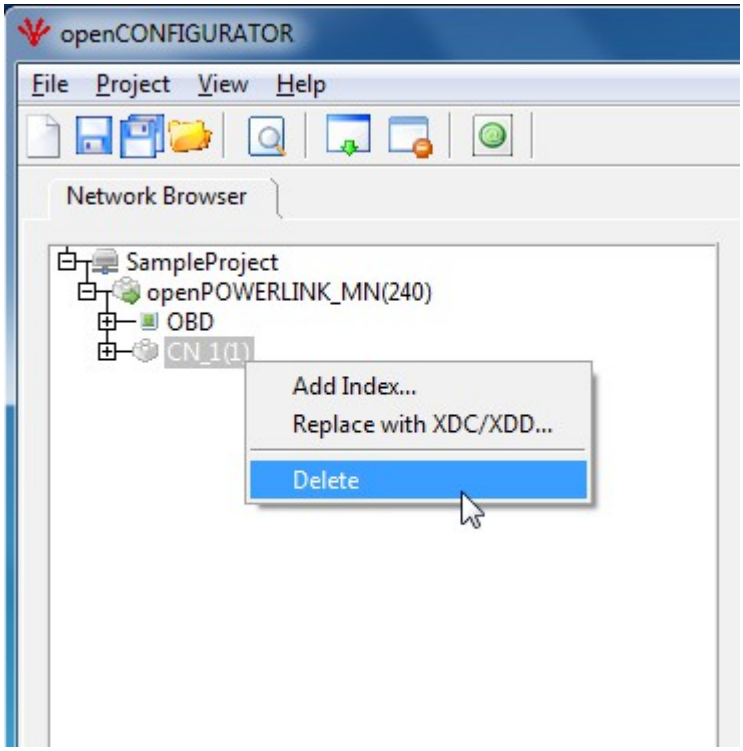

Figure 26: Delete a CN

5. Output Files

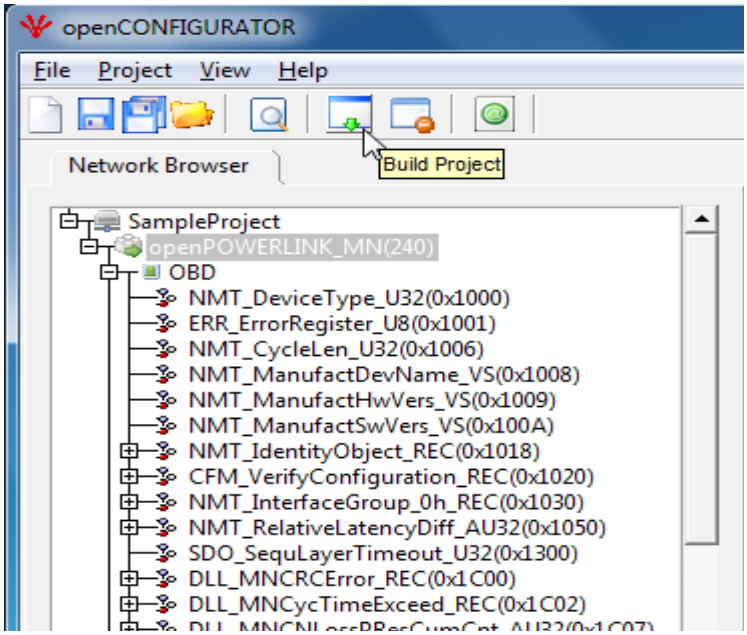

Figure 27: Build Project Icon

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

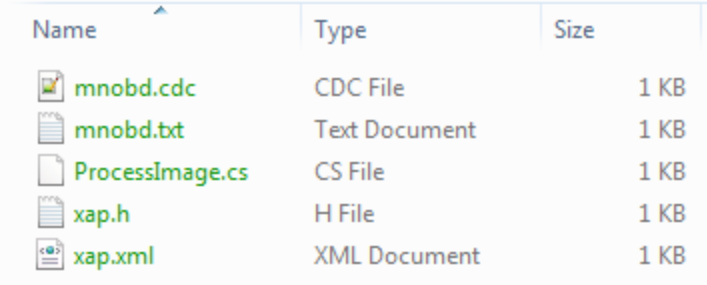

Figure 28: cdc_xap Folder View

File name	Description
mnobd.cdc	CDC 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
XAP.xml	XML file that describes the process image with the variables names, Datatype, Datasize, ByteOffsets, BitOffsets
ProcessImage.cs	C# class that describes the process image for .NET projects

Table 13: Output Files

6. Uninstall

6.1.1. Linux

- Un-tar the openCONFIGURATOR_linux.tar.gz file
- Change to the directory and run the command given below in the Terminal
  - *./configure*
- Run the command given below in the terminal
  - sudo make uninstall

6.1.2. Windows (XP)

- Go to Start Menu > All Programs > openCONFIGURATOR
- Click uninstall shortcut and follow the uninstaller instructions.

Figure 29: Windows XP - Start Menu

Note: If the user had chosen to not create shortcuts during installation, the start menu entry will not be available. In such a case the user can uninstall openCONFIGURATOR by double clicking on the Uninstall.exe from the installation directory that was set during the installation.

6.1.3. Windows (Vista and 7)

- Right Click **'Uninstall'** and click 'Run as Administrator'.
- Follow the uninstaller instructions to completion.

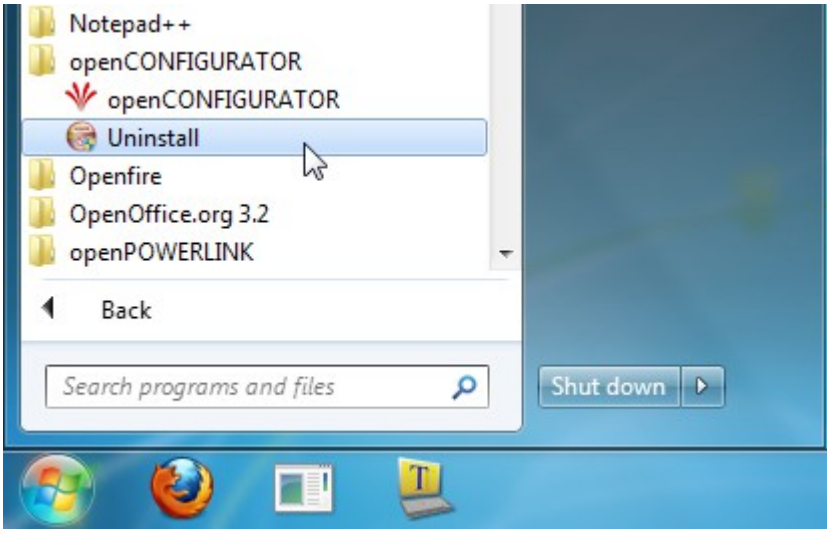

Figure 30: Uninstall - Start Menu

Note: In Vista, if Un-installation is not done as 'Administrator', the installed files will not be deleted and any further installations may not be proper. In such a case, the user shall manually delete the files installed in the corresponding directory.

7. Compiling openCONFIGURATOR from source

7.1. Windows

7.1.1. Pre-requisites

- openCONFIGURATOR source package
  - Download the package from http://sourceforge.net/projects/openconf/
- Microsoft Visual C++ 2008 or above
- CMake version 2.8.8 or above
  - Download and install the package from http://www.cmake.org/
- ActiveTCL 8.5
  - Download only the Windows x86 version of ActiveTCL from http://www.activestate.com/activetcl/downloads
  - Install and set the path of the installed directory as environmental variable
    - TCL_PATH = “<ActiveTCL Installed Directory>”
    - Note: Make sure that the environmental path is set correctly.
- Swig-2.0.10 or above
  - Download the Swig-2.0.10 package from http://sourceforge.net/projects/swig/
  - Unzip package and append the path to the environmental variable
    - Path = “<Swig extracted Directory>”
- Doxygen (optional)
  - It can be downloaded from http://www.doxygen.org/index.html.

7.1.2. DLL compilation

- Unzip the downloaded source package.
- Move to '<openCONFIGURATORSoln.zip\openCONFIGURATORSoln>'.
- Run CMake, set the source and binary paths and click on **'Configure'**.
- Select your preferred C/C++ compiler and Click **'Finish'** to complete the configuration process.

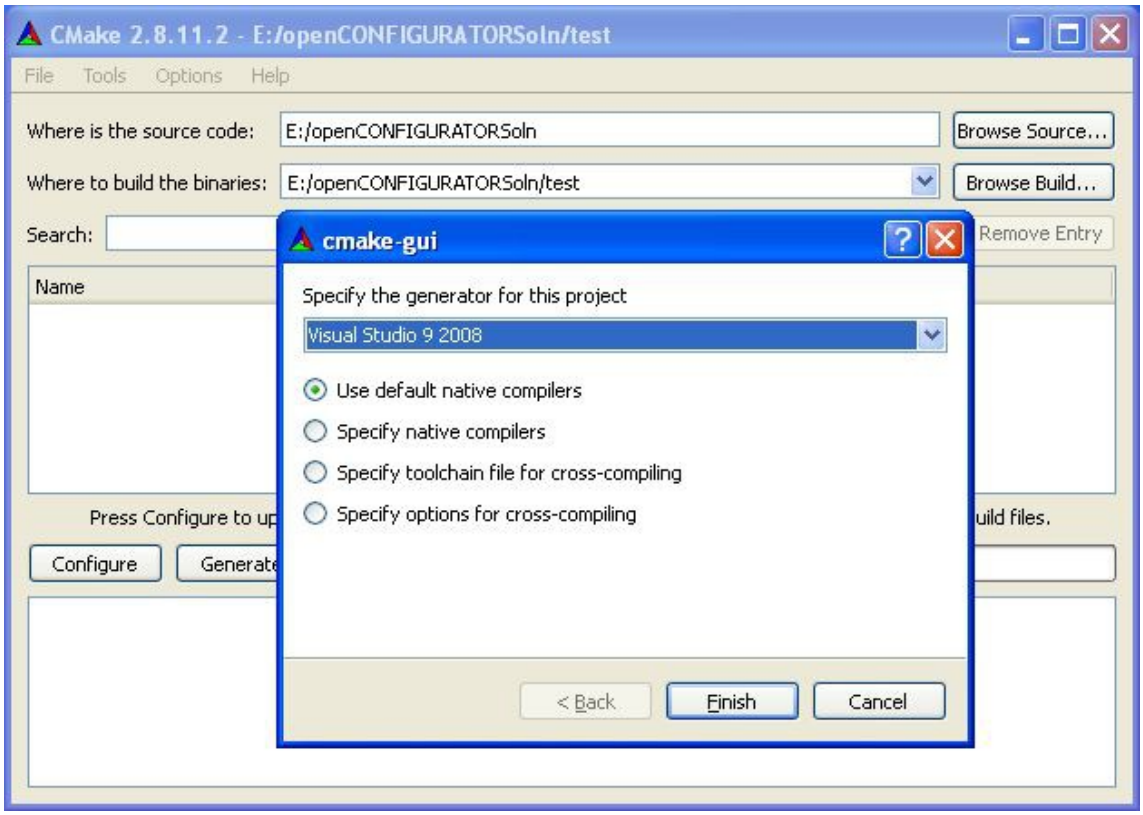

Figure 31: Windows DLL compilation - Initial

- On successful selection of the compiler, the text 'Configuring done will be visible in the console window'.
- Now press **'Generate'** to generate the Microsoft Visual Studio solution file.

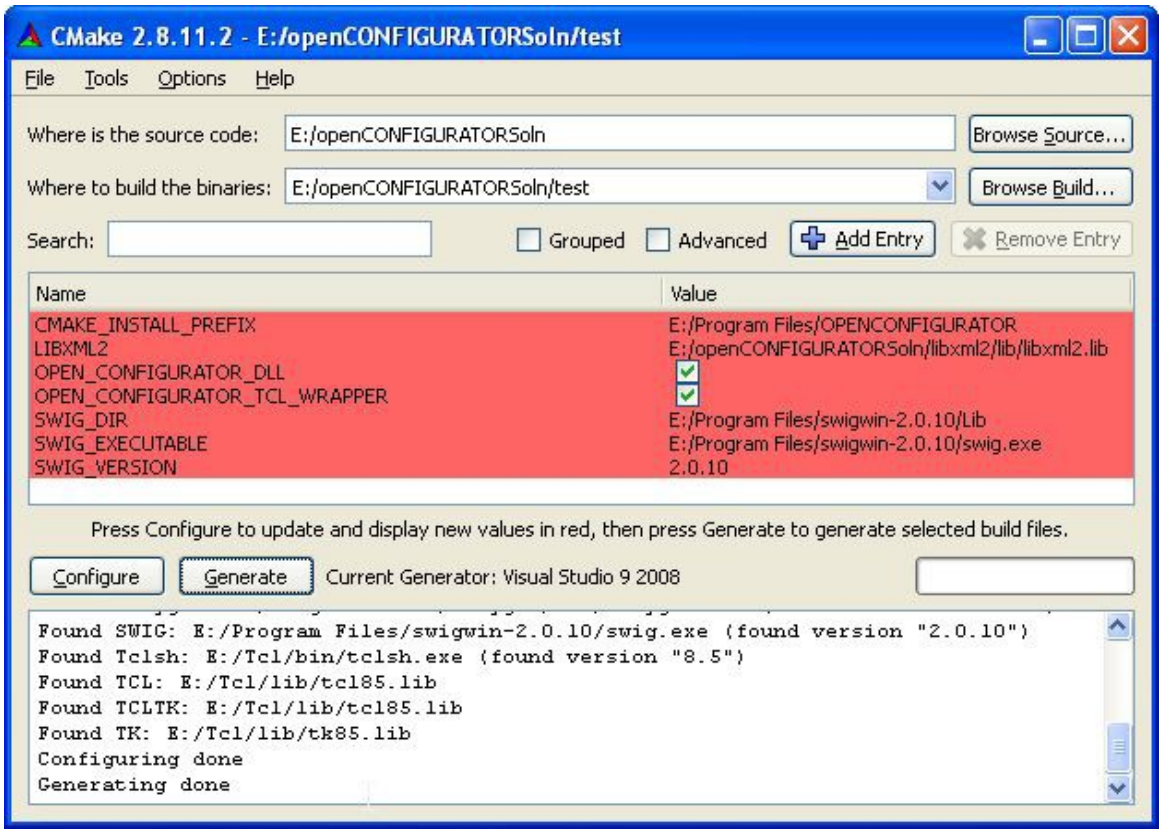

Figure 32: Windows DLL compilation – Generating done

- The solution files will be created in the chosen binary path.
- Now open the solution file named 'openCONFIGURATOR' with the chosen compile environment (e.g. MS Visual C++).
- Choose the mode of build (Debug/Release). Then build the **'ALL_BUILD'** project to compile the binaries.

- After successful completion, build the **'INSTALL'** project to generate the binaries.

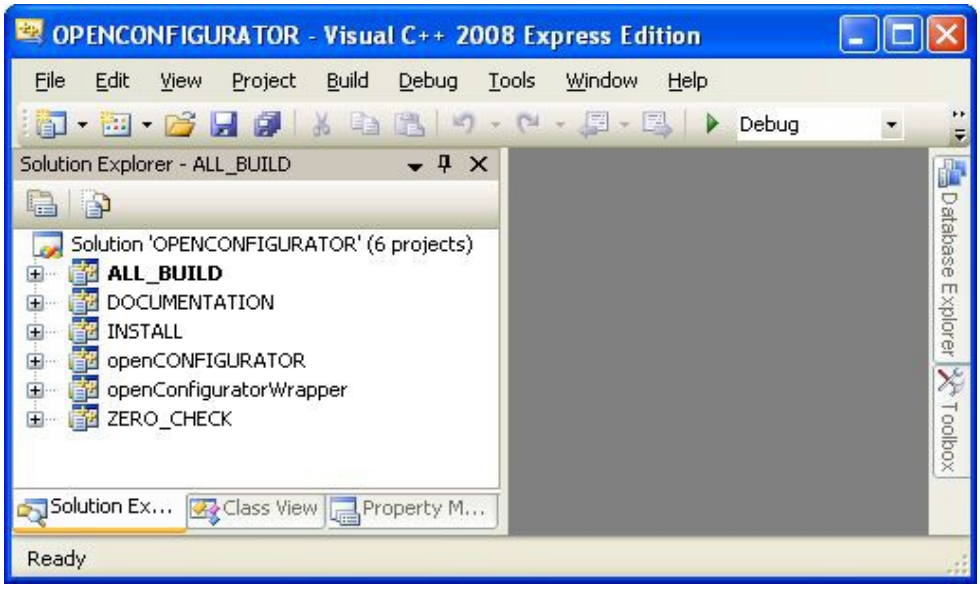

Figure 33: Windows DLL compilation – Visual Studio Project

- Now, the binaries will be built inside the chosen binary path.
- Copy all the files present in the bin directory into the openCONFIGURATOR installed path (default install path: <Program Files (x86)>/openCONFIGURATOR-V1.3.0/)

  Note: Close openCONFIGURATOR application before copying.

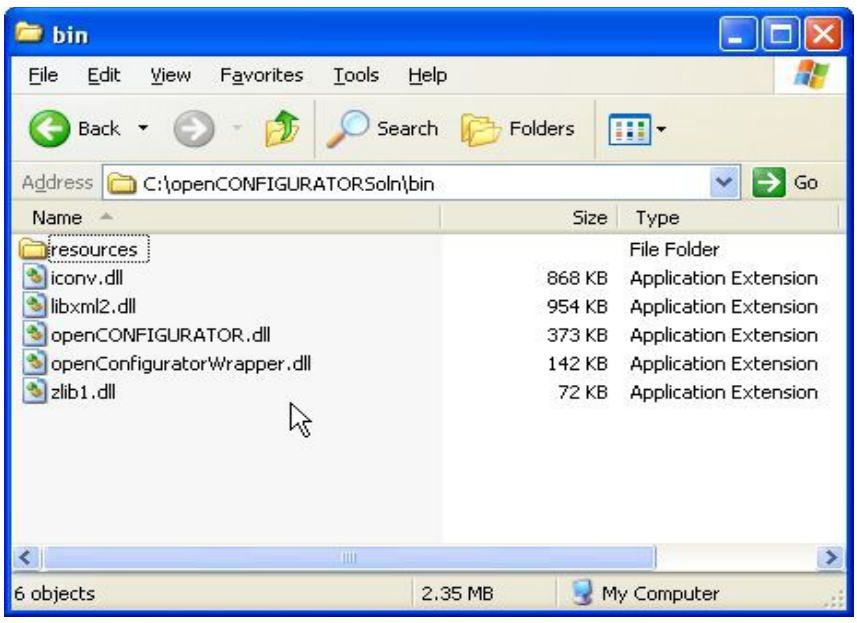

Figure 34: Windows DLL compilation – binaries

7.2. Linux

7.2.1. Pre-requisites

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:

- libxml2 version 2.7.2 or greater
- libxml2-dev
- tcl8.5
- tk8.5
- tcl8.5-dev
- tk8.5-dev
- swig or above (swig2.0 is preferred)
- g++ (v4.7.3 has been tested, other versions may not work)
- doxygen (optional)
- cmake-gui version 2.8.8 or above

  If a previous version of CMake is installed, please update it to at least v2.8.8 in order to compile openCONFIGURATOR. The steps given below can be used to upgrade to v2.8.8.

  1. Uninstall the previous version of CMake if it is installed.

     sudo apt-get remove cmake

  2. Download the CMake package version **'2.8.11 or higher'** from http://www.cmake.org
  3. Extract it to any folder

     Eg: /home/User/cmake-2.8.11-Linux-i386

  4. Add the line given below to your **.bashrc** which is hidden inside your user '*</home/username/>*' directory.

     Eg: export PATH=/home/User/cmake-2.8.11-Linux-i386/bin:$PATH

  5. Now close the terminal window for completing the change to PATH variable.

Note: If you have a 64 bit OS installed, you also need to install the package **ia-32libs** for CMake**.** Use 'sudo apt-get install ia-32libs' to install.

7.2.2. Shared library compilation

- Download the openCONFIGURATOR source package.
  - The package is available at http://sourceforge.net/projects/openconf/
  - Unzip **'<openCONFIGURATOR_1.3.0_Src.tar.gz>'**.
  - Open the terminal and change path to **'<openCONFIGURATORSoln\>'** directory after it is extracted.
- From the terminal open cmake-gui.
  - Enter '***sudo cmake-gui***'
- Set the source and binary paths.
- Click configure and in the window that appears, select the options **'Unix Makefiles'** and **'Use default native compilers'**.

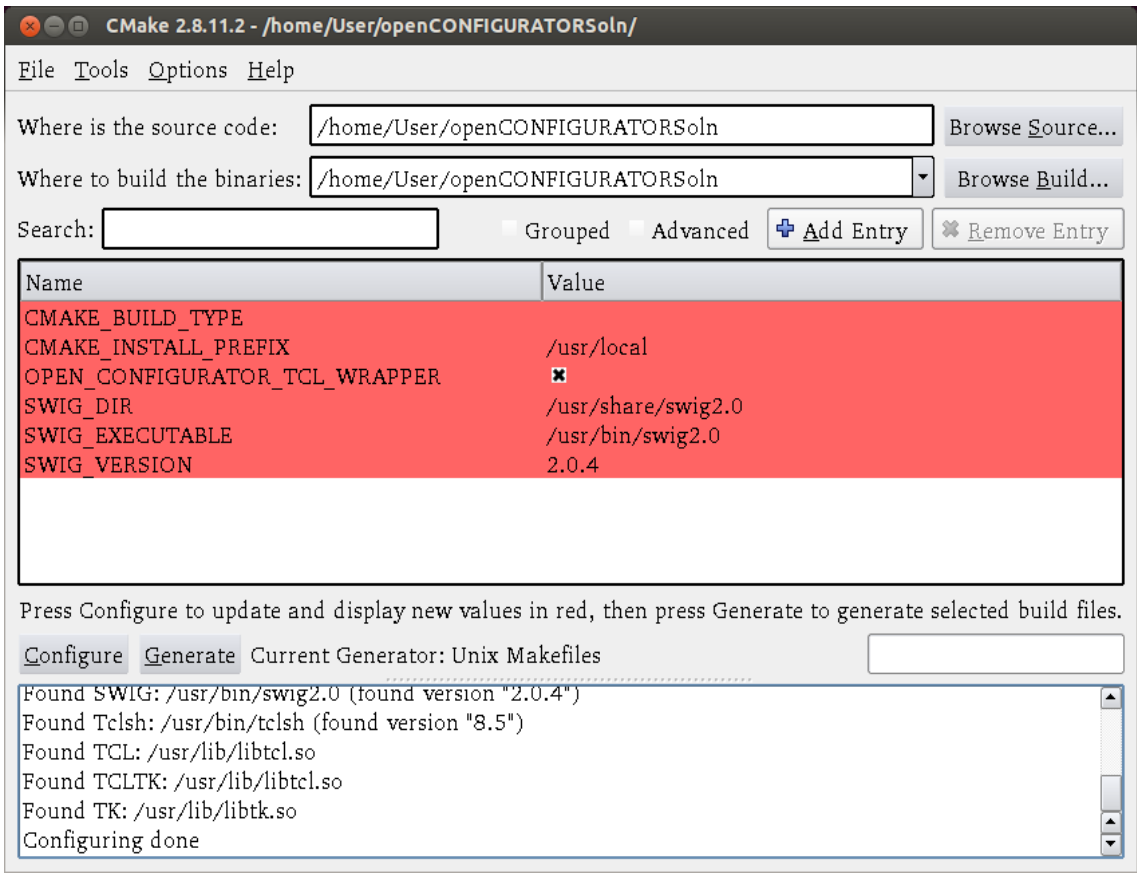

Figure 35: Linux shared library compilation – Configuring done

- Click 'Generate' to generate the Makefile.

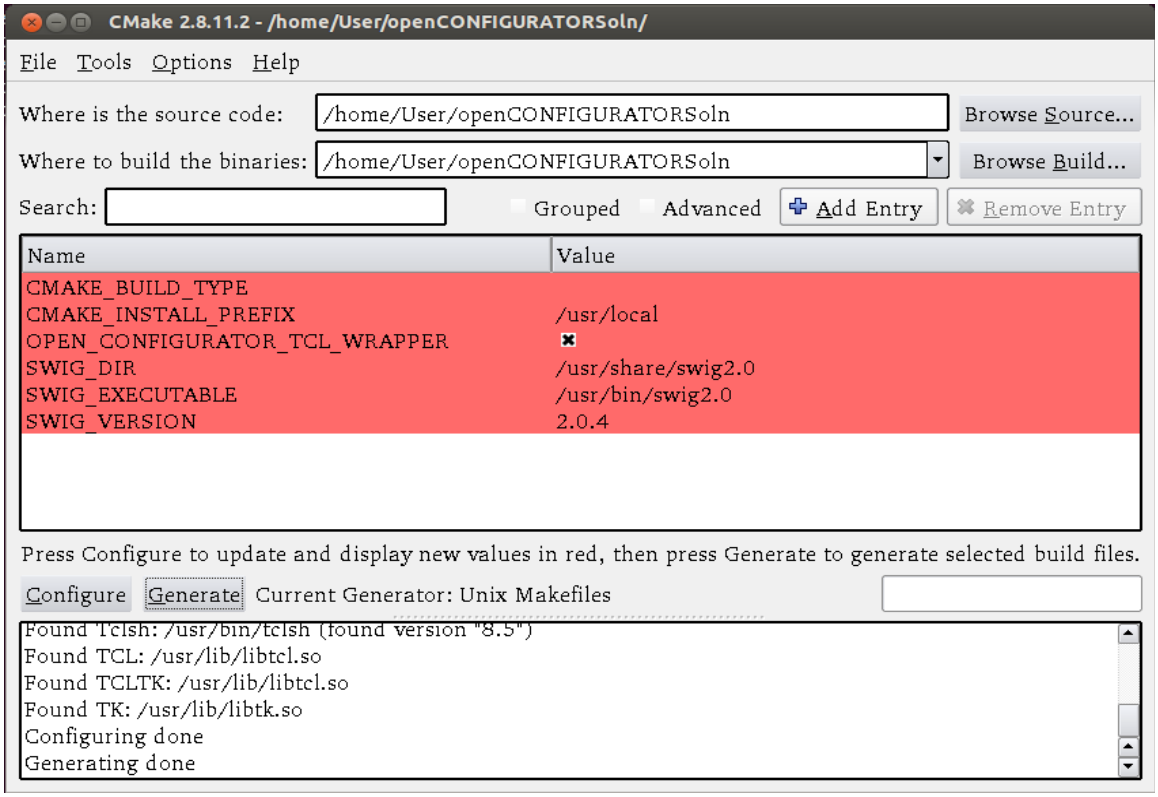

Figure 36: Linux shared library compilation – Generating done

- Open a terminal window and change path to the chosen binary path.
- To compile the libraries
  - make all
- To generate the libraries
  - make install
- The above commands will create shared libraries and other resources into the folder named 'bin' under the chosen binary path.
- Copy all the contents of the 'bin' folder into the openCONFIGURATOR installation folder.
- Copy 'openCONFIGURATOR.so' from 'bin' folder to 'usr/lib'. Replace this file if already present in 'usr/lib'.
- The compilation process has been completed.

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/cdc_xap*
- Convert the edited text file to CDC using the command given below
  - */usr/share/openCONFIGURATOR-1.3.0/txt2cdc mnobd.txt mnobd.cdc*

Note: The txt2cdc executable can be found in the openCONFIGURATOR installation directory.

9. FAQs

Q: The PDO Mapping is done but the mapping values are not found in mnobd.txt after the build process. Why is this so?

A: This is because of any one of the following reasons(Consider the PDO tree node):

- The “include in CDC” checkbox is not selected for the required index and sub-index properties.
- The XDD/XDC has the default value configured and not the actual value (Only the actual value which differs from the default value will be added to the CDC).
- Check for the value configured in the “NumberOfEntries (0x00)” SubIndex. This determines the number of sub-indexes to be taken into account for the PDO generation.

10. 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.ethernet-powerlink.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/

11. Support

11.1. Sourceforge forum

If you need help on using openCONFIGURATOR, please post on help forum at http://sourceforge.net/p/openconf/discussion/help/

11.2. Release note

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