User Manual

For
openCONFIGURATOR-V-1.0.0
[POWERLINK Configuration Tool]

Prepared By
Kalycito Infotech Pvt Ltd.,
Coimbatore

Identifier	UserManual_openCONFIGURATOR	**Version**	1.00
Prepared By	Kalycito Powerlink Team	**Date**	16-Apr-2009
Approved By		**Confidentiality**	Public domain document

Revision History

Version	Date	Modified By	Remarks
0.01	31-Mar-2009	Kalycito Powerlink Team	Initial Draft
0.02	09-Apr-2009	Kalycito Powerlink Team	Incorporates information for Vista
1.00	16-Apr-2009	Kalycito Powerlink Team	Incorporates Known bugs and their work around and compilation - how to

License

Registered Office in India:

Kalycito Infotech Private Limited.

Address: 331, Cross-cut Road, Gandhipuram,
Coimbatore - 641012, India.

Email: info@kalycito.com
Sales Contact: sales@kalycito.com
Phone: 00 91 422 4518454
Fax: 00 91 422 4518445

Research & Development Office:

Kalycito Infotech Private Limited.

Address: E-LAB, Science & Technology Entrepreneurial Park I, PSG College of Technology,
Coimbatore - 641004, India.

European Office:

Kalycito Europe.

Address : 239 bis avenue du bois de Verrieres
92160 ANTONY, FRANCE.
General Information : info.europe@kalycito.com
Sales Contact : sales.europe@kalycito.com
Phone : +33 9 70 44 52 54

Table of Contents

1 Introduction

1.1 Purpose

This document is intended for the users of openCONFIGURATOR-V-1.0.0 tool. This document will not describe the compilation of APIs.

1.2 Document Conventions

Text in Type Writer font is commands that can be executed on the shell.

1.3 Intended Audience and Reading Suggestions

This document is intended for the users of openCONFIGURATOR tool. User is assumed to possess basic knowledge on openPOWERLINK.

1.4 Document Scope

This document limits its scope with explaining the openCONFIGURATOR tool and the sample project(s) provided along with it. Refer the respective documents of openPOWERLINK CFM.
[http://www.kalycito.com/downloads.html#powerlinkxp]

1.5 References

1.openCONFIGURATOR High Level Design document.

2 Overall Description

2.1 Design Goal

Powerlink is an open, Ethernet based industrial field bus technology that guarantees deterministic data transfer even for high bandwidth applications. Apart from its real-time data handling capabilities Powerlink can also be best described by the statement “Powerlink = CANOpen over Ethernet” as the configuration layers of Powerlink are designed to benefit from the Standard device profiles specified in CANopen EN 50325-4 for automation.

Though Powerlink is extremely open and it brings together the advantages of both Ethernet and CANOpen, it demands a certain degree of expertise in Ethernet and CANOpen for the users of this technology, for e.g. they may have to understand the Powerlink Object Dictionary and may have to understand the index, sub-indexes and PDO mappings before they can see meaningful data exchange between their nodes. openCONFIGURATOR is designed to eliminate this need and enable an uninitiated user of Ethernet and CANOpen to configure and run a Powerlink network in a very short time. At the same time, for an experienced user, the tool offers complete control over the configuration in each stage of the build process. openCONFIGURATOR is also designed to run on both Linux and Windows.

3 Features Summary

3.1 For the Uninitiated

- Building a Powerlink network is now as simple as
  - Importing the XML Device Description / Configuration file (XDD / XDC file) provided by the vendors of each device (MN and CNs) in the network
  - choosing auto-generate MN PDO mapping option
  - hitting a build button
  - whats more, everything can be done using the GUI provided along with the openCONFIGURATOR
  - in fact the tool works on both Linux and Windows

- At the end of the build process, the following files are automatically generated by the tool:
  - A Concise Device Configuration file (CDC file) with the entire Object Dictionary Configuration. The format of the CDC file is specified in the Powerlink specification and the Powerlink stack can read this information and configure itself.
  - an XML file and a C Header containing the Application Variable names and offsets (XAP.xml file and XAP.h file) for easy reference by the application. The application designer may choose how they want to use these files. As part of openCONFIGURATOR delivery, we deliver an example application.

3.2 For the Expert Users

- A basic Powerlink network configuration can be built by importing the XML Device Description / Configuration file (XDD / XDC file) provided by the vendors of each device in the network.

- The Powerlink Managing Nodes's PDO mappings can be generated in multiple ways:
  - Using the Auto generation feature of openCONFIGURATOR, PDO Mapping objects and subobjects of MN will be automatically computed from the CN's PDO mapping information (manually edited on the CN side, or imported from the XDD/XDC file).
  - Manually adding the PDO mapping information of the MN to have maximum control over the configuration.
  - Initially Auto generate and then manually edit to customize

- Building the configuration after manually editing or an import generates:
  - A Concise Device Configuration File (CDC file) with the actual values for MN's Object Dictrionary and CNs' Object Dictionary entires. This file is available in two formats one is a binary file file and second one is a text file(in a readable format) so that user can view all the information which is present in binary file.
  - An XML file (XAP.XML) and C Header file (XAP.h) specifying the names of the network variables, their offsets, datasize, datatype etc. These files can be directly imported for use by the application designers to build their applications.

- For users interested in having cross traffic between CNs', it can be easily achieved by manually editing the PDO mapping of each CN and then building the project to generate the CDC files.

- The users can add any object, subobject that is specified in the Powerlink specification or that is the manufacturer specific area. Except manufacturer specific objects, when a user adds an object/subobject, it will automatically fills the default parameters like Name, Datatype, ObjectType, AccessType according to the Powerlink specification.

- There is also an option on the GUI to add or leave out a particular index when a cdc file is generated. This can be done just by check/uncheck of the checkbox provided.

- openCONFIGURATOR is a package of C++ library with a GUI front end develop using Tcl/Tk. This means:
  - Any user who wish to develop their own GUI can access the API interface provided by the C++ dlls.
  - The tool and the GUI runs on both Windows XP/Vista and Linux operating systems

- openCONFIGURATOR source code is available in sourceforge under BSD license, which means, it is:
  - available for free
  - can evolve over a long period of time without the issues of orphaning or vendor-lockup
  - can be customized as per your organizations needs in a proprietary manner
  - permits you to retain the option of commercializing final results with minimal legal issues.

4 Product Feature

4.1 Project Wizard

4.1.1 Create New Project

New project creation can be done in two ways

a)When a user launches an openConfigurator tool it will ask either to Create New Project or Open Existing Project as shown in Figure.1

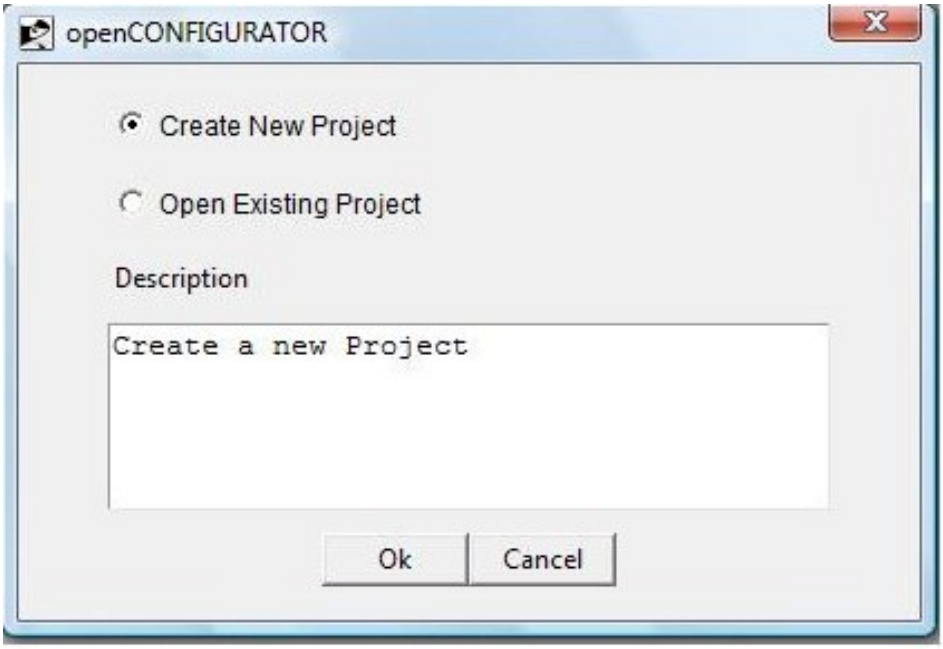


Figure 1

b)From the top menu bar ***File → NewProject***

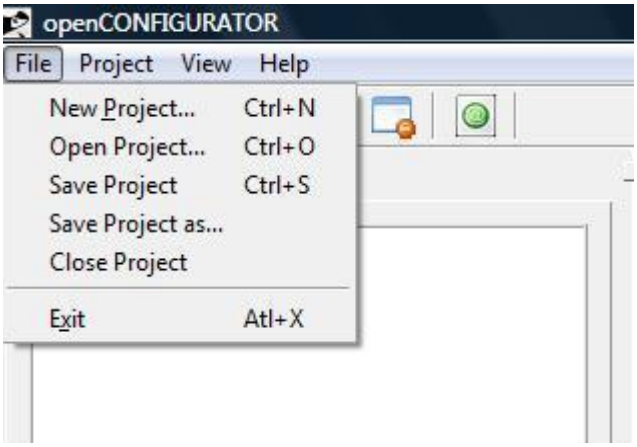


Figure 2

4.1.2 Project Settings

Project Name - User can choose any name of the project. The maximum limit of the Project Name is 32 characters and spaces are not allowed

Project Location - User can select any location by clicking the browse button

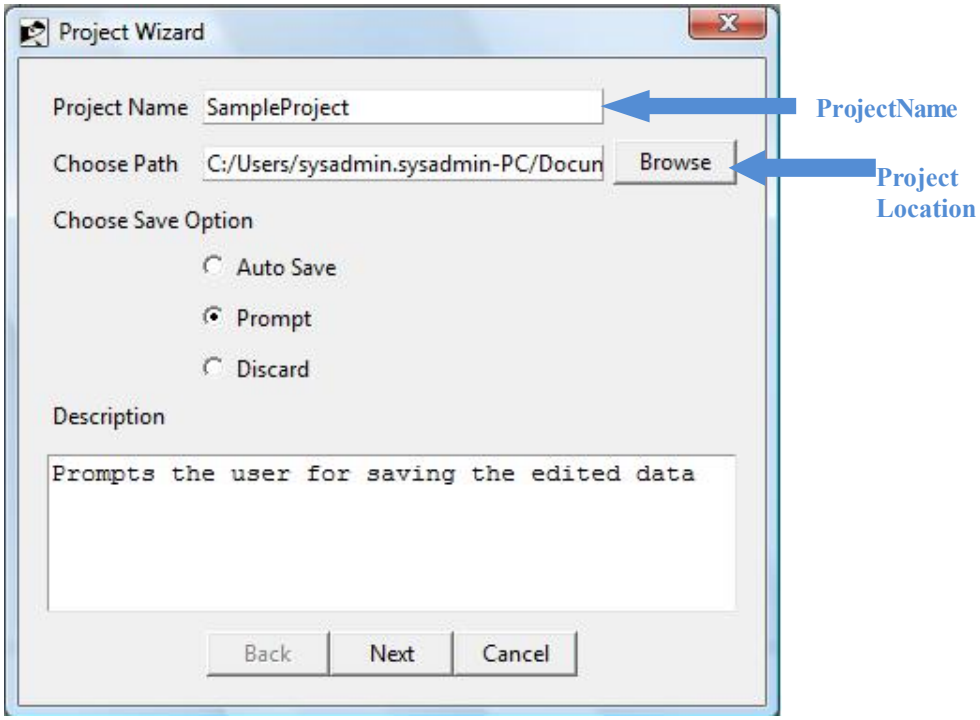


Figure 3

Project Save Options
After User selects an option of create new project from the project wizard, next screen popups which asks for the Project save option as

a)Auto Save:
In this mode tool will automatically save the data whenever user click anywhere in the tree
b)Prompt
Prompt to an user, to ask if user wants to save the data or not before exiting from the screen
c)Discard
Changes will not be saved, if user wants to save the data, he/she has to click the save button

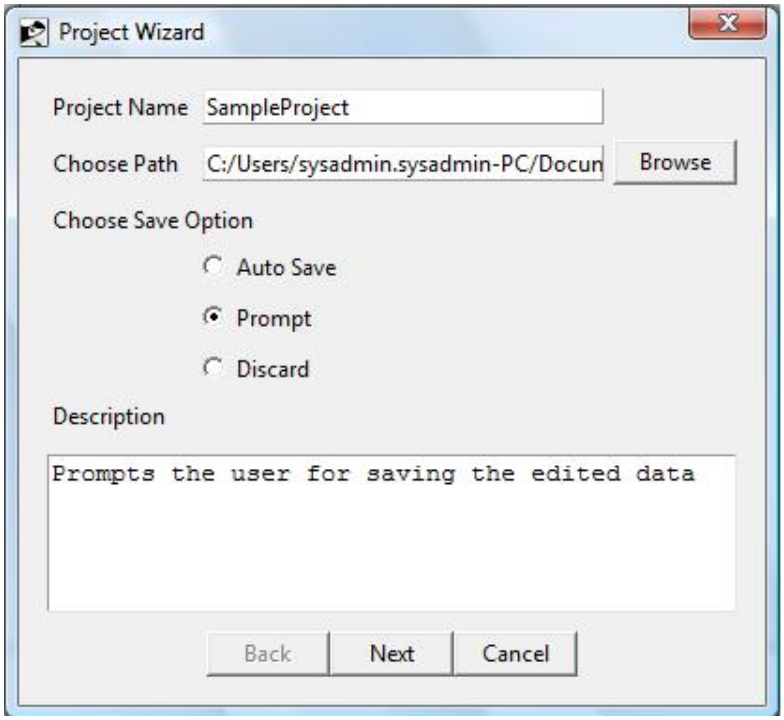


Figure 4

Project Save option can also be any time during the project from the top menu bar ***Project → Project Settings*** as shown in Figure3.

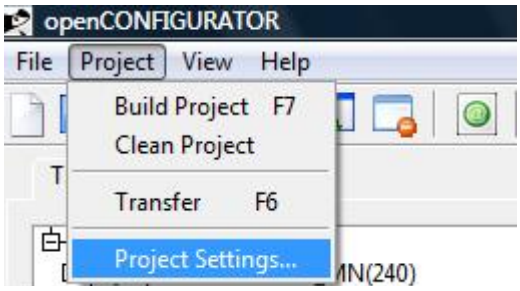


Figure 5

After selecting project settings menu item a pop-up will appear, there also user can change the user setting. **This pop-up will only appear if any project is present in the tree, else pop-up will not appear**

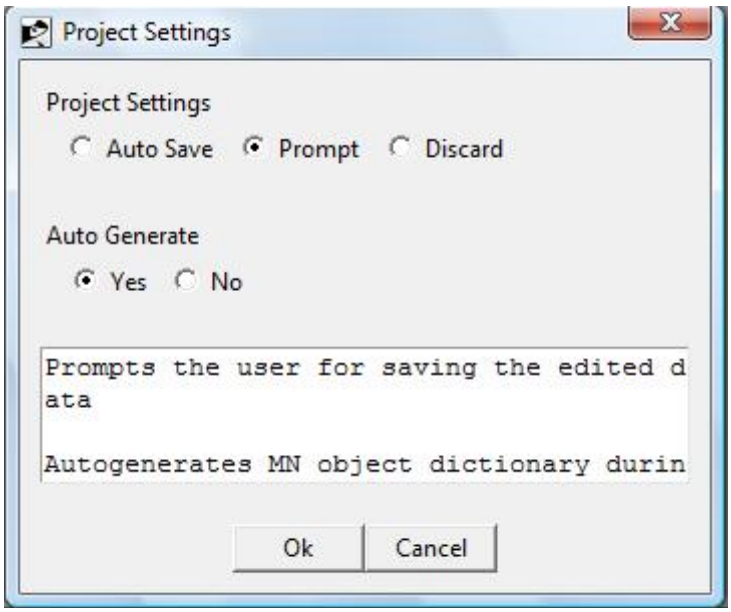


Figure 6

4.1.3 Choose MN Configuration

There are two options by which MN OBD can be created
a) Default - A default XDD for MN will be used in this case. This file will be installed with the installation of the openConfigurator. This file is named as **openPOWERLINK_MN.**

b)Import XDC/XDD - User can import his/her own XDD/XDC file for the MN.

Autogeneration Mode -
If autogenerate is set to 'Yes', Following things will be automatically generated

•PDO mapping of MN

•Object 1F26h (CFM_ExpConfDateList_AU32)
Date will be calculated by the tool according to the specifications and as this object is an array of subobjects where each subindex corresponds to the CN with the Node ID equal to the sub-index is added.
Each Subobject value will be set to the calculated date value.

•Object 1F27h (CFM_ExpConfTimeList_AU32)
Time will be calculated by the tool according to the specifications and as this object is an array of subobjects where each subindex corresponds to the CN with the Node ID equal to the sub-index is added.
Each Subobject value will be set to the calculated time value

•Object 1F84h (NMT_MNDeviceTypeIdList_AU32)
This object is an array of subobjects where each subindex corresponds to the CN with the Node ID equal to the sub-index is added. Each Subobject value is set to **'Default value'**

•Object 1F8Bh(NMT_MNPReqPayloadLimitList_AU16)
This object is an array of subobjects where each subindex corresponds to the CN with the Node ID equal to the sub-index is added. Each Subobject value is set to **'C_DLL_ISOCHR_MAX_PAYL (1490)'**

•Object 1F8Dh (NMT_PResPayloadLimitList_AU16)
This object is an array of subobjects where each subindex corresponds to the CN with the Node ID equal to the sub-index is added. Each Subobject value is set to **'C_DLL_ISOCHR_MAX_PAYL (1490)'**

•Object 1F92h (NMT_MNCNPResTimeout_AU32)
This object is an array of subobjects where each subindex corresponds to the CN with the Node ID equal to the sub-index is added. Each Subobject value is set to **'2000000'**

User always need not to set the project settings for autogeneration, he can also autogenerate by just right clicking MN Node as shown in the Figure below

Figure 7

If autogenerate mode is set to 'NO' , PDO mapping of MN will not be generated automatically.

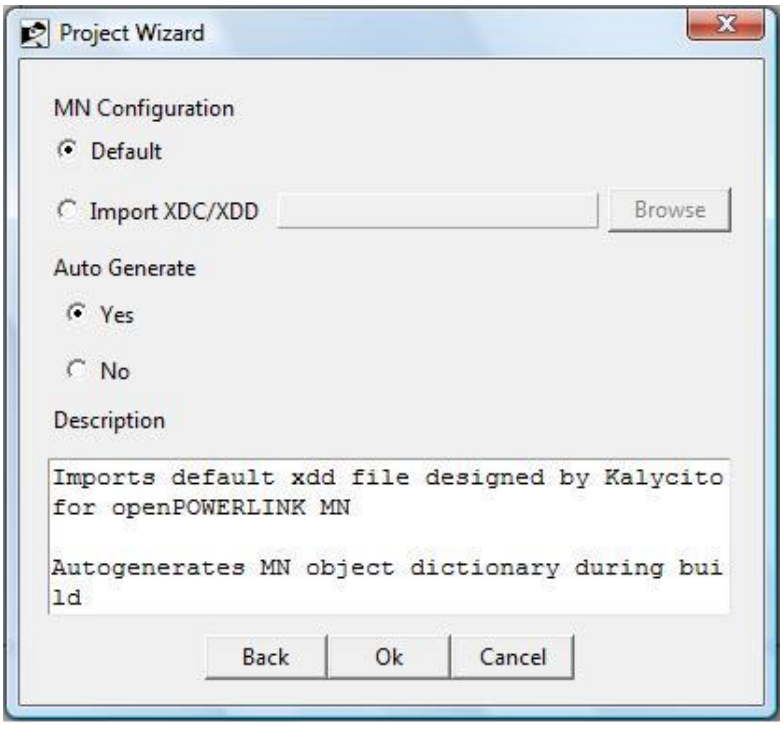


Figure 8

This feature is also available on the right click of MN Node after it is created as shown in Figure 6.

After User Presses 'OK' in Figure8 MN project will be created with MN Node as shown in Figure 9.

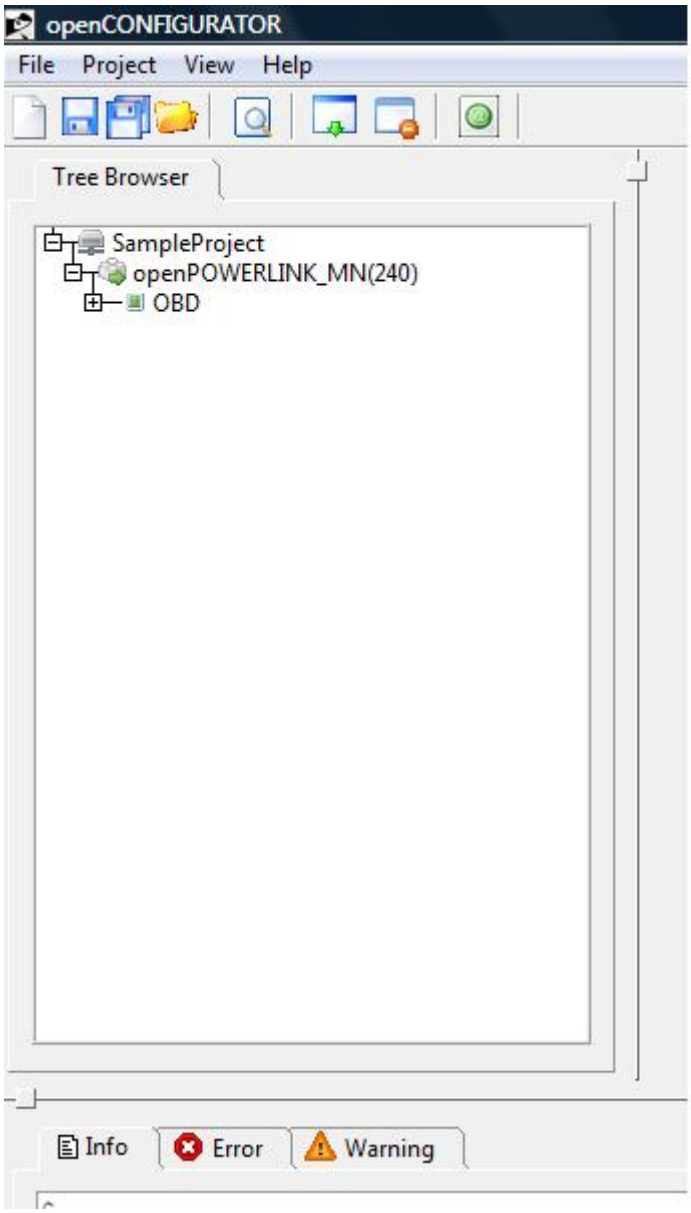

openCONFIGURATOR
File
Project
View
Help
Tree Browser
SampleProject
openPOWERLINK_MN(240)
OBD
Info
Error
Warning

Figure 9

4.2 Adding a CN Node

CN Node can be added by right click on the MN Node a small menu will appear which will have the first menu item ‘Add CN’ as shown in Figure 10

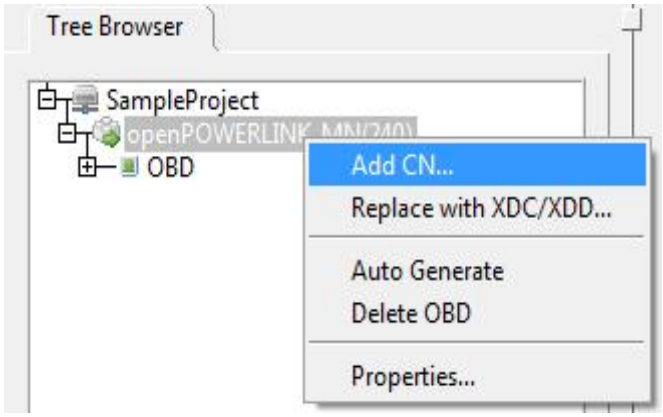


Figure 10

After clicking on ‘Add CN’ a popup will appear where user can enter CN related information and can select the xdc/xdd files for that CN as shown in Fugure10.

4.2.1 Properties of CN

CN Name - Name of the CN, as Node ID name will be visible in the tree. The maximum length of CN Name is 32 characters.
CN ID (Node ID) - NodeID of the CN. NodeID can only be between 1 to 239

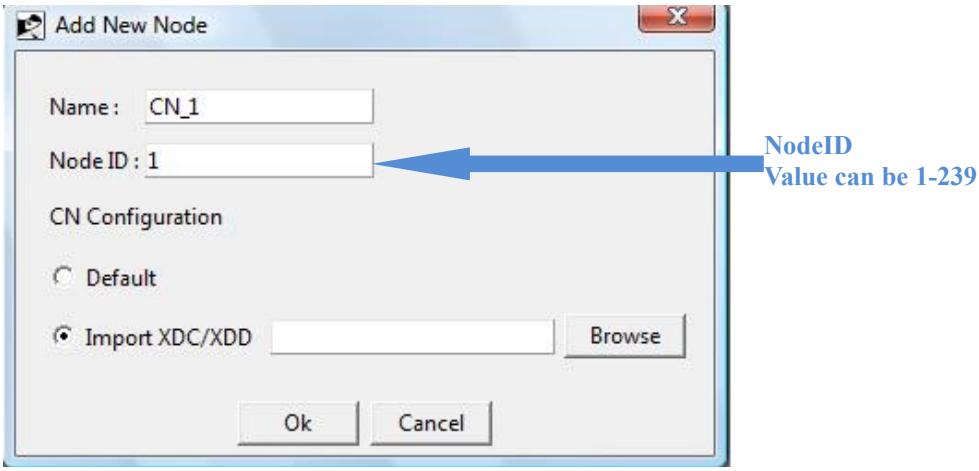


Figure 11

4.2.2 CN Configuration

AS MN, CN can also be added with two options as shown in Figure 12:

a)Default - A default XDD for CN will be used in this case. This file will be installed with the installation of the openConfigurator. This file is named as **openPOWERLINK_CN.**

b)Import XDC/XDD - User can import his/her own XDD/XDC file for the CN.

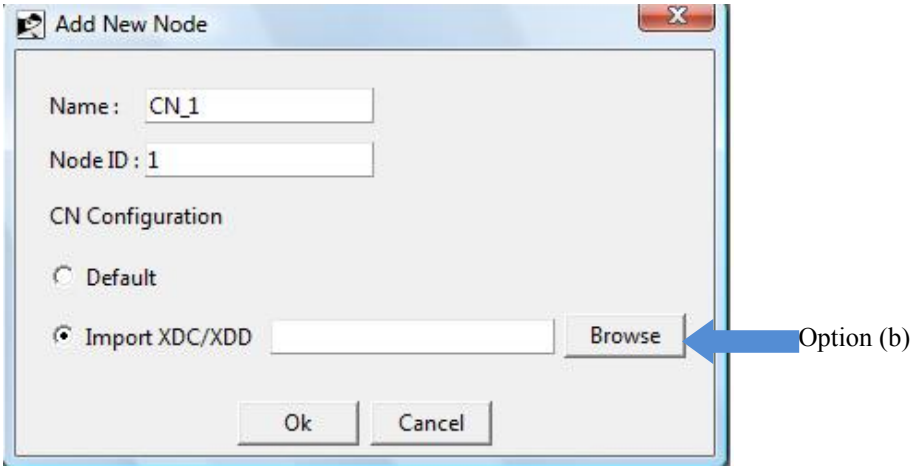


Figure 12

4.2.3 Import XDC/XDD

User selects the xdd/xdc file and then presses ‘OK’

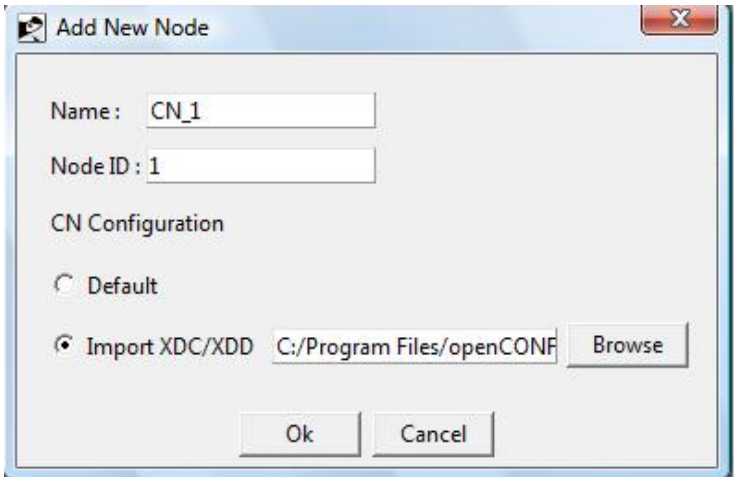


Figure 13

CN Node will be added to the tree as shown in Figure 14

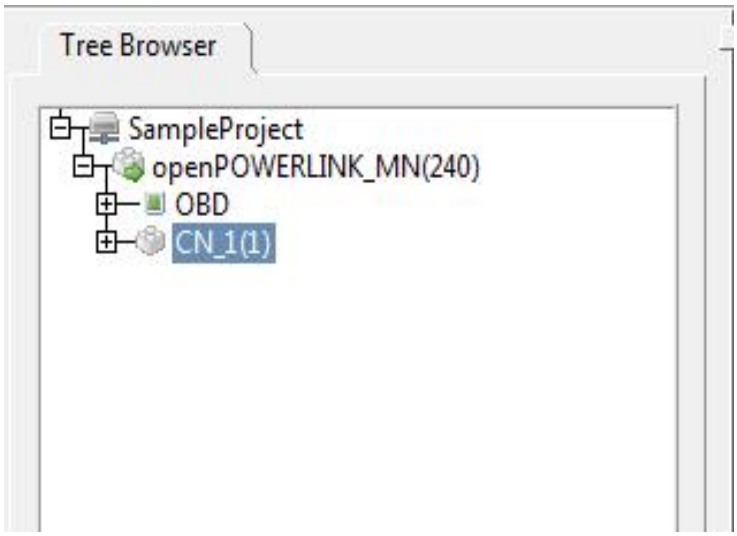


Figure 14

Expanded view of CN Node can be seen by clicking the '+' sign next to the CN node. The view of the tree will be visible as in the Figure15

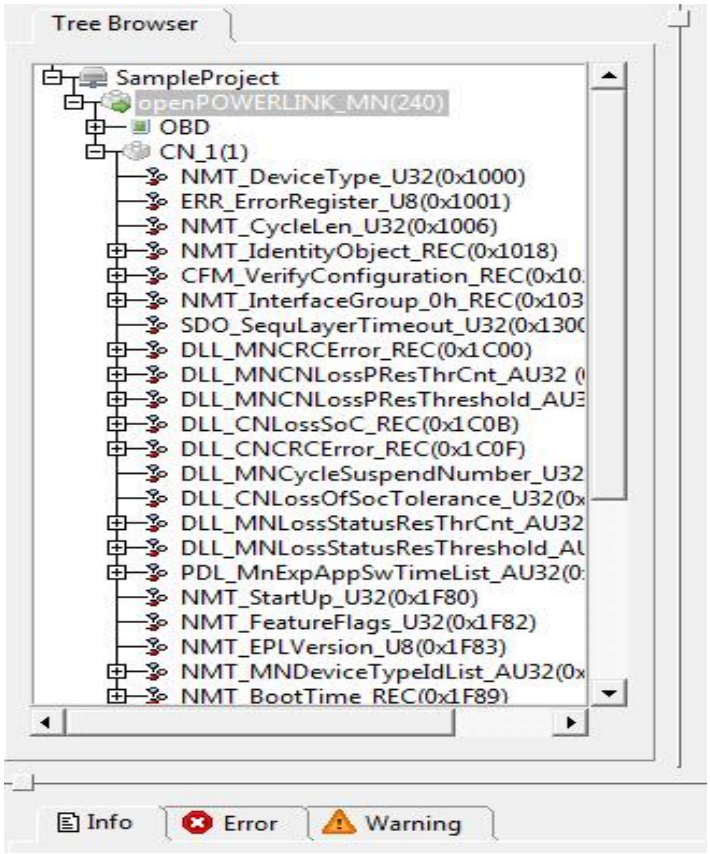


Figure 15

4.3 Adding an Index

Index can be added by right clicking the node in which Index has to be added as shown in Figure16
Node can either be MN or CN.

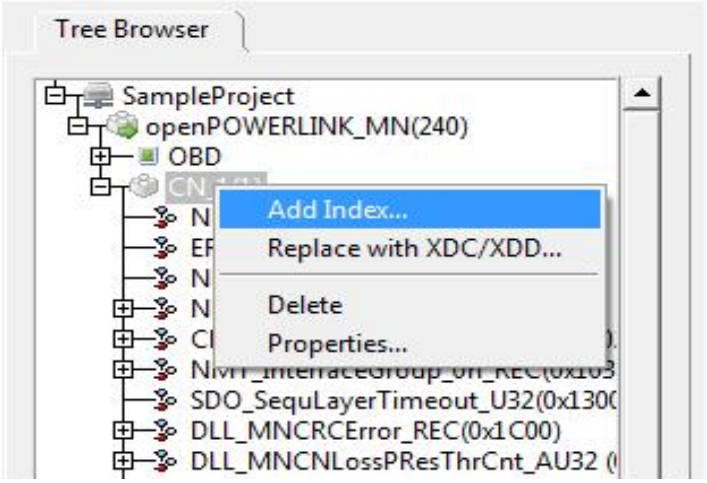


Figure 16

After clicking “Add Index” a popup with a textbox will appear as shown in Figure 17. Enter the Index and press ‘OK’ index will be added in the Node

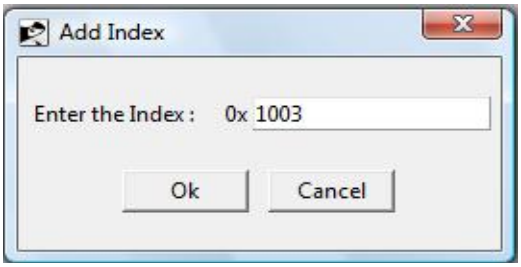


Figure 17

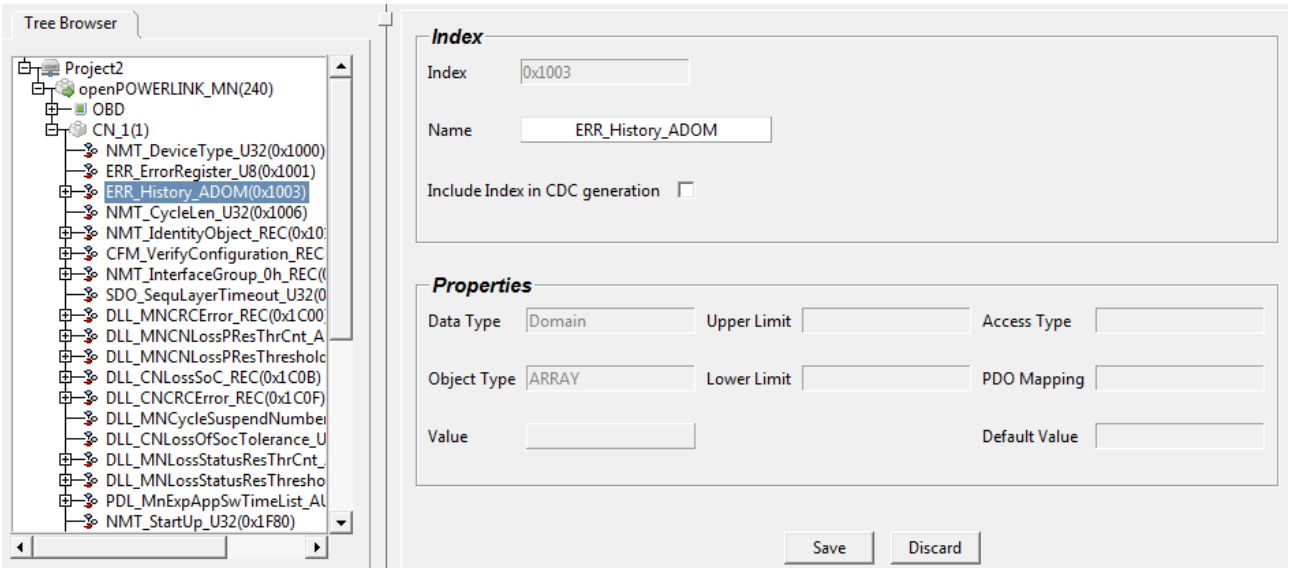


Figure 18

Objects which can added are as

•Communication Profile Area Objects
Communication Profile Area Objects will be added automatically with the Default properties as mentioned in the EPSG specification EPSG DS 301 V1.1.0. Range is **1000-1FFF**

As shown in Figure18 Default properties like Name, Object Type, DataType is automatically added.

•Manufacture Specific Profile Area Objects
User can also add objects which are manufacturer specific lie in the range of **2000 – 5FFF.** As these objects are manufacturer specific no property will be set by the tool, user can edit all the properties of these objects

•Standardized Device Profile Area Objects
User can also add objects which are manufacturer specific lie in the range of **6000 – 9FFF.** No default properties will be set by the tool.

•Standardized Interface Profile Area Objects
User can also add objects which are manufacturer specific lie in the range of **A000 – BFFF.** No default properties will be set by the tool.

•Reserved for further use
User can also add objects which are manufacturer specific lie in the range of **C000 – FFFF** No default properties will be set by the tool.

4.4 Adding an SubIndex

SubIndex can be added by right clicking the object in which subindex has to be added as shown in Figure19.

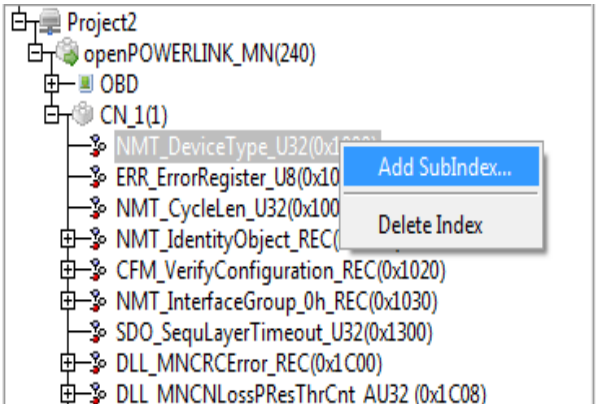


Figure 19

After clicking "Add SubIndex" a popup with a text box will appear as shown in Figure 20. Enter the SubIndex and press 'OK' subindex will be added in the Node

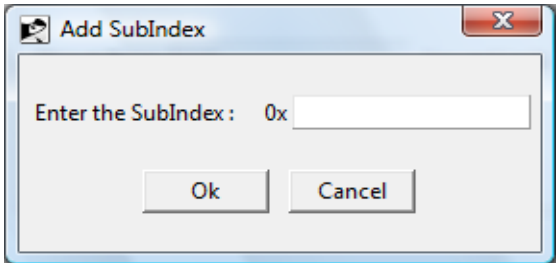


Figure 20

Communication Profile Area Objects
Communication Profile Area SubObjects will be added automatically with the Default properties as mentioned in the EPSG specification EPSG DS 301 V1.1.0.

Default properties like Name, ObjectType, DataType, AccessType, Default Value is automatically added.

For the Other Objects(Manufacturer Specific Profile Area, Standarised Device Profile Area, Standardised Interface Profile Area Objects, Reserved for further use), if subindex is added no default property will be set.

4.5 Editing Object/Subobject Properties

The Objects/ SubObjects with access type value ro/const/empty from XDCs cannot be modified.

•Communication Profile Area Objects

User can edit only **NAME and Value of these objects** as shown in Figure22

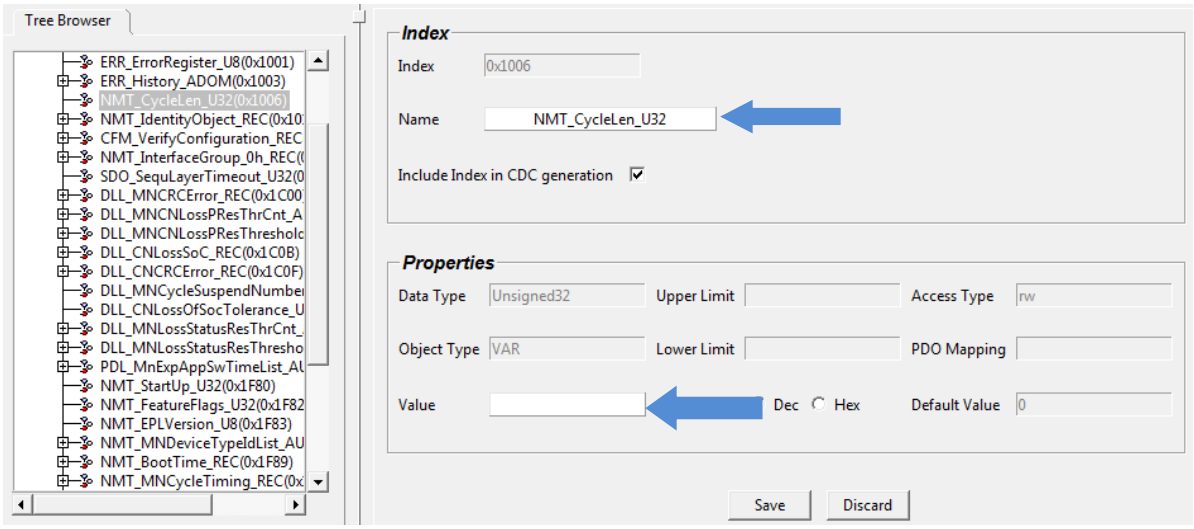


Figure 22

•Manufacture Specific Objects

User can edit **ALL** the properties of these objects as shown in Figure23

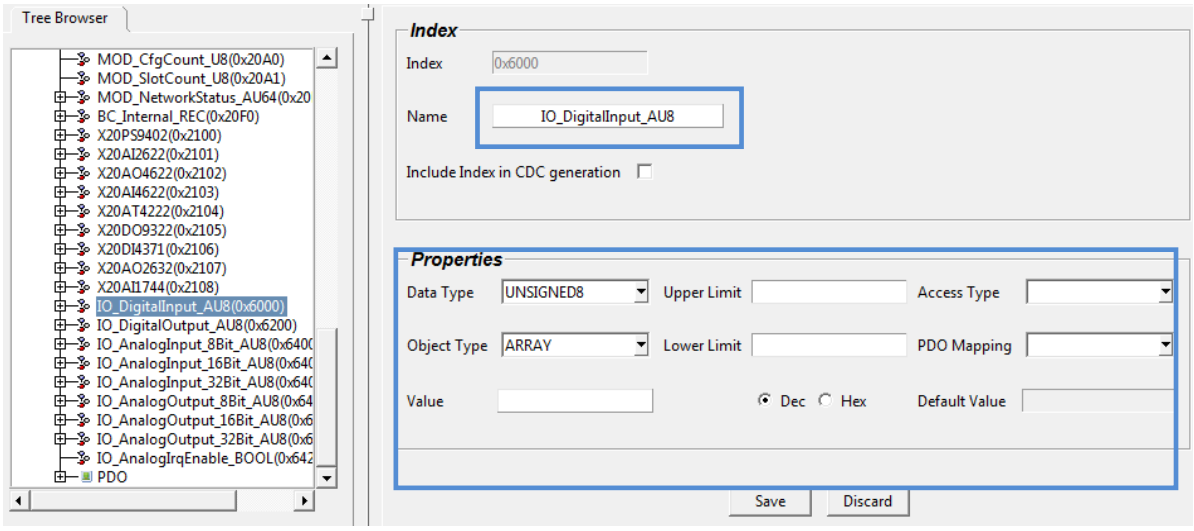


Figure 23

4.6 PDO Screens

PDO objects will be presented in a tabular structure. This table is visible by clicking '+' sign of PDO Node or double clicking the PDO node, and then select RDPO or TPDO as shown in Figure 24.

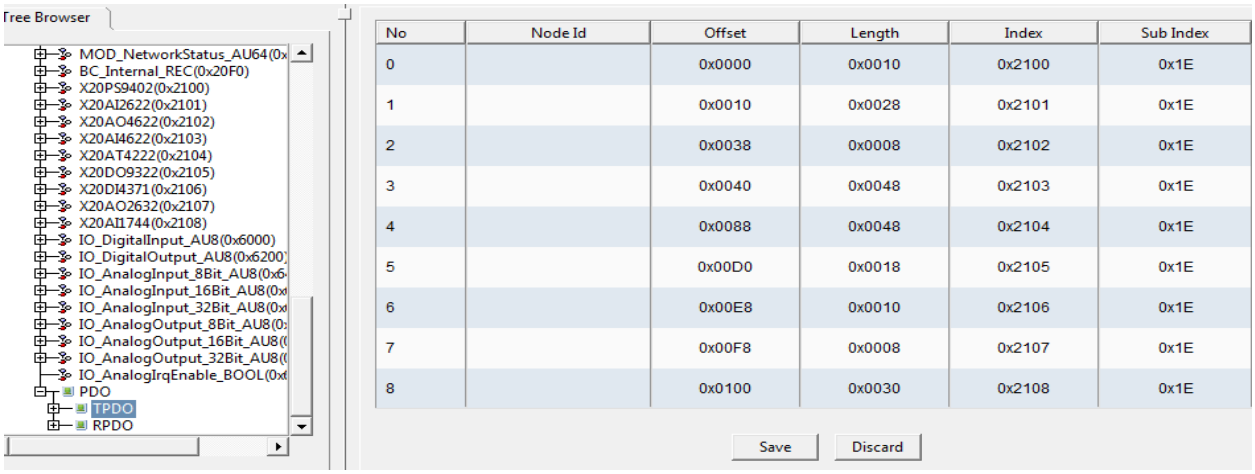


Figure 24

4.6.1 Editing the PDO Objects

a)In the tabular structure - User can change the Maping entry in the table, the same change will be reflected if user selects that particular object

Feature of pdo table

(a)Table will be filled with the data only if pdo have **ACTUAL VALUES**, default values will not be visible in the table.
(b)User can edit the following columns of the table, provided the access type is not 'ro'

- Offset
- Length
- Index
- SubIndex

(a)If user makes any change in the above mentioned column, same change will be automatically reflected in the mapping entry

b)User can select that particular object in the tree and edit it as its for other objects as shown in Figure25

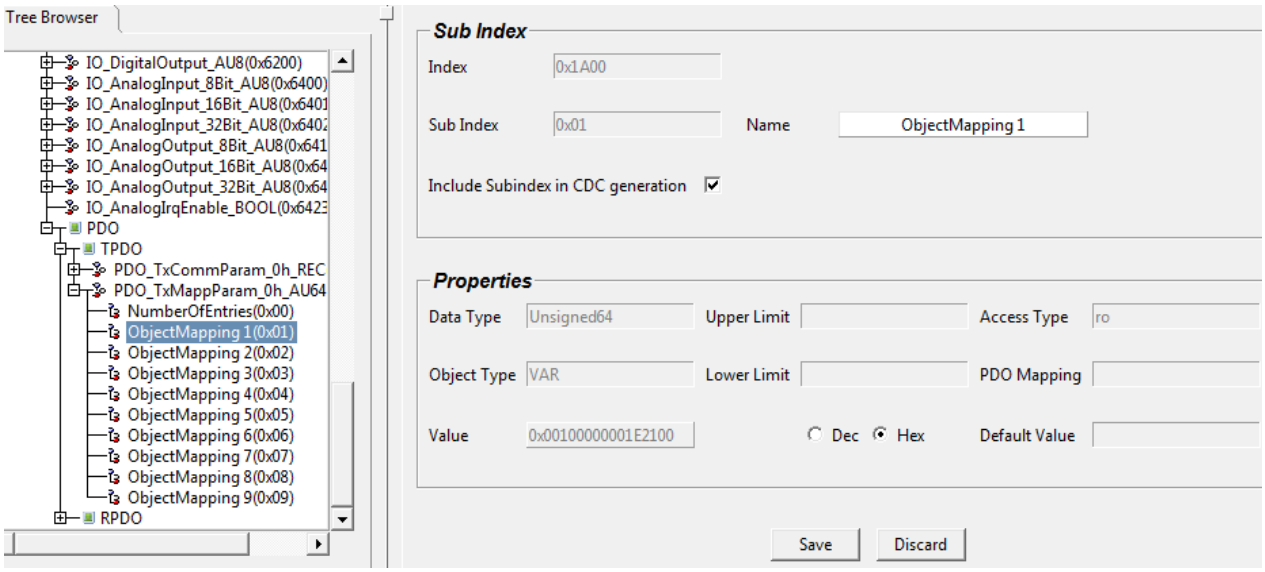


Figure 25

4.7 Include in CDC Generation

User will have the option of selecting which objects should go in cdc and which should not.
This can be achieved by selecting that particular object and in the right side window; a checkbox is provided “Include Index in CDC generation” as shown in Figure26
If the check-box is checked, then that object will be included in the cdc generation.
If the check-box is **NOT** checked, then that object will not be included in the cdc generation.

This is as available for all Objects of both MN and CN

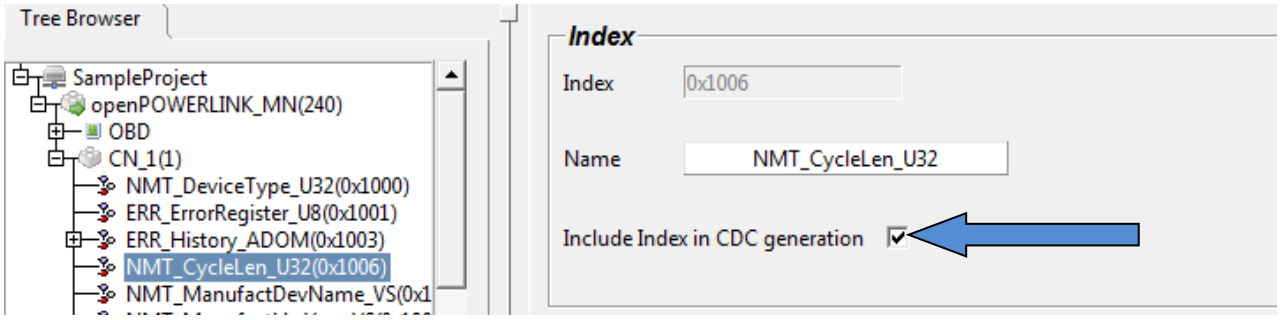


Figure 26

4.8 Build the Project

User can build the project either from the top menu bar as shown in Figure 27

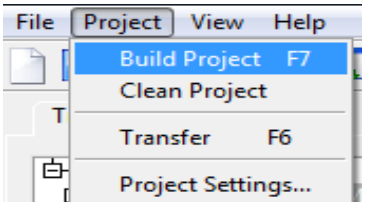


Figure 27

User can also build the project by clicking icon  in the icon bar

If the user has set auto-generate is to 'yes', so a message pop-up appears to tell user if any change is made by user in PDOs, it will be lost as auto-generate option is set to yes, pdo mapping will be calculated automatically

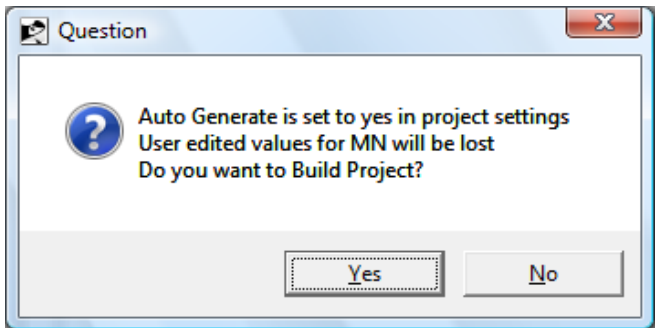


Figure 28

Following files will be created after the build of project. This files will be present in <Project location >/<Project Name>/cdc_xap folder

•mnobd.txt – Text version of the binary cdc file.
•mnobd.cdc - CDC binary file used with the stack
•XAP.h - Header file for the application

•XAP.xml - XML file with variable names, Datatype, datasize, ByteOffsets, BitOffsets

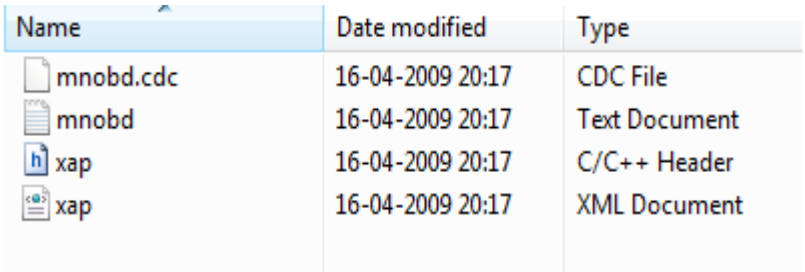


Figure 29

4.9 Transfer CDC & XAP

4.9.1 Copy CDC

After the build, mnobd.cdc generated by the tool can be manually copy to
<openPOWERLINK_CFM_Dir>/openPOWERLINK_CFM_V1.3.0-3/mnobd.cdc

4.9.2 Copy XAP

XAP.h can be manually copied to the following locations:-

For Linux
openPOWERLINK_CFM_V1.3.0-3/Examples/X86/Windows/VC8/demo_pcap/xap.h

For Windows
openPOWERLINK_CFM_V1.3.0-3/Examples/X86/Linux/gnu/demo_mn_8139_kernel/

4.10 Delete the Node

4.10.1 Delete MN OBD

MN OBD can be deleted by right clicking MN Node, a menu will appear as shown in Figure 30

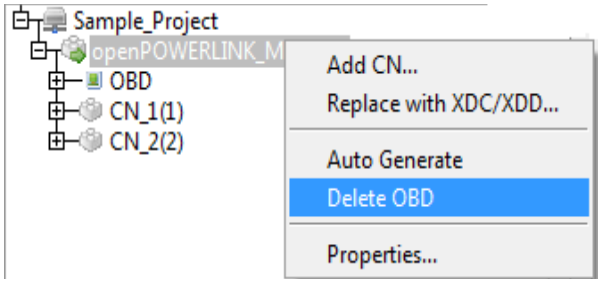
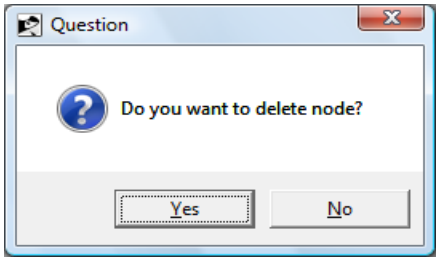


Figure 30

Select ‘Delete OBD’ menu item, , a popup appears to ask whether he/she wants to delete the node



4.10.2 Delete CN Node

CN node can be deleted by right clicking on the node, a menu will appear as shown in Figure31

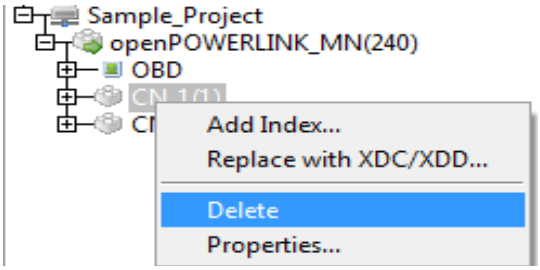
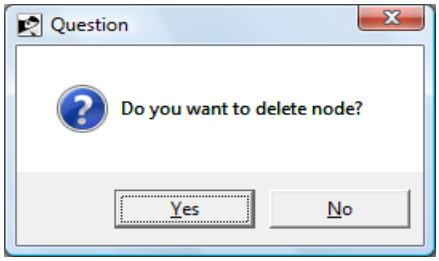


Figure 31

Select ‘Delete’ menu item , a popup appears to ask whether he/she wants to delete the node



If User presses ‘Yes’ , Node will be deleted.
If User presses ‘NO’ , Node will not be deleted.

4.11 Delete Index

Index of a particular node can be deleted by clicking on the node, node tree will be expanded, then user can right click on the Index which has to be deleted, a menu will appear as shown in Figure32

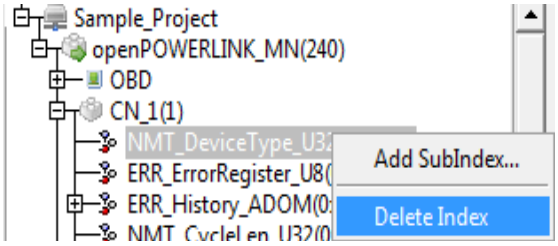


Figure 32

Select ***‘Delete Index’*** Index will be deleted.

4.12 Delete SubIndex

Sub Index of a particular Index and of a node can be deleted by right clicking on the subIndex, a menu will appear as shown in Figure33.

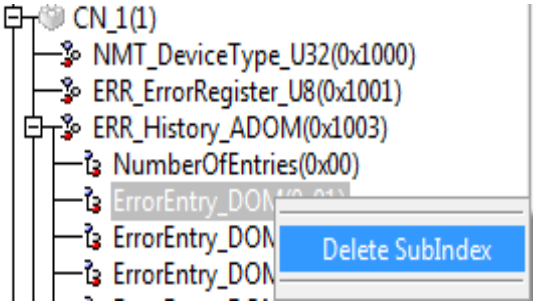


Figure 33

Select ***‘Delete SubIndex’*** subIndex will be deleted.

4.13 Save Project

Project can be saved from the top menu bar by clicking File menu and select Save Project as shown in Figure34

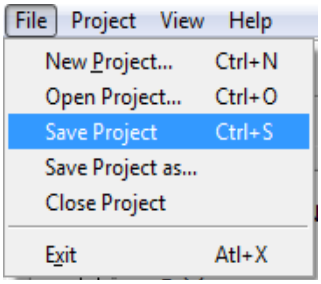


Figure 34

Project can also be saved by clicking  icon in the icon bar

Save project will save the following files in the project directory

/octx folder
a)One octx file for each CN. Name of the file is the Node Id of the CN
b)One octx file for MN. Name of the file is 240(NodeId of MN)

Project folder
a) One OCT file for the project

4.14 Clean Project

Clean project will delete cdc and xap files from the project folder. This can be done either from the top menu ***Project→Clean Project*** as shown inFigure35

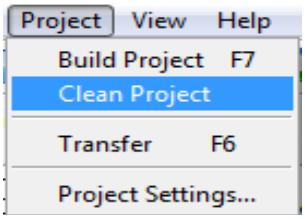


Figure 35

This can also be done by clicking  icon in the icon bar

4.15 Close Project

This will close the project and can be done from the top menu ***File→CloseProject*** as shown inFigure36

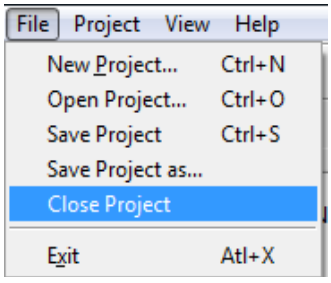
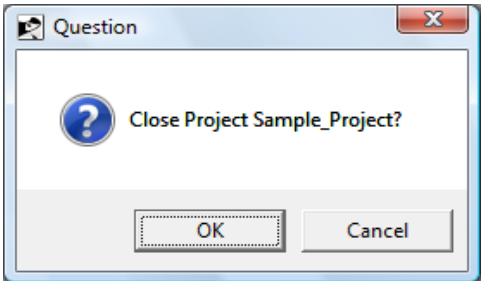


Figure 36

A pop-up will appear which will ask user whether to close the project or not as shown below



5 Operating System

This tool is designed for the following operating environments:
•Linux
•Windows XP
•Windows Vista

Tested operating environments:
•Linux - Ubuntu 8.xx
•Windows XP - Business
•Windows Vista - Business

6 Install & Uninstall

6.1 System Requirements

The following are the requirements for the tool to run. The requirements for POWERLINK and dependencies are not under the scope of this document.

6.1.1 Linux

•Tcl 8.5
•Tk 8.5
•tclthread 2.6.5
•libxml2

6.1.2 Windows(XP and Vista)

•Tcl 8.5
•Tk 8.5
•tclthread 2.6.5 and higher
[You may get the executable for the above from http://www.activestate.com/]
•libxml2

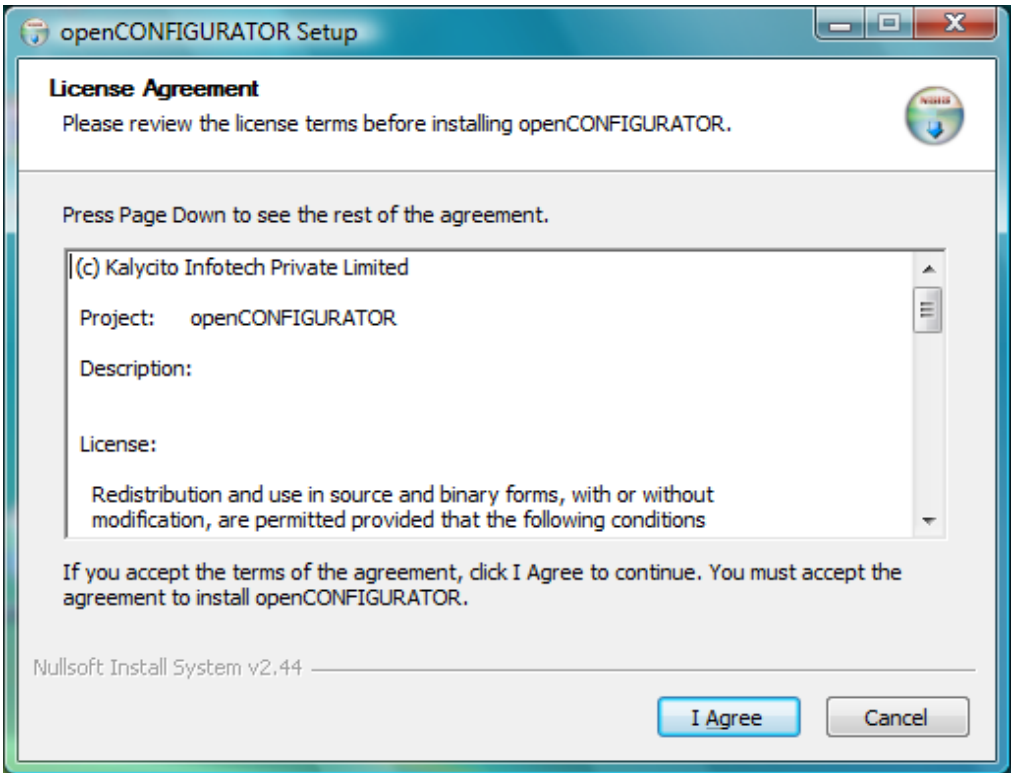
6.2 Install

6.2.1 Linux

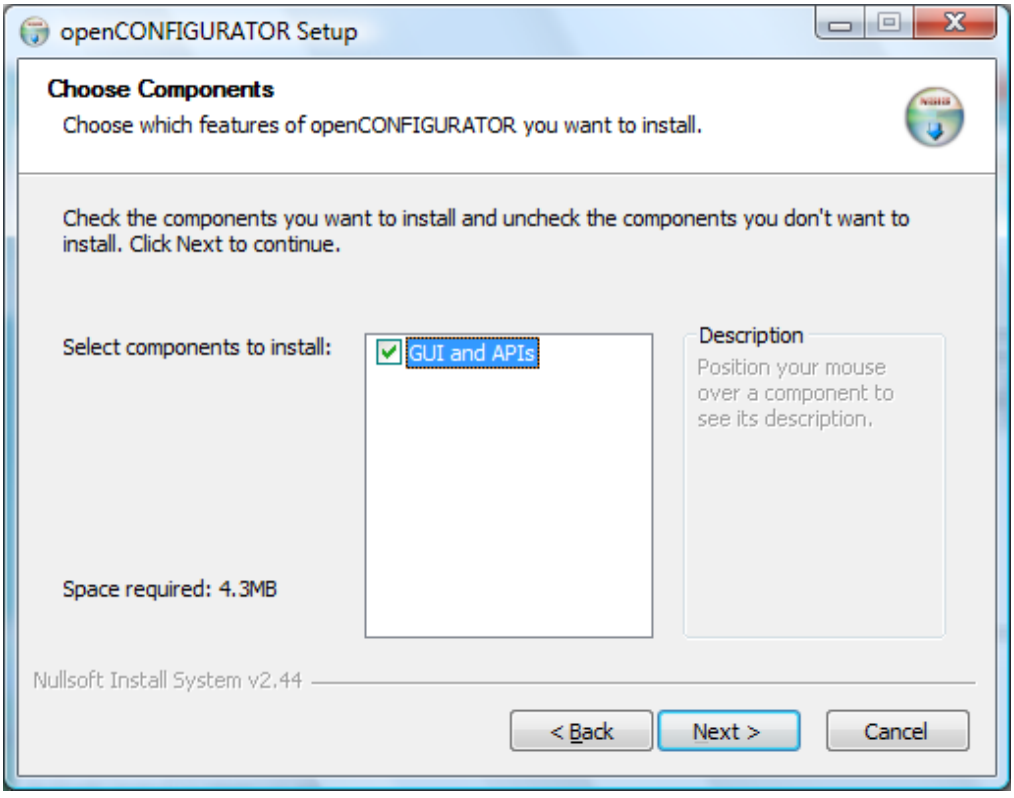
- Un-tar the openCONFICURATOR.tar file
- To check for required packages, run
  - `./configure`
- If configuration succeeds, Makefile will be created
- To install openCONFIGURATOR, run
  - `make install`

6.2.1 Windows (XP and Vista)

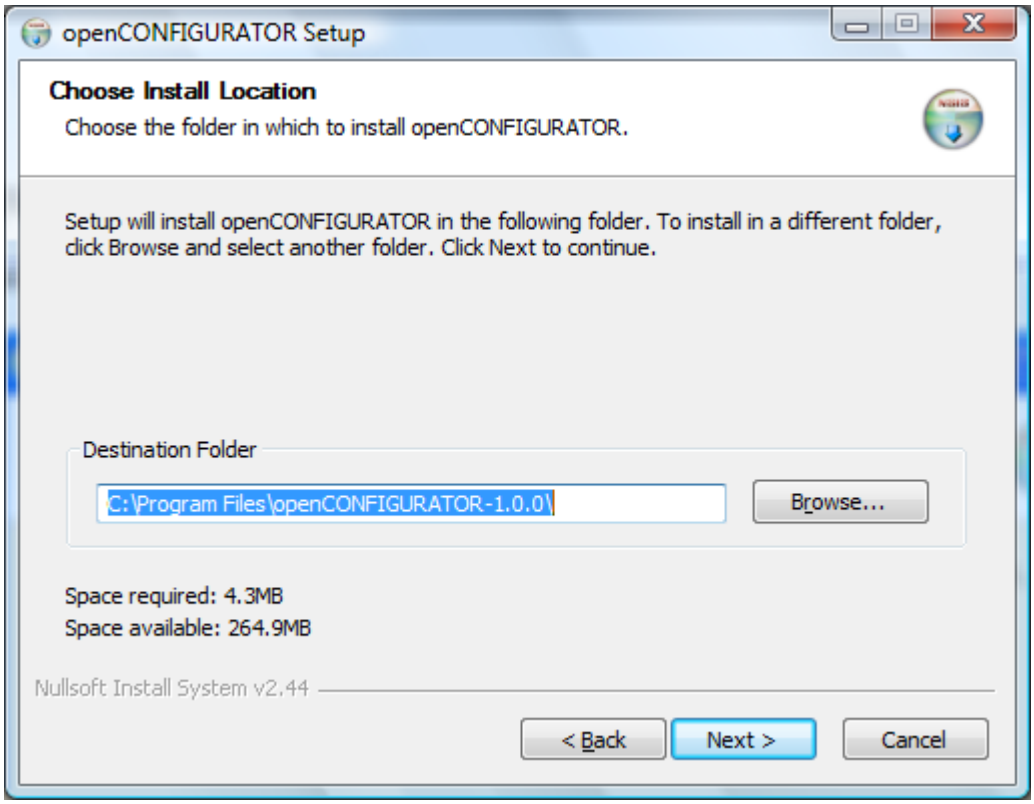
- Unzip the openCONFICURATOR.zip file
- Windows XP
  - Run the openCONFIGURATOR_Setup file and follow the instructions.
- Windows Vista
  - Run the openCONFIGURATOR_Setup file as Administrator [right click on the setup file and click on 'Run as Administrator'] and follow the instructions.



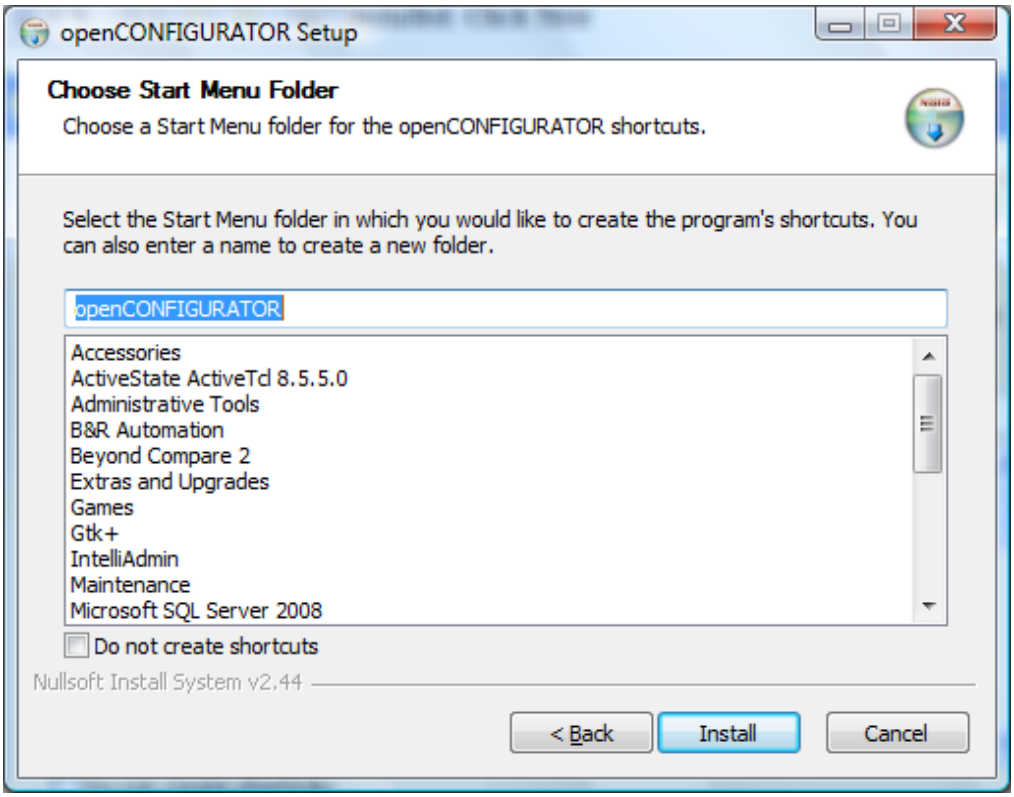
Read through the License and if you agree, press **'I Agree'** button and proceed the installation.



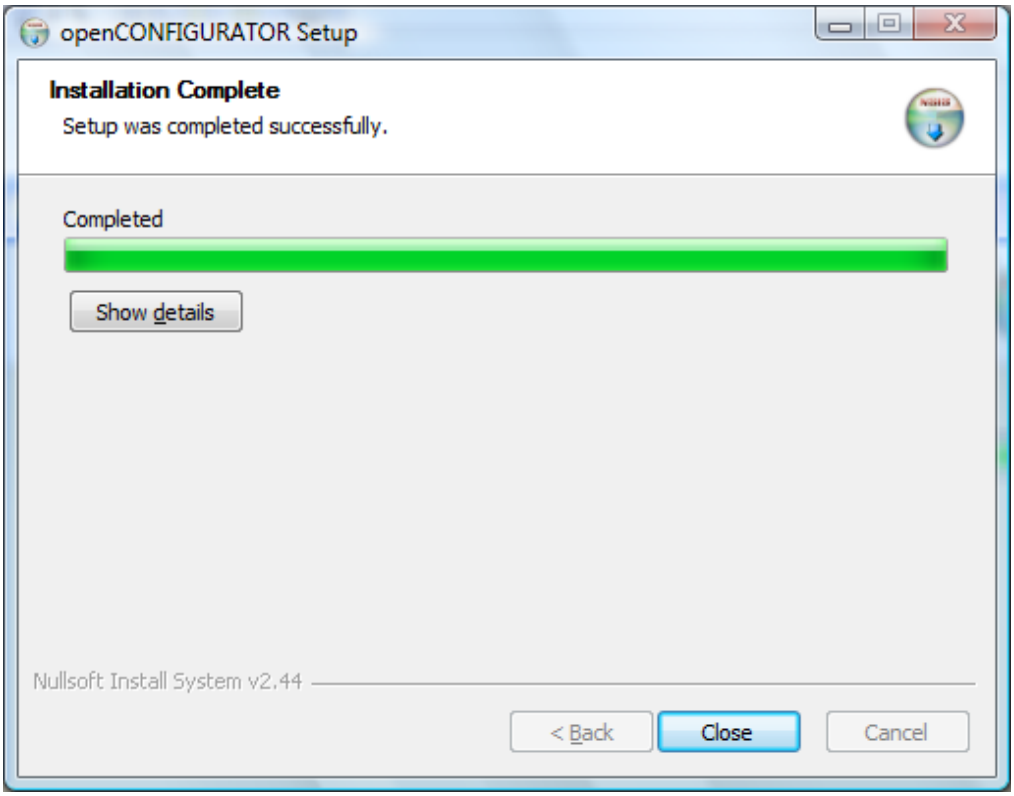
Press **NEXT** button



Select the Directory where the tool should be installed. Click Next



Press **Install**



Press **'Close'.** openCONFIGURATOR is successfully installed.

6.3 Uninstall

6.3.1 Linux

1.Un-tar the openCONFICURATOR.tar file
2.To uninstall openCONFIGURATOR, run

```
1) make  uninstall
```

6.3.2 Windows(XP and Vista)

1.Go to Start Menu > All Programs > openCONFIGURATOR
2.Click uninstall and follow the instructions.

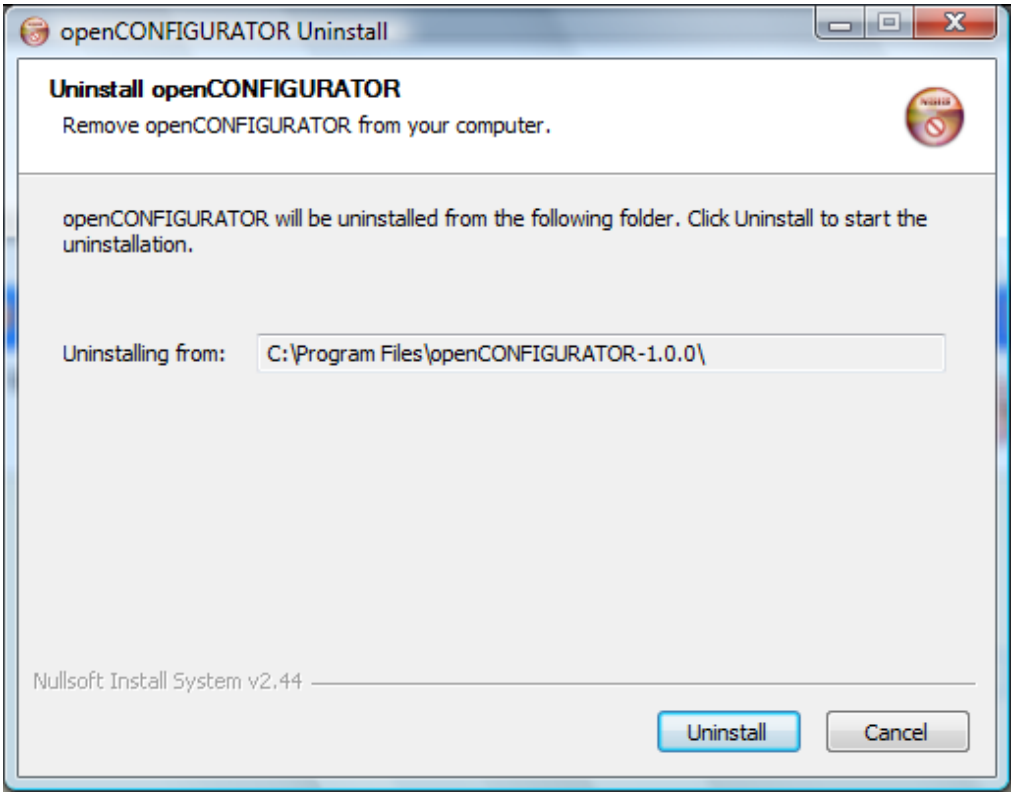
6.3.2.1 XP

Click **'Unistall'**

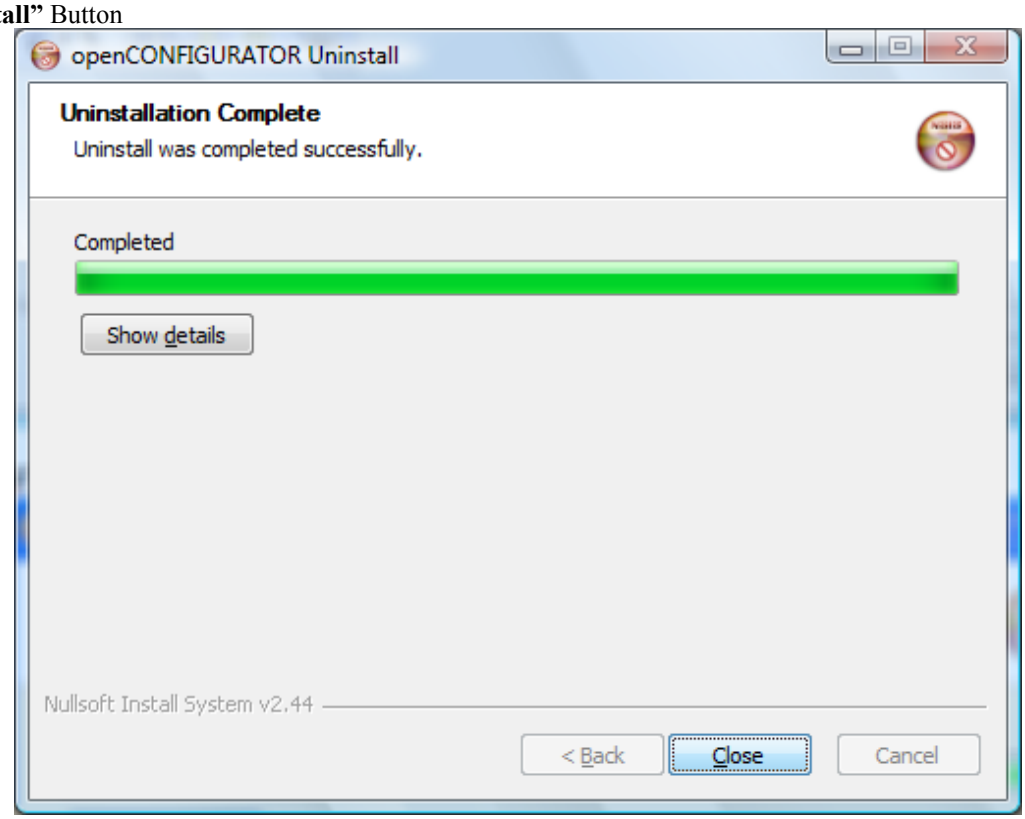
6.3.2.2 Vista

Right Click **'Unistall'** and click 'Run as Administrator'

Note: In Vista, if Uninstallation 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 delete the files installed in the corresponding directory.



Press "**Uninstall"** Button



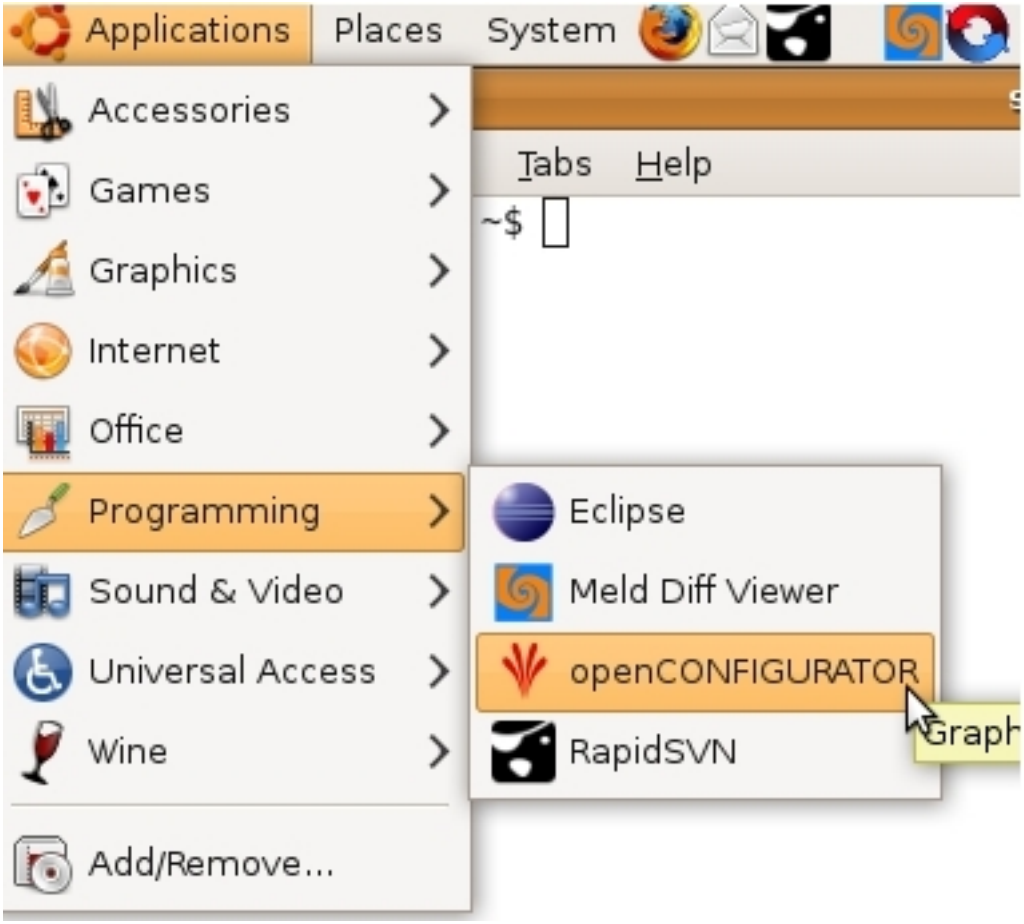
Press **'Close'.** OpenCONFIGURATOR is successfully uninstalled

7 Launch

7.1 Linux

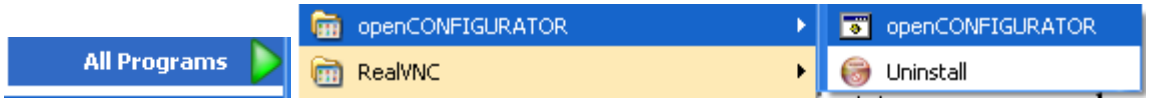
From command prompt:
1.Go to the command prompt
2.To launch, type
1.openCONFIGURATOR

From GUI:
1.Go to Applications> Programming
2.Click on 'openCONFIGURATOR'.



7.2 Windows

1.Go to Start Menu > All Programs > openCONFIGURATOR
2.Click on openCONFIGURATOR



8 Compilation

For compiling the tool follow the below:

8.1 Linux

- Unzip openCONFIGURATORSoln.zip
- Goto “openCONFIGURATORSoln.zip\openCONFIGURATORSoln\openCONFIGURATOR\src”
- Type
  ```
  make
  ```
- The above will compile and create openCONFIGURATOR.so in the same directory
- Copy the openCONFIGURATOR.so into /usr/lib/openCONFIGURATOR.so
  ```
  sudo cp -rvf ./openCONFIGURATOR.so /usr/lib/
  ```
- Goto “openCONFIGURATORSoln.zip\openCONFIGURATORSoln\openConfiguratorWrapper”
- Type
  ```
  make
  ```
- The above will compile and create Tcl_WrapperMain.so in the same directory
- Copy the Tcl_WrapperMain.so into the installed directory of openCONFIGURATOR.
  ```
  sudo cp -rvf ./Tcl_WrapperMain.so /usr/share/openCONFIGURATOR-V-1.0.0/
  ```

Note: The compilation of wrapper needs the openCONFIGURATOR.so in /usr/lib location. If the location is changed, it should also be reflected in the Makefile for Tcl_WrapperMain.so. An issue in any one of the above steps may lead to “Error loading shared library/ Please re-install the tool” error pop-up on the GUI.

8.2 Windows

- Unzip and goto “openCONFIGURATORSoln.zip\openCONFIGURATORSoln”
- open “openCONFIGURATOR.sln” with MS Visual Studio Express Edition
- [The solution was created and tested with MS – Visual C++ 2008 Express Edition]
- Install Active Tcl mentioned in the pre-requisites and set the path of the installed directory as environmental variable
  - $TCL_PATH = “Tcl Installed Dir”
- Install Swigwin and set the installed path of swig
  - $SWIG_PATH = “Swig installed Dir”
- Build the solution
- Inside the Release directory of the solution, two dlls will be created viz., openConfigurator.dll and openConfiguratorWrapper.dll
- Copy the openConfigurator.dll and openConfiguratorWrapper.dll into the installed directory

9 Known bugs and Work around

The following are the list of known bugs in the openCONFIGURATOR-V-1.0.0.

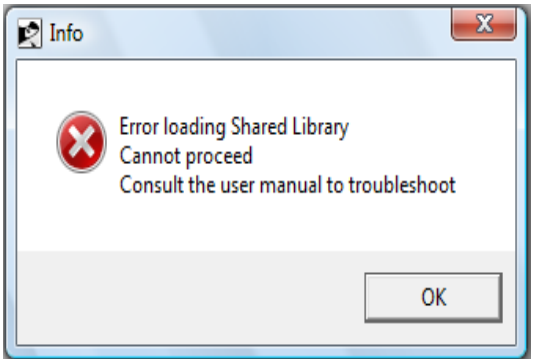


Figure 37

- "Error Loading Shared library \n Cannot proceed\n Consult the user manual to troubleshoot" for binaries downloaded from sourceforge on some machines.
  - Description:
    The above error happens when the shared libraries are corrupted or when the shared libraries are missing. On some Windows machine, the error is happening in a undefined way. We are in the process of fixing it. Till we make a release for this issue, please follow the work around mentioned below.
  - Work Around:
    Recompile the shared libraries on the machine in which the error happens. Follow section "Compilation" for recompilation.

10 Trouble Shooting

1. ***"The file od.xml is missing\n Cannot Proceed\nPlease consult User Manual"***
   The file od.xml is used for storing the objects for the tool. If the file is missing, new objects cannot be added and, the autogeneration will also fail, so the tool will not allow the user to proceed if the file is missing.

   Copy the od.xml from the zip file downloaded from sourceforge into the installed directory and the issue will be fixed.
2. ***"Tranfer CDC/XAP file"***
   If the transfer cdc and xap file is having any issue, please read through the comments described on top of the Transfer.bat/Transfer.sh file.