ECHELON®

OpenLNS Commissioning Tool User's Guide

078-0488-01A

Neuron Chips and other OEM Products were not designed for use in equipment or systems which involve danger to human health or safety or a risk of property damage and Echelon assumes no responsibility or liability for use of the Neuron Chips in such applications.

Parts manufactured by vendors other than Echelon and referenced in this document have been described for illustrative purposes only, and may not have been tested by Echelon. It is the responsibility of the customer to determine the suitability of these parts for each application.

ECHELON MAKES NO REPRESENTATION, WARRANTY, OR CONDITION OF ANY KIND, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE OR IN ANY COMMUNICATION WITH YOU, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR ANY PARTICULAR PURPOSE, NONINFRINGEMENT, AND THEIR EQUIVALENTS.

Printed in the United States of America.
Copyright ©1997–2012 by Echelon Corporation.
Echelon Corporation
www.echelon.com

Table of Contents

Preface xi

Purpose xii
Audience..xii
System Requirements xii
OpenLNS CT Documentation...xii
For More Information and Technical Support...............................xiii
Content xv

1 Introduction 1

Introduction to the OpenLNS Commissioning Tool 2
New Features 2
No-Cost Device Installation...2
Automated Product Activation...3
Annual Product Maintenance 3
Increased Device Compatibility..3
Longer Database Directory Paths...3
Improved Windows Compatibility..3
Additional OpenLNS Events 3
New OpenLNS CT Menus 3
OpenLNS CT Versions 4
OpenLNS CT Network Designs...4
OpenLNS Network Database...4
OpenLNS CT Drawing 5
LONWORKS Basics 5
Networks 5
Devices 5
Protocol 5
Device Templates 5
Channels 5
Routers...6
Applications..6
Network Variables 6
Configuration Properties 7
Functional Blocks 7
Functional Profiles...7
Standard Network Variable and Configuration Property Types 8
User-defined Standard Network Variable and Configuration Property Types 8
Subsystems...8
Supernodes...9
Visio Basics 9

2 Installing and Activating OpenLNS CT... 11

Ordering OpenLNS CT 12
Installing and Activating OpenLNS CT 12
Manually Activating OpenLNS CT 19

3 Getting Started 25

Design Overview 26
Defining Network Requirements and Organization.....................26
Selecting a Network Installation Scenario 26
Engineered System Scenario 26

Ad-Hoc System Scenario ... 27
Determining User Permissions ... 27
Optimizing OpenLNS CT Network Performance ... 27
Drawing Files ... 27
Network Changes ... 28
Connections ... 28
Sharing the OpenLNS Interface with the LNS DDE Server ... 28
Functional Blocks ... 28
Subsystems ... 29
OpenLNS CT Design Manager Overview ... 29
Options ... 32
New Network Options ... 34
OpenLNS CT Stencils ... 36
OpenLNS CT Default Options ... 37
Setting Up a Network Interface ... 38
Optimizing Network Interface Performance ... 38
Using an IP-852 Network Interface ... 38
Creating and Opening OpenLNS CT Networks ... 40
Creating an OpenLNS CT Network Design ... 40
Working with Digital Signatures ... 46
Creating an OpenLNS CT Network from an Existing OpenLNS Database ... 47
Copying an OpenLNS CT Network Design ... 51
Opening an OpenLNS CT Network Design ... 53
OpenLNS CT Client Types ... 55
Local Client ... 55
Remote Full Client ... 56
Remote Lightweight Client ... 60
Using Network Service Device SmartShapes ... 63
Listing Network Service Devices ... 64
Upgrading Network Service Devices ... 65
Replacing a Local Network Service Device ... 66
User Profiles ... 66
Creating a New User Profile ... 67
Changing Passwords ... 69
Modifying and Deleting User Profiles ... 70
Changing User Profiles in an OpenLNS CT Drawing ... 70
Using OpenLNS CT Remotely with User Profiles ... 71

4 Designing Networks ... 73

Creating a LONWORKS Network ... 74
Creating an OpenLNS CT Drawing ... 74
Creating Application Devices ... 75
Using Automatic Channel Selection for Devices ... 79
Changing the Channel of an Application Device ... 80
Creating Functional Blocks ... 81
Using Automatic Device Selection ... 85
Reassigning a Functional Block ... 85
Configuring a Functional Block ... 86
Copying a Functional Block Configuration ... 86
Creating a New Functional Block from a Configured Functional Block ... 86
Updating a Functional Block from a Configured Functional Block ... 86
Creating a Functional Block Master SmartShape ... 89
Creating Dynamic Functional Blocks ... 89

Deleting Dynamic FBs 91
Creating Dynamic Functional Block Master SmartShapes 91
Creating a Virtual Functional Block 91
Adding a Message Tag SmartShape to a Virtual Functional Block 92
Deleting Message Tag SmartShapes 92
Creating Network Variables 92
Creating a Network Variable SmartShape 93
Creating a Network Variable Master SmartShape 93
Adding Network Variable SmartShapes to a Functional Block 93
Creating Network Variables Using Generic Network Variable SmartShapes 94
Creating Dynamic Network Variables Using Network Variables SmartShapes from the OpenLNS CT NV SmartShapes Stencil 98
Creating Dynamic Network Variables Using Network Variable Master SmartShapes 98
Changing a Network Variable Name 99
Changing Network Variable Position 99
Deleting a Network Variable SmartShape 99
Deleting Dynamic Network Variables 101
Creating a Router 101
Using Automatic Channel Selection for Routers 105
Changing a Router Channel 106
Creating a Channel 106
Creating a Subsystem 108
Creating a Supernode 110
Renaming and Deleting a Supernode Network Variable 111
Copying a Subsystem or Supernode 111
Creating Connections 112
Creating a Connection with the Connector SmartShape 113
Creating a Connection with the Connector Tool 113
Creating a Connection with the Network Variable Connection Dialog Box 114
Using Reference Connection SmartShapes 117
Aligning Reference Connection SmartShapes 117
Using Connection Descriptions 118
Connection Description Properties 118
Message Service Type 118
Addressing Mode 119
Viewing and Creating Connection Descriptions 119
Using Automatic Connection Description Selection 121
Changing the Connection Description for a Connection 123
Hiding and Showing Connector SmartShapes 124
Hiding and Showing All Connector SmartShapes in a Subsystem 124
Hiding and Showing Selected Connector SmartShapes Attached to Functional Blocks or Supernodes 124
Hiding and Showing Connector SmartShapes Attached to a Functional Block 126
Viewing and Navigating OpenLNS CT Network Design 128
Opening any Subsystem 128
Opening a Parent Subsystem 129
Using Network Navigators 129
Navigating and Managing a Network with the Network Explorer 130
Navigating a Network with the Network Explorer 132

Managing a Network with the Network Explorer 132
Managing and Editing Device Templates with the Network Explorer 132
Managing Device Templates 132
Editing Device Template Properties 133
Working with OpenLNS CT Layers 135
Working with OpenLNS CT SmartShapes 136
Viewing and Setting OpenLNS CT SmartShape Properties 136
Moving an OpenLNS CT SmartShape 136
Repositioning an OpenLNS CT SmartShape 136
Changing the Subsystem for an OpenLNS CT SmartShape 136
Changing the Channel for an OpenLNS CT SmartShape 137
Moving a Device or Router to a Different Channel 137
Moving an OpenLNS Computer to a Different Channel 139
Copying an OpenLNS CT SmartShape 140
Deleting an OpenLNS CT SmartShape 140
Customizing the User Interface 141
Editing the Title Block 141
Using OpenLNS CT SmartShape Menus 141
Using AutoCAD Drawings 142
Importing an AutoCAD Drawing 142
Exporting an AutoCAD Drawing 143

5 Installing Networks 145

Network Installation Overview 146
Commissioning a Device 146
Selecting Devices for Commissioning 147
Loading a Device Application Image 148
Setting the Initial Device State and Source of Configuration Property Values 150
Setting the Initial Application Device State and Source of Configuration Property Values 151
Setting the Initial Router State 153
Setting the Neuron ID 154
Selecting the Device Identification Method 154
Using the Service Pin Method 155
Using the Manual Entry Method 157
Entering a Neuron ID Manually 157
Entering a Neuron ID Using a Bar Code Scanner 157
Commissioning an IP-852 Router 158
Commissioning Using Device Discovery 158

6 Monitoring and Controlling Networks 165

Monitoring and Controlling Overview 166
Using Monitored Connections 166
Displaying a Network Variable Value 167
Using the OpenLNS CT Browser 168
Starting the OpenLNS CT Browser 168
The OpenLNS CT Browser Toolbar 170
Customizing the Browser 170
Customizing Browser Columns 171
Hiding or Changing the Order of Browser Columns 171
Adjusting the Width of Browser Columns 171
Customizing Browser Rows 171
Selecting Browser Rows to be Displayed 171
Hiding Browser Rows 172

Saving Browser Customization 173
Monitoring Network Variables 173
Enabling Network Variable Monitoring 173
Disabling Network Variable Monitoring 173
Getting Network Variable Values 173
Using Bound Updates 174
Binding Network Variables to the Host 174
Updating Network Variable and Configuration Property Values 175
Setting Values 175
Setting Values for Structured Objects 175
Setting Values for Configuration Property Arrays 177
Clearing Values 178
Getting Values 178
Changing a Network Variable or Configuration Property Type 178
Changing a Network Variable or Configuration Property Format 179
Displaying Error Messages 181
Managing Functional Blocks 181
Using Data Point SmartShapes 182
Adding and Monitoring a Data Point SmartShape 182
Updating a Data Point 185
Updating a Scalar Data Point 186
Updating a Structured Data Point 186
Updating an Enumerated Data Point 187
Getting a Data Point Value 187
Creating and Using a Custom Data Point Master SmartShape 188
Creating a Custom Data Point Master SmartShape 188
Using a Custom Data Point Master SmartShape 188
Copying and Creating a Data Point SmartShape 189
Deleting a Data Point SmartShape 189
Creating an HMI with Data Point SmartShapes 189
Writing Data Point SmartShape Values 190
Using an Add-On to Write Values 190
Using a Macro to Write Values 191
Reading Data Point SmartShape Values 192
Using an Add-On to Read Values 192
Using a Macro to Read Values 193
Organizing HMIs 194

7 Maintaining Networks 195

Maintaining Networks Overview 196
Loading Devices 196
Selecting the Devices to Load 197
Selecting or Creating a Device Template 197
Selecting an Application Image and a Neuron Firmware Image 198
Selecting Initial Device State and Source of Configuration Property Values 200
Replacing Devices 202
Attaching a New Device to the Network 202
Replacing a Device in the OpenLNS CT Network 202
Removing the Old Device 205
Decommissioning Devices 205
Resynchronizing Configuration Properties 205
Using OpenLNS CT as a Passive Configuration Tool 207
Backing up an OpenLNS CT Network Design 207
Manually Backing Up an OpenLNS CT Network 208
Creating a Backup from an OpenLNS CT Drawing 208

Creating a Backup from the OpenLNS CT Design Manager 210
Scheduling Drawing Saves and Database Backups............................ 210
Scheduling Backups with OpenLNS CT 210
Scheduling Backups with the Windows Task Scheduler 212
Restoring an OpenLNS CT Network.. 214
Restoring a Network Database Backup 214
Restoring an OpenLNS CT Drawing Backup 216
Restoring a Full Network Backup 217
Recovering an OpenLNS CT Network 221
Subsystem Recovery Options... 222
Using Subsystem Paths 224
Using Subsystem IDs 224
Using the OpenLNS Database Recovery Wizard................................ 225
Network Recovery vs. Database Backup 234
Resynchronizing an OpenLNS CT Network 235
Automatic OpenLNS CT Drawing Synchronization 235
OpenLNS Event Tracking.. 236
OpenLNS CT Event Log.. 236
Viewing the OpenLNS CT Event Log 236
Exporting the OpenLNS CT Event Log.................................. 237
Manual Network Resynchronization 239
Refreshing the OpenLNS CT Network... 247
Merging OpenLNS CT Networks 247
Network Merge Considerations.. 248
Limitations 248
Information Loss 248
Merging OpenLNS CT Networks 249

8 Managing Networks 253

Using the OpenLNS CT Device Manager 254
Opening the OpenLNS CT Device Manager 254
Managing Devices.. 255
Managing Functional Blocks 256
Managing Routers.. 258
Device Manager Settings.. 259
Using OpenLNS CT Styles Overview.. 260
OpenLNS CT Device Styles... 260
OpenLNS CT Functional Block Styles 261
Network Variable and Message Tag SmartShape Styles..................... 263
OpenLNS CT Connector SmartShape Styles...................................... 263
Generating a Device Status Summary Report 264
Using Resource Usage Reports... 267
Network Resource Report.. 267
Alias Table Summary... 268

9 Exporting and Importing Networks with XML........................... 271

Using XML Export/Import Overview 272
Exporting a LONWORKS Network XML File 272
Viewing an OpenLNS CT Network Report.. 274
Importing a LONWORKS Network XML File... 277

10 Managing OpenLNS CT Licenses 279

Commissioning Devices with OpenLNS CT 280
Overview of OpenLNS CT Licensing.. 280
Maintaining OpenLNS CT.. 280

Upgrading OpenLNS CT Standard Edition to OpenLNS CT Professional Edition 286
Renewing OpenLNS CT Annual Maintenance 289
Transferring OpenLNS CT Licenses 289

11 Using Plug-ins 293

Using Plug-ins Overview 294
Starting a Plug-in 294
Viewing Plug-in Information 296
Viewing Plug-in Status 297
Re-Registering Plug-ins 298
Disabling and Enabling Plug-ins 299
Disabling a Plug-in 299
Enabling a Plug-in 299

12 Creating and Using Custom OpenLNS CT SmartShapes and Stencils 301

OpenLNS CT Stencils 302
Creating a Custom OpenLNS CT Stencil 302
Creating Custom OpenLNS CT Master SmartShapes 303
Using Custom OpenLNS CT Master SmartShapes 304
Device Master SmartShapes 304
Functional Block Master SmartShapes 305
Subsystem or Supernode Master SmartShapes 305
Connection Master SmartShapes 306
Creating Additional Channels 308
Editing Master SmartShape User Defined Cells 308
Additional Device User Cells 309
Additional Functional Block User Cells 314
Additional Router SmartShape User Cells 315
Setting User Functional Block Scopes and Types 315
Adding a Bitmap to a Device Master SmartShape 315
Viewing and Editing VBA Code Associated with an OpenLNS CT Network Drawing 316

Appendix A Setting OpenLNS CT Default Options 317

Setting OpenLNS CT Default Options 318
Backup/Restore Options 319
Configuration Properties Options 322
Device Options 325
Functional Block Options 327
General Options 330
OpenLNS Event Options 331
Naming Options 333
Network Explorer Options 335
NV Browser/Monitoring Options 336
OpenLNS CT Font Options 337
Recovery Options 338
Service Pin Options 341
Shape ToolTips Options 342
Synchronization Options 343
Warnings Options 344

Appendix B Glossary 347

Appendix C OpenLNS Software License Agreement 367

Appendix D Software License Transfer Agreement........................ 375

Preface

The OpenLNS Commissioning Tool (OpenLNS CT) is a software package for designing, installing, and maintaining multi-vendor, open, interoperable LONWORKS® control networks. Based on Echelon's OpenLNS network operating system, OpenLNS CT combines support for open LONWORKS control networks with a user-friendly Microsoft Visio interface. The result is a software tool that's robust enough to work with all your devices, yet economical enough to leave behind as an operations and maintenance tool. OpenLNS CT, OpenLNS Server, and the OpenLNS SDK are the three primary components of OpenLNS. OpenLNS CT complies with the OpenLNS plug-in standard, and it is compatible with the LNS plug-in standard, making it compatible with the wide variety of plug-ins available from Echelon and many other vendors.

Purpose

This guide outlines the new features in the OpenLNS Commissioning Tool (OpenLNS CT), and it describes how to use OpenLNS CT to design, commission, monitor and control, maintain, and manage a network. OpenLNS CT includes online help that provides context-sensitive documentation that supplements the information in this guide.

Audience

This guide is intended for system designers and integrators with an understanding of control networks.

System Requirements

System requirements for computers running the OpenLNS CT are as follows:

- Microsoft Windows® 7 (64-bit and 32-bit), Windows Server 2008 SR2 64-bit, Windows Vista® with Service Pack (SP) 1, or Windows XP with SP3 (32-bit).
- 500 MHz processor or faster. 2 GHz processor recommended.
- 2 GB or more of free disk space.
- 512 MB RAM. 2 GB RAM recommended.
- 1,024 MB page file minimum. 2,048 page file recommended.
- 1,024 x 768 or higher-resolution display with at least 256 colors.
- Mouse or compatible pointing device
- DVD-ROM drive.
- OpenLDV 4.0-compatible local, remote, or IP-852 network interface.
    - Compatible local network interfaces include the U10/U20 USB network interface; PCC-10, PCLTA-20, or PCLTA-21 network interface cards; and the SLTA-10 Serial LonTalk Adapter. The PCC/PCLTA and SLTA-10 network interfaces are compatible with 32-bit versions of Windows only.
    - Compatible remote network interfaces include the SmartServer, i.LON 100 Internet Server, i.LON 600 IP-852 Router, and i.LON 10 Ethernet Adapter.
    - Compatible IP-852 network interfaces include the SmartServer (with IP-852 routing option), i.LON 100 Internet Server (with IP-852 routing option), and i.LON 600 IP-852 Router.

OpenLNS CT Documentation

The documentation for OpenLNS CT is provided as Adobe Acrobat PDF files and online help files. The PDF file for this document is installed in the **Echelon OpenLNS CT** program folder when you install the Echelon OpenLNS CT software. You can also download the latest OpenLNS documentation, including the latest version of this guide, by going to the Echelon OpenLNS Web site at *www.echelon.com/openlns*.

OpenLNS CT XML Programmer's Guide	Describes how to create and modify an OpenLNS network using the OpenLNS XML Plug-in.
OpenLNS Programmer's Guide	Describes how to use the OpenLNS Object Server ActiveX Control to develop OpenLNS apps.

OpenLNS Programmer's Reference
Provides reference information for writing OpenLNS tools, applications, and plug-ins. Describes the objects in the OpenLNS Object hierarchy, and details their properties, methods, and events.

LNS®Plug-in Framework Developer's Guide
Describes how to write system and device plug-ins using .NET programming languages such as C# and Visual Basic .NET.

The following documents supplement the material provided in this guide. You can download these documents from Echelon's Web site at *www.echelon.com/docs*.

i.LON 600 LonWorks/IP Server User's Guide
Describes how to install, configure, use, and manage the i.LON 600 IP-852 routers, and how to use the Echelon IP-852 Configuration Server.

Introduction to the LonWorks® Platform
Provides a high-level introduction to LonWorks networks and the tools and components that are used for developing, installing, operating, and maintaining them.

IP-852 Channel User's Guide
Describes how to configure an IP-852 channel with the *Echelon IP-852 Configuration Server*. You will need this information if you plan on attaching your OpenLNS CT computer to an IP-852 channel.

LonMark® SNVT and SCPT Guide
Documents the standard network variable types (SNVTs) and standard configuration property types (SCPTs) used by LonWorks device applications. For more information, go the LonMark International Web site at www.lonmark.org/technical_resources/resource_files.

LonWorks® USB Network Interface User's Guide
Describes how to install and use the U10 and U20 USB Network Interfaces to connect an OpenLNS or OpenLDV application to a LonWorks network.

Mini FX User's Guide
Describes how to use the Mini kit to develop a prototype or production control system that requires networking, particularly in the rapidly growing, price-sensitive mass markets of smart light switches, thermostats, and other simple devices and sensors.

NodeBuilder® FX User's Guide
Describes how to use the NodeBuilder tool to develop LonWorks device applications and build and test prototype and production LonWorks devices

PCC/PCLTA Network Interface User's Guide
Describes how to install, configure, and test the PCC-10, PCLTA-20, and PCLTA-21 network interface cards that you can use to connect an OpenLNS or OpenLDV application to a LonWorks network.

SmartServer 2.0 User's Guide
Describes how to configure the SmartServer and use its applications to manage control networks.

For More Information and Technical Support

The **Echelon OpenLNS CT ReadMe** document provides descriptions of known problems, if any, and their workarounds. To view the **Echelon OpenLNS CT ReadMe** document, click **Start**, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **Echelon OpenLNS CT ReadMe**.

If you have technical questions that are not answered by this document, the OpenLNS CT online help files, or the **Echelon OpenLNS CT ReadMe** document, you can contact Echelon technical support. There is no charge for software installation-related questions during the first 30 days after you receive the OpenLNS CT DVD or purchase an OpenLNS CT activation key. To receive technical support from Echelon, you must purchase support services from Echelon or an Echelon support partner. See *www.echelon.com/support* for more information on Echelon support. Your OpenLNS CT distributor may also provide customer support.

You can also enroll in training classes at Echelon or an Echelon training center to learn more about using OpenLNS CT. You can find additional information about device development training at *www.echelon.com/training*.

You can obtain technical support via phone, fax, or e-mail from your closest Echelon support center. The contact information is as follows:

Region	**Languages Supported**	**Contact Information**
The Americas	English Japanese	Echelon Corporation Attn. Customer Support 550 Meridian Avenue San Jose, CA 95126 Phone (toll-free): 1.800-258-4LON (258-4566) Phone: +1.408-938-5200 Fax: +1.408-790-3801 *lonsupport@echelon.com*
Europe	English German French Italian	Echelon Europe Ltd. Suite 12 Building 6 Croxley Green Business Park Hatters Lane Watford Hertfordshire WD18 8YH United Kingdom Phone: +44 (0)1923 430200 Fax: +44 (0)1923 430300 *lonsupport@echelon.co.uk*
Japan	Japanese	Echelon Japan Holland Hills Mori Tower, 18F 5-11.2 Toranomon, Minato-ku Tokyo 105-0001 Japan Phone: +81.3-5733-3320 Fax: +81.3-5733-3321 *lonsupport@echelon.co.jp*
China	Chinese English	Echelon Greater China Rm. 1007-1008, IBM Tower Pacific Century Place 2A Gong Ti Bei Lu Chaoyang District Beijing 100027, China Phone: +86-10-6539-3750 Fax: +86-10-6539-3754 *lonsupport@echelon.com.cn*
Other Regions	English Japanese	Phone: +1.408-938-5200 Fax: +1.408-328-3801 *lonsupport@echelon.com*

Content

This guide includes the following content:

- *Introduction*: Provides an introduction to the OpenLNS CT, new features; and the basics of OpenLNS CT network designs, LONWORKS, and Visio.
- *Installing and Activating OpenLNS CT*: Describes how to order and install OpenLNS CT and Microsoft Visio 2010, and then how to activate OpenLNS CT.
- *Getting Started*: Provides information on the planning an OpenLNS CT network design; using the OpenLNS CT Design Manager; setting up a network interface; creating and opening an OpenLNS CT network design; OpenLNS CT client types; using OpenLNS CT remotely; and creating and using user profiles.
- *Designing Networks*: Describes how to design a network using OpenLNS CT. Covers how to create the following objects in an OpenLNS CT drawing: application devices, functional blocks, network variables, routers, channels, and subsystems. Explains how to connect network variables. Explains working with OpenLNS CT SmartShapes® and layers, customizing the OpenLNS CT user interface, and using OpenLNS CT with AutoCAD drawings.
- *Installing Networks*: Describes how to install devices using OpenLNS CT, including how to load applications into them, set the initial state of their applications, set the source of their configuration properties, and select how they manage device-specific configuration properties. It also explains the different methods for acquiring device Neuron IDs and how to alternatively use the device discovery method to install a network.
- *Monitoring and Controlling Networks*: Describes how you can monitor and control the devices in your network with OpenLNS CT. Describes the three methods you can use to read and/or write network variables and configuration properties: using monitored connections, browsing with the OpenLNS CT Browser, and using Data Point SmartShapes. Covers how to bind network variables to the host in order to receive event-driven updates. Describes how to change the types and formats of network variables and configuration properties. Explains how to create simple HMIs in your OpenLNS CT drawing with Data Point and Visio SmartShapes.
- *Maintaining Networks*: Provides an overview of network maintenance tasks that you can perform with OpenLNS CT. Describes loading, replacing, and decommissioning devices. Explains how to resynchronize and propagate configuration properties values. Explains how to back up and restore an OpenLNS CT network; create an OpenLNS CT network by recovering information from the physical network; and resynchronize the network database, OpenLNS CT drawing, and physical devices. Explains how to merge two OpenLNS CT networks.
- *Managing Networks*: Explains how to test and verify application devices, functional blocks, and routers; describes OpenLNS CT styles; and details how to generate device status summary reports, network resource reports, and OpenLNS network reports.
- *Managing OpenLNS CT Licenses*: Provides an overview of OpenLNS CT licensing. Describes how to upgrade OpenLNS CT, upgrade an OpenLNS CT Standard Edition to the Professional Edition, renew your OpenLNS CT annual maintenance contract, and transfer an OpenLNS CT license.
- *Exporting and Importing a Network Using XML*: Describes how to export a LONWORKS network to an OpenLNS CT network XML file; view a LONWORKS network XML file in a Web browser; edit an OpenLNS CT network XML file; and import a LONWORKS network XML file to update a network.
- *Using Plug-ins*: Provides an overview of plug-ins and then describes how to start a plug-in, get plug-in information, and re-register, enable, and disable plug-ins.

- *Creating and Using OpenLNS CT SmartShapes and Stencils*: Describes how to create an OpenLNS CT stencil and create and use custom master SmartShapes for devices, functional blocks, subsystems, and connections. Explains how to modify a master SmartShape by changing its user-defined cells.
- *Appendices*: Includes the OpenLNS CT default options, a glossary, the OpenLNS CT Software License Agreement, and the OpenLNS CT License Transfer Agreement.

1

Introduction

This chapter provides an introduction to the OpenLNS Commissioning Tool, describes new features, and explains the basics of OpenLNS CT Network Designs, LONWORKS, and Visio.

Introduction to the OpenLNS Commissioning Tool

The OpenLNS Commissioning Tool (OpenLNS CT) is a LONWORKS® network tool that runs on the OpenLNS network operating system and uses Microsoft Visio® 2010 or Visio 2003 (Professional or Standard editions) as a graphical user interface. The OpenLNS network operating system implements a client/server architecture with directory, installation, management, monitoring, and control services provided by an OpenLNS Server that is included with the OpenLNS CT. The OpenLNS Server allows multiple users running OpenLNS CT and other OpenLNS tools, applications, and plug-ins on separate computers to access the OpenLNS Server simultaneously. This means that managers, system integrators, installers, and maintenance personnel can all work on the same LONWORKS network at the same time. OpenLNS CT is compatible with Windows 7 (64-bit and 32-bit), Windows Server 2008 R2 (64-bit), Windows Vista with SP1 (32-bit), and Windows XP with SP3 (32-bit). The OpenLNS Server and OpenLNS CT are backwards compatible with all existing LNS Turbo Edition databases and LonMaker Turbo Edition drawings providing simple migration for existing LNS and LonMaker networks.

OpenLNS CT can be used to manage all phases of a network's life cycle—from the initial design and commissioning to the ongoing operation. It provides the functionality of several network tools in one single solution:

- *Network Design Tool*. You can design a network offsite (without actually being connected to the network) and/or onsite, and modify it anytime.
- *Network Installation Tool*. You can rapidly install a network designed offsite once it is brought onsite. The device definitions can be quickly and easily associated with their corresponding physical devices to reduce on-site commissioning time. The OpenLNS CT Browser provides complete access to all network variables and configuration properties.
- *Network Documentation Tool*. You can create an OpenLNS CT drawing during the network design and installation process. This OpenLNS CT drawing is an accurate, logical representation of the installed physical network. The OpenLNS CT drawing is therefore an essential component of as-built reports.
- *Network Operation Tool*. You can operate the network using the operator interface pages contained within the OpenLNS CT drawing.
- *Network Maintenance Tool*. You can easily add, test, remove, modify, or replace devices, routers, channels, subsystems, and connections to maintain the network.

New Features

This section describes the major features included with the OpenLNS Server and OpenLNS CT.

- No-cost device installation.
- Automated product activation (Internet connection required).
- Annual product maintenance.
- Increased device compatibility.
- Longer database directory paths.
- Improved Microsoft Windows® Compatibility.
- Additional OpenLNS events.
- New OpenLNS CT Menus.

No-Cost Device Installation

You can install devices without commissioning fees (known as "credits" in LNS Turbo and LonMaker Turbo software). OpenLNS CT features no-cost installation for all LONWORKS devices that comply with the ISO/IEC 14908-1 Control Network Protocol. This includes devices based on the FT 5000 Smart Transceiver, Neuron® 5000 Processor, FT 3150®/3120® Smart Transceiver, or PL

3170/3150/3120 Smart Transceiver, and also includes devices based on third-party ISO/IEC 14908-1 protocol processors. This reduces network installation costs, makes installation and maintenance costs more predictable, and simplifies the installation process.

Automated Product Activation

You can quickly install and activate the OpenLNS CT software. When you install the OpenLNS CT software on an Internet-connected computer, the installer automatically connects to the Echelon License Server, the License Server issues activation licenses for OpenLNS Server and OpenLNS CT, and the licenses are installed on your computer. If your computer does not have access to the Internet, you can manually activate your software via e-mail or phone using the Echelon License Wizard, which provides easy-to-follow instructions for activating Echelon software products. See Chapter 2, *Installing and Activating OpenLNS CT*, for more information on installing and activating OpenLNS CT.

Annual Product Maintenance

The OpenLNS CT software products each include one-year maintenance during which you can download and install OpenLNS CT software updates and upgrades for free. You can renew your annual maintenance anytime before it expires. Renewing your maintenance enables you to continue installing software updates. If you do not renew the product's maintenance, you can still use the product; however, you will not be able to install any updates or upgrades released after the expiration of your maintenance period.

Increased Device Compatibility

OpenLNS supports network variables with up to 225 bytes. This expands OpenLNS compatibility to include devices with network variables longer than 31 bytes.

Longer Database Directory Paths

OpenLNS CT supports network database paths up to 230 characters (the previous limit in the LNS Turbo Server was 23 characters). This means that OpenLNS data can now be stored in any user data directory on your computer.

Improved Windows Compatibility

The OpenLNS Server and OpenLNS CT are now installed in the **C:\Program Files\LonWorks** directory by default, which is a more compatible location with Windows conventions for program file installation. Windows has become increasingly more restrictive about default access permissions on the computer’s root directory. These restrictions caused compatibility issues with LNS Turbo and LonMaker Turbo Editions, which were installed in the **C:\LonWorks** directory by default. If you have previously installed the LonMaker tool or other LNS application on your computer and you already have a LONWORKS directory, OpenLNS CT will continue to use your existing directory.

Additional OpenLNS Events

To improve synchronization between OpenLNS CT and other OpenLNS apps, the OpenLNS Server includes new events for when device templates and extensions are updated.

New OpenLNS CT Menus

If you are using OpenLNS CT with Visio 2010, you now click **Add-ins** to access the options previously available in the **LonMaker** menu (for example, network options, network properties, synchronization, and documents [backup/restore, XML export/import]). This manual assumes you are using Visio 2010—if you are using Visio 2003, open the **LonMaker** menu when this document instructs you to click **Add-ins** and then select the desired option.

OpenLNS CT Versions

There are five versions of OpenLNS CT: OpenLNS CT Professional, OpenLNS CT Professional Without Visio, OpenLNS CT Standard, OpenLNS CT Standard Without Visio, and OpenLNS CT Trial. OpenLNS CT Professional Without Visio and OpenLNS CT Standard Without Visio require a separate purchase of Visio 2010 or 2003.

The advantage of OpenLNS CT Professional is that you can have an unlimited number of OpenLNS networks—OpenLNS CT Standard is limited to five networks. In addition, OpenLNS CT Professional includes Visio 2010 Professional, which contains all of the business diagramming tools of the Visio 2010 Standard edition, as well as additional comprehensive technical and drawing solutions. You can purchase OpenLNS CT Standard and then upgrade it later to OpenLNS CT Professional by purchasing the OpenLNS CT Standard to Professional Upgrade Key (Echelon model number 38070-400). The upgrade key does not include and upgrade to Visio 2010 Professional—it only upgrades the number of networks that you can access with OpenLNS CT. See *Upgrading OpenLNS CT Standard Edition to OpenLNS CT Professional Edition* in Chapter 10 for more information.

OpenLNS CT Professional and OpenLNS CT Standard each include one-year maintenance during which you can download and install OpenLNS CT software updates and upgrades for free. You can renew your annual maintenance anytime before it expires by purchasing an OpenLNS CT Professional One-Year Maintenance Renewal (Echelon model number 93800) or an OpenLNS CT Standard One-Year Maintenance Renewal (Echelon model number 93810). Renewing your license enables you to continue installing software updates and upgrades. If you do not renew the product's maintenance, you can still use the product; however, you will not be able to install updates or upgrades released after your annual maintenance expired.

You can download a free trial edition of OpenLNS CT from the Echelon Web site at *www.echelon.com/openlns*. The trial edition is limited to 2 five-device networks, runs for a maximum of 60 minutes at a time, and expires after 60 days. Visio 2010 is required to use the trial edition, but it is not included with the trial edition. The DVD version of the trial edition does include a trial version of Visio 2010. You can convert the trial edition to OpenLNS CT Professional or OpenLNS CT Standard by purchasing the OpenLNS Commissioning Tool Professional Edition Without Visio Activation Key (Echelon model number 38060-400), OpenLNS Commissioning Tool Professional Activation Key (Echelon model number 38050-400), OpenLNS Commissioning Tool Standard Edition Without Visio Activation Key (Echelon model number 38160-400), or the OpenLNS Commissioning Tool Standard Activation Key (Echelon model number 38150-400).

OpenLNS CT Network Designs

An OpenLNS CT Network Design consists of an *OpenLNS network database* and an *OpenLNS CT drawing*.

OpenLNS Network Database

An OpenLNS network database contains definitions of the devices contained within a LONWORKS network and an OpenLNS CT drawing, including information such as the name, address, application configuration, and network connections for each device. Whenever you change an OpenLNS CT drawing, OpenLNS CT automatically uses OpenLNS services to update the information in the OpenLNS network database associated with the OpenLNS CT drawing.

Every OpenLNS CT drawing is associated with an OpenLNS network database, and the OpenLNS network database is always located on the computer with the OpenLNS Server. The OpenLNS Server may be located on the same computer as OpenLNS CT, or it may be located on a different computer. You can maintain backup copies of the OpenLNS network database, and you can move the OpenLNS Server and OpenLNS network database to a backup computer. See *Backing Up an OpenLNS CT Network Design* in Chapter 7, *Maintaining Networks*, for more information.

OpenLNS CT Drawing

An *OpenLNS CT drawing* contains the graphical representation of the LONWORKS network. An OpenLNS CT drawing consists of one or more drawing files, and the drawing files consist of one or more pages that each represent a *subsystem*. One drawing file is designated as the *top-level drawing file*; this drawing file is the one created when you create a new OpenLNS CT network design. A subsystem within the top-level drawing file is designated as the top-level subsystem; this subsystem is represented by the first page of the drawing that you create when you create a new OpenLNS CT network design.

An OpenLNS CT drawing is always located on the computer running OpenLNS CT or on a remote file share. When multiple OpenLNS CT users are accessing the same OpenLNS CT drawing, OpenLNS CT automatically keeps the drawing synchronized. Multiple-user operation is described in *Automatic OpenLNS CT Drawing Synchronization* in Chapter 7.

LONWORKS Basics

This section provides an overview of LONWORKS networks and defines related terms. The *Glossary* in Appendix B provides a quick reference for specific terms.

Networks

A LONWORKS network consists of intelligent *devices* (such as sensors, actuators, and controllers) that communicate with each other using a common *protocol* over one or more *channels*. Network devices are sometimes called *nodes*.

Devices

Each *device* includes one or more processors and a *transceiver*. The processor(s) provide the device’s intelligence and implement the ISO/IEC 14908-1 Control Network Protocol (CNP). The transceiver serves as the device’s electrical interface to the communications channel.

Protocol

A device publishes and consumes information as instructed by the application that it is running. The applications on different devices are not synchronized, and it is possible that multiple devices may all try to talk at the same time. Meaningful transfer of information between devices on a network, therefore, requires organization in the form of a set of rules and procedures. These rules and procedures are the *communication protocol*, which may be referred to simply as the *protocol*. The protocol defines the format of the messages being transmitted between devices and defines the actions expected when one device sends a message to another. The protocol normally takes the form of embedded software or firmware code in each device on the network. The CNP is an open protocol defined by the ISO/IEC 14908-1 standard (defined nationally in the United States, Europe, and China by the ANSI/EIA 709.1, EN 14908, and GB/Z 20177 standards, respectively).

Device Templates

A *device template* contains all the attributes of a given device type, such as its functional blocks, network variables, and configuration properties.

Channels

A *channel* is the physical media between devices upon which the devices communicate. The Control Network Protocol is media independent; therefore, numerous types of media can be used for channels such as twisted pair, power line, fiber optics, IP, and radio frequency (RF). Channels are categorized into *channel types,* and the channel types are characterized by the device transceiver. Common channel types include TP/FT-10 (ISO/IEC 14908-2 twisted pair free topology channel), TP/XF-1250 (high-speed twisted pair channel), PL-20 (ISO/IEC 14908-3 power line channel), FO-20 (ANSI/CEA-709.4 fiber optics channel), and IP-852 (ISO/IEC 14908-4 IP-communication).

Different transceivers may be able to interoperate on the same channel; therefore, each transceiver type specifies the channel type or types that it supports. The choice of channel type affects transmission speed and distance as well as the network topology.

Routers

Multiple channels can be connected using *routers*. Routers are used to manage network message traffic, extend the physical size of a channel (both length and number of devices attached), and connect channels that use different media (channel types) together. Unlike other devices, routers are always attached to at least two channels.

Applications

Every LONWORKS device contains an *application* that defines the device's behavior. The application defines the inputs and outputs of the device. The inputs to a device can include information sent on LONWORKS channels from other devices, as well as information from the device hardware (for example, the temperature from a temperature sensing device). The outputs from a device can include information sent on LONWORKS channels to other devices, as well as commands sent to the device hardware (for example, a fan, light, heater, or actuator).

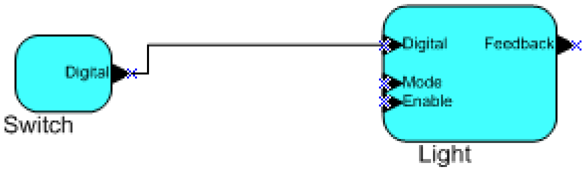
The application may be in the device when you purchase it, or you may load it into the device from application files (.**nld**, .**apb**, and .**nxe** extensions) using OpenLNS CT.

Applications in devices are divided into one or more *functional blocks*. A functional block is a collection of *network variables* and *configuration properties* that are used together to perform one task. For example, a four-port digital input module may have four digital input functional blocks that contain the configuration properties and output network variable for each of the four hardware digital inputs on the device.

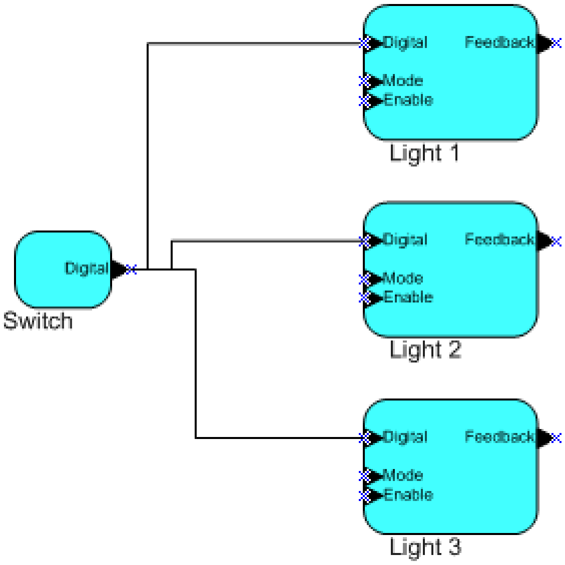
Network Variables

Applications exchange information with other LONWORKS devices using *network variables*. Every network variable has a *direction*, *type*, and *length*. The network variable direction can be either input or output, depending on whether the network variable is used to receive or send data. The network variable type determines the format of the data.

Network variables of identical type and length but opposite directions can be connected to allow the devices to share information. For example, an application on a lighting device could have an input network variable based on the **SNVT_switch** type, while an application on a dimmer-switch device could have an output network variable of the same **SNVT_switch** type. A network management tool such as OpenLNS CT could be used to connect these two devices, allowing the switch to control the lighting device, as shown in the following figure:



A single network variable may be connected to multiple network variables of the same type but opposite direction. The following example shows the same switch being used to control three lights:



The application program in a device does not need to know where input network variable values come from or where output network variable values go. When the application program has a changed value for an output network variable, it simply assigns the new value to the output network variable.

Through a process called *binding* that takes place during network design and installation, the device is configured to know the logical address of the other device or group of devices in the network expecting that network variable's values. The device's embedded firmware assembles and sends the appropriate packet(s) to these destinations. Similarly, when the device receives an updated value for an input network variable required by its application program, its firmware passes the data to the application program. The binding process thus creates logical *connections* between an output network variable in one device and an input network variable in another device or group of devices.

Connections may be thought of as virtual wires. For example, the dimmer-switch device in the dimmer-switch-light example could be replaced with an occupancy sensor, without making any changes to the lighting device.

Configuration Properties

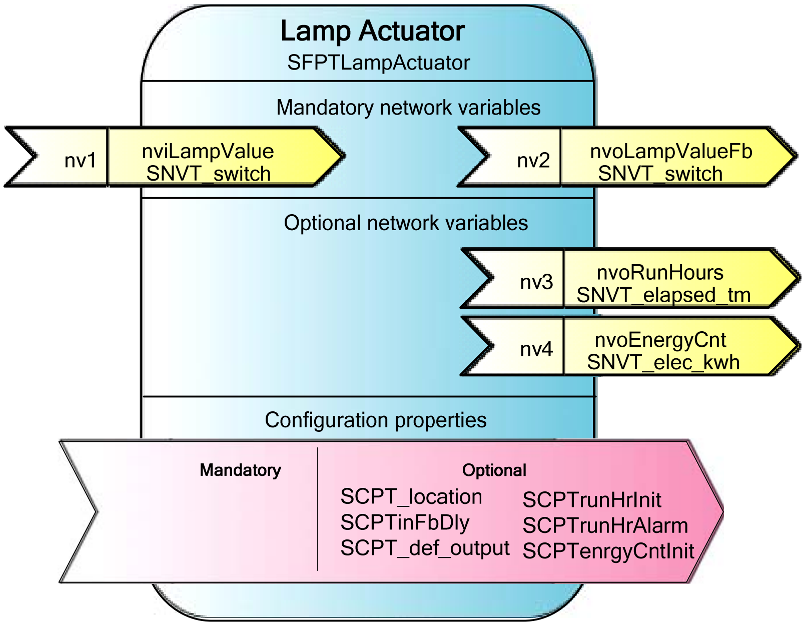
Configuration properties define how an application device behaves by determining the manner in which data is manipulated and when it is transmitted. Configuration properties determine the functions to be performed on the values stored in the network variables. For example, a configuration property may specify a minimum change that must occur on a physical input to a device before the corresponding output network variable is updated. Like network variables, configuration properties have types that determine the type and format of the data they contain.

Functional Blocks

Applications in devices are divided into one or more *functional blocks*. A functional block is a collection of network variables and configuration properties that work together to perform a single task. These network variables and configuration properties are called the *functional block members*. For example, a digital input device could have four digital input functional blocks that contain the configuration properties and output network variable members for each of the four hardware digital inputs on the device. A functional block is an implementation of a functional profile.

Functional Profiles

A *functional profile* defines mandatory and optional network variable and configuration property members for a type of functional block. For example, the standard functional profile for a light actuator has mandatory **SNVT_switch** input and output network variables, optional **SNVT_elapsed_tm** and **SNVT_elec_kwh** output network variables, and a number of optional configuration properties. The following diagram illustrates the components of the standard light actuator functional profile:



When a functional block is created from a functional profile, the application designer can determine which of the optional configuration properties and network variables to implement.

Standard Network Variable and Configuration Property Types

Every network variable and configuration property has a type, which determines the content and structure of the data. To enable devices from different manufacturers to interoperate, the following standard types are defined:

- *Standard Network Variable Types (SNVTs, pronounced snivits).* SNVTs contain many common operational data types. For example, **SNVT_temp_p** is a network variable type for network variables containing temperature as a fixed-point number.
- *Standard Configuration Property Types (SCPTs, pronounced skipits).* SCPTs contain many common configuration data types. For example, **SCPT_location** is a configuration property type for configuration properties containing the device location as a text string.

  See *types.lonmark.org* for a list and description of all SNVTs and SCPTs.

User-defined Standard Network Variable and Configuration Property Types

Applications may use manufacturer-defined non-standard types—user network variable types (UNVTs) and user configuration property types (UCPTs)—which are defined in user resource files. Device manufacturers may provide additional resource files that define these types. See the *NodeBuilder Resource Editor User's Guide* for additional information on creating or using these files.

Subsystems

Devices, routers, and functional blocks are contained in *subsystems*. With OpenLNS CT, each subsystem corresponds to one page within an OpenLNS CT drawing. Subsystems allow you to place devices, routers, and functional blocks onto separate pages for organizational purposes. You may also nest subsystems in other subsystems, allowing you to create a subsystem hierarchy for large networks. For example, a network may consist of HVAC, lighting, security, and operator subsystems. These may be further divided into subsystems for each floor, and each floor divided into subsystems for each room.

Supernodes

Using OpenLNS CT, you can also use subsystems to create *supernodes*. A supernode is a subsystem with its own network variable interface. You can use supernodes to organize groups of devices into logical units and to hide complex subsystem details, exposing only the most important network variables. This structure reduces errors and decreases the time required for network engineering and commissioning. A network variable interface for a supernode may contain any network variable on any device functional block found within the supernode or in any of its nested subsystems.

Visio Basics

OpenLNS CT is built on the Visio drawing tool to provide a robust technical drawing environment for network design. An OpenLNS CT drawing consists of one or more drawing files, and each drawing file contains multiple subsystems that are each displayed on individual Visio pages. The Visio documentation provides detailed descriptions of Visio commands and capabilities.

Two key Visio concepts are *shapes* and *stencils*. Shapes are reusable drawing objects. A shape may represent a simple drawing object such as a line, arc, circle, or square, or it may represent a complex drawing object with special behavior such as OpenLNS CT SmartShapes®. OpenLNS includes SmartShapes for subsystems, application devices, functional blocks, network variables, message tags, connections, routers, and channels.

To simplify finding and reusing shapes, Visio defines a special type of drawing called a *stencil*. A shape contained on a stencil is called a *master shape*. When you drag a master shape from a stencil to one of your drawing pages, Visio automatically makes a copy of the master shape on your drawing and leaves the master shape unchanged on the stencil.

You can create *custom master shapes* and stencils for any set of OpenLNS CT SmartShapes. For example, OpenLNS CT includes custom master SmartShapes and a custom stencil for SmartServer devices and functional blocks. You may wish to create your own custom master SmartShapes and stencils to speed up network design. See Chapter 12, *Creating and Using Custom OpenLNS CT Master SmartShapes*, for more information.

To simplify access to your most commonly used stencils and drawing options, Visio defines another type of drawing called a *template*. Templates are drawings that may be used as the starting point when you create a new drawing. OpenLNS CT includes a LONWORKS template that is used automatically when you create an OpenLNS CT drawing. If you wish, you can create your own custom OpenLNS CT template and select it as the default template in the *OpenLNS CT Design Manager*. See *New Network Options* in Chapter 3 for more information on selecting the default drawing template.

2

Installing and Activating OpenLNS CT

This chapter describes how to order OpenLNS CT, install the various software and components included in your copy of OpenLNS CT, and how to activate OpenLNS CT.

Ordering OpenLNS CT

You can order OpenLNS CT directly from Echelon, from an Echelon distributor, or from the Echelon store at *store.echelon.com*. If you order from the Echelon store, make sure you provide a valid e-mail address when you order OpenLNS CT products. Once you complete your order, Echelon will send you an e-mail message to the provided e-mail address with the activation key for your OpenLNS CT products. You can order the following products from the Echelon store:

Echelon Part No.	**OpenLNS CT Product**
38000-400	OpenLNS Commissioning Tool Professional Edition (DVD)
38010-400	OpenLNS Commissioning Tool Professional Edition Without Visio (DVD)
38050-400	OpenLNS Commissioning Tool Professional Edition Activation Key
38060-400	OpenLNS Commissioning Tool Professional Edition Without Visio Activation Key
38100-400	OpenLNS Commissioning Tool Standard Edition (DVD)
38150-400	OpenLNS Commissioning Tool Standard Edition Activation Key
38160-400	OpenLNS Commissioning Tool Standard Edition Without Visio Activation Key
38300-400	OpenLNS Commissioning Tool Trial Edition Without Visio Activation Key

You can download your OpenLNS CT products from the Echelon Web site at *www.echelon.com/openlns*, and then install and activate them as described in the next section.

Installing and Activating OpenLNS CT

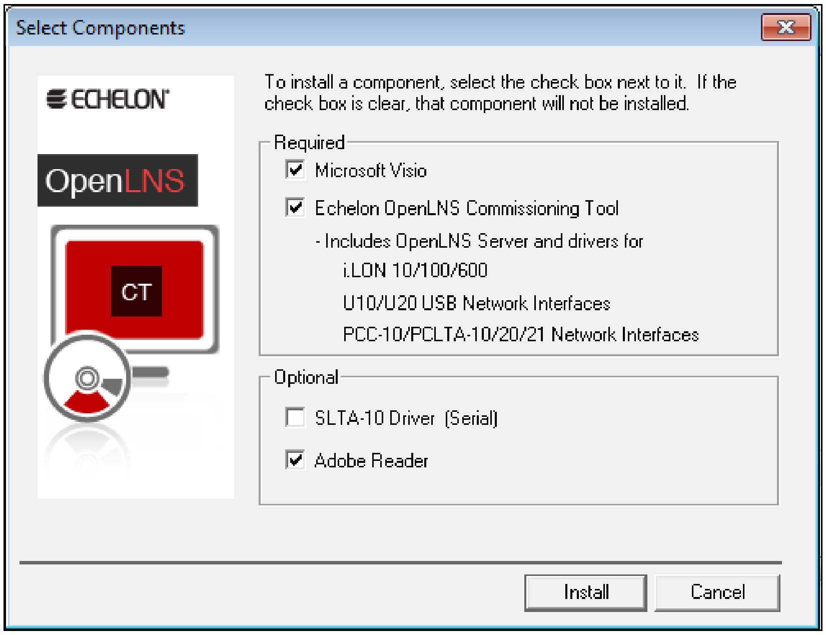
You can install a licensed copy of OpenLNS CT on up to two computers, a primary and a secondary computer, provided that the following requirements are met: only the OpenLNS CT user on the primary computer can use the copy of OpenLNS CT on the secondary computer and the OpenLNS CT software may only be used on one computer at a time. See the *OpenLNS CT Software License Agreement* in Appendix C for more information. To begin your installation, follow these steps:

1. If you are installing a version of OpenLNS CT that does not include Visio, you must separately purchase and install the 32-bit version of Visio 2010 or Visio 2003 before installing the OpenLNS CT software.

2. If you do not have an Echelon download account, create one on the Echelon Web site at *www.echelon.com/support/downloads/accounts*. Your account will enable you to download OpenLNS software from the Echelon Web site; make activating and managing licenses for multiple Echelon products easier; enable you to return, transfer, and recover licenses; and help you receive technical support from Echelon faster.

3. Download OpenLNS CT from the Echelon Web site and run the **OpenLNSCTSetup.exe** file, or insert the OpenLNS CT DVD into your DVD-ROM drive.

   If you are installing OpenLNS CT from a DVD and the OpenLNS CT setup application does not launch immediately, click **Start** on the taskbar and then and click **Run**. Browse to the **OpenLNSCTSetup** application on the OpenLNS CT DVD and click **Open**. The **Select Components** dialog box opens.



Select Components

To install a component, select the check box next to it. If the check box is clear, that component will not be installed.

Required

☑ Microsoft Visio

☑ Echelon OpenLNS Commissioning Tool

- Includes OpenLNS Server and drivers for
  i.LON 10/100/600
  U10/U20 USB Network Interfaces
  PCC-10/PCLTA-10/20/21 Network Interfaces

Optional

☐ SLTA-10 Driver (Serial)

☑ Adobe Reader

4. Select the components that you wish to install by selecting the appropriate check boxes on the **Select Components** dialog box.

   - **Microsoft Visio:** Installs the 32-bit version of the Visio 2010 drawing tool. To install the 32-bit version of Visio 2010, your computer cannot have the 64-bit version of any Office 2010 application installed on it.

     OpenLNS CT is not compatible with the 64-bit version of Visio 2010 (even if your computer is using a 64-bit operating system), and it is not compatible with any version of Visio prior to Visio 2003. If you already have a 64-bit version of any Office 2010 application, you must uninstall it before installing the 32-bit version of Visio. You can reinstall the 32-bit version of any 64-bit Office software that you have installed.

   - **Echelon OpenLNS Commissioning Tool:** Installs the required components of OpenLNS CT. Installing OpenLNS CT automatically installs OpenLNS Server, the OpenLNS CT XML Utility, and the drivers for the following OpenLDV 4.0-compatible network interfaces:

     - Remote Network Interfaces (RNIs): SmartServer , i.LON 100 Internet Server, i.LON 600 IP-852 Router, and i.LON 10 Ethernet Adapter
     - Local USB network interfaces: U10/U20 USB network interfaces.
     - Local PC card network interfaces: PCC-10 and PCLTA-10, 20, and 21 PCI network interfaces. The PCC/PCLTA and SLTA-10 network drivers are compatible with 32-bit versions of Windows only and are not installed on 64-bit versions of Windows.

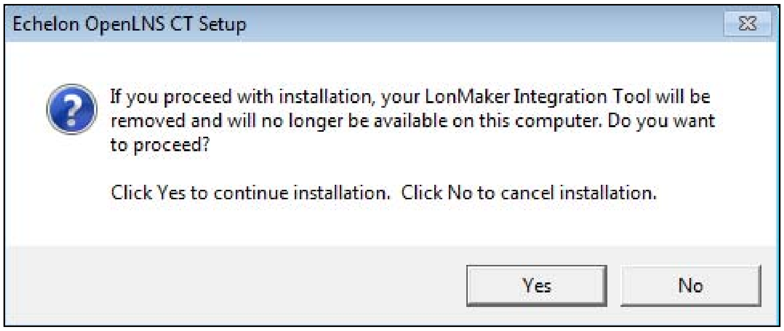
     If you are using OpenLNS CT on an IP-852 channel, you will need an IP network interface card (such as Ethernet or WiFi) or modem with PPP software. An IP network interface is also required if you are using an IP-852 interface such as a SmartServer (with IP-852 routing option), i.LON 100 Internet Server (with IP-852 routing option), or i.LON 600 IP-852 Router. If you are using another OpenLNS interface, install the network interface hardware and driver following the instructions provided with the hardware.

     Most network interfaces include an application that handles network interface configuration. You must perform this configuration before you can use OpenLNS CT while attached to a network. The application may handle host details such as IRQ assignment, and it may also manage network parameters such as buffer counts and transceiver types.

You can use the **LONWORKS Interfaces** application in the Control Panel to define, configure, and test RNIs, IP-852 network interfaces, USB 10/20 network interfaces, and PCC/PCLTA network interfaces.

- **SLTA-10 Driver (Serial):** Installs the network driver for the SLTA-10 serial network interface. Click this check box if you are using an SLTA-10 serial adapter as your network interface. This option is not available on 64-bit versions of Windows.
- **Adobe Reader:** Installs Adobe Reader 9.2, which is a free program that allows you to read Adobe Acrobat portable document files (**.pdf** extension). Many of the LONMARK reference documents are exclusively in PDF format. You can clear this check box if you already have a PDF reader application.

5. Click **Install**. If you are upgrading from the LonMaker Integration Tool, a message opens informing you that installing OpenLNS CT will remove any version of the LonMaker tool on your computer.

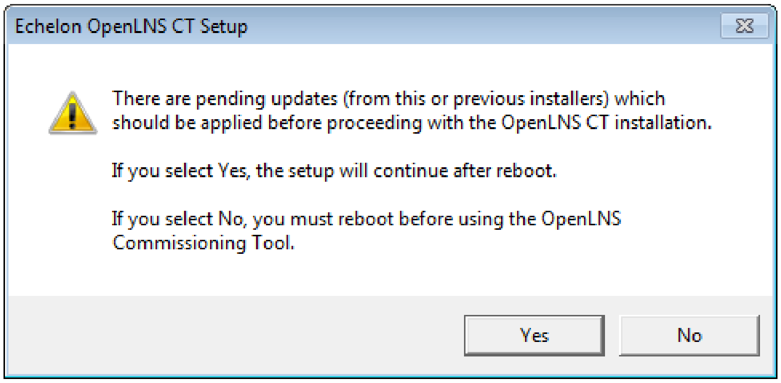


Echelon OpenLNS CT Setup

If you proceed with installation, your LonMaker Integration Tool will be removed and will no longer be available on this computer. Do you want to proceed?

Click Yes to continue installation. Click No to cancel installation.

6. Click **Yes** to continue installing OpenLNS CT. The installation program installs the selected components sequentially, prompting you for required reboots.



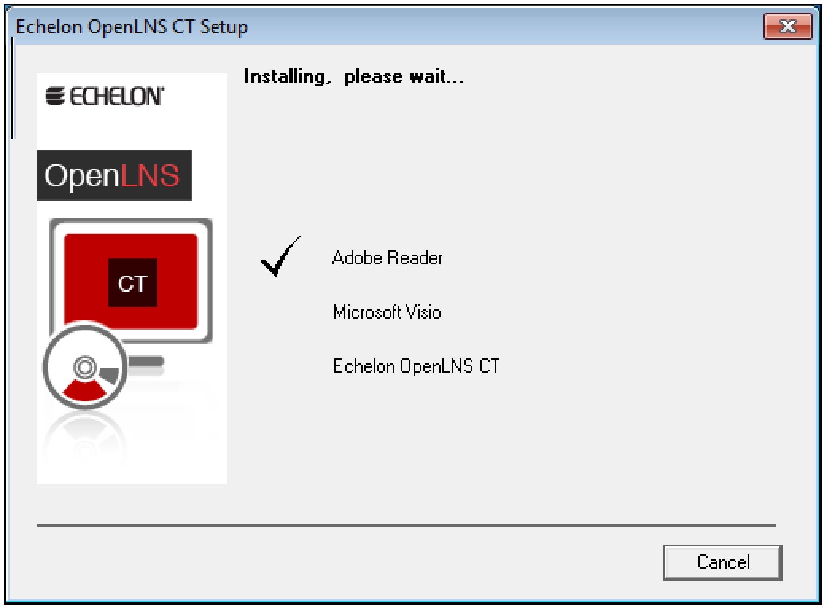
Echelon OpenLNS CT Setup

There are pending updates (from this or previous installers) which should be applied before proceeding with the OpenLNS CT installation.

If you select Yes, the setup will continue after reboot.

If you select No, you must reboot before using the OpenLNS Commissioning Tool.

7. During the installation, the following dialog opens, displaying the current progress of the OpenLNS CT setup:



Echelon OpenLNS CT Setup

Installing, please wait...

✓ Adobe Reader

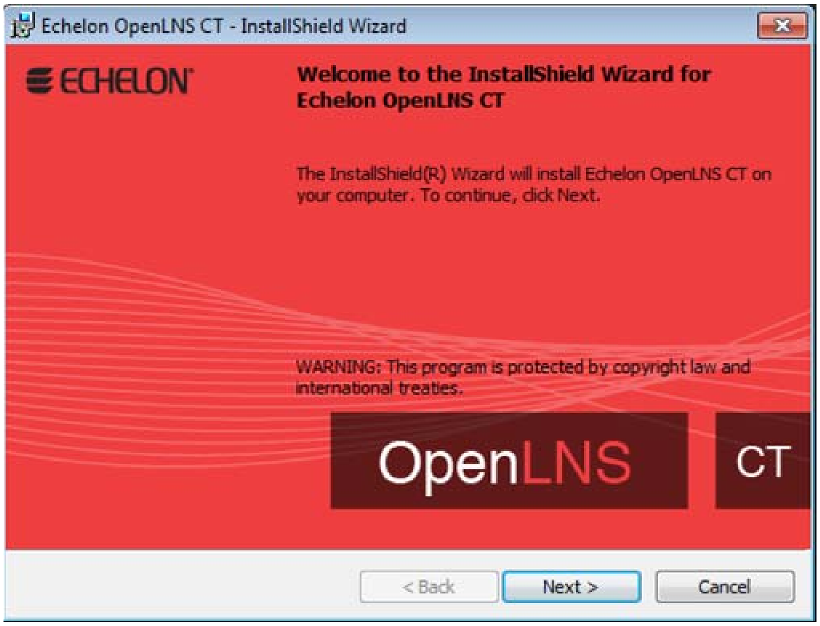
Microsoft Visio

Echelon OpenLNS CT

If you selected the Visio component in the **Select Components** window, the Microsoft Visio setup wizard opens. Follow the instructions in the installation program to complete the Visio setup.

8. Install OpenLNS CT following these steps:

   a. The OpenLNS CT setup program begins automatically with the Welcome window opening.



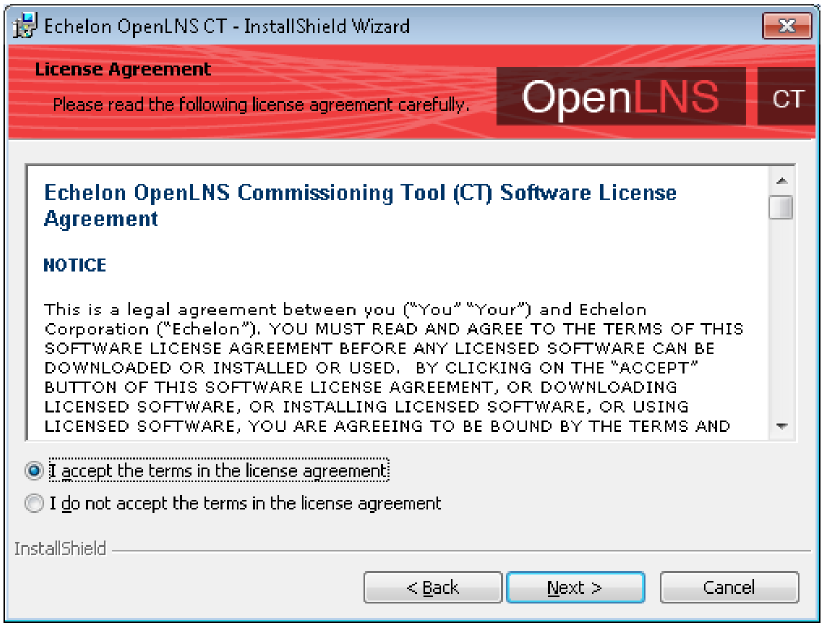
Echelon OpenLNS CT - InstallShield Wizard

Welcome to the InstallShield Wizard for Echelon OpenLNS CT

The InstallShield(R) Wizard will install Echelon OpenLNS CT on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

   b. Click **Next**. The License Agreement window opens.



License Agreement

Please read the following license agreement carefully.

Echelon OpenLNS Commissioning Tool (CT) Software License Agreement

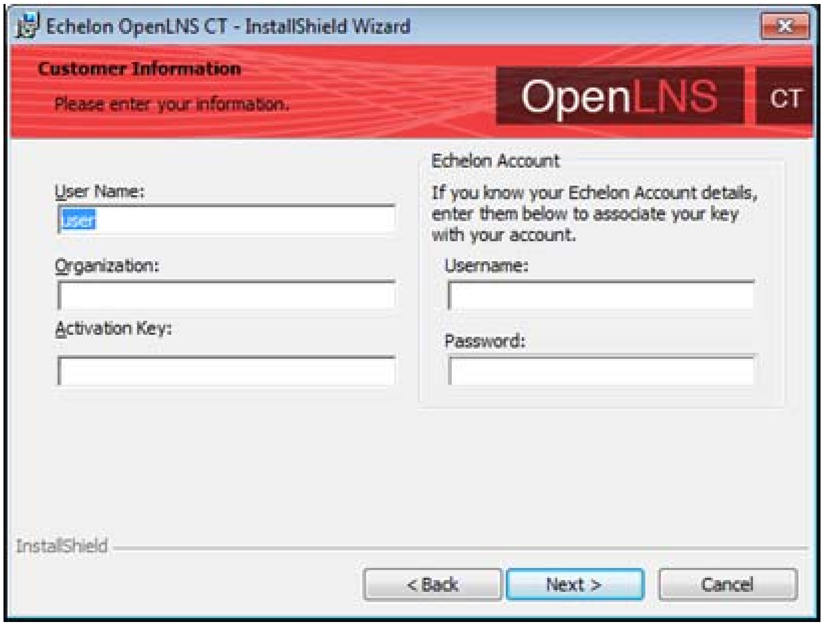
NOTICE

This is a legal agreement between you ("You" "Your") and Echelon Corporation ("Echelon"). YOU MUST READ AND AGREE TO THE TERMS OF THIS SOFTWARE LICENSE AGREEMENT BEFORE ANY LICENSED SOFTWARE CAN BE DOWNLOADED OR INSTALLED OR USED. BY CLICKING ON THE "ACCEPT" BUTTON OF THIS SOFTWARE LICENSE AGREEMENT, OR DOWNLOADING LICENSED SOFTWARE, OR INSTALLING LICENSED SOFTWARE, OR USING LICENSED SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE TERMS AND

● I accept the terms in the license agreement

○ I do not accept the terms in the license agreement

c. Read the license agreement and click **Yes** if you agree with the terms. The Customer Information window opens.



Customer Information

Please enter your information.

User Name: User

Organization:

Activation Key:

Echelon Account

If you know your Echelon Account details, enter them below to associate your key with your account.

Username:

Password:

d. Enter your name, organization, and the OpenLNS CT activation key in the appropriate fields. The name and organization may be entered automatically based on the user currently logged on and whether other Echelon products are installed on your computer.

If you renewed your annual maintenance and you are installing your upgrade on the same computer as your original OpenLNS CT software, your activation key is automatically displayed. If you are installing an upgrade on a different computer, you must find the activation key that you saved from your original version of OpenLNS CT.

Under **Echelon Account**, enter the user name (e-mail address) and password you created for your Echelon download account (see step 2 if you do not have an Echelon download account).

Click **Next**.

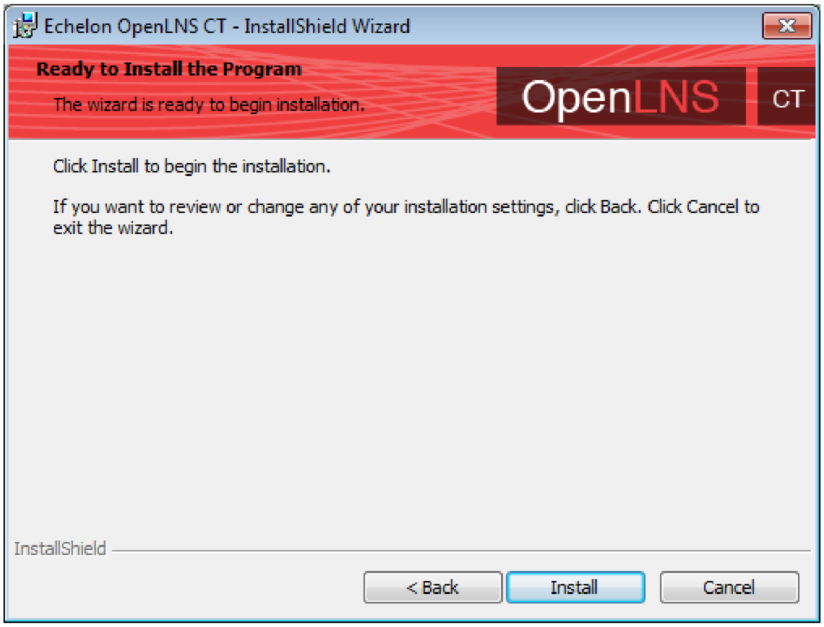
e. If your computer is connected to the Internet, the OpenLNS CT installer connects to the Echelon License Server, the License Server issues activation licenses for OpenLNS Server and OpenLNS CT, and the installer stores the licenses on your computer.

   If your computer is not attached to the Internet, you can use the Echelon License Wizard to manually activate OpenLNS CT via e-mail or phone after installing the software (e-mail is recommended). See the next section, *Manually Activating OpenLNS CT*, for more information.

f. If the Destination Location window opens, choose a LONWORKS folder in which you want the OpenLNS CT software installed, and then click **Next**. The OpenLNS CT software will be installed in an OpenLNS CT folder below the LONWORKS folder that you specify. If you previously installed other Echelon software on this computer, use the same LONWORKS folder.

   The default LONWORKS folder is **C:\Program Files\LonWorks** (on a 32-bit version of Windows) or **C:\Program Files (x86)\LonWorks** (on a 64-bit version of Windows). If you previously installed other Echelon software on this computer, the last LONWORKS folder that you selected will be displayed as the default, and you cannot change this value.

g. The Ready to Install window opens. Click **Install** to begin the OpenLNS CT installation.



Echelon OpenLNS CT - InstallShield Wizard

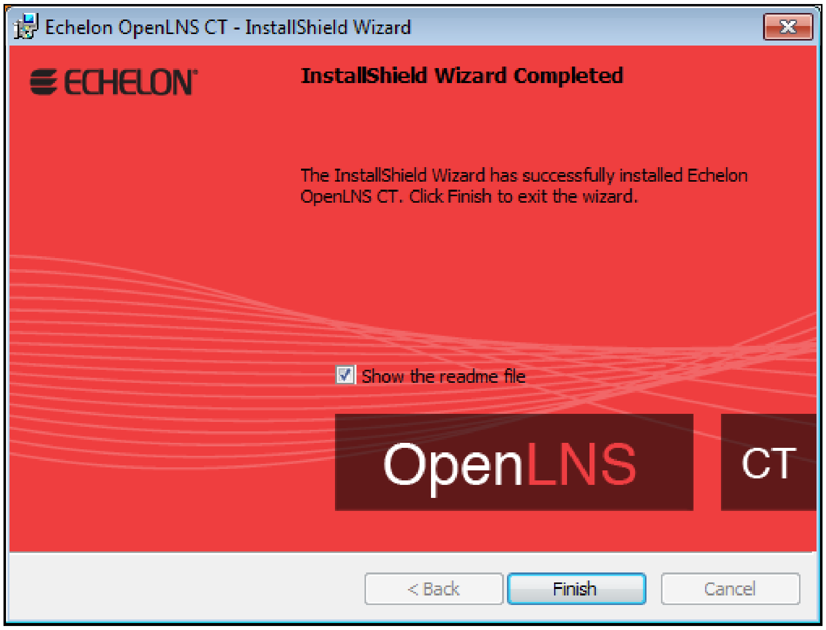
Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

h. If you need to reboot your computer, a window will appear prompting you to select to reboot your computer now or later. After OpenLNS CT installation and any required reboots have been completed, a window will appear, stating that OpenLNS CT has been installed successfully and providing you with the option to view the ReadMe file. The ReadMe file contains information that is not included in this user’s guide.


Echelon OpenLNS CT - InstallShield Wizard

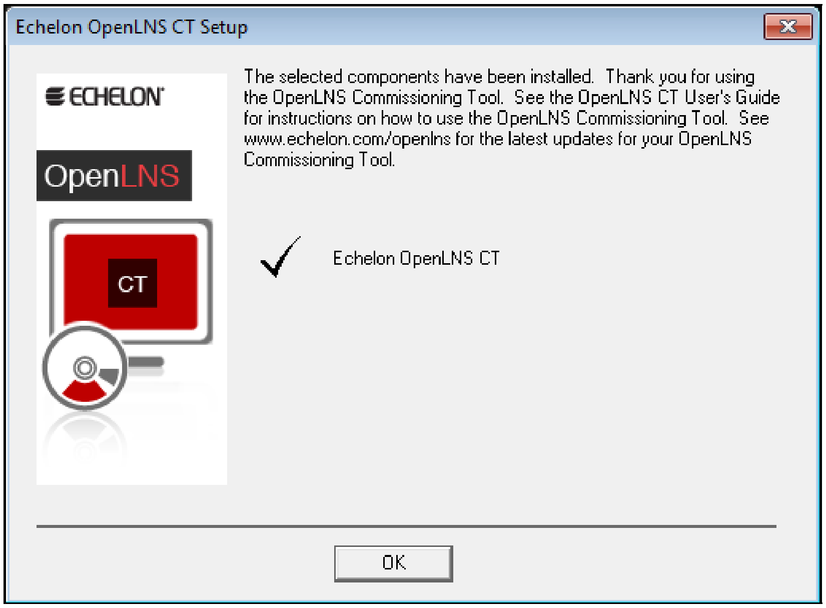
InstallShield Wizard Completed

The InstallShield Wizard has successfully installed Echelon OpenLNS CT. Click Finish to exit the wizard.

☑ Show the readme file

i. Click **Finish**. The ReadMe file opens if the **Show the Readme File** check box was selected. When you finish reading the ReadMe file, close the window.

j. A confirmation dialog opens stating that the selected components have been installed. Click **OK**.


Echelon OpenLNS CT Setup

The selected components have been installed. Thank you for using the OpenLNS Commissioning Tool. See the OpenLNS CT User's Guide for instructions on how to use the OpenLNS Commissioning Tool. See www.echelon.com/openlns for the latest updates for your OpenLNS Commissioning Tool.

✓ Echelon OpenLNS CT

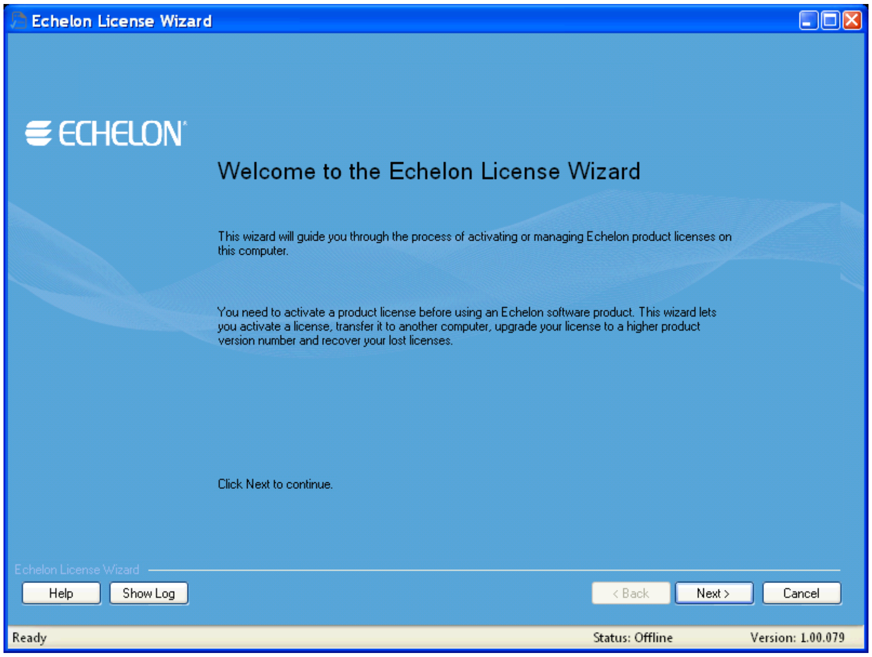
Note: Do not uninstall any of the following applications and components that are used by OpenLNS CT: OpenLNS Server, OpenLNS XML Utility, OpenLDV, Visio, LonMark Resource Files, Multi-Port Router 50, and the IP-852 Configuration Server. Uninstalling any of these applications and components will cause OpenLNS CT to stop running.

9. If your computer was not attached to the Internet when you were installing OpenLNS CT, proceed to the next section to manually activate your software. OpenLNS CT will not run until it is activated.

Manually Activating OpenLNS CT

If your computer was not attached to the Internet when you installed OpenLNS CT, you can use the Echelon License Wizard to activate OpenLNS Server and OpenLNS CT after you complete the software installation. You can activate your software via the Internet if your computer is now online, or you can activate your software via e-mail or phone if your computer is still offline. To use the License Wizard to activate your OpenLNS CT software, follow these steps:

1. Open the Echelon License Wizard. Click **Start**, click **Programs**, point to **Echelon License Wizard**, and then click **License Wizard**. The **Echelon License Wizard** opens. Click **Next**.

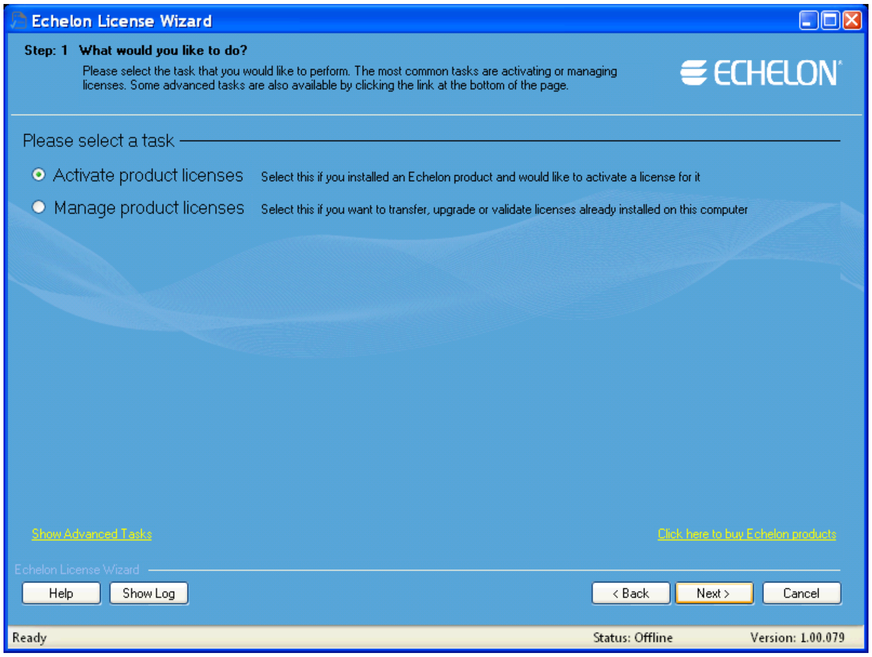


This wizard will guide you through the process of activating or managing Echelon product licenses on this computer.

You need to activate a product license before using an Echelon software product. This wizard lets you activate a license, transfer it to another computer, upgrade your license to a higher product version number and recover your lost licenses.

Click Next to continue.

2. The **Step 1: What would you like to do?** dialog opens.



Step: 1 What would you like to do?

Please select the task that you would like to perform. The most common tasks are activating or managing licenses. Some advanced tasks are also available by clicking the link at the bottom of the page.

Please select a task

● Activate product licenses Select this if you installed an Echelon product and would like to activate a license for it

○ Manage product licenses Select this if you want to transfer, upgrade or validate licenses already installed on this computer

3. To activate your OpenLNS CT software using the over the Internet or via e-mail, click **Activate Product Licenses** and then click **Next**.

   To activate over the phone, click **Show Advanced Tasks**, **Upgrade a Product License over the Phone**, and then follow the instructions in the License Wizard. If your computer is offline, activate via e-mail if you have e-mail access because activating over the phone requires manually entering many lengthy alpha-numeric strings.

4. If your computer is online, the **Step 2: What is Your Account Information** dialog opens. Complete the following steps to activate your software (if your computer is still offline, proceed to step 5):

   a. Enter your account information and/or your OpenLNS CT activation key, and then click **Next**.

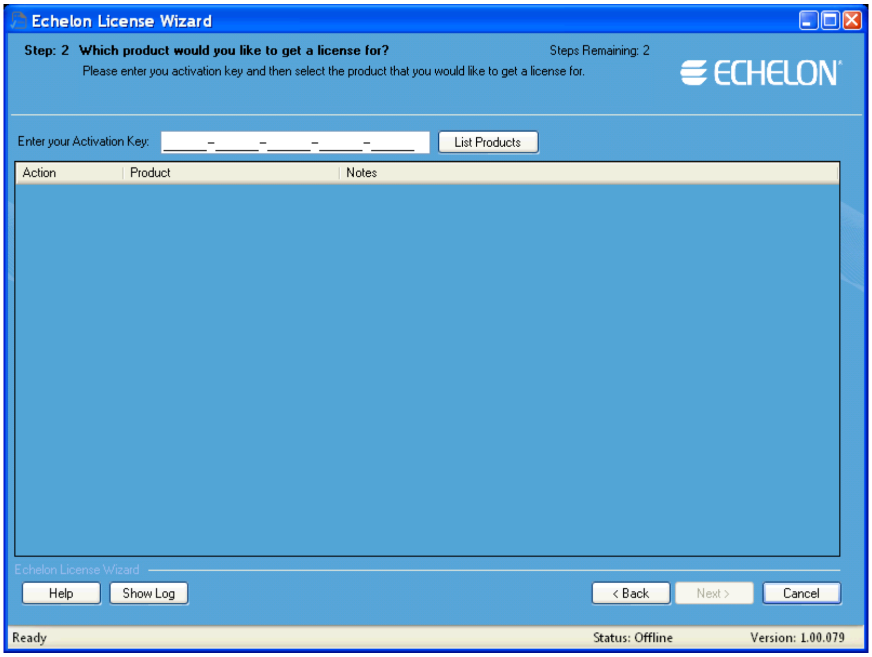
   b. The **Step 3: Which License Would You Like to Activate?** dialog opens.

   c. Select **Activate** from the **Action** list for both OpenLNS Server and OpenLNS CT, and then click **Next**.

   d. The **Step 4: Status** dialog opens, and the License Wizard activates your licenses.

   e. Click **Exit** to close the License Wizard.

5. If your computer is still offline, the **Step 2: Which Product Would You Like to Get a License For** dialog opens.



Echelon License Wizard

Step: 2 Which product would you like to get a license for?

Steps Remaining: 2

Please enter you activation key and then select the product that you would like to get a license for.

Enter your Activation Key: _____-_____-_____-_____-_____

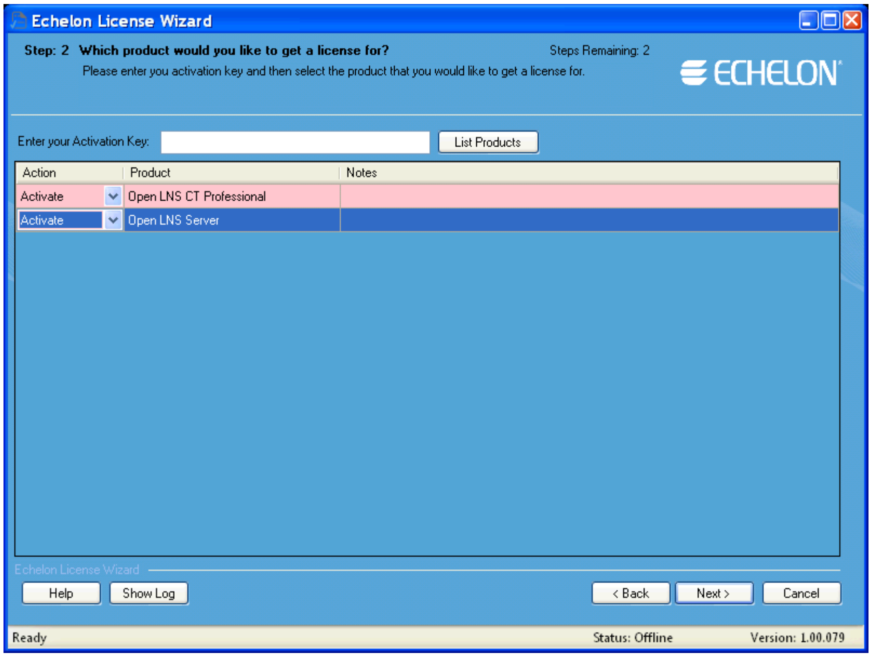
Action	Product	Notes

Ready

Status: Offline

Version: 1.00.079

6. Enter your OpenLNS CT activation key in the **Enter Your Activation Key** box and then click **List Products**. The License Wizard lists your OpenLNS Server and OpenLNS CT products. Select **Activate** from the **Action** list for both OpenLNS Server and OpenLNS CT, and then click **Next**.



Step: 2 Which product would you like to get a license for? Steps Remaining: 2

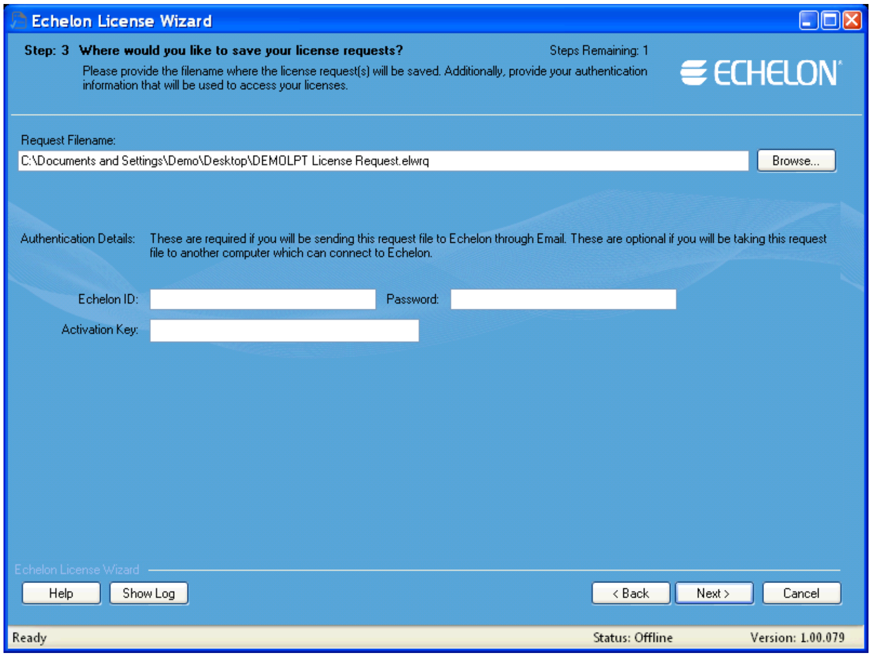
Please enter you activation key and then select the product that you would like to get a license for.

Enter your Activation Key:

Action	Product	Notes
Activate	Open LNS CT Professional	
Activate	Open LNS Server	

Ready Status: Offline Version: 1.00.079

7. The **Step 3: Where Would You Like to Save Your License Requests?** dialog opens.



Step: 3 Where would you like to save your license requests? Steps Remaining: 1

Please provide the filename where the license request(s) will be saved. Additionally, provide your authentication information that will be used to access your licenses.

Request Filename:

C:\Documents and Settings\Demo\Desktop\DEMOLPT License Request.elwrq

Authentication Details: These are required if you will be sending this request file to Echelon through Email. These are optional if you will be taking this request file to another computer which can connect to Echelon.

Echelon ID: Password:

Activation Key:

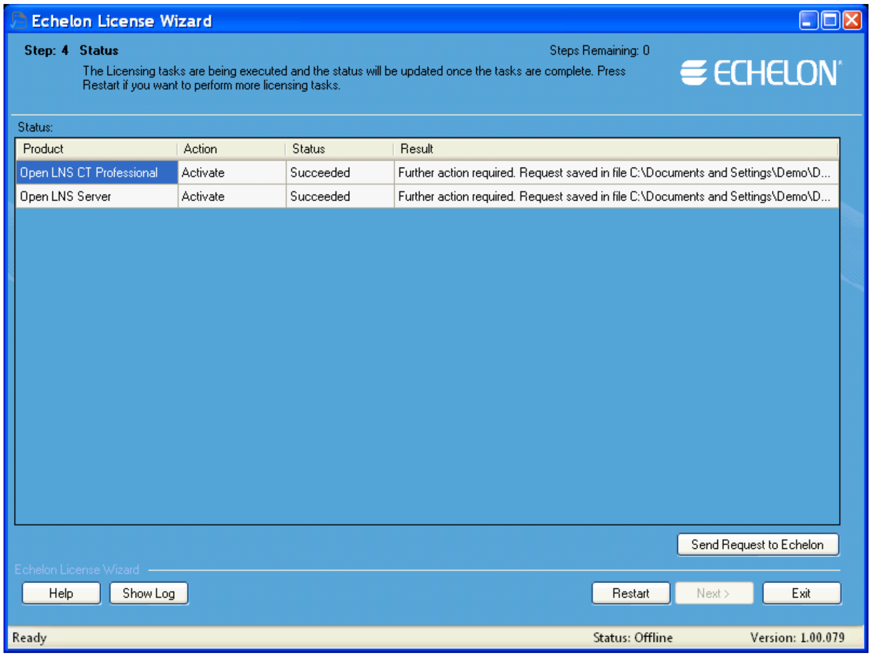
Ready Status: Offline Version: 1.00.079

8. In the **Request File Name** property, enter a full path where a license request file (**.elwrq** extension) is to be saved, or click **Browse**, specify the directory where the license request file is to be saved, and then enter a name for the file.

   Under **Authentication Details**, enter the Echelon ID (e-mail address) and Password you created for your Echelon download account (see step 2 if you do not have an Echelon download account). The **Activation Key** box displays the 25-character alphanumeric string for the Echelon product to be activated. Do not modify this property.

   Click **Next**.

9. The **Status** dialog opens. The License Wizard confirms the creation of the license request file.



Step: 4 Status

Steps Remaining: 0

The Licensing tasks are being executed and the status will be updated once the tasks are complete. Press Restart if you want to perform more licensing tasks.

Status:

Product	Action	Status	Result
Open LNS CT Professional	Activate	Succeeded	Further action required. Request saved in file C:\Documents and Settings\Demo\D...
Open LNS Server	Activate	Succeeded	Further action required. Request saved in file C:\Documents and Settings\Demo\D...

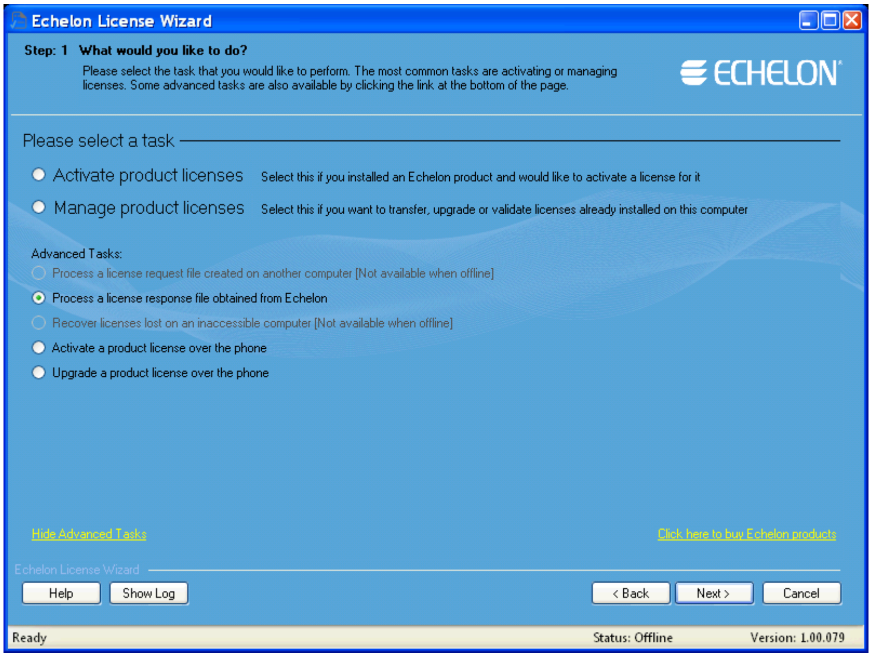
Ready Status: Offline Version: 1.00.079

10. Click **Send Request to Echelon** to e-mail the license request file to Echelon at *activation@echelon.com*. Echelon will send you an e-mail message with a license response (**.elwrs** extension) file. For faster activation, copy the license request file to an Internet-connected computer with the License Wizard installed on it and importing it into the License Wizard. The License Wizard creates the license response file, which you can then copy to the original computer.

11. Import the license response (**.elwrs** extension) file into the License Wizard and activate your OpenLNS Server and OpenLNS CT licenses following these steps:

    a. Start the Echelon License Wizard.

    b. In the **Task Selection** dialog, click the **Show Advanced Topics** link, and then click the **Process a License Response File Obtained from Echelon** option.



Step: 1 What would you like to do?

Please select the task that you would like to perform. The most common tasks are activating or managing licenses. Some advanced tasks are also available by clicking the link at the bottom of the page.

Please select a task

○ Activate product licenses Select this if you installed an Echelon product and would like to activate a license for it

○ Manage product licenses Select this if you want to transfer, upgrade or validate licenses already installed on this computer

Advanced Tasks:

○ Process a license request file created on another computer [Not available when offline]

● Process a license response file obtained from Echelon

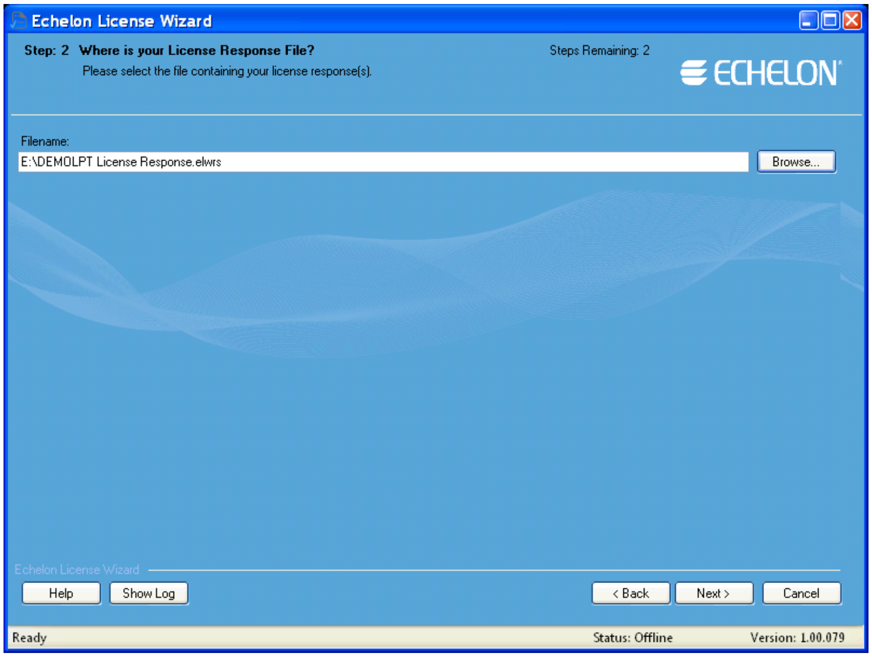
○ Recover licenses lost on an inaccessible computer [Not available when offline]

○ Activate a product license over the phone

○ Upgrade a product license over the phone

Ready Status: Offline Version: 1.00.079

c. The **Import License Request or Response File** dialog opens. In the **File Name** property, enter the full path of the license response file (**.elwrs** extension), or click **Browse** and then select the file. Click **Next**.



Echelon License Wizard

Step: 2 Where is your License Response File?

Please select the file containing your license response(s).

Steps Remaining: 2

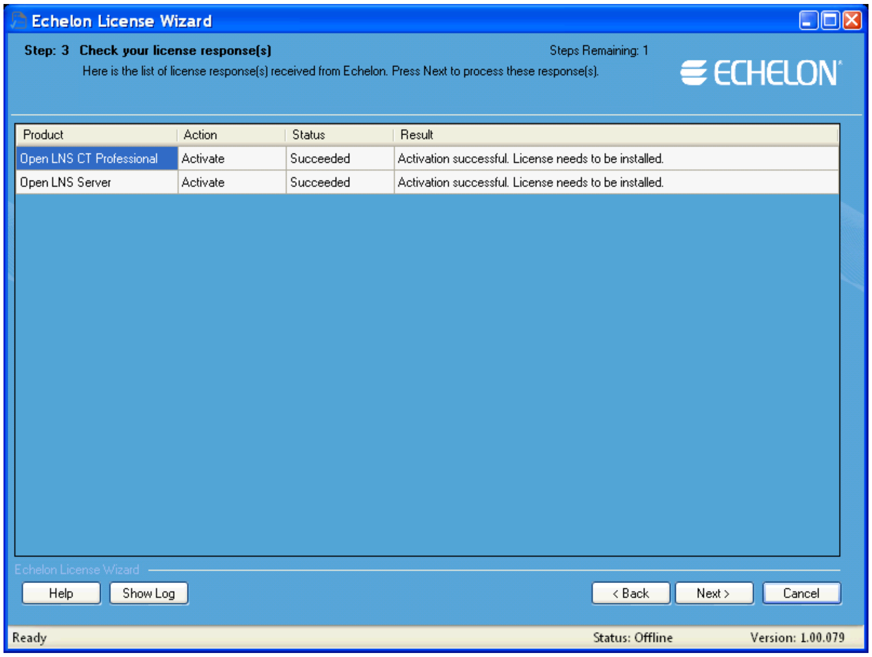
Filename: E:\DEMOLPT License Response.elwrs

Ready

Status: Offline

Version: 1.00.079

d. The **Check License Response File** dialog opens. Confirm that the Echelon product licenses in the license response file are the ones you want to activate. Click **Next**. The License Server installs and activates the licenses in the license response file.



Echelon License Wizard

Step: 3 Check your license response(s)

Here is the list of license response(s) received from Echelon. Press Next to process these response(s).

Steps Remaining: 1

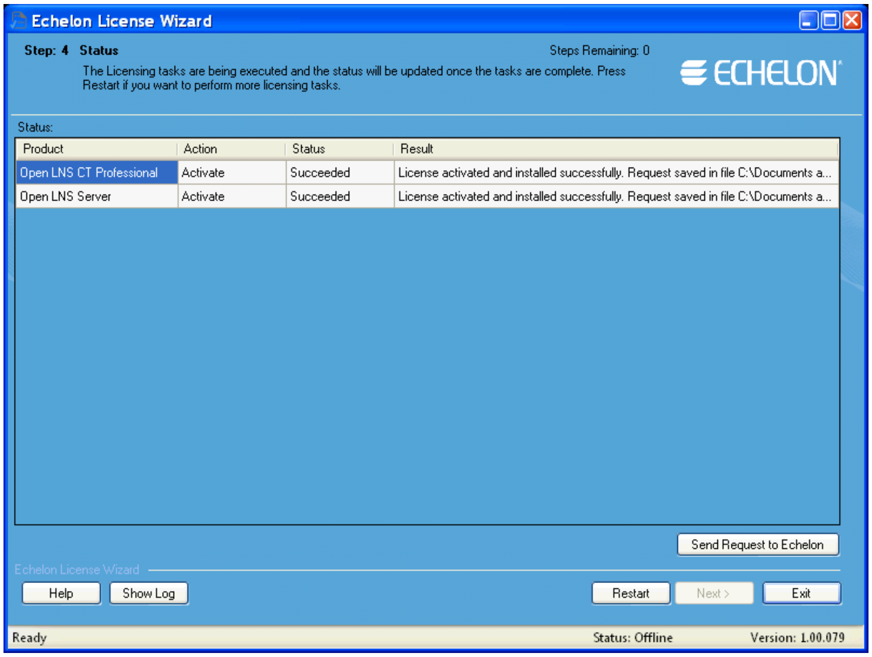
Product	Action	Status	Result
Open LNS CT Professional	Activate	Succeeded	Activation successful. License needs to be installed.
Open LNS Server	Activate	Succeeded	Activation successful. License needs to be installed.

Ready

Status: Offline

Version: 1.00.079

e. The **Status** dialog opens and confirms that the OpenLNS Server and OpenLNS CT licenses have been successfully activated.



Step: 4 Status Steps Remaining: 0

The Licensing tasks are being executed and the status will be updated once the tasks are complete. Press Restart if you want to perform more licensing tasks.

Status:

Product	Action	Status	Result
Open LNS CT Professional	Activate	Succeeded	License activated and installed successfully. Request saved in file C:\Documents a...
Open LNS Server	Activate	Succeeded	License activated and installed successfully. Request saved in file C:\Documents a...

Ready Status: Offline Version: 1.00.079

12. Click **Exit** to close the License Wizard.

3

Getting Started

This chapter describes how to plan and create OpenLNS CT network designs. It describes how to use the OpenLNS CT Design Manager to create, open, and manage OpenLNS CT network designs, and it explains how to set up a network interface that enables OpenLNS CT to communicate with the network. It summarizes the different OpenLNS CT client types, and it explains how to use OpenLNS CT remotely, and how to create and use user profiles.

Design Overview

To prepare to design an OpenLNS CT network, follow these steps:

1. Define network requirements and organization.
2. Select a network installation scenario.
3. Determine user permissions for viewing and changing the network.
4. Optimize network performance.

Defining Network Requirements and Organization

You can define your network's requirements by evaluating which devices (sensors, actuators, and controllers) are needed to achieve the desired network functions and how the devices will be integrated into the network. For example, your network may need a simple fan coil unit, which could require a fan start/stop relay, a cooling coil valve, and a supply air temperature sensor. You could use digital output, analog output, and analog input devices, respectively, to integrate these devices (the analog output module could also includes a PID controller you can use to control the fan coil unit).

Once you define the network requirements, you can figure out how to organize the network in OpenLNS CT. You can help organize the network by answering the following questions:

- How will devices and functional blocks be grouped within subsystems?
- Which functional blocks are needed for the devices to accomplish desired tasks?
- How do functional blocks need to be configured and connected to correctly interpret values from sensors, perform any necessary calculations, and drive actuators?

Selecting a Network Installation Scenario

You can design your entire network without commissioning any devices until after your design is complete, or you can design your network and commission devices at the same time. These two approaches are called *installation scenarios*. The first network installation scenario is called the *engineered system* and the second is the *ad-hoc system*. You can use both scenarios within a single system, or you can use either exclusively. The best scenario for a given network depends on many factors including the skill level of the installer, desired flexibility for the network, and the end-user requirements.

Engineered System Scenario

The *engineered system scenario* allows you to design the network without having access to the physical devices

First, you design the network offsite using OpenLNS CT. You drag OpenLNS CT SmartShapes representing *devices*, *routers*, *IP-852 routers, functional blocks*, and *connections* from an OpenLNS CT stencil to an OpenLNS CT drawing and arrange them on the drawing, and into multiple subsystems if desired.

Once you are onsite and have attached an OpenLNS CT computer to the network, you then commission the physical devices. OpenLNS CT commissions a device by configuring the physical device to match the configuration that you specified in the OpenLNS CT drawing.

Engineered systems must often be modified during on-site installation due to differences between as-drawn plans and the physical network. OpenLNS CT supports on-site changes to the engineered system to allow for these changes during commissioning.

The advantage of the engineered system scenario is that it makes network installation quick, easy, and error-free because most of the time-consuming data entry and processing is done offsite during the design phase. This scenario is often used for building and industrial automation systems, in which the original design closely matches the actual installation.

Ad-Hoc System Scenario

The *ad-hoc system scenario* allows you to design and install the network onsite. In this scenario, OpenLNS CT loads the network configuration information into each device as you define devices, configurations, and connections. This is different from the engineered system scenario in that information is incrementally loaded to the physical devices.

The advantage of the ad-hoc system scenario is its flexibility, as the installer can make decisions onsite. It is most appropriate for simpler systems in which the details of the system to be installed are not known prior to commissioning.

The disadvantages of the ad-hoc system scenario are that it's slower for large systems and requires the same person to do both the network design and installation.

Determining User Permissions

When you create an OpenLNS CT network design, OpenLNS CT does not assign any security to the network drawing. It defaults to one user—the administrator of the OpenLNS CT drawing. While this default is permissible if one person is performing the network installation in a secure environment, it may not be suitable for multi-user or insecure environments. In these environments, you need to create and manage user profiles to control access to your OpenLNS CT network design. This will prevent any unauthorized changes to your OpenLNS CT network design and enable you to better manage the installation or repair of a system.

User profiles allow you to create access controls for a network drawing and its subsystems and set privileges for actions that can be performed on OpenLNS CT objects. For example, you could have a scenario in which a system integrator has access to the entire network and can add or change application devices and routers, while the installers can also access the entire network, but only can commission the application devices and routers.

You might use a different scenario if you are working with multiple system integrators who are all supplying devices for your network. In this case, you can give each integrator access to their subsystem only but allow them to freely change the subsystem drawing and commission devices. This prevents one integrator from accidentally modifying a drawing created by another integrator.

By default, user profiles are not enabled for new networks. To enable user profiles, you must first assign an Administrator password and create new user profiles as described in *User Profiles*. The Administrator can create, modify, and delete user profiles.

Once you logon as a user other than the Administrator, you can still create new user profiles, as described in *Creating a New User Profile*, and you can change your password as described in *Changing Passwords*.

For information on using user profiles with OpenLNS CT remotely, see *Using OpenLNS CT Remotely with User Profiles*.

Optimizing OpenLNS CT Network Performance

Many factors affect the performance of an OpenLNS CT network design. The following are guidelines for several aspects of your OpenLNS CT network design that can help optimize network performance— especially for large networks.

Drawing Files

A single drawing file should not be larger than 3MB, or consist of more than 20 typical pages or 10 complex pages. If you are creating a network exceeding these guidelines, design the subsystems in *multiple drawing files*. For optimum performance, devices and their associated functional blocks should be contained within the same drawing file, as should all members of the same connection.

When designing multiple drawing files, organize the individual drawing files as subsystems within the top-level drawing file, and limit the top-level drawing to contain only those subsystems. This minimizes the number of drawing files that must be opened as you descend the subsystem hierarchy.

Avoid deep subsystem hierarchies with long subsystem names that results in long subsystem paths. Subsystem paths may be stored in devices to help the network recovery process if you lose the network database, but devices have limited memory available to store the network path. Furthermore, visualization tools such as Wonderware's InTouch® have limits to item name lengths that can be exceeded with long subsystem paths.

Network Changes

To improve network performance when making numerous network changes at one time, set OpenLNS CT to OffNet. This will allow you to make the changes quickly, as the changes will be saved in the OpenLNS network database instead of being sent to the physical devices. When you finish making the changes, set OpenLNS CT to OnNet and the changes are propagated to the physical devices.

Connections

If you will be making changes to a subsystem, limit the number of connections monitored by OpenLNS CT to less than five per subsystem. Having more than five monitored connections in a subsystem will increase the time required to perform certain network management operations. Alternatively, you can turn off all monitoring while making changes by right-clicking a clear space in the subsystem page and clicking **Enable Monitoring** if it is currently set. You can turn monitoring on again after making the changes.

If you are using an OpenLNS application that does not support monitor sets (groups of data points that OpenLNS CT uses to monitor and control network variables) and you are making many bound connections to the OpenLNS application, make the bound connections persistent. This will improve network performance. Conversely, non-persistent bound connections, which can be created in an OpenLNS CT network with OpenLNS applications other than OpenLNS CT, can significantly increase the time required to open and close an OpenLNS CT drawing. OpenLNS tools that support monitor sets, such as OpenLNS CT, enable monitoring of monitor points to be setup quickly once an OpenLNS CT network is opened. This is because points in monitor sets are always persistent. OpenLNS CT automatically creates monitor sets for connection monitoring when monitoring is enabled on an OpenLNS CT network. To enable monitoring, right-click a clear space in the subsystem page and clicking **Enable Monitoring** if it is not currently set. For more information on monitor sets, see the *OpenLNS Programmer's Guide*.

Sharing the OpenLNS Interface with the LNS DDE Server

If the LNS DDE Server and OpenLNS CT are running concurrently and sharing the same OpenLNS network interface, limit the polling rate of the LNS DDE server to 10 polls per second on a LONWORKS channel. Having the LNS DDE Server polling above this rate will degrade network performance. By default, the LNS DDE server polls active network variables every 1 second, so more than 10 active network variables running at the default poll interval can impact network performance.

You will see similar performance degradation if you have any other OpenLNS monitoring application sharing the OpenLNS network interface with OpenLNS CT, or if you are using connection monitoring in OpenLNS CT.

If you will be making numerous network changes, either disable the LNS DDE Server and any other OpenLNS monitoring applications, or reduce the polling rate to a maximum of 10 polls per second while you are making changes.

Functional Blocks

If you have a large network with many unconnected network variables on the functional blocks, delete the unused network variable SmartShapes. Network variable SmartShapes add performance overhead, yet they are not required if they are not connected. You can re-add network variable SmartShapes to a functional block if you need to connect them later. To simplify following this guideline, you can create functional block master SmartShapes with your most commonly used network variables. See *Creating a Functional Block Master SmartShape* in Chapter 4, *Designing Networks* for more information.

Subsystems

Subsystems allow you to divide a large network into more manageable parts. For example, in a building control network, one subsystem could contain lighting controls and another temperature controls. Each subsystem is represented by a page of an OpenLNS CT drawing and by a subsystem SmartShape on the drawing page of its parent subsystem. You can nest subsystems like directories on your computer, so the temperature control subsystem could contain other subsystems representing, for example, each floor of the building. You can also create subsystems in separate OpenLNS CT drawings. Using this feature, you can create large networks while maximizing Visio's performance by limiting the number of pages in a single drawing. You can also create a separate OpenLNS CT drawing for each user for large networks with multiple installers.

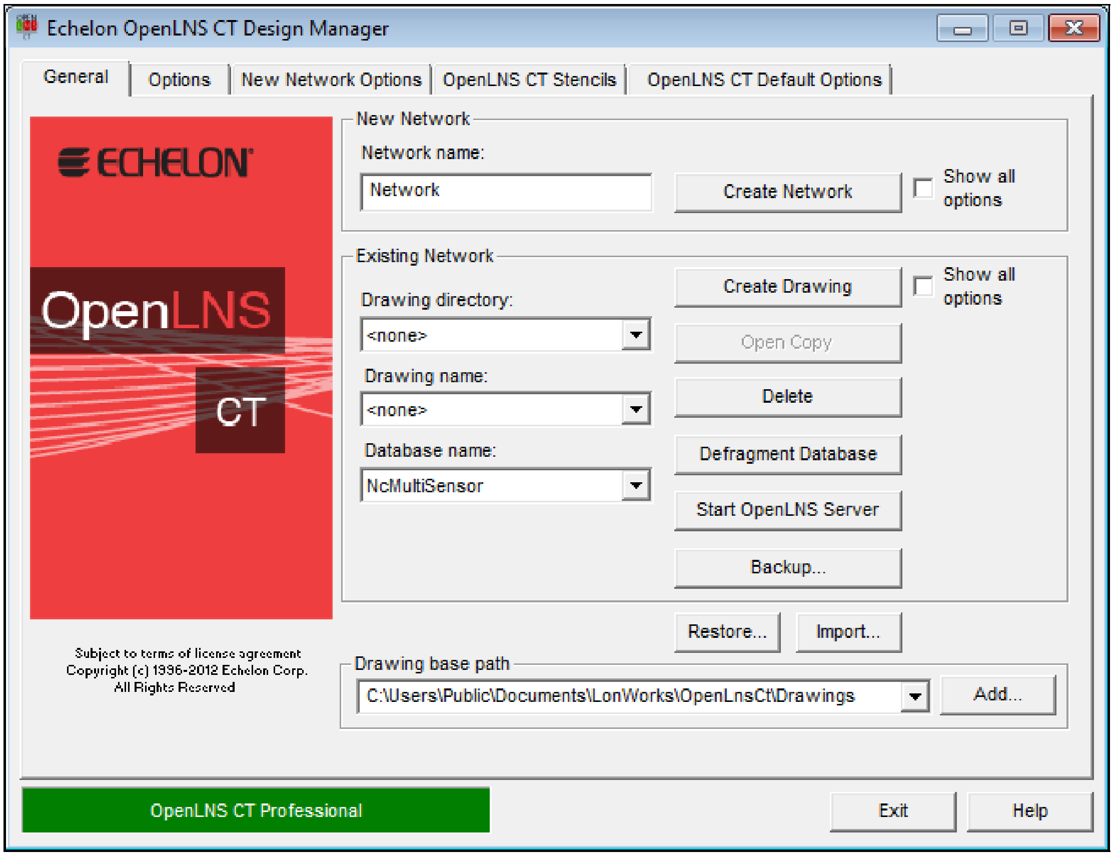
If you are creating a network that needs more subsystems than 20 typical pages or 10 complex pages, you should divide it into two or more drawing files. For optimum performance, devices and their associated functional blocks should be contained within the same drawing file, as should all members of the same connection.

A *supernode* is a special type of subsystem SmartShape that has one or more network variable interfaces on it. Any network variable or message tag on any functional block in the subsystem may be exported to the subsystem SmartShape. This does not create new network variables or message tags, but exports the network variable or message tag's connection point one or more levels up the subsystem hierarchy. Using a supernode allows you to provide a simple interface to an arbitrarily complex subsystem. For example, a supernode that contains lighting controls may have a network variable interface that allows the devices contained within it to be put into emergency override. This allows the integrator to easily identify the critical interfaces into the subsystem, while ignoring those that are typically only used internally by the subsystem.

OpenLNS CT Design Manager Overview

The OpenLNS CT Design Manager is an essential tool for managing your OpenLNS CT networks. With the OpenLNS CT Design Manager, you can create, open, and delete networks; back up and restore networks to prevent against loss; and configure default settings for new networks to speed up the network creation process.

To open the OpenLNS CT Design Manager, click **Start** on the taskbar, point to **Programs**, point to the **Echelon OpenLNS CT** folder, and then click **OpenLNS CT**. The OpenLNS CT Design Manager opens.



New Network

Network name: Network

□ Show all options

Existing Network

□ Show all options

Drawing directory: <none>

Drawing name: <none>

Database name: NcMultiSensor

Drawing base path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Drawings

Subject to terms of license agreement
Copyright (c) 1996-2012 Echelon Corp.
All Rights Reserved

The OpenLNS CT Design Manager consists of five tabs that let you perform the following functions:

- **General**. Create, open, copy, delete, and backup/restore OpenLNS CT network designs; start the OpenLNS Server application; and import and defragment OpenLNS network databases.
- **Options**. Set a prioritized list of languages for LONMARK resource files.
- **New Network Options**. Configure default properties for the pages in the Network Wizard and options to skip the pages, and set the network components.
- **OpenLNS CT Stencils**. Add, remove, or modify the properties of OpenLNS CT stencils.
- **OpenLNS CT Default Options**. Configure the default properties and behavior for OpenLNS CT objects (for example, adding devices, adding functional blocks, and backing up and synchronizing the OpenLNS database).

General

You can use the **General** tab in the OpenLNS CT Design Manager to create, open, copy, delete, backup, restore OpenLNS CT network designs; start the OpenLNS Server application so remote clients can access the OpenLNS network database; import a database and then create a new OpenLNS CT drawing from it; and defragment OpenLNS network databases. To use the **General** tab, enter the following data in the boxes and click the following buttons:

New Network

Network Name	Specifies the name of the new OpenLNS CT network design to be created. The name must be unique to all the OpenLNS CT networks on this computer. You can enter up to 16 characters and include embedded spaces. The name is not case sensitive; therefore, you

cannot create two networks with names that differ only in capitalization.

The initial default network name is “Network 1”. OpenLNS CT automatically increments the index in the default network name so that the network names are unique for subsequent networks you create using the default name. After you create a network with a different name than the initial default, the default network name will become the last network created with an incremented index. For example, if you create a new network named “Test”, the default network name will be “Test 1”. If you create the “Test 1” network, the default network name will become “Test 2”, and so on.

Create Network

Creates and opens a new OpenLNS CT network design consisting of a new OpenLNS network database and a new OpenLNS CT drawing.

Show all options

Displays all the pages in the Network Wizard as the network is opened, regardless if the **Skip this Prompt** option is selected for a page in the **New Network Options** tab.

To create networks with one-click, clear this check box and select the options in the **New Network Options** tab to skip the pages in the Network Wizard.

Existing Network

Drawing Directory

Selects a file from the folder indicated in the **Drawing Base Path** list. To open an existing OpenLNS CT network design, select the folder containing the OpenLNS CT drawing file.

Drawing Name

Selects an OpenLNS CT drawing from all the OpenLNS CT drawings contained in the selected drawing folder. Select the OpenLNS CT drawing you want to open, delete, or backup.

Database Name

Selects an OpenLNS network database from all the OpenLNS network databases on this computer. Select the database you want to defragment, delete, or backup, or from which you want to start the OpenLNS Server. This field is not used for the **Open Network**, **Open Copy**, and **Restore** operations.

Open Network

Opens the OpenLNS CT drawing and associated OpenLNS network database specified in the **Drawing Name** list. The **Database Name** list is ignored by this operation.

Open Copy

Opens a copy of the OpenLNS CT drawing specified in the **Drawing Name** list, making a copy of the OpenLNS CT drawings and the associated OpenLNS network database. This copy is created like a new network. (You will be prompted to choose a new top-level drawing name and database path. The **Database Name** list is ignored by this operation.)

Delete

Deletes the OpenLNS CT drawing and OpenLNS network database specified in the **Drawing Name** and **Database Name** lists, respectively. To delete only a drawing or a database, select **<none>** in the appropriate list.

Defragment Database

Defragments and recreates the index for the OpenLNS CT database specified in the **Database Name** list. This can reduce the size of and decrease the access time to a network database that has grown in size because of many changes. Backup the database before defragmenting it in the event an error occurs during the defragmentation process.

Start OpenLNS Server — Starts the OpenLNS Server application so that remote client computers can access the OpenLNS network database.

Backup — Opens the OpenLNS CT **Backup** dialog, in which you can back up the network database specified in the **Database Name** lists and all of associated drawing files.

See *Manually Backing Up an OpenLNS CT Network* in Chapter 7, *Maintaining Networks*, for more information on how to use the **OpenLNS CT Backup** dialog. See *Backup/Restore Options* in Appendix A for the default backup options you can set.

Show all options — If this check box is selected, all of the network option windows will be presented as the network is opened regardless if the **Skip this Prompt** option had been previously selected.

Restore — Restores an OpenLNS CT drawing and OpenLNS network database from an OpenLNS CT backup file. See *Restoring an OpenLNS CT Network* in Chapter 7, *Maintaining Networks*, for more information. See *Backup/Restore Options* in Appendix A for the default restore options you can set.

Import — Imports an OpenLNS network database (objects.dat file) into OpenLNS so that it opens in the **Database Name** box. The imported database may have an OpenLNS CT drawing associated with it.

Select the folder containing the OpenLNS network database you want to import. If the database does not have an OpenLNS CT drawing associated with it, you can then click **Create Drawing** to create an OpenLNS CT drawing based on the imported database.

Settings

Drawing Base Path — Determines the directory that will be used by the **Drawing Directory** list. All of the folders in this directory will be listed in the **Drawing Directory** list. You can browse through the folders to find the desired path or enter the pathname.

Add — Lets you select a drawing folder to add to the **Drawing Base Path** list.

Exit — Closes the OpenLNS CT Design Manager. This does not close Visio or any open networks.

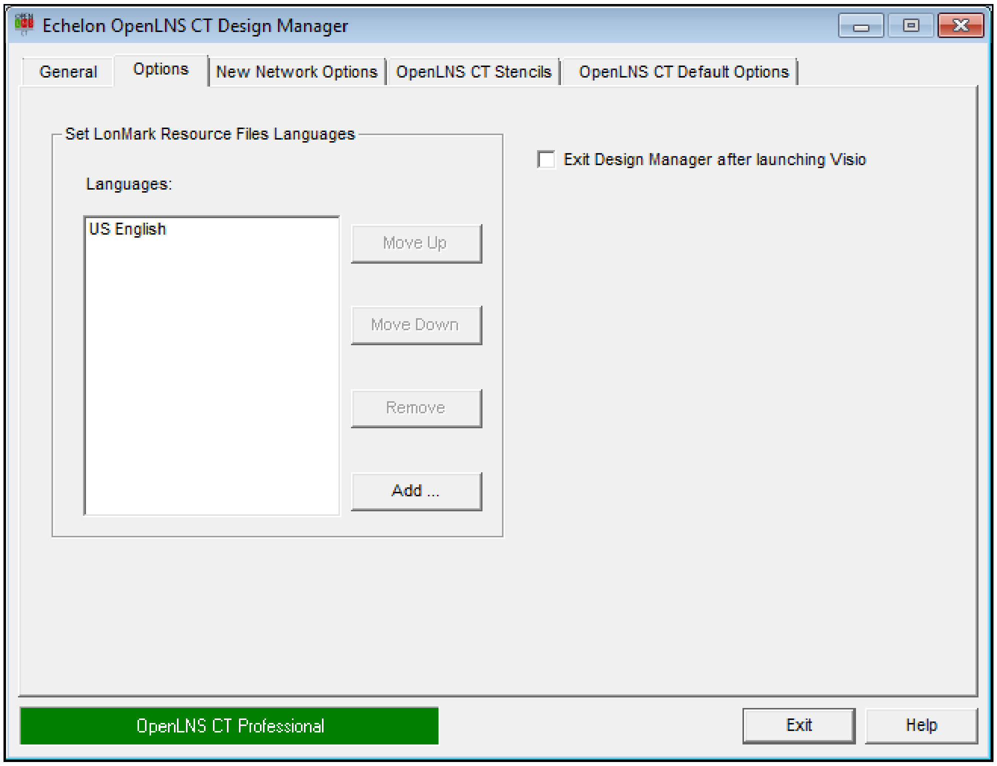
To automatically exit the Design Manager after opening an OpenLNS CT drawing, click the **Options** tab in the OpenLNS CT Design Manager and select the **Exit Design Manager After Launching Visio** check box.

Options

The OpenLNS CT Design Manager’s **Options** tab allows you to set a prioritized list of languages for LONMARK resource files. This list is then applied to all new networks you create.

When OpenLNS CT displays documentation for selected devices, functional blocks, and network variables, it uses the definitions contained in LONMARK resource files. You can display the LONMARK resource file information in different languages by specifying a prioritized list of languages for LONMARK resource files in OpenLNS CT.

You can set the resource file language for a specific network while in OpenLNS CT by clicking **Add-ins**, clicking **OpenLNS CT**, clicking **LonWorks Network**, clicking **Network Properties**, and then clicking the **Resource File Language** tab.



Languages:

US English

☐ Exit Design Manager after launching Visio

The OpenLNS CT Design Manager's **Options** tab contains the following options:

Set LonMark Resource Files Languages

Allows available language-specific LonMark resource files to be added, removed, or reordered in the languages list. When searching for a resource definition such as an NV field name, OpenLNS CT searches the first language file in the list for the device information, then the second language file in the list, and so on until it finds the information for the specified device. If no match is found and the device information is found in the corresponding U.S. English file, the English text will be used to display the information.

U.S. English is the default language and cannot be removed from the list. The language-specific LonMark resource files added to the list must be installed and available to OpenLNS CT. Usually they will have been installed if available and you click **Add** to add them to the list.

Languages

Lists the languages in the order they will be searched for device information.

Move Up

Moves the selected language up one position in the list.

Move Down

Moves the selected language down one position in the list.

Remove

Removes the selected language from the list.

Add

Adds a language to the end of the list.

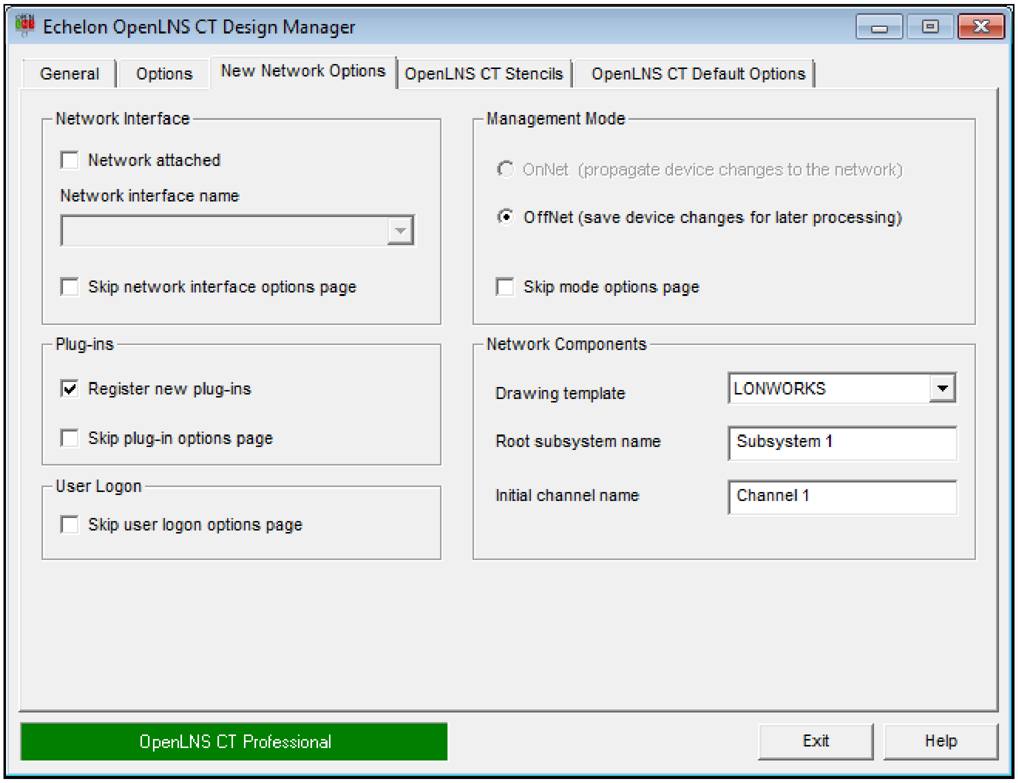
Exit Design Manager after Launching Visio

Selecting this option automatically closes the OpenLNS CT Design Manager after launching Visio.

New Network Options

You can use the **New Network Options** tab to set the default network interface, automatically register plug-ins, set the default network management mode, and set the default drawing template. The properties you set are then applied to each new network you create.

You can also select options to have OpenLNS CT skip the pages in the Network Wizard when opening a network. If you additionally clear the **Show all Options** check box under the **New Network** box in the **General** tab, you can create new networks with one-click, skipping all the pages in the Network Wizard.



Network Interface

☐ Network attached

Network interface name

☐ Skip network interface options page

Management Mode

○ OnNet (propagate device changes to the network)

● OffNet (save device changes for later processing)

☐ Skip mode options page

Plug-ins

☑ Register new plug-ins

☐ Skip plug-in options page

Network Components

Drawing template: LONWORKS

Root subsystem name: Subsystem 1

Initial channel name: Channel 1

User Logon

☐ Skip user logon options page

The OpenLNS CT Design Manager’s **New Network Options** tab contains the following options:

Network Interface

Network Attached — Select this check box to specify that the OpenLNS CT computer will be attached to the physical network so that OpenLNS CT can communicate with the physical devices.

Clear this check box if you the OpenLNS CT computer will not be attached to the network. This check box is cleared by default.

Network Interface Name — Select the OpenLNS network interface to be used for communication between the OpenLNS CT computer and the physical network. This option is only available if you selected the **Network Attached** check box.

Skip Network Interface Options Page — Skips the network interface page in the Network Wizard when you create a new OpenLNS CT network.

Plug-Ins

Register Plug-Ins — Select this check box to register all enabled plug-ins when you create a new OpenLNS CT network.

Skip Plug-ins Options Page — Skips the plug-in registration page in the Network Wizard when you create a new OpenLNS CT network.

User Logon

Skip Logon Options Page — Skips the user logon page in the Network Wizard when you open an OpenLNS CT network.

Management Mode — Select when changes made in your OpenLNS CT drawing are transmitted to the physical devices on the network.

OnNet — Changes to the OpenLNS CT drawing are sent immediately to the physical devices on the network. Select OnNet if you are commissioning an engineered system or if you are working on any phase of an ad-hoc system.

OffNet — Changes to the OpenLNS CT drawing are saved in the OpenLNS network database, but are not propagated to physical devices until OpenLNS CT is placed OnNet. Select OffNet if you are designing an engineered system.

Once you have commissioned devices with OpenLNS CT OnNet, you can make real-time changes to network variables when OpenLNS CT is OffNet.

You can read and make changes to configuration properties while OpenLNS CT is OffNet, but changes will not be propagated until you set OpenLNS CT OnNet or you manually propagate them (by right-clicking a device, pointing to **Commissioning**, and selecting **Propagate CP Values to Device** on the shortcut menu).

Skip Mode Options Page — Skips the management mode page in the Network Wizard when you create a new OpenLNS CT network.

Network Components

Drawing Template — Lists the templates available for new OpenLNS CT networks. The template is an empty OpenLNS CT drawing that specifies a default drawing background, styles, stencils, and other OpenLNS CT settings. You can select from the following three default OpenLNS CT templates that are listed:

- **LonWorks** is the initial default template used for new networks. It uses U.S. measurement units.
- **LonDemo** is the template used to create a demonstration network. In a demo network, you can only create a maximum of five application devices.
- **LonWorksMetric** is a template that uses SI measurement units for drawing pages.

Custom OpenLNS CT templates installed in the template directory are also listed.

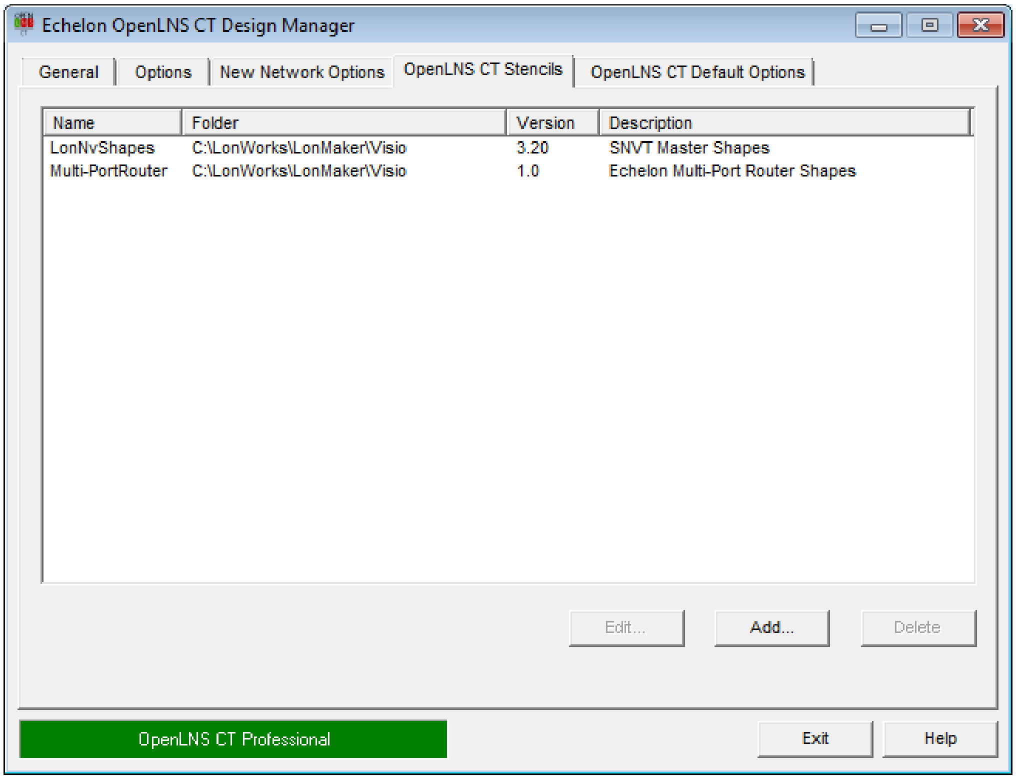
Root Subsystem Name — Specifies the top-level subsystem name, which opens on the title block on the top-center of the first page of each new OpenLNS CT drawing. Subsystem names may be up to 85 characters, may include embedded spaces, but may not include the period, backslash, colon, forward slash, or double quote characters. Subsystem names are case sensitive. The default top-level subsystem name is “Subsystem 1”.

Initial Channel Name — Specifies the name of the initial channel that is attached to the OpenLNS Network Interface in each new network drawing.

Channel names may be up to 85 characters, may include embedded spaces, but may not include the period, backslash, colon, forward slash, or double quote characters. Channel names are case sensitive. The default root channel name is “Channel 1”.

OpenLNS CT Stencils

The OpenLNS CT Design Manager’s **OpenLNS CT Stencils** tab lists the OpenLNS CT stencils that are initially available in the LONWORKS template upon opening or creating a network drawing. It allows you to add, remove, or modify the properties of OpenLNS CT stencils.



Name	Folder	Version	Description
LonNvShapes	C:\LonWorks\LonMaker\Visio	3.20	SNVT Master Shapes
Multi-PortRouter	C:\LonWorks\LonMaker\Visio	1.0	Echelon Multi-Port Router Shapes

The **OpenLNS CT Stencils** tab contains the following options:

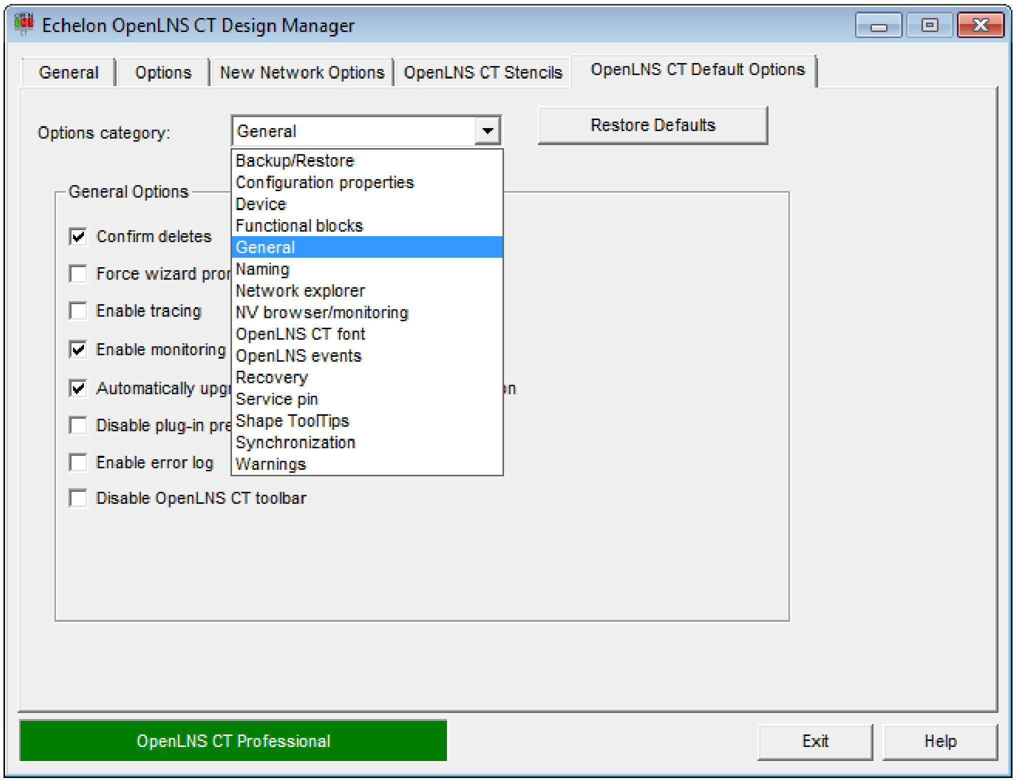
Edit Click the name of an OpenLNS CT stencil you wish to modify and then click **Edit**. The **Edit Stencil Properties** dialog opens. You can change the name of the OpenLNS CT version or modify the description associated with an OpenLNS CT stencil file. Conversely, you can change the stencil file associated with the OpenLNS CT version or description of a stencil file.

Add Click **Add** to open the **Add New Stencil** dialog. Click **Browse**, select a Visio stencils (.vss) file from **C:\Program Files\LonWorks\LonMaker\Visio** (32-bit system), **C:\Program Files (x86)\LonWorks\LonMaker\Visio** (64-bit system), or from another folder in your computer, and then click **Open**. You can then enter the OpenLNS CT version and a description of the selected Visio stencil. There is no limit on the number of characters, and you can include embedded spaces, periods, backslashes, colons, forward slashes, and double quotes. When you are done, click **OK**.

Delete Click the name of an OpenLNS CT stencil you want to remove from the list and then click **Delete** (the stencil file is not deleted).

OpenLNS CT Default Options

You can use the **OpenLNS CT Default Options** tab to set the OpenLNS CT default properties for the new networks you create. This tab contains a list of OpenLNS CT options categories such as options for configuration properties, devices, functional blocks and for backing up and synchronizing the OpenLNS CT network database. The default options you set will be applied to each new OpenLNS CT network you create. Once you create a new network, you can change any of the default options in the OpenLNS CT drawing by clicking **Add-ins**, clicking **OpenLNS CT**, and then clicking **OpenLNS CT Options**.



Options category: General

General Options

☑ Confirm deletes

□ Force wizard pro

□ Enable tracing

☑ Enable monitoring

☑ Automatically upg

□ Disable plug-in pre

□ Enable error log

□ Disable OpenLNS CT toolbar

Click **Restore Defaults** to reset the options and values in the currently displayed options category to their defaults.

You can set the default options for new networks in the following options categories:

- Backup/Restore
- Configuration Properties
- Device
- Functional Blocks
- General
- Naming Options
- Network Explorer
- NV Browser/Monitoring
- OpenLNS Events
- OpenLNS CT Font
- Recovery
- Service Pin
- SmartShape ToolTips

- Synchronization
- Warnings

Appendix A details the defaults you can set under each network properties options category.

Setting Up a Network Interface

To attach your OpenLNS CT computer to a LONWORKS network, you must define and configure an OpenLDV 4.0-compatible network interface or an IP-852 network interface before starting OpenLNS CT. Network interfaces include the following:

- RNIs: SmartServer, i.LON 100 Internet Server, i.LON 600 IP-852 Router, and i.LON 10 Ethernet Adapter.
- IP-852 network interfaces: SmartServer (with IP-852 routing option), i.LON 100 Internet Server (with IP-852 routing option), and i.LON 600 IP-852 Router.
- Local USB network interfaces: U10/U20 USB network interfaces.
- Local PC card network interfaces: PCC-10 and PCLTA-10, 20, and 21 PCI network interfaces (32-bit versions of Windows only).
- Serial network interface: SLTA-10 serial interface (32-bit versions of Windows only).

You can use the **LONWORKS Interfaces** application in the Control Panel to define, configure, and test these OpenLDV 4.0 network interfaces.

Optimizing Network Interface Performance

To optimize the performance of OpenLNS CT when your OpenLNS CT computer is attached to a LONWORKS network, use an OpenLNS high performance network interface (RNI, U10/U20 USB network interface, IP-852 network interface, or other network interface with a Layer 2 MIP), if available. If you are using a PL-20 power line channel or another slow channel, you can use a legacy interface to limit network overhead (a high performance network interface may saturate the channel).

Using an IP-852 Network Interface

You can use an IP-852 channel to implement a LONWORKS control network over an IP network and integrate multiple native LONWORKS networks into one large network that uses a high-speed IP-852 channel as a backbone. If you are attaching your OpenLNS CT computer to an IP-852 channel, you can set up an IP-852 interface and create the IP-852 channel following these steps:

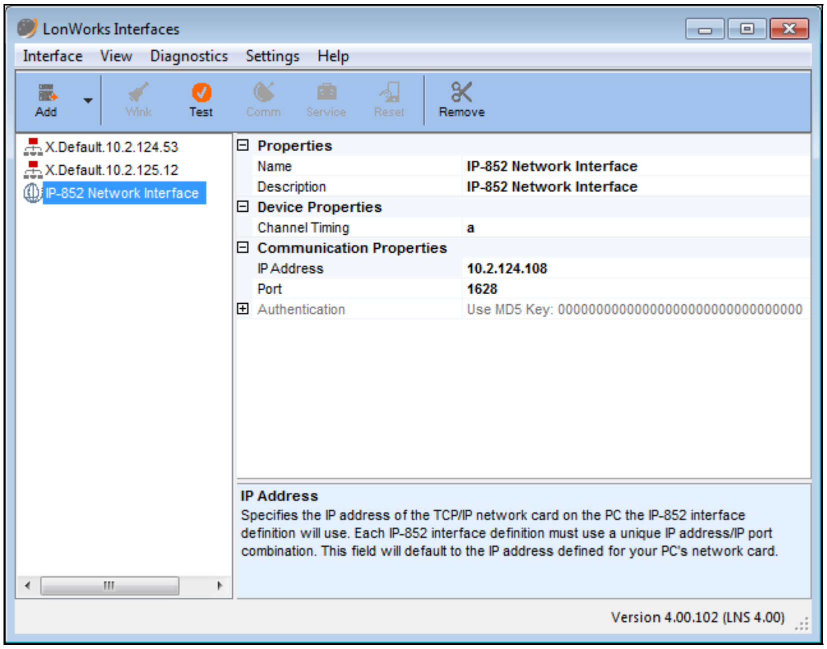
1. Ensure that all of your IP-852 devices (for example, the OpenLNS Server computer, one or more OpenLNS CT computers, and one or more IP-852 routers) all have static IP addresses or have dynamic IP addresses that do not change. A static IP address is an address that is not dynamically assigned by a DHCP server.
2. Create an IP-852 interface on each computer that is to be connected to the IP-852 channel. This may include the OpenLNS Server computer and one or more OpenLNS CT computers. To create an IP-852 interface, follow these steps on each of the computers:
    a. Open the **LONWORKS Interfaces** application in the Control Panel.

    b. Click the **IP-852** tab.

    c. Click **Interface**, point to **Add**, and then either click **IP-852 Interface** or click **New Interface** and then click **IP-852** in the **Select Interface Type** dialog.

    d. The **Add Network Interface Wizard** dialog opens.

    e. Specify the name of the new IP-852 network interface; the IP address, port, and channel timing parameters; and authentication parameters. Click **OK**.



⊟ **Properties**	
Name	**IP-852 Network Interface**
Description	**IP-852 Network Interface**
⊟ **Device Properties**	
Channel Timing	**a**
⊟ **Communication Properties**	
IP Address	**10.2.124.108**
Port	**1628**
⊞ Authentication	Use MD5 Key: 00000000000000000000000000000000

IP Address
Specifies the IP address of the TCP/IP network card on the PC the IP-852 interface definition will use. Each IP-852 interface definition must use a unique IP address/IP port combination. This field will default to the IP address defined for your PC's network card.

Version 4.00.102 (LNS 4.00)

For more information on creating IP-852 network interfaces, see the **LONWORKS Interfaces** online help file.

3. Define the IP-852 channel and devices in the IP-852 Configuration Server. To do this, follow these steps:

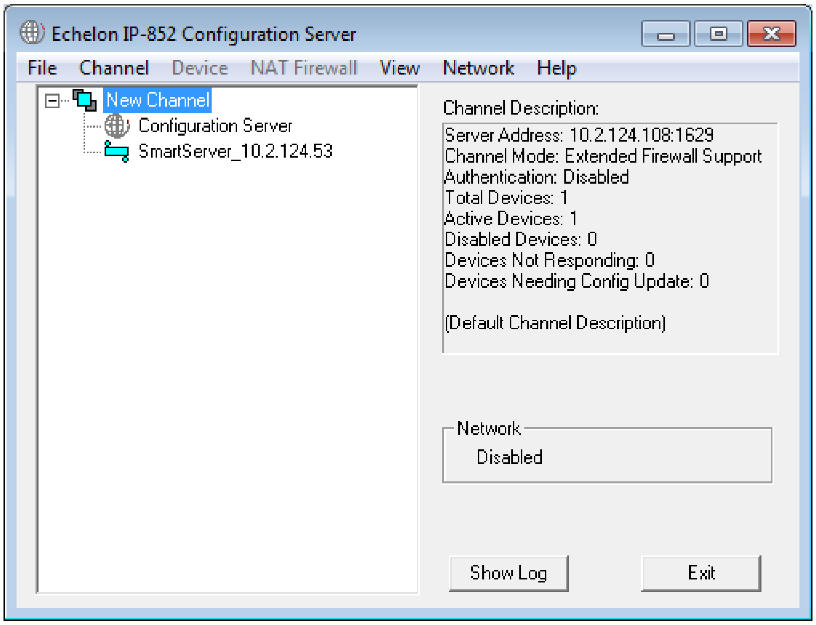
   a. Click **Start** on the taskbar, point to **Programs**, point to **Echelon IP-852 Configuration Server**, and then click **Echelon IP-852 Configuration Server**.

   b. Click **Channel**, and then click **New Channel**. Enter a name for the new IP-852 channel.

   c. Right-click the new IP-852 channel you just created and then click **New Device**. Enter a name for the new device.

   d. Right-click the new device and then click **Device Properties**. Enter the IP address for the device.

   e. Repeat steps c and d for each OpenLNS Server computer, OpenLNS CT computer, and IP-852 router on the IP-852 channel.



Channel Description:

Server Address: 10.2.124.108:1629
Channel Mode: Extended Firewall Support
Authentication: Disabled
Total Devices: 1
Active Devices: 1
Disabled Devices: 0
Devices Not Responding: 0
Devices Needing Config Update: 0

(Default Channel Description)

Network
Disabled

The devices on the IP-852 channel will not be fully commissioned until you run OpenLNS applications such as the OpenLNS Server or OpenLNS CT on each of the OpenLNS Server or OpenLNS CT computers. For more information, see the *IP-852 Channel User's Guide*.

Creating and Opening OpenLNS CT Networks

An OpenLNS CT network consists of an OpenLNS CT *drawing* and an *OpenLNS network database*. OpenLNS CT provides several ways to create an OpenLNS CT network, including the following methods:

- Create a new OpenLNS CT network design. Use this method if you are engineering a new network, or if you want to redo the logical design of an existing network.
- Use an existing OpenLNS database to create an OpenLNS CT drawing. Use this method to use OpenLNS CT with a network that was previously installed with an OpenLNS application other than OpenLNS CT.
- Create an OpenLNS CT network design from an existing physical network, which creates a new OpenLNS network database and an OpenLNS CT drawing. This method is called *network recovery*. Use this method if the network is operational and was originally installed with an application not based on OpenLNS or was self-installed. You can also use this method with a network that was previously installed with an OpenLNS application, but does not have its original network database. OpenLNS CT can recover a network without changing the device applications or network configuration. See *Recovering an OpenLNS CT Network* in Chapter 7, *Maintaining Networks*, for more information on how to use network recovery and for a summary of the limitations of network recovery.
- Copy an existing OpenLNS CT network design. Use this method to create a new network that is similar to an existing network.

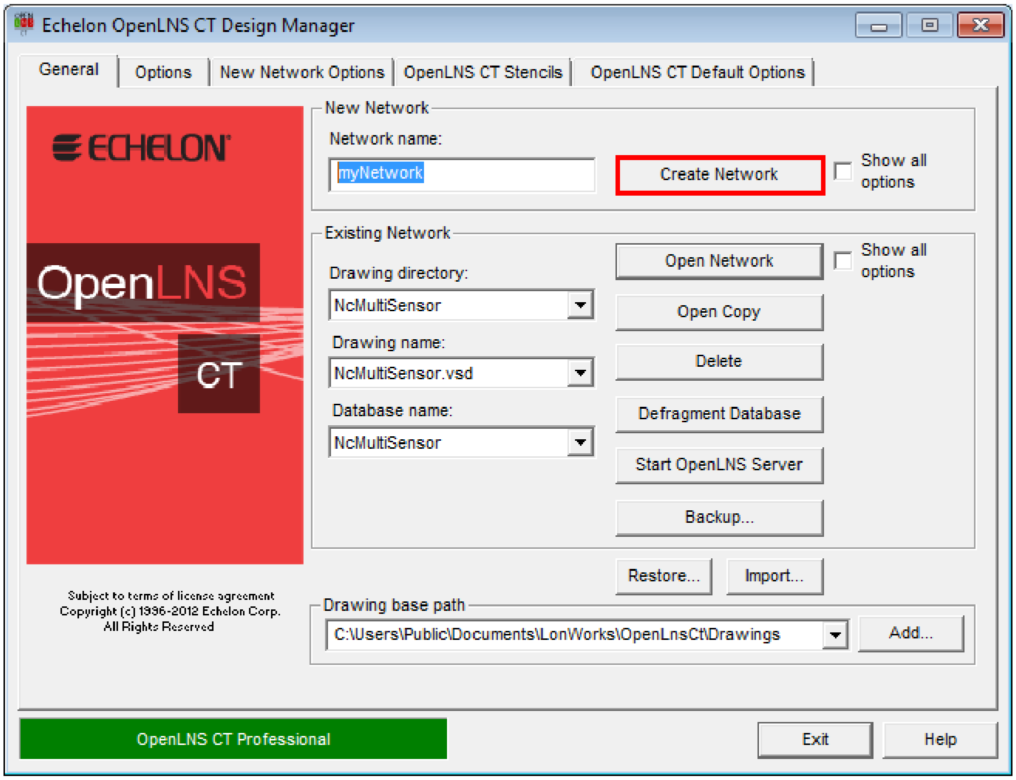
Creating an OpenLNS CT Network Design

You can create a new OpenLNS CT network design to engineer a new network or to redo the logical design of an existing network. To create a new network design, you must use OpenLNS CT as a local client (OpenLNS CT is running on the same computer as the OpenLNS server).

To create a new OpenLNS CT network, follow these steps:

1. Click **Start** on the taskbar, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **OpenLNS CT**. The **OpenLNS CT Design Manager** opens.

2. Under **New Network**, enter the name of the network in the **New Network** box, or accept the default network name.



Network name: myNetwork

Drawing directory: NcMultiSensor

Drawing name: NcMultiSensor.vsd

Database name: NcMultiSensor

Drawing base path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Drawings

Subject to terms of license agreement
Copyright (c) 1996-2012 Echelon Corp.
All Rights Reserved

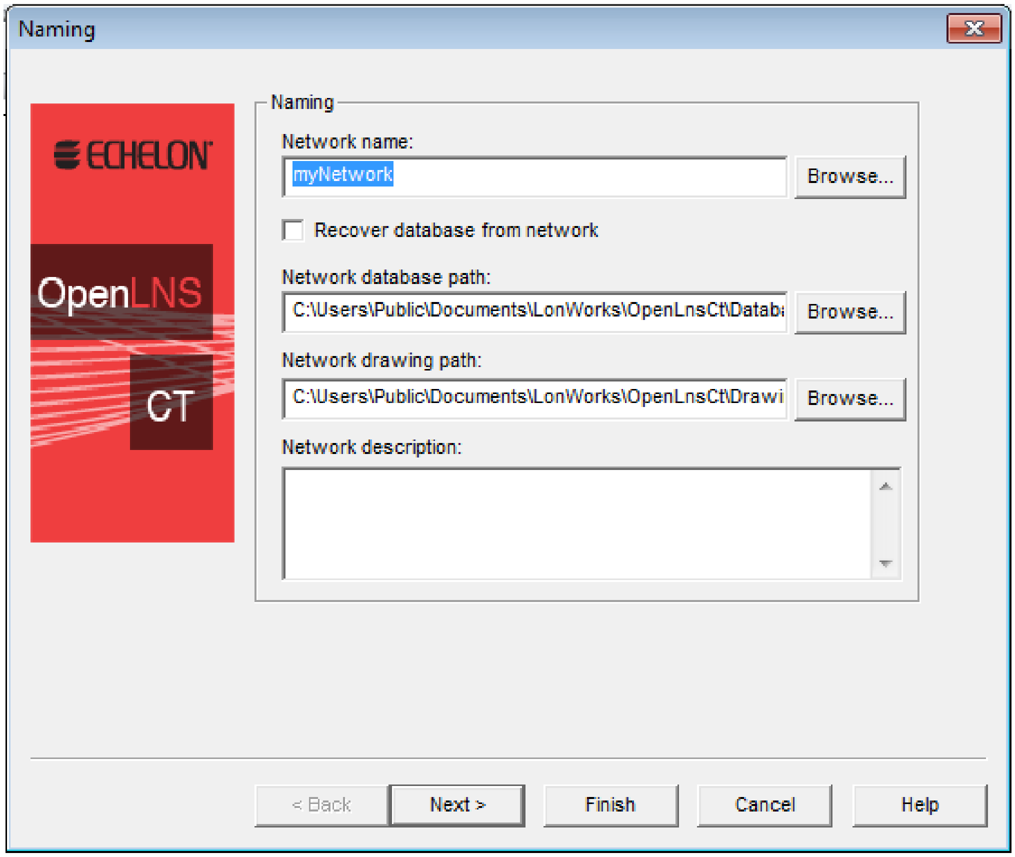
3. Click **Create Network** to create the new network. Visio starts.

   - When you initially start Visio 2010, a message opens informing you that Visio must activated. Click **OK** and follow the activation instructions.

   - When you initially attempt to create a network, a message opens informing you that Visio must be launched and initialized so that it can work with OpenLNS CT.

   - In addition, a warning opens asking you if you want to enable macros, which are required for OpenLNS CT to function. Click **Enable Macros**. See *Working with Digital Signatures* for more information.

4. If you selected the **Show all Options** check box under the **New Network** box in the **OpenLNS CT Design Manager**, or you cleared the check boxes for skipping the network interface, management mode, and plug-in options pages of the Network Wizard in the **New Network Options** tab, the **Naming** dialog in the Network Wizard opens. Otherwise, you can skip to step 6.



Naming

Network name: myNetwork

☐ Recover database from network

Network database path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Databa

Network drawing path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Drawi

Network description:

Tips:

- You can click **Finish** at any point in the Network Wizard. Doing so will skip the subsequent series of pages, and your new OpenLNS CT network design will be created using the current settings in the **New Network Options** tab of the OpenLNS CT Design Manager.
- By default, the subsequent pages in the Network Wizard will appear each time you re-open the OpenLNS CT drawing. Select the **Skip This Prompt when Re-opening this Drawing** check box on any of the pages you want to skip when re-opening the drawing.
- You can change any options you set in the Network Wizard in the OpenLNS CT drawing. Open the OpenLNS CT menu, click **Network Properties**, and then click the tab containing the options you want to modify.

5. Edit the following information as required:

Network Name	Displays the network name. By default, this is the current name in the **Network Name** box under **New Network** in the **General** tab of the OpenLNS CT Design Manager. The default network name is incremented if a naming conflict exists. You can change the name of the network. The name must be unique to the network database, and it can have a maximum of 16 characters. The network name in the **Network Database Path** and **Network Drawing Path** fields will automatically be updated to match the name you specify.
Recover Database from Network	Opens the OpenLNS Database Recovery Wizard after completing the Network Wizard. You can use the recovery wizard to create a new OpenLNS CT drawing and network database from an existing operational network.

See *Recovering an OpenLNS CT Network* in Chapter 7, *Maintaining Networks*, for more information.

Network Database Path — Specifies the folder that will contain the network database. By default, the database path is **C:\Users\Public\Documents\LonWorks\OpenLnsCt\Databases\**<*network name*>.

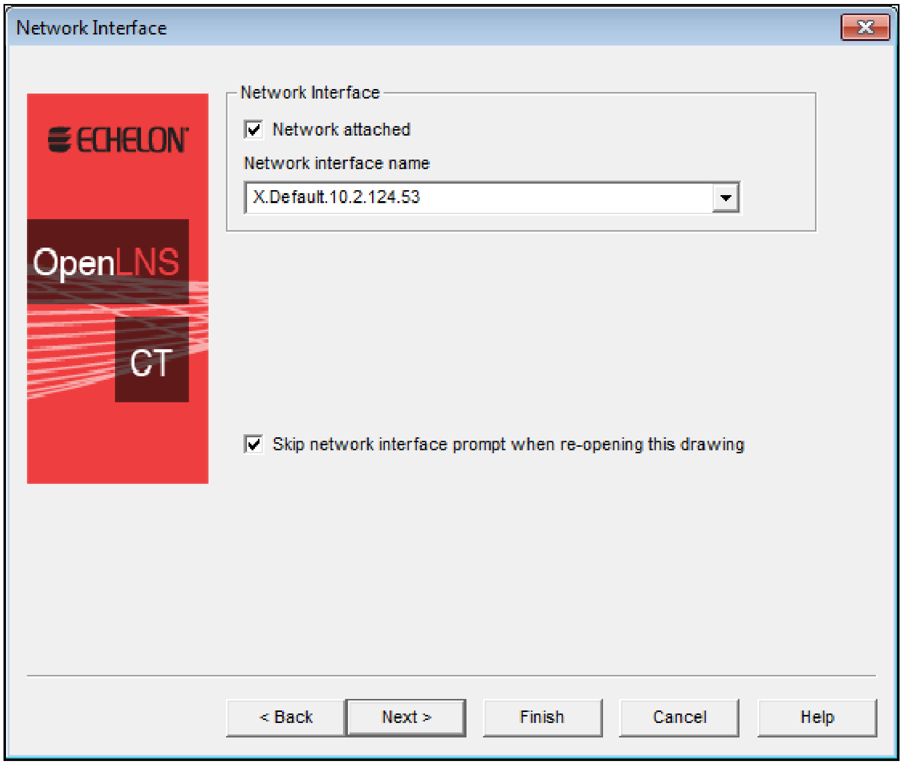
You can specify a different to place the OpenLNS CT network database; however, the network name in the database path must match the one in the **Network Name** box.

Network Drawing Path — Specifies the folder that will contain the OpenLNS CT drawing. By default, the drawing path is **C:\Users\Public\Documents\LonWorks\OpenLnsCt\Drawings**\<*network name*>.

You can specify a different to place the OpenLNS CT drawing; however, the network name in the drawing path must match the one in the **Network Name** box.

Network Description — Contains an optional description of the network being created. This field has no effect on network operation, but it can provide additional documentation for as-built reports.

6. Click **Next**. The **Network Interface** dialog opens.



Network Interface

Network Interface

☑ Network attached

Network interface name: X.Default.10.2.124.53

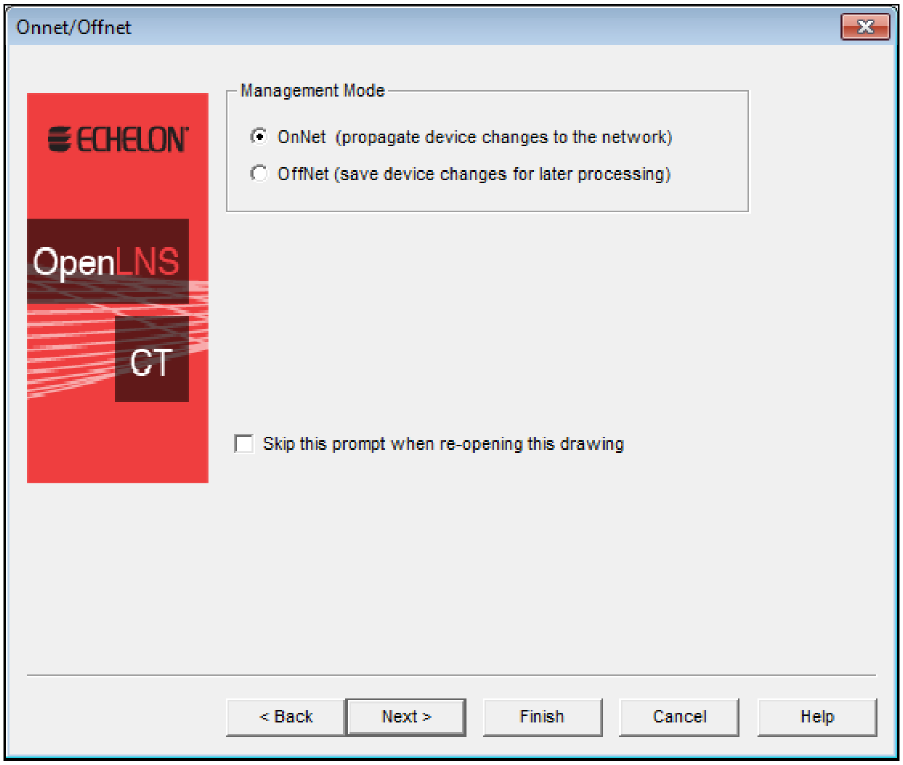
☑ Skip network interface prompt when re-opening this drawing

7. If your OpenLNS CT computer is attached to the physical network and you want OpenLNS CT to communicate with the physical devices, select **Network Attached**. If you select this option and you have more than one network interfaces in your computer, select the name of the OpenLDV 4.0 or IP-852 network interface under **Network Interface Name**.

Notes:

- The network interface you select must already have been defined and configured, as described in *Setting up a Network Interface*.
- For OpenLNS CT to communicate on an IP-852 channel, the OpenLNS CT computer must be configured as an IP-852 device on the target channel using the IP-852 Configuration Server. If an IP-852 device name has not been defined, create one in the IP-852 Configuration Server as described in *Using an IP-852 Network Interface* earlier in this chapter.

8. Click **Next**. If you specified that OpenLNS CT computer is attached to the physical network, the **OnNet/OffNet** dialog opens. If it is not attached, the **Plug-In Registration** dialog opens and you can skip to step 11.


Onnet/Offnet

Management Mode

● OnNet (propagate device changes to the network)

○ OffNet (save device changes for later processing)

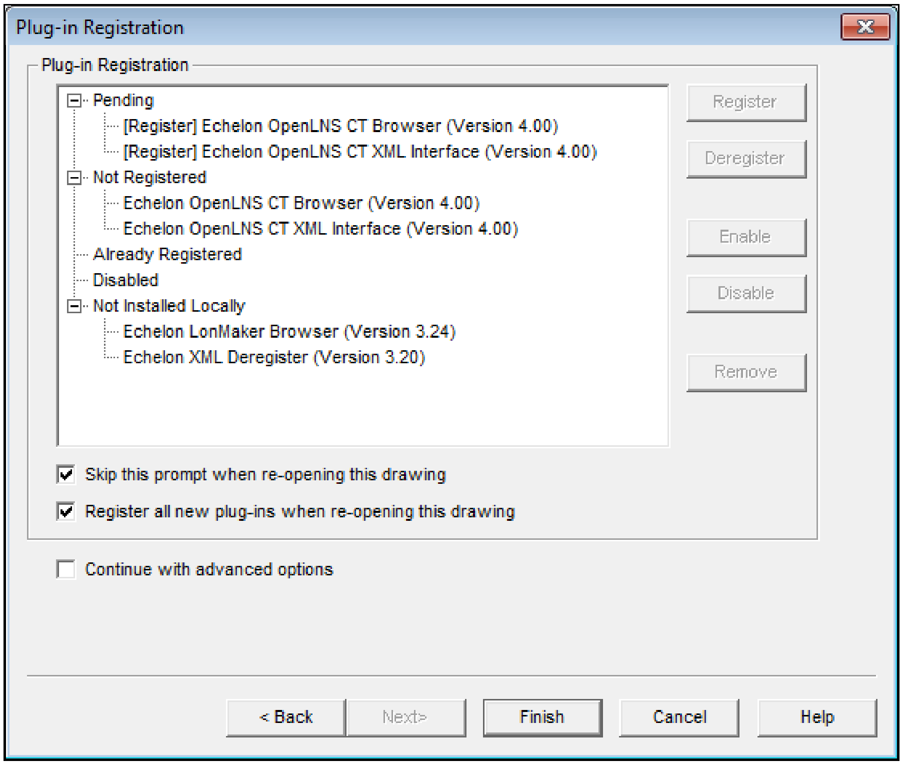
□ Skip this prompt when re-opening this drawing

9. Select when changes made in your OpenLNS CT drawing are sent to the physical devices on the network. You have two choices:

- **OnNet**. Changes to the OpenLNS CT drawing are sent immediately to the physical devices as you make the changes. Select OnNet if you are commissioning an engineered system or if you are working on any phase of an ad-hoc system.
- **OffNet**. Changes to the OpenLNS CT drawing are saved in the network database and then sent to the physical devices once OpenLNS CT is set OnNet. Select OffNet if you are designing an engineered system. You can make real-time changes to network variables when OpenLNS CT is OffNet once you have commissioned devices with OpenLNS CT OnNet. You can read and make changes to configuration properties while OpenLNS CT is OffNet, but changes will not be propagated until you set OpenLNS CT OnNet or you manually propagate them (by right-clicking a device, pointing to **Commissioning**, and selecting **Propagate CP Values to Device** on the shortcut menu).

  Select this **Skip the Prompt** check box if you want the Network Wizard to skip the Management Mode page each time you open the OpenLNS CT network. If the **Show all Options** check box under **Existing Networks** in the OpenLNS CT Design Manager **General** tab is selected, the prompt will appear regardless if this check box is selected.

10. Click **Next.** The **Plug-In Registration** dialog opens.



Plug-in Registration

Plug-in Registration

- Pending
  - [Register] Echelon OpenLNS CT Browser (Version 4.00)
  - [Register] Echelon OpenLNS CT XML Interface (Version 4.00)
- Not Registered
  - Echelon OpenLNS CT Browser (Version 4.00)
  - Echelon OpenLNS CT XML Interface (Version 4.00)
- Already Registered
- Disabled
- Not Installed Locally
  - Echelon LonMaker Browser (Version 3.24)
  - Echelon XML Deregister (Version 3.20)

☑ Skip this prompt when re-opening this drawing

☑ Register all new plug-ins when re-opening this drawing

☐ Continue with advanced options

11. Use the following commands to register, deregister, enable, and disable plug-ins in your OpenLNS CT network.

Register

Adds a plug-in to the Pending list, indicating that the plug-in is to be registered. A plug-in must be registered for it to be available within OpenLNS CT.

This button is available when a plug-in listed under any category or a category label (except Pending) is selected. If a category label is selected, all the plug-ins under it will be registered.

Deregister

Adds a plug-in to the Pending list, indicating that the plug-in is to be deregistered. Once a plug-in is deregistered, it is no longer available within OpenLNS CT.

This button is available when a plug-in listed under Already Registered is selected, can be deregistered, and supports being deregistered.

Enable

Adds a plug-in listed under Disabled to the Pending category, indicating that the plug-in is to be enabled. You can enable a plug-in that you previously disabled.

This button is available when a plug-in listed under Disabled or the Disabled category label is selected. If the Disabled category label is selected, all the plug-ins under it will be enabled.

Disable

Adds a plug-in listed under Already Registered or Not Registered to the Pending category, indicating that the plug-in is to be disabled. You can disable a plug-in to prevent OpenLNS CT from prompting you to register the plug-in.

This button is available when a plug-in listed under Already

Registered or Not Registered or either of the category labels is selected. If the Already Registered or Not Registered category label is selected, all the plug-ins under the selected category label will be disabled.

Remove

Removes a plug-in from the Pending list. This button is available when a plug-in listed under Pending or the Pending category label is selected. If the Pending category label is selected, all the plug-ins under it will be removed.

Skip this Prompt when Re-opening Drawing

Select this check box if you want the Network Wizard to skip the Plug-In Registration page each time you open the OpenLNS CT network. If the **Show all Options** check box under **Existing Networks** in the OpenLNS CT Design Manager **General** tab is selected, the prompt will appear regardless if this option is selected.

Register all New Plug-ins when Re-opening this Drawing

Select this check box if you want new plug-ins in the plug-in directory to be automatically registered each time you open the OpenLNS CT drawing.

New plug-ins must be registered with Windows. The installation program for the plug-ins typically does this automatically. If you are designing your own plug-ins see the *OpenLNS Plug-in Framework User's Guide* for more information on how to create plug-ins and register them with Windows.

Continue with Advanced Options

Opens and lets you set options in the authentication, domain definition, timing, and resource file language pages in the Network Wizard. See *Recovering an OpenLNS CT Network* in Chapter 7, *Maintaining Networks*, for setting the options in these pages.

12. Click **Finish** to open your new OpenLNS CT drawing.
13. If you are creating or opening an OpenLNS CT network for the first time, a message opens informing you that the drawing being opened contains macros from the Echelon Corporation, and asking you if you would like to enable macros for this drawing. Verify that the macros were signed by Echelon Corporation, and then enable macros, which is required for using OpenLNS CT. Set **Always Trust Macros from this Source** to automatically enable macros for all OpenLNS CT drawings that are signed by Echelon

Working with Digital Signatures

Visio drawings can have VBA code associated with them through the use of macros and ActiveX objects. Visio drawings that use this capability use Microsoft VBA signatures to provide security from VBA viruses. Each time you create an OpenLNS CT network design, the template that you selected in the OpenLNS CT Design Manager (**LonWorks.vst** by default) is used to create the network drawing. This template uses VBA macros to implement the network drawing's LonWorks functionality. This template and all Visio drawings created from it are signed by the Echelon Corporation.

Network drawings created with LonMaker releases before 3.1 will not have a digital signature. Network drawings created with LonMaker Turbo Editions and LonMaker 3.1 or later may have expired digital signatures. In addition, you can cause an OpenLNS CT drawing to lose its Echelon signature by adding VBA code to it. This can happen if you add an ActiveX object or VBA code to the OpenLNS CT drawing. Once a network drawing has lost its signature, it cannot get it back automatically. If you have lost Echelon's signature on a network drawing, you have three options: (1) ignore the macro warnings as described in the next paragraph; (2) disable macro warnings as described in the next paragraph; or (3) sign the drawing with your company's digital signature as described at *http://office.microsoft.com/en-us/excel-help/digitally-sign-a-macro-project-HA001231781.aspx*.

If an OpenLNS CT drawing does not have a digital signature, you will see the macro warning dialog and will have to enable macros every time the network drawing is opened, provided Visio's macro security is set to medium (the default). In order to change Visio's macro security, open the **Tools** menu, point to **Macro** and then select **Security**. A dialog opens allowing you to select **Low**, **Medium**, or **High** security.

- If you select **Low** security, you will never see the macro warning.
- If you select **Medium** security, you will be prompted to enable macros for unsigned network drawings or network drawings signed by companies that have not been added to the trusted signatures list.
- If you select **High** security, macros will automatically be disabled (and OpenLNS CT will not function) for unsigned network drawings, and you will be prompted to enable macros for network drawings signed by companies that have not been added to the trusted signatures list.

Visio's security level is persistent. When you change Visio's security level, it will be the security level for Visio every time it is started until you change it again.

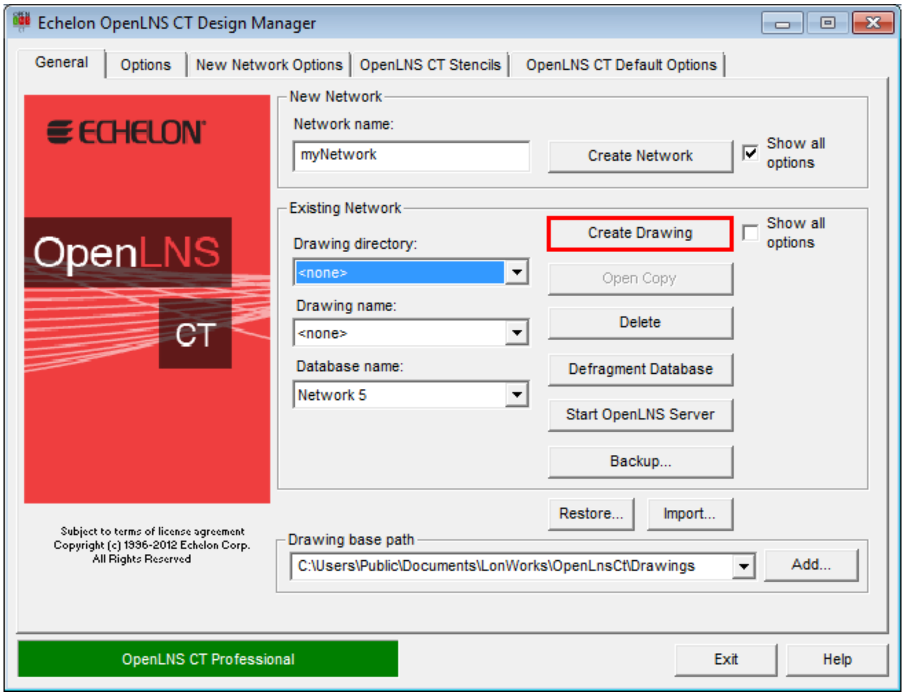
Tip: If your drawing contains ActiveX controls, change the security setting to **Low**. Otherwise, Visio will prompt you each time the drawing is opened as to whether to activate the controls.

Creating an OpenLNS CT Network from an Existing OpenLNS Database

You can create a new OpenLNS CT network design by opening an OpenLNS network database that is not already associated with a drawing. OpenLNS CT will then create a drawing for the database. You can use this procedure to import an existing network into OpenLNS CT when the network was created with an OpenLNS application other than OpenLNS CT. To create a new OpenLNS CT network from an existing OpenLNS network database, you must use OpenLNS CT on a local client.

To create a new OpenLNS CT network design from an existing OpenLNS database, follow these steps:

1. Click **Start** on the taskbar, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **OpenLNS CT**. The **OpenLNS CT Design Manager** opens.
2. Under **Existing Network**, select which database you want to open in the **Database Name** list and then select **<none>** in the **Drawing Directory** list. A **Create Drawing** command will appear in place of the **Open Network** command.



Network name: myNetwork

☑ Show all options

☐ Show all options

Drawing directory: <none>

Drawing name: <none>

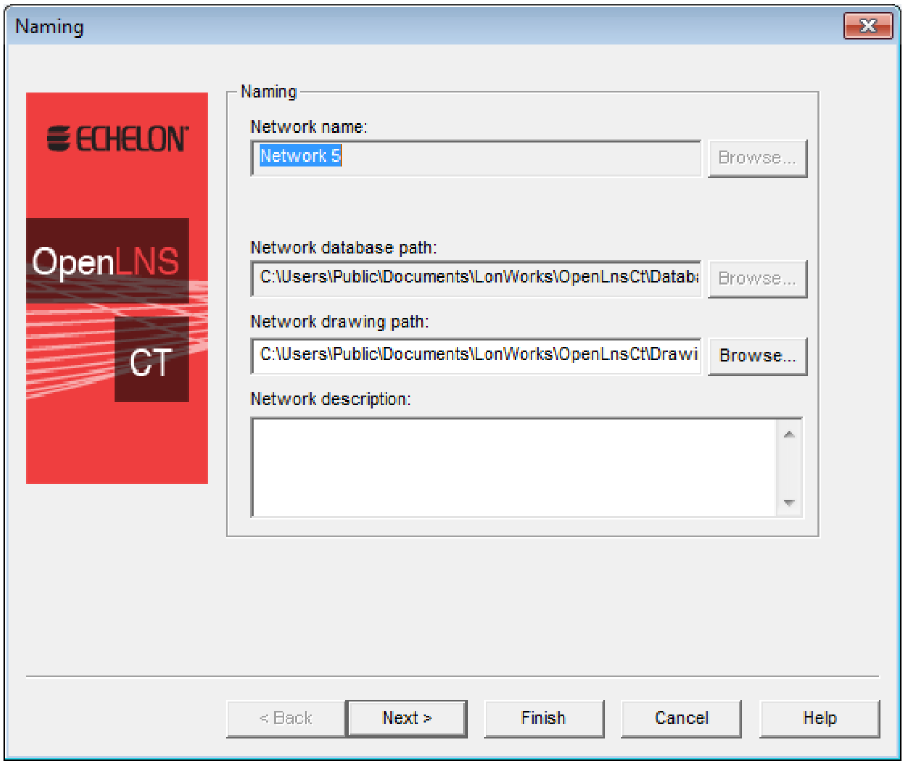
Database name: Network 5

Drawing base path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Drawings

Subject to terms of license agreement
Copyright (c) 1996-2012 Echelon Corp.
All Rights Reserved

3. Click **Create Drawing**. Visio starts and the **Naming** dialog of the Network Wizard opens.

4. Specify the folder that will contain the OpenLNS CT drawing in the **Network Drawing Path** box. The folder must have the same name that opens in the **Network Name** box, but it may be contained within another folder. The **Network Name** and **Network Database Path** boxes are read-only, as these settings are determined by the OpenLNS network database. Click **Next**. The Network Interface dialog opens.



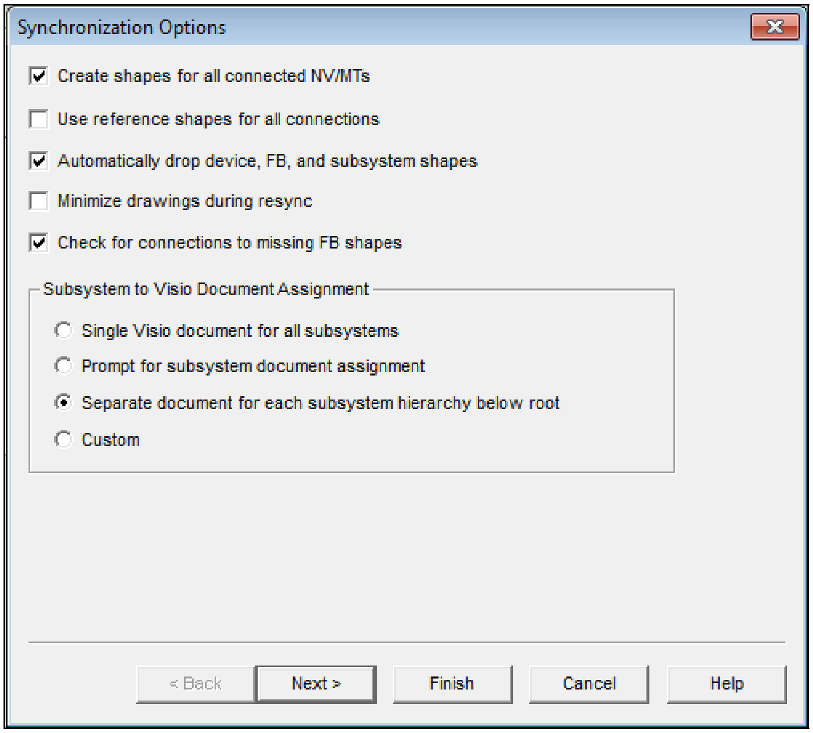
Network name: Network 5

Network database path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Datab

Network drawing path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Drawi

Network description:

5. Following steps 7 through 9 of *Creating a New OpenLNS CT Network Design*, select whether OpenLNS CT is attached to the network, and if it is attached, select the management mode (OnNet or OffNet). Click **Finish**. The **Synchronization Options** dialog box opens.


Synchronization Options

☑ Create shapes for all connected NV/MTs

☐ Use reference shapes for all connections

☑ Automatically drop device, FB, and subsystem shapes

☐ Minimize drawings during resync

☑ Check for connections to missing FB shapes

Subsystem to Visio Document Assignment

○ Single Visio document for all subsystems

○ Prompt for subsystem document assignment

● Separate document for each subsystem hierarchy below root

○ Custom

6. Select the desired synchronization options and subsystem assignments.

Create Shapes for all Connected NVs/MTs
Adds SmartShapes to functional blocks for all connected network variables or message tags created by the resynchronization.

If this check box is cleared, OpenLNS CT will use generic connections when possible. It will still create network variable or message tag SmartShapes for connections to supernodes and for those functional blocks that already contain network variable or message tag SmartShapes.

Use Reference Shapes for all Connections
Represents all network variable connections created by the resynchronization with reference connection SmartShapes.

If this check box is cleared, OpenLNS CT will use standard connection SmartShapes for connections for network variable hubs and targets located in the same subsystem. Reference connection SmartShapes are always used for connections that span subsystems.

Automatically Drop Device, FB, and Subsystem Shapes
Automatically adds SmartShapes to the drawing for all devices, functional blocks, and subsystems created by the resynchronization.

If this check box is cleared, OpenLNS CT will prompt the user to select whether a SmartShape should be added for a device, functional block, or subsystem created by the resynchronization.

Minimize Drawings during Resync
Minimizes all the drawings in the OpenLNS CT network during the resynchronization. This speeds up the resynchronization as it reduces the time Visio spends updating the drawings.

Check for Connections for
Enables OpenLNS CT to check for connections to functional blocks that are in the OpenLNS database but not represented by a

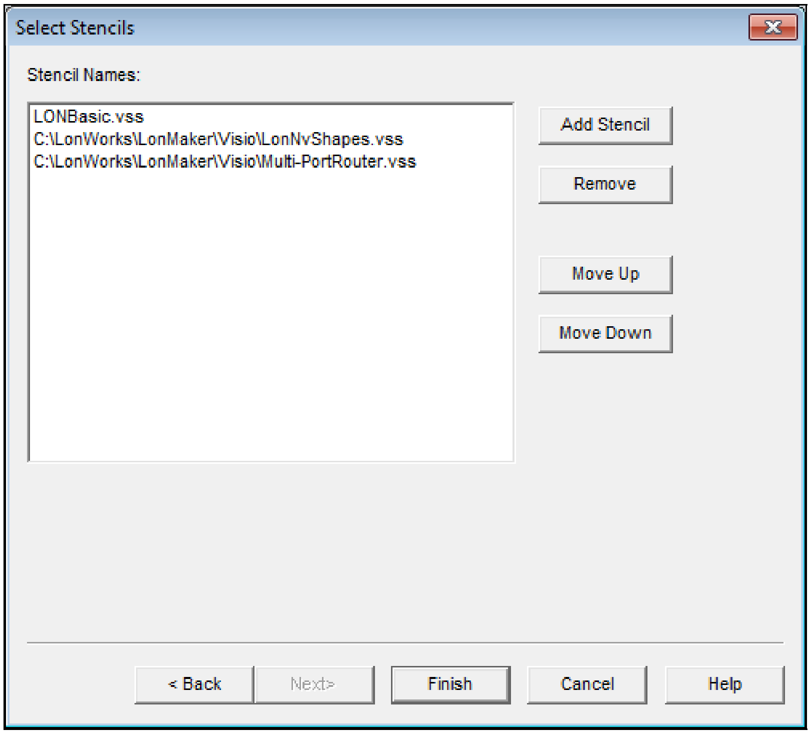
Missing FB Shapes SmartShape in the OpenLNS CT drawing.

Subsystem to Visio Document Assignment

Select in which Visio drawing file to place the subsystems in your network. You have four choices:

- **Single Visio Document for all Subsystems**. Places all subsystems in a single Visio document.
- **Prompt for Subsystem Document Assignment**. Prompts you to specify the Visio document for each subsystem.
- **Separate Document for each Subsystem Hierarchy below Root**. Places each subsystem hierarchy below the top-level subsystem in a separate, new Visio document.
- **Custom**. Opens the **Subsystem Document Assignment** dialog. In this dialog, you specify the Visio document for each subsystem.

7. Click **Next**. If you selected the **Automatically Drop Device, FB, and Subsystem Shapes** check box, the **Select Stencil** dialog box opens. If you cleared this check box, skip to step 9.



Stencil Names:

LONBasic.vss
C:\LonWorks\LonMaker\Visio\LonNvShapes.vss
C:\LonWorks\LonMaker\Visio\Multi-PortRouter.vss

8. Select the stencils that you want to be scanned for master SmartShapes.

   - To add a stencil, click **Add Stencil**, select the desired Visio stencil file (.vss), and then click **Open** or double-click the stencil.
   - To remove a stencil, select the desired stencil and then click **Remove**. You can also re-order the stencil positions by selecting a stencil and clicking **Move Up** or **Move Down**. Click **Finish**.

   OpenLNS CT will then search, in order, all the listed stencils. When objects are found within the database that are not already represented in the drawing, OpenLNS CT scans the listed stencils and automatically adds the appropriate SmartShape to the drawing. If the particular master SmartShape for an object is not found, an appropriate generic SmartShape (a

LONWORKS Device SmartShape, for example) will be used. After OpenLNS CT is done scanning stencils, the **Synchronization Status** dialog box opens.

9. You can monitor the progress of the resynchronization. When the resynchronization is complete, click **OK** to open your OpenLNS CT network.

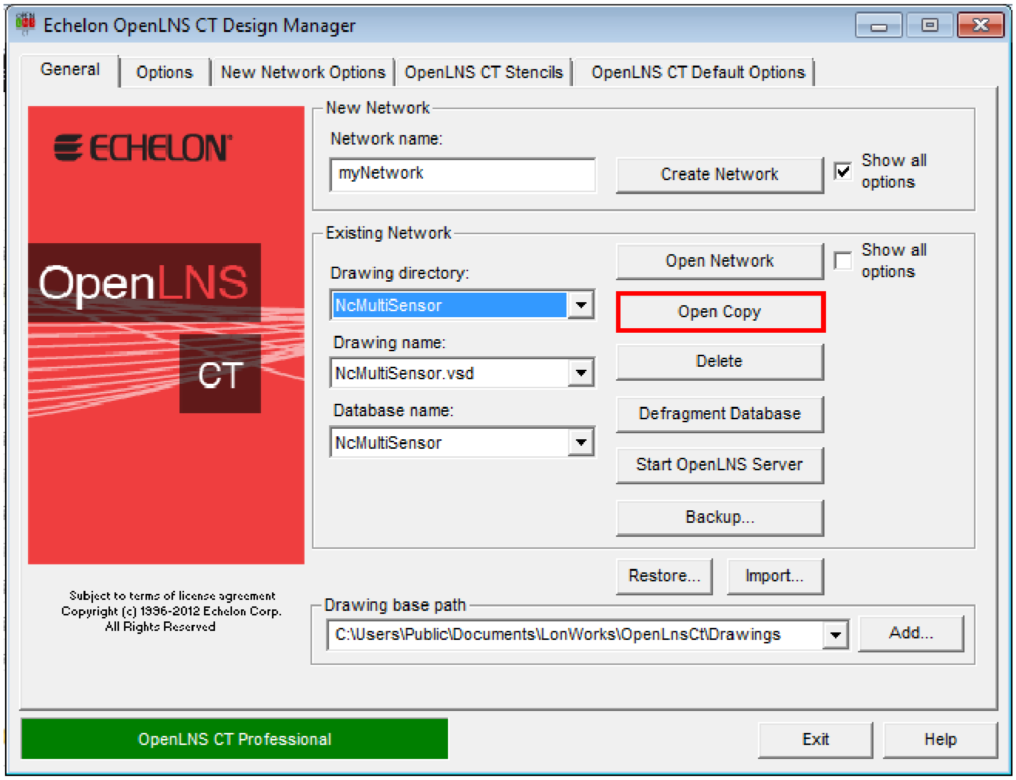
Copying an OpenLNS CT Network Design

You can create a new OpenLNS CT network design by copying an existing one. You can use this procedure to reuse the drawings, options, plug-ins, master SmartShapes, and device templates of an existing network design. You can also use this procedure to reduce the time required to create a new network. To do this, create an empty network design with all your desired plug-ins already registered, and then copy the empty network design for each new network that you create. To copy an existing OpenLNS CT network design, you must use OpenLNS CT on a local client

To create a new OpenLNS CT network design by copying an existing one, follow these steps:

1. Click **Start** on the taskbar, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **OpenLNS CT**. The OpenLNS CT Design Manager opens.

2. Choose an OpenLNS CT network to copy. Under **Existing Network**, select a drawing directory from the **Drawing Directory** list, and then select the network you want to copy from the **Drawing Name** list. Click **Open Copy**. A message opens prompting you to confirm that you want to open a copy of the selected network, which will create copies of the OpenLNS network database and all the drawings in the network.



Network name: myNetwork

☑ Show all options

Drawing directory: NcMultiSensor

☐ Show all options

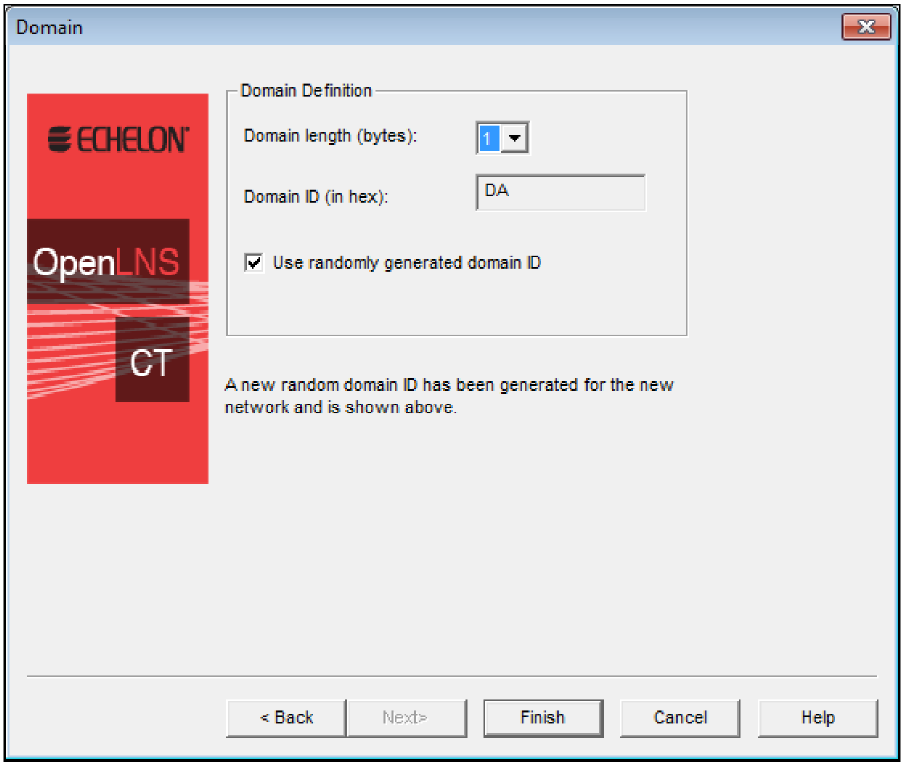
Drawing name: NcMultiSensor.vsd

Database name: NcMultiSensor

Drawing base path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Drawings

3. Click **Yes**. The **Network Wizard** opens and the **Naming** page opens

4. Follow steps 5–11 in *Creating an OpenLNS CT Network Design*. By default, the options selected in the existing network will be used in your new network; however, you can change any option. When you have completed the Plug-In Registration page, click **Next**. The **Network Properties: Domain** dialog box opens.



Domain Definition

Domain length (bytes): 1

Domain ID (in hex): DA

☑ Use randomly generated domain ID

A new random domain ID has been generated for the new network and is shown above.

5. A new random domain ID is generated. To use a specific domain ID, clear the **Use Randomly Generated Domain ID** check box and enter your ID in hexadecimal format in the **Domain ID** box. Click **Finish**.

6. If you are copying a network created with OpenLNS CT, your new OpenLNS CT network will open and you can skip the remaining steps.

7. If the existing OpenLNS CT network design you are copying was created in a previous version of LonMaker Turbo Editions (or earlier versions of the LonMaker tool), the **Synchronize OpenLNS CT for Windows Drawing** dialog box opens. This is because the OpenLNS CT SmartShapes in the drawing need to be updated to reflect the current OpenLNS CT.

8. In the **Synchronize OpenLNS CT for Windows Drawing** dialog box, the **Update Shapes to Current OpenLNS CT Version** check box will be selected by default. Select any other desired synchronization options and click **Next**.   The **Choose Synchronization Scope** dialog box opens.

9. Accept the defaults and click **Next**. The **Choose Shapes and Items to Update** dialog box will appear.

10. Accept the defaults and click **Finish**. The **Synchronization Status** dialog box opens as the synchronization process begins.

11. The **Synchronization Status** dialog box displays the progress of the synchronization. Once “Synchronization Complete” opens, click **OK** to open your OpenLNS CT network design.

For more information about the synchronization options in the preceding dialog boxes, see *Manual Network Resynchronization* in Chapter 7, *Maintaining Networks*.

Notes: If you open the copy of the OpenLNS CT network with OpenLNS CT set OnNet, a message opens reminding you that the copy of the OpenLNS CT network should not be attached to the physical network associated with the original. This is because the devices and routers will be updated to match the new network database. Verify that OpenLNS CT is not attached to the original network and then click **Yes** to proceed with your OpenLNS CT set OnNet. Click **No** to return to the Management Mode page and set OpenLNS CT OffNet.

Any devices you created and commissioned in your original network will have the same Neuron IDs in the copied version. To avoid conflicts, make sure the domain IDs of the original and copied OpenLNS CT network designs are unique, or make sure that the copied OpenLNS CT network design is not attached to the physical network associated with the original version when you are using it.

Monitoring will be disabled in the copy of the OpenLNS CT network. To enable monitoring, right-click a clear space in the drawing page and click **Enable Monitoring**. See Chapter 6, *Monitoring and Controlling Networks*, for more information on how to monitor and control application devices.

Opening an OpenLNS CT Network Design

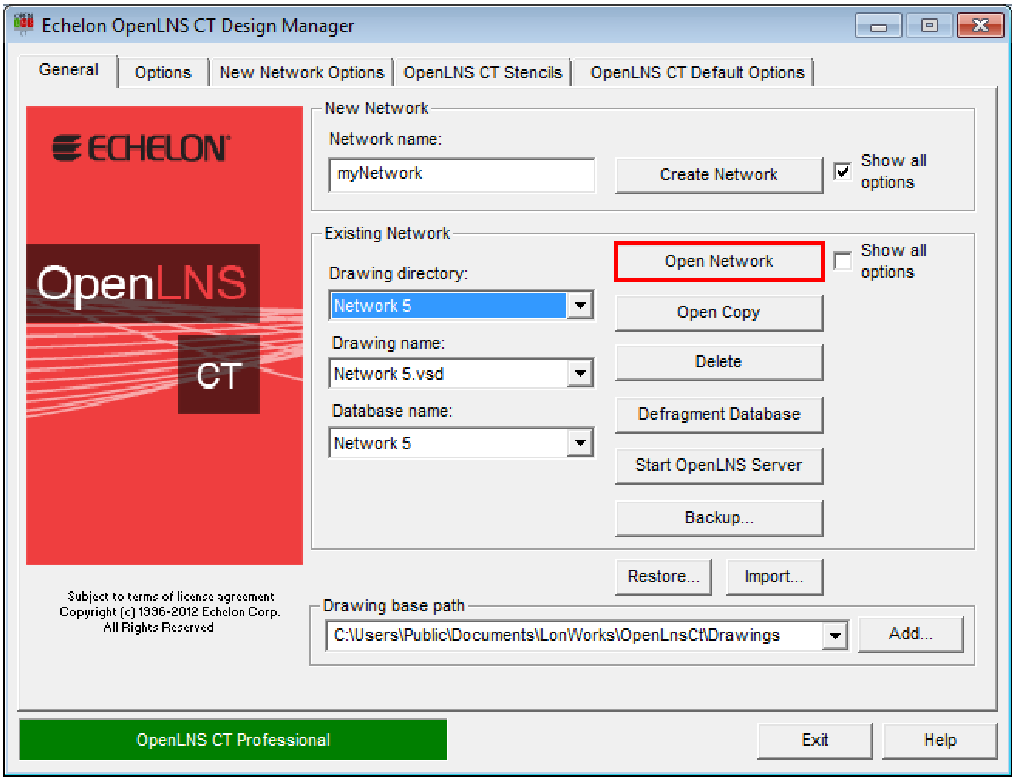
To open an existing OpenLNS CT network, follow these steps:

1. Click **Start** on the taskbar, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **OpenLNS CT**. The OpenLNS CT Design Manager opens.

2. Choose the OpenLNS CT network you want to open. Under **Existing Network**, select the desired drawing directory from the **Drawing Directory** list, and then select the network you want to open from the **Drawing Name** list. For networks with multiple drawings files, you can select any of the drawing files associated with the network.

3. To skip any or all of the pages in the Network Wizard when opening the existing network, clear the **Show all Options** check box under the **Existing Network** box. The pages in the Network Wizard will be skipped when you re-open the network provided that you previously selected the check box at the bottom of the page (the **Automatically Determine Server Location** check box for the Server Location page and the **Skip This Prompt** check box for all other pages). You can also select the options to skip the pages in the Network Wizard in the OpenLNS CT drawing (to do this, click **Add-ins**, click **LonWorks Network**, and click **Network Properties**). You cannot skip the User Logon page if you have defined user profiles for the network (see *User Profiles* in this chapter for more information).

4. Click **Open Network**.



Network name: myNetwork

☑ Show all options

Drawing directory: Network 5

Drawing name: Network 5.vsd

Database name: Network 5

☐ Show all options

Drawing base path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Drawings

Subject to terms of license agreement
Copyright (c) 1996-2012 Echelon Corp.
All Rights Reserved

5. If you are opening an OpenLNS CT network after starting OpenLNS CT for the first time, a message may appear asking you if you want to enable macros. You must enable macros for OpenLNS CT to function. See *Working with Digital Signatures* for more information.

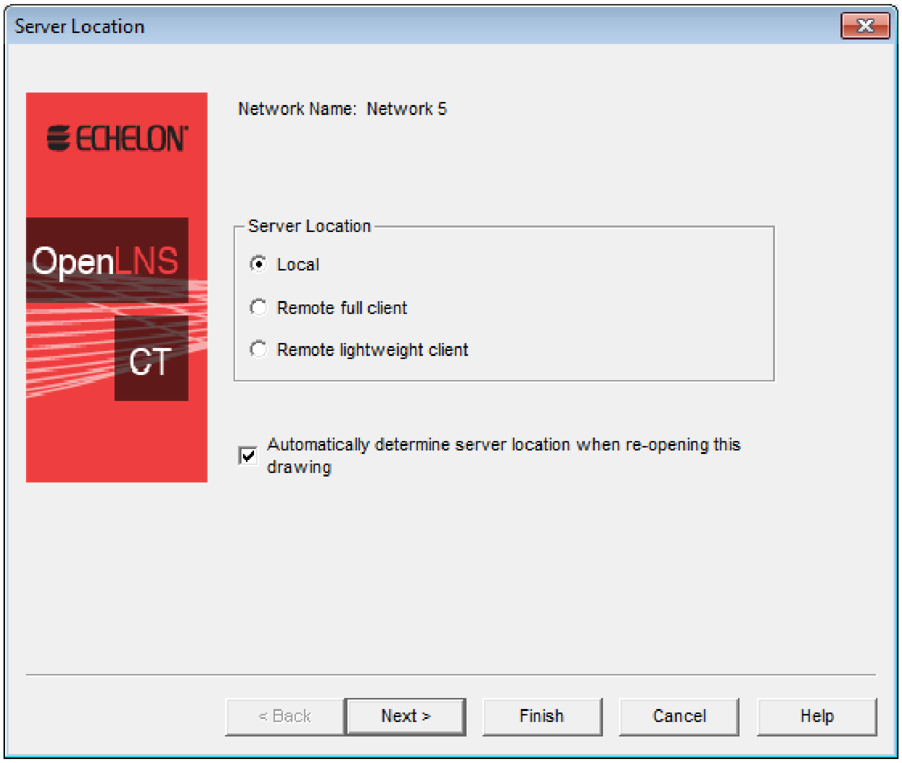
6. Visio starts, your OpenLNS CT drawing opens, and the **Server Location** page of the Network Wizard opens (unless you selected the **Automatically Determine Server Location** check box at the bottom of this page when you previously opened the network).

   By default, each subsequent page in the Network Wizard will appear every time you open the OpenLNS CT drawing unless you previously selected the **Skip This Prompt** check box on the page from the Network Wizard or from the **Network Properties** dialog in the drawing.

   Tip: You can click **Finish** at any point in the Network Wizard to skip the subsequent series of pages and open your OpenLNS CT network. Your OpenLNS CT network will be opened using the settings in the **Network Properties** dialog in the drawing at the time it was last saved.

7. Under **Server Location**, select the client type of the computer on which OpenLNS CT is running. See *OpenLNS CT Client Types* for more information on each of these client types.

   - Select **Local** if OpenLNS CT is running on the same computer as the OpenLNS Server.
   - Select **Remote Full Client** if OpenLNS CT is running on a computer different than the OpenLNS Server and the OpenLNS CT computer is attached to a LonWorks channel through an IP-852 channel.
   - Select **Remote Lightweight Client** if OpenLNS CT is running on a computer different than the OpenLNS Server and the OpenLNS CT computer communicates with the OpenLNS Server over TCP/IP.



Server Location

Network Name: Network 5

Server Location

● Local

○ Remote full client

○ Remote lightweight client

☑ Automatically determine server location when re-opening this drawing

   Select the **Automatically Determine Server Location** check box to have OpenLNS CT automatically determine the location of the OpenLNS Server when you re-open your OpenLNS CT network. If this check box is cleared, OpenLNS CT will prompt you to select the server location each time you open the drawing.

8. Click **Next**. The Network Interface page opens, unless the **Skip this Prompt** check box was previously selected. Follow steps 7–12 in *Creating an OpenLNS CT Network Design*. By default, the options previously selected for the network will be used; however, you can change any option.

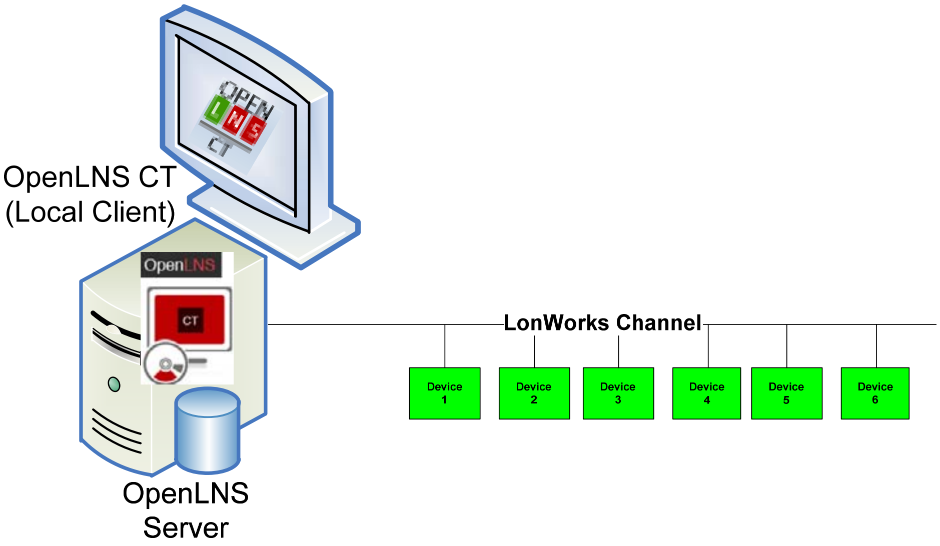
   Note: You may open multiple OpenLNS CT network designs by repeating the preceding steps. Each OpenLNS CT network design that you open will have its own instance of Visio.

OpenLNS CT Client Types

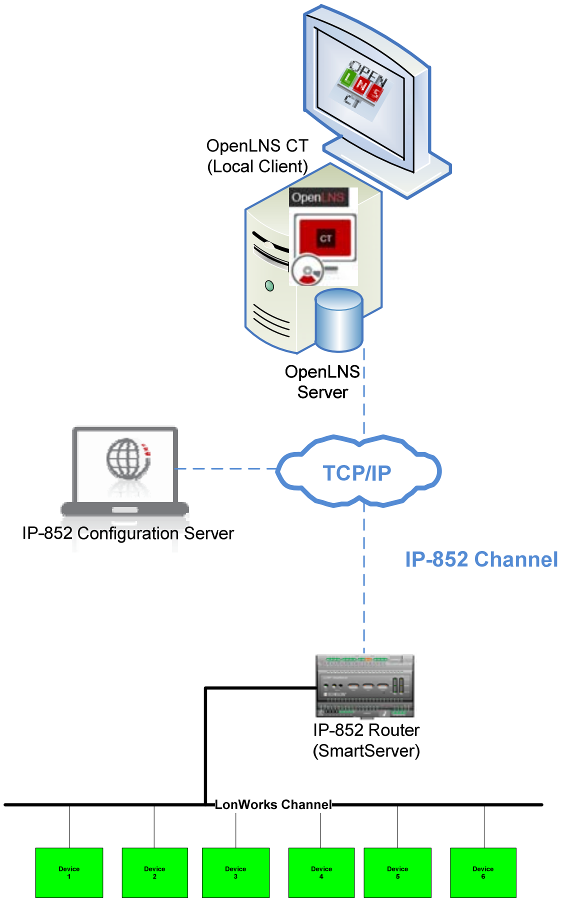
The OpenLNS CT *client type* determines the location of OpenLNS CT in relation to the OpenLNS server. OpenLNS CT supports three client types: *local client*, *remote full client*, and *remote lightweight client*.

Local Client

A *local client* is an OpenLNS CT computer that is also running OpenLNS Server. In this configuration, a local network interface is attached to the OpenLNS Server computer, and OpenLNS CT uses the local network interface to communicate directly with the physical network. The following figure demonstrates how OpenLNS CT connects to the physical network when it is running as local client.

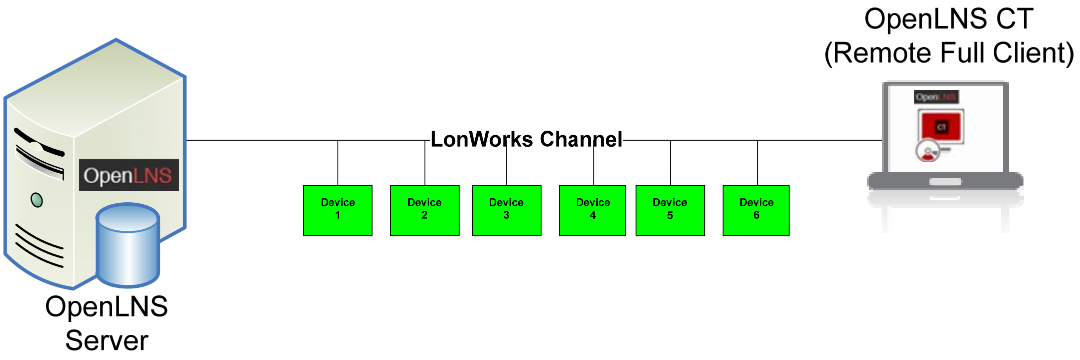


A local client can also be configured as a *local IP client.* In this configuration, the OpenLNS CT computer is attached to an IP-852 channel and communicates remotely with the network through an IP-852 router (SmartServer [with IP-852 routing option], i.LON 100 Internet Server [with IP-852 routing option], or i.LON 600 IP-852 Router). OpenLNS CT is still considered a local client because is it running on the same computer as the OpenLNS Server.



Remote Full Client

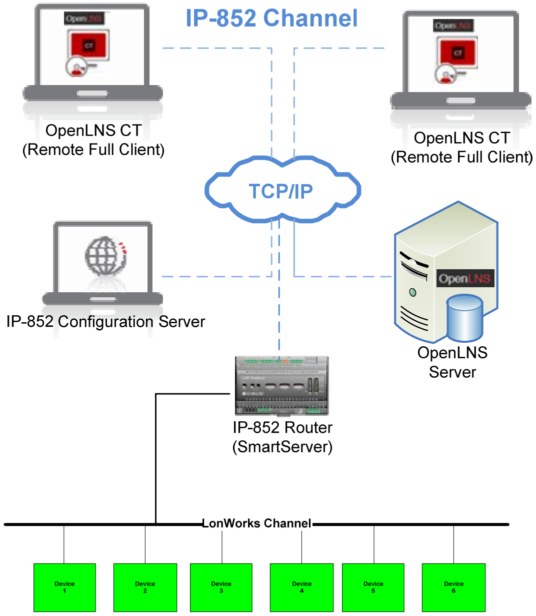
A *remote full client* is an OpenLNS CT computer that communicates with the OpenLNS Server (running on a separate computer) over a LONWORKS channel. In this configuration, the OpenLNS CT computer uses a network interface to communicate with the network (for example, a U10/U20 USB network interface or PCC-10, PCLTA-20, or PCLTA-21 network interface card). As a remote full client, network management tasks performed by OpenLNS CT are routed through the OpenLNS Server, but monitoring and controlling tasks are sent directly to the network.



One or more OpenLNS CT computers and the OpenLNS Server may also be connected to the network.

The following example demonstrates a scenario in which the OpenLNS CT computers communicate with the OpenLNS Server and the network over an IP-852 channel. The OpenLNS computers directly route management tasks to the network via an IP-852 router.

The IP-852 Configuration Server creates virtual channels that direct packet traffic and enable the IP-852 devices (OpenLNS CT computers, the OpenLNS Server, the IP-852 router, and other IP-852 devices) to communicate with each other over the TCP/IP network. To create the IP-852 channel, you must use the IP-852 Configuration Server to add and configure all the IP-852 devices to the network. For more information on using the IP-852 Configuration Server to create IP-852 channels, see *Using an IP-852 Network Interface* earlier in this chapter.



To use OpenLNS CT on a remote full client follow these steps:

1. Prepare for remote operations.

   a. Copy any custom stencils that you will need to the OpenLNS CT drawing folder on the OpenLNS Server computer.

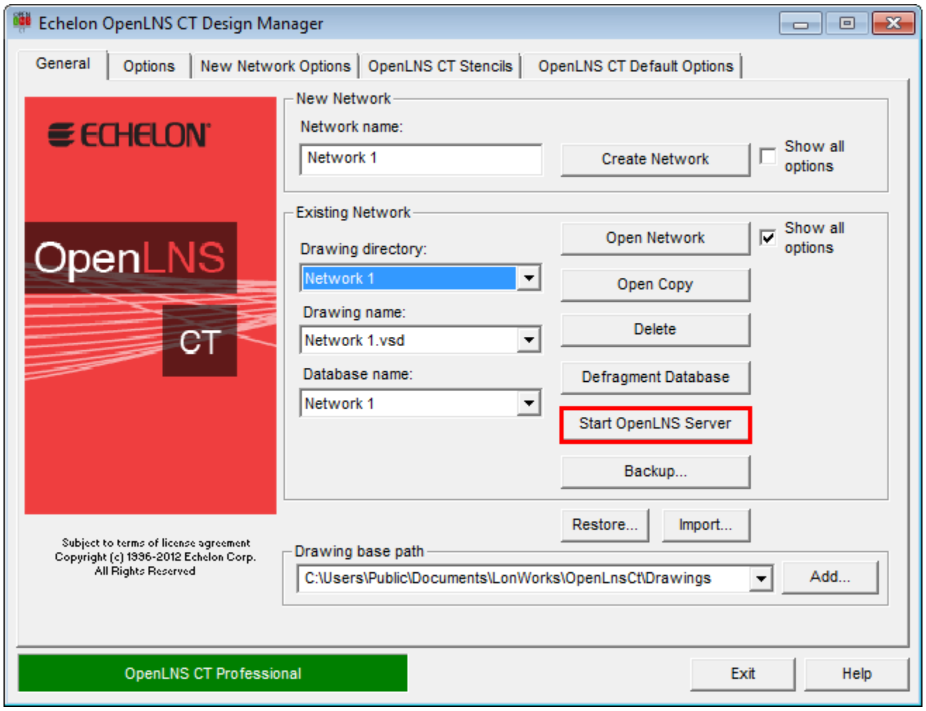
   b. Back up the OpenLNS CT drawing for the network you will be using and all other files in the drawing directory. See *Backing Up an OpenLNS CT Network Design* in Chapter 7, *Maintaining Networks*, for instructions.

   c. Save the OpenLNS CT backup file to a shared network folder. If you do not have a shared network folder, save the backup file to a CD, DVD, USB drive, floppy disk, or other storage media.

2. Start the OpenLNS Server.

   a. Click **Start** on the taskbar, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **OpenLNS CT**. The OpenLNS CT Design Manager opens.

   b. Choose the OpenLNS network database you want to open from the **Database Name** list and click **Start OpenLNS Server**.



New Network

Network name: Network 1

□ Show all options

Existing Network

Drawing directory: Network 1

Drawing name: Network 1.vsd

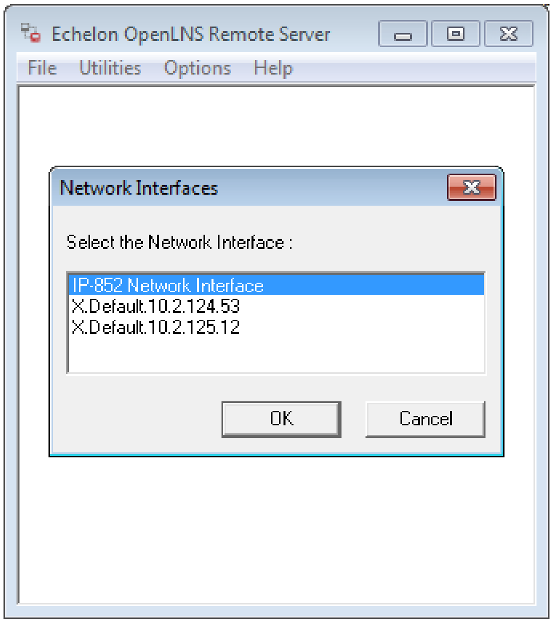
Database name: Network 1

☑ Show all options

Drawing base path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Drawings

Subject to terms of license agreement
Copyright (c) 1996-2012 Echelon Corp.
All Rights Reserved

c. The **Echelon OpenLNS Remote Server** application opens with the **Network Interfaces** dialog.

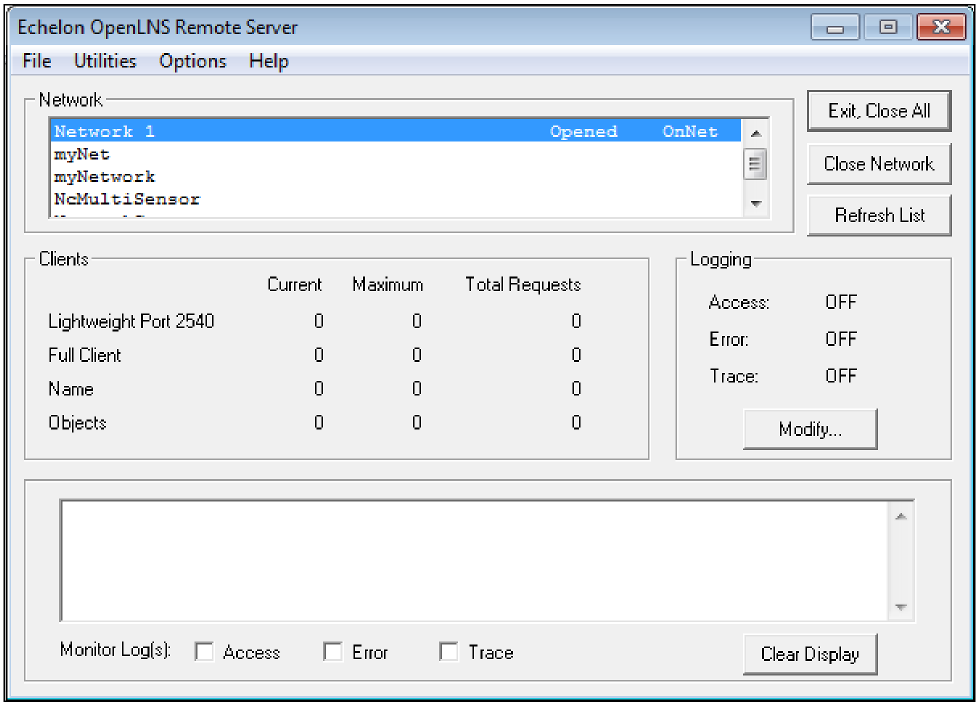


Select the Network Interface :

IP-852 Network Interface
X.Default.10.2.124.53
X.Default.10.2.125.12

d. Select the OpenLNS network interface or IP-852 network interface to be used to connect to the LONWORKS channel. An OpenLNS Server or remote client can be attached to an IP-852 channel using an IP network interface, such as an Ethernet network interface card (NIC), WiFi interface, or a modem channel with PPP software. The network interface to the LONWORKS channel must be defined and configured before opening an existing OpenLNS CT network on a remote full client. Click **OK**. The **OpenLNS Server** dialog opens.

e. The **OpenLNS Server** dialog displays the status for the open network. For more information on using the OpenLNS Server, open the **Help** menu and click **OpenLNS Server Help** to view the *OpenLNS Utilities Online Help System* file.



```
Network 1                                        Opened     OnNet
myNet
myNetwork
NcMultiSensor
```

	Current	Maximum	Total Requests
Lightweight Port 2540	0	0	0
Full Client	0	0	0
Name	0	0	0
Objects	0	0	0

Access: OFF

Error: OFF

Trace: OFF

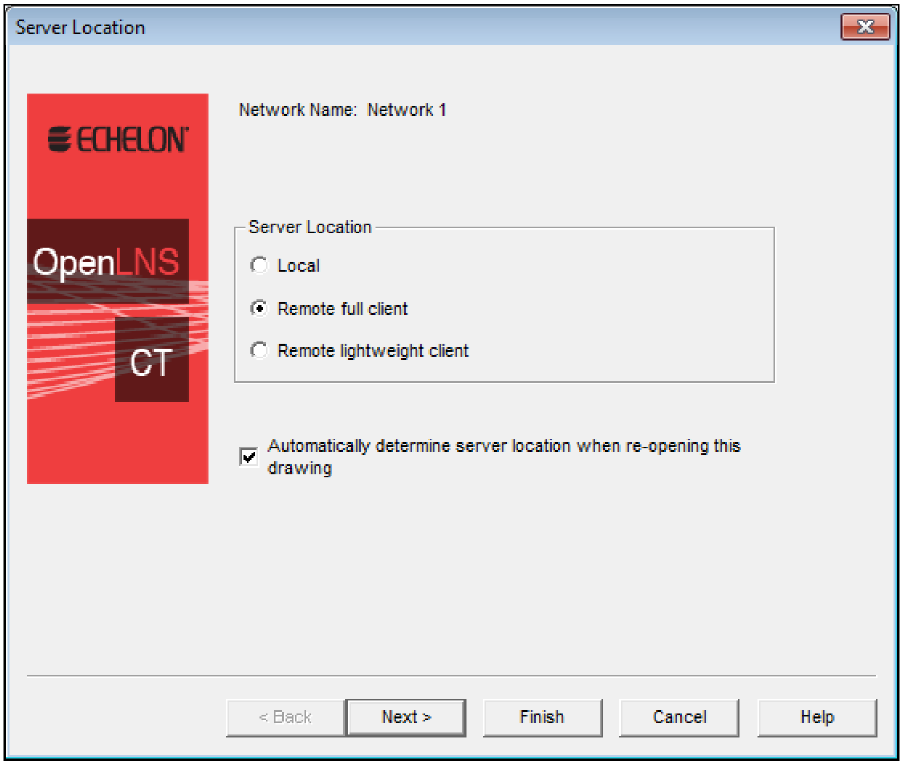
Monitor Log(s): □ Access □ Error □ Trace

f. To open an additional network, click **Start OpenLNS Server** on the OpenLNS CT Design Manager. The OpenLNS Server window opens. Highlight the next network to open. Click **Open Network**. The Network Interface page opens. Select the network interface. Click **OK**. Repeat this process for each additional network you want to open.

3. Copy the OpenLNS CT backup file that you created in step 1 to the remote client.

4. Start OpenLNS CT on the remote full client. Click **Start** on the taskbar, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **OpenLNS CT**. The OpenLNS CT Design Manager opens.

5. Restore the OpenLNS CT drawings from the backup you created in step 1. See *Restoring an OpenLNS CT Network* in Chapter 7, *Maintaining Networks*, for instructions. After restoring the drawing files, a prompt opens asking whether you want to open the network drawing. Click **Yes** to open the network drawing. The Network Wizard opens with the **Server Location** dialog.



Network Name: Network 1

Server Location

○ Local

● Remote full client

○ Remote lightweight client

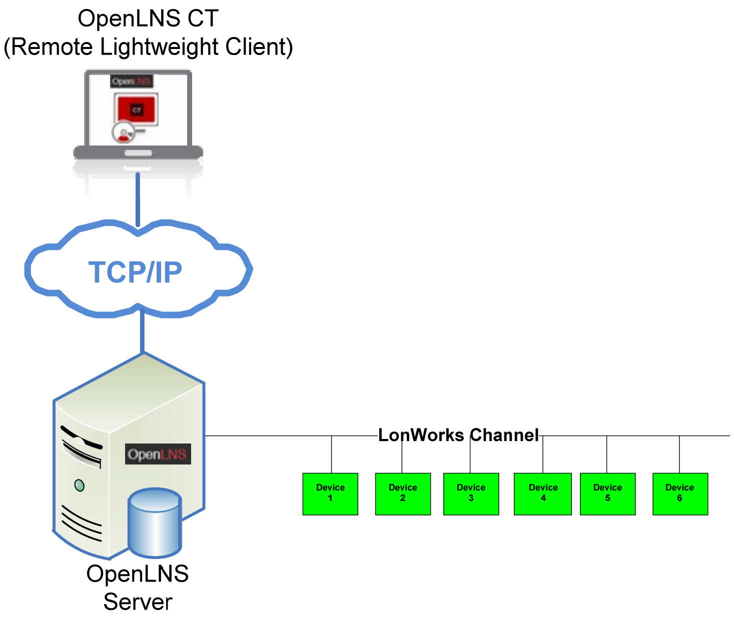
☑ Automatically determine server location when re-opening this drawing

6. Select **Remote Full Client** to access the network and then click **Next**. The Network Interface page opens.

7. Follow steps 7–12 in *Creating an OpenLNS CT Network Design*.

Remote Lightweight Client

A *remote lightweight client* is an OpenLNS CT computer that communicates with the OpenLNS Server (running on a separate computer) via TCP/IP. As a remote lightweight client, the network management, monitoring, and controlling tasks performed by OpenLNS CT are all routed through the OpenLNS Server. This means that all network variable reads and writes are routed through the OpenLNS Server, which can create bottlenecks in the OpenLNS Server if multiple remote lightweight clients are monitoring and control multiple data points simultaneously. A remote lightweight client can connect to the network without an IP-852 router.



To use OpenLNS CT on a remote lightweight client follow these steps:

1. Prepare for Remote Operations.

   a. Copy any custom stencils that you will need to the OpenLNS CT drawing folder on the OpenLNS Server computer.

   b. Back up the OpenLNS CT drawing for the network you will be using and all other files in the drawing directory. See *Backing Up an OpenLNS CT Network Design* in Chapter 7, *Maintaining Networks*, for instructions.

   c. Save the OpenLNS CT backup file to a shared network folder. If you do not have a shared network folder, save the backup file to a CD, DVD, floppy disk, or other storage media.

2. Start the OpenLNS Server application.

   a. Ensure that the OpenLNS Server computer is not set up to go into standby mode. Standby mode will prevent communication with remote clients.

   b. Click **Start** on the taskbar, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **OpenLNS CT**. The OpenLNS CT Design Manager opens.

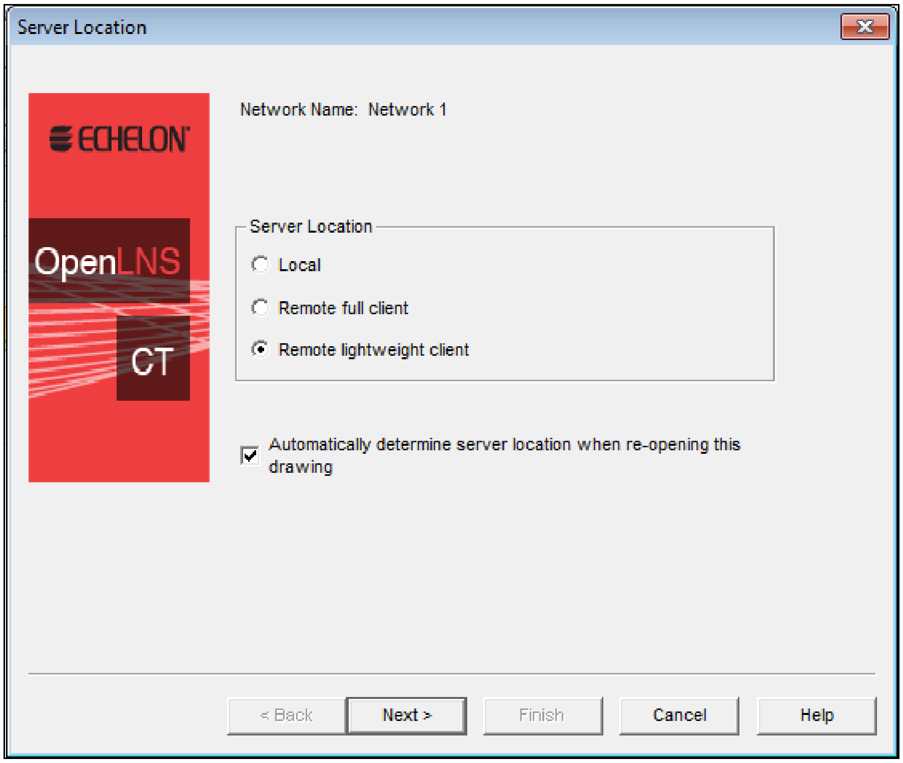
   c. Choose the OpenLNS network database you want to open from the Database Name list and click **Start OpenLNS Server**. The **Network Interfaces** dialog box opens.

   d. Select the OpenLNS Network Interface or an IP-852 Network Interface you want to use to connect to the LONWORKS channel. Click **OK**. The **OpenLNS Server** dialog opens.

3. Copy the OpenLNS CT backup file that you created in step 1 to the remote lightweight client.

4. Start OpenLNS CT on the remote lightweight client. Click **Start** on the taskbar, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **OpenLNS CT**. The OpenLNS CT Design Manager opens.

5. Restore the OpenLNS CT drawings from the backup you created in step 1. See *Restoring an OpenLNS CT Network* in Chapter 7, *Maintaining Networks*, for instructions. After restoring the drawing files, a prompt opens asking whether you want to open the network drawing. Click **Yes** to open the network drawing. The Network Wizard opens with the **Server Location** dialog.



Network Name: Network 1

Server Location

○ Local

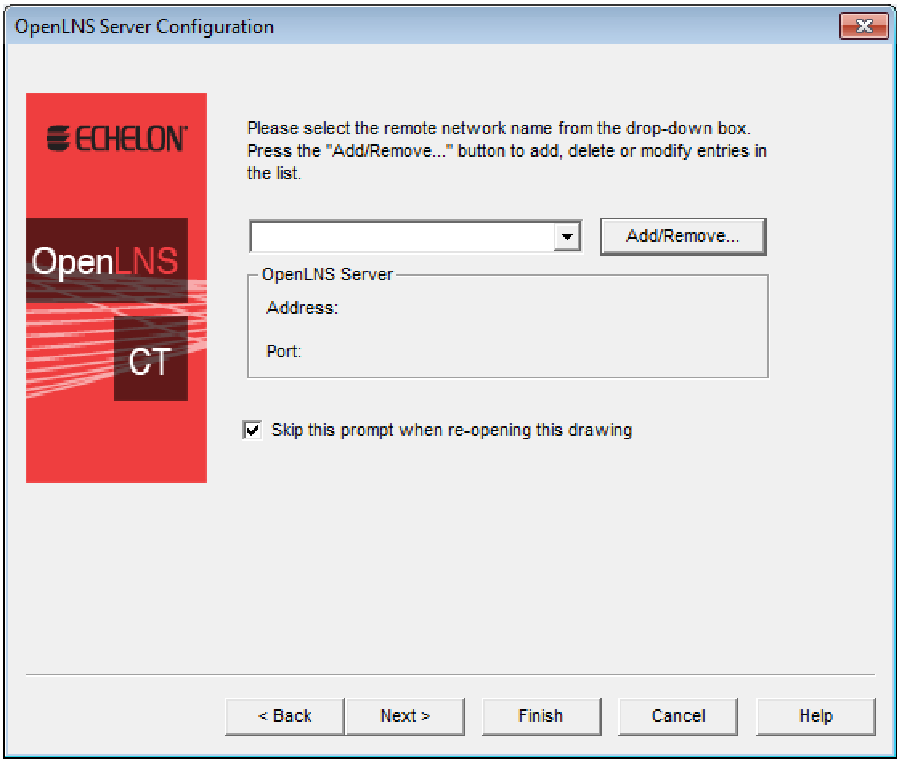
○ Remote full client

● Remote lightweight client

☑ Automatically determine server location when re-opening this drawing

6. Select **Remote Lightweight Client** to access the network and then click **Next**.

7. If the server address is configured correctly, you can skip to step 10, in which the **User Logon** window opens. If there is a problem with the server address, or you are opening a drawing created using LonMaker, the **Selecting a Remote Network Name** dialog opens.



Please select the remote network name from the drop-down box. Press the "Add/Remove..." button to add, delete or modify entries in the list.

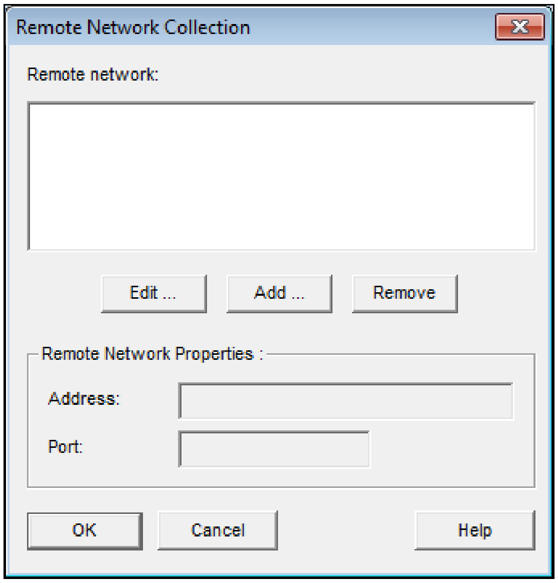
OpenLNS Server

Address:

Port:

☑ Skip this prompt when re-opening this drawing

8. Select an OpenLNS CT network to open. If a connection to this network has already been established, click **Next**, and skip to step 10. If you are accessing the selected network over an OpenLNS/IP interface for the first time, click **Add/Remove**. The **Remote Networks Collection** dialog box opens.



9. Under **Remote Network Properties**, the **Address** box displays the address of the OpenLNS Server, and the **Port** box displays the port used for the OpenLNS/IP connection (the default is 2540, which is the default port for the OpenLNS Server). Both boxes are read-only. Click **Add** or **Edit** to create a new remote network or modify an existing one. Click **OK**, select the newly created remote network, and then click **Next**.

10. Follow steps 9–12 in *Creating an OpenLNS CT Network Design*.

Using Network Service Device SmartShapes

A *network service device* (NSD) is a local or remote client capable of monitoring and controlling the network. Every OpenLNS Server and OpenLNS CT remote full client in a network may be represented in an OpenLNS CT drawing with an NSD SmartShape. When an OpenLNS CT network is created, it initially contains an NSD SmartShape for the OpenLNS Network Interface that represents the OpenLNS Server.

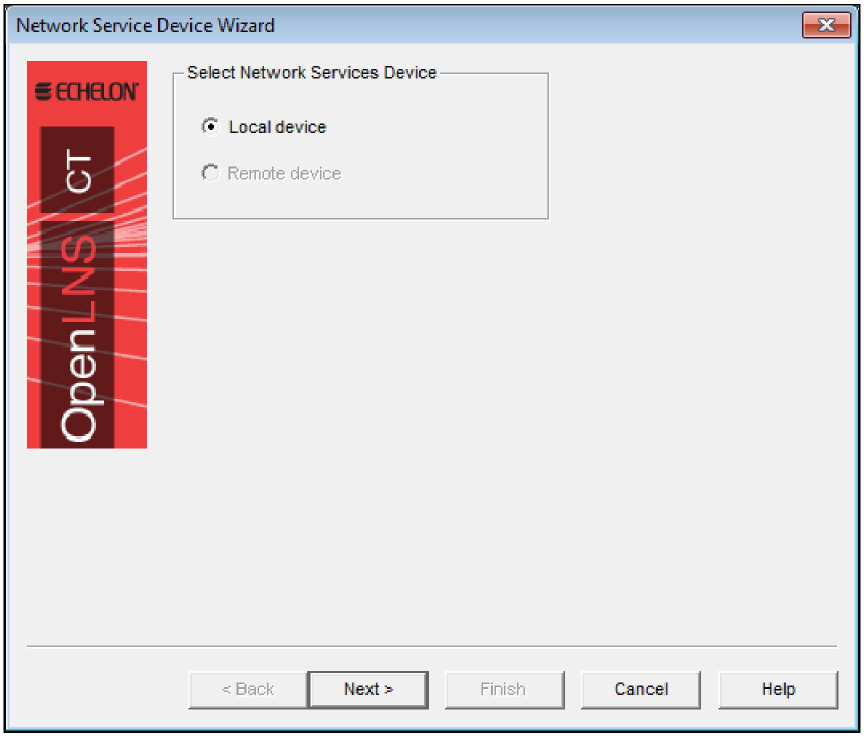
If a remote full client is connected to a network, you can add an NSD SmartShape that represents the remote full client to the OpenLNS CT drawing. Creating an NSD SmartShape makes that remote full client permanent in the drawing; otherwise, OpenLNS automatically creates and deletes it each time you open and close the network.

After you have added a network service device, you can add a virtual functional block to that NSD, add network variables to the virtual functional block, and then connect the network variables. This enables the NSD to monitor devices directly as described in *Binding Network Variables to the Host* in Chapter 6, *Monitoring and Controlling Devices.* Additionally, this lets you perform the network management tasks described in Chapter 8, *Managing Networks*.

To add a NSD SmartShape to an OpenLNS CT drawing, follow these steps:

1. Drag the Network Service Device SmartShape (🖥) from the **OpenLNS CT Basic Shapes** stencil to your OpenLNS CT drawing.

2. A dialog opens that allows you to select whether the network service device being added represents the local NSD (only available if you are a remote client and you are using a NSD that does not have a SmartShape associated with it) or a remote NSD.



Network Service Device Wizard

Select Network Services Device

● Local device

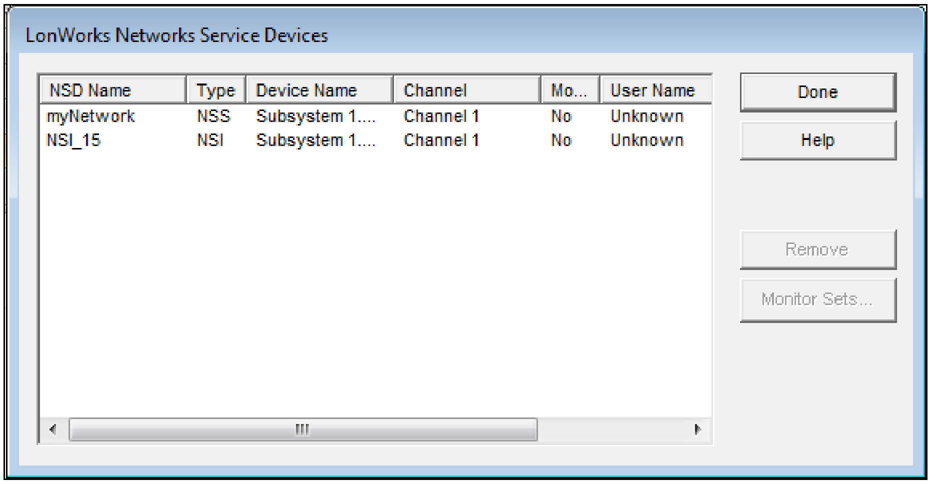
○ Remote device

3. If you selected **Remote Device** in step 2, a dialog opens which allows you to select the local or remote network service device to be associated with this network service device SmartShape. If there is only one network service device with no corresponding SmartShape, it will be automatically selected.

Note: If you need to remove a NSD from the physical network, first remove the network service device SmartShape from the drawing by right-clicking it and selecting **Delete** from the shortcut menu.

Listing Network Service Devices

If you are using OpenLNS CT with remote clients, you can view a list of all OpenLNS clients connected to the network. Each client is represented by a network service device. To obtain a list of all NSDs in a network, click **Add-ins**, point to **LonWorks Network**, and then select **Network Service Devices**. The **LonWorks Network Service Devices** dialog box opens.



NSD Name	Type	Device Name	Channel	Mo...	User Name
myNetwork	NSS	Subsystem 1....	Channel 1	No	Unknown
NSI_15	NSI	Subsystem 1....	Channel 1	No	Unknown

This dialog box lists all NSDs in this network, regardless of whether a SmartShape has been created for each NSD. For each NSD, the following information is listed:

NSD Name — Specifies the name of the NSD, which is assigned by the OpenLNS Server.

Type — Specifies the NSD type. This can either be **NSS** (the OpenLNS Server) or **NSI** (a remote full client).

Device Name — Specifies the NSD name in the format of <*subsystem name*>.<*network service device name*>.

Channel — Specifies the channel on which the NSD is attached.

Monitor Sets? — Indicates whether monitor sets are defined on the NSD.

User Name — Specifies the user name associated with the NSD. See *User Profiles* for more information on user names.

Application — Specifies the name of the application currently active on the NSD.

If there is no application, or the application does not provide its name, “Unknown” is displayed.

If the network service device is not responding, “No Response” is displayed.

Click **Remove** to delete the selected network service device from the network. This option is generally used to remove remote network service devices that are no longer in use, which improves the performance of OpenLNS CT.

Click **Monitor Sets** to open the **Monitor Sets** dialog box, in which you can view and remove monitor sets in the OpenLNS CT network. To delete a monitor set, locate the monitor set to delete from the list, right-click the monitor set, and then click **Remove** on the shortcut menu.

Upgrading Network Service Devices

When you upgrade OpenLNS CT or switch between an OpenLNS high performance network interface (LonTalk layer 2) and an OpenLNS network interface (LonTalk layer 5), you must upgrade your network service device. When you open the network and either of these conditions is detected, OpenLNS CT will ask whether you want to upgrade. You should typically perform the upgrade. However, if you are only using a different network interface temporarily, you may choose not to upgrade. In this case, communication with physical devices is limited until the network is reopened with the original network interface. If initially you choose not to upgrade the network service device and decide later to upgrade, select the NSD SmartShape, right-click it, and then click **Upgrade NSD** on the shortcut menu.

When moving from an OpenLNS high performance network interface (LonTalk layer 2) to an OpenLNS network interface (LonTalk layer 5), all connections and monitor sets may not be created on the new network interface. If this happens, a dialog displays the items that will be lost and gives you an opportunity to cancel the upgrade.

Replacing a Local Network Service Device

When working as a remote full client, you can configure your local NSD to assume the configuration (monitor sets, connections, and so on) of another, previously defined network service device. You can do this by replacing a local NSD with the desired NSD SmartShape. The following conditions must be met:

- Your current local NSD cannot have an NSD SmartShape associated with it. If it does, you must first delete the NSD SmartShape from the drawing along with any associated functional block SmartShapes and connections.
- The desired NSD must have a SmartShape in the drawing. If necessary, select one from the template and drag it to the drawing.

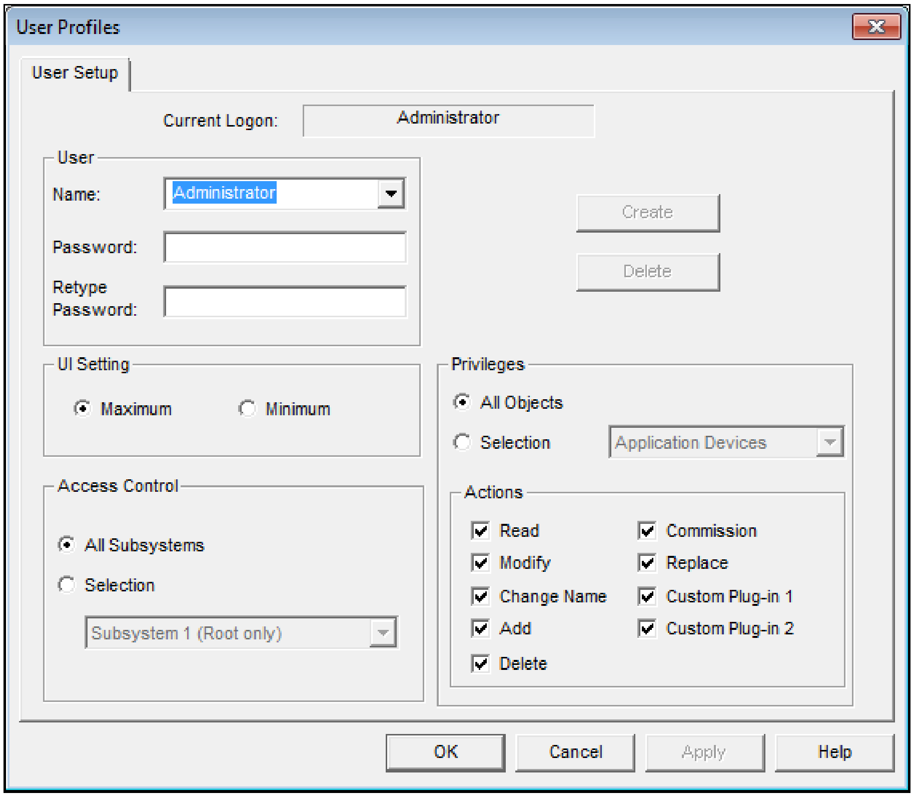
When these conditions are met, select the desired NSD SmartShape, right-click it, and then click **Make Local NSD** on the shortcut menu. You have full access to the configuration of the selected NSD.

Note: The selected NSD must not currently be in use by another remote client. If it is, performing this operation will disconnect the other user from the network.

User Profiles

With user profiles, you can set which parts of an OpenLNS CT drawing users can access and the actions they can perform. For example, you could have a scenario in which a system integrator has access to the entire network and can add, delete, or change channels, routers, devices, functional blocks and all other objects. Operators, though, would only be able to access the subsystem they are installing and be restricted to commissioning devices and routers in that subsystem.

To enable user profiles on an OpenLNS CT drawing, you must first assign an administrator password. To set the administrator password, click **Add-ins**, point to **OpenLNS CT**, select **User Profiles**, enter the administrator password in the **Password** and **Retype Password** boxes, and then click **Apply**. Passwords are case sensitive.



Current Logon: Administrator

User

Name: Administrator

Password:

Retype Password:

UI Setting

● Maximum ○ Minimum

Access Control

● All Subsystems

○ Selection

Subsystem 1 (Root only)

Privileges

● All Objects

○ Selection Application Devices

Actions

☑ Read ☑ Commission

☑ Modify ☑ Replace

☑ Change Name ☑ Custom Plug-in 1

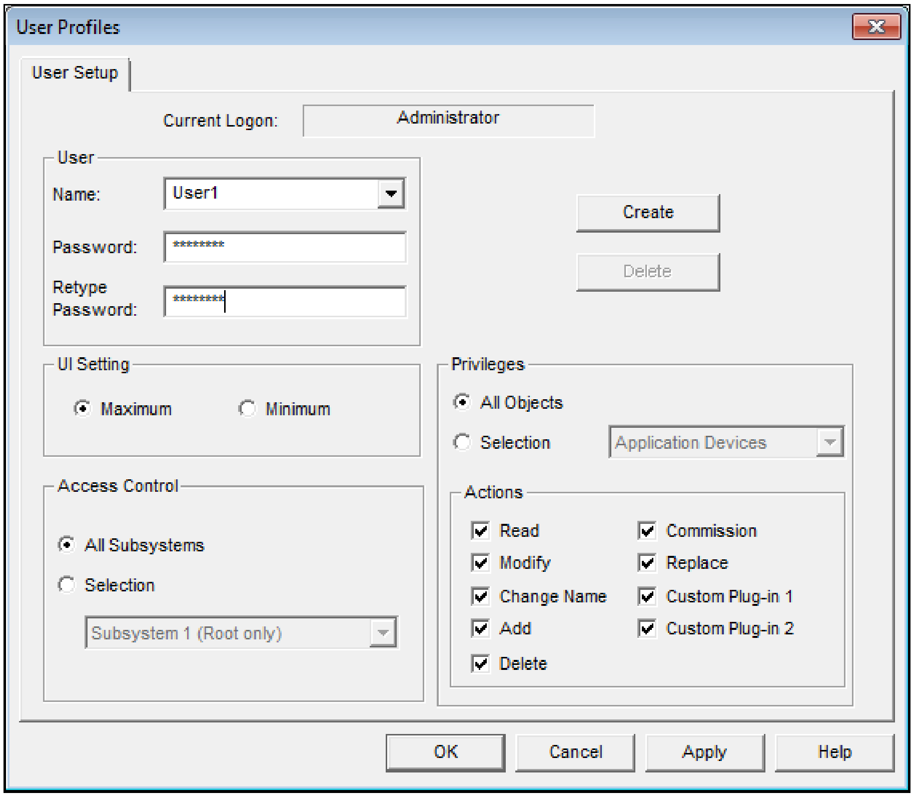
☑ Add ☑ Custom Plug-in 2

☑ Delete

Creating a New User Profile

To create a new user profile, follow these steps:

1. In the OpenLNS CT drawing, click **Add-ins**, point to **OpenLNS CT**, and then select **User Profiles**. The **User Profiles** dialog box opens.
2. Enter the name of the new user name in the **Name** box, and the password in the **Password** and **Retype Password** boxes. Both the user name and password are case sensitive.



User Profiles

User Setup

Current Logon: Administrator

User

Name: User1

Password: ********

Retype Password: ********

UI Setting

● Maximum ○ Minimum

Privileges

● All Objects

○ Selection Application Devices

Access Control

● All Subsystems

○ Selection

Subsystem 1 (Root only)

Actions

☑ Read ☑ Commission

☑ Modify ☑ Replace

☑ Change Name ☑ Custom Plug-in 1

☑ Add ☑ Custom Plug-in 2

☑ Delete

3. Set the user interface, access, and privileges of the new user.

UI Setting

If you are using Visio 2003, you can select the set of OpenLNS CT and Visio menus that will be available to the user. This option is not available if you are using Visio 2010. If you are using Visio 2003, you have two choices:

- **Maximum**. This setting provides the user a full set of toolbars, menus, and menu items for advanced use of OpenLNS CT.
- **Minimum**. This setting provides the user only the basic menus, menu items, and toolbars necessary to use OpenLNS CT.

Access Control

Select the type of access the user has in the OpenLNS CT drawing. You have two choices:

- **All Subsystems**. The user has unrestricted access to all of the subsystems in the network.
- **Selection**. The user's access is restricted to the selected subsystem. This option can only be applied to the root subsystem or any one of the subsystems one level below it. If you select the root subsystem, the user cannot access any of the subsystems below it.

Privileges

Select which actions the user can perform on objects in the OpenLNS CT drawing.

You can set privileges for reading, modifying, changing the name, adding, deleting, commissioning, and replacing objects, as well as the use of plug-ins for the objects. The objects for which you can set privileges consist of the network, subsystems, channels, application devices, routers, functional blocks, connections, network variables, and configuration properties.

While **All Objects** is selected, the privileges you set apply to all objects. For example, to prevent this user from modifying any object, click **All Objects** and then clear the **Modify** check box.

While **Selection** is selected, the privileges you set apply only to the object specified in the **Selection** list. For example, to enable this user to be able to modify functional blocks but not devices, click **Selection**, select **Functional Blocks**, and then select the **Modify** check box. Then select **Application Devices** from the **Selection** list and clear the **Modify** check box.

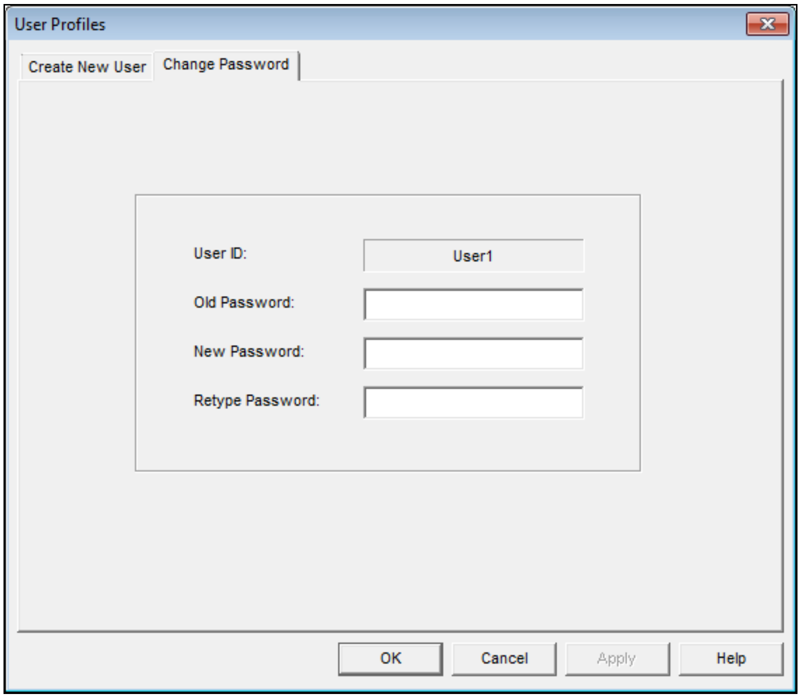
4. Click **Create** if you are logged on as the Administrator; otherwise, click **OK**.

Note: Once the new user profile has been created, that user can open the network drawing and then change their password or create new user profiles. A user can grant a new user only those privileges that they have themselves.

Changing Passwords

All users can change their passwords. To change a password, follow these steps:

1. In the OpenLNS CT drawing, click **Add-ins**, point to **OpenLNS CT**, and then select **User Profiles**. The **User Profiles** dialog box opens.
2. Click the **Change Password** tab.



User ID: User1

3. Type the old password in the **Old Password** box, enter the new password in the **New Password** and **Retype Password** boxes, and then click **OK**. The password will be changed.

Modifying and Deleting User Profiles

Only the Administrator can modify or delete a user profile. To modify or delete a user profile, follow these steps:

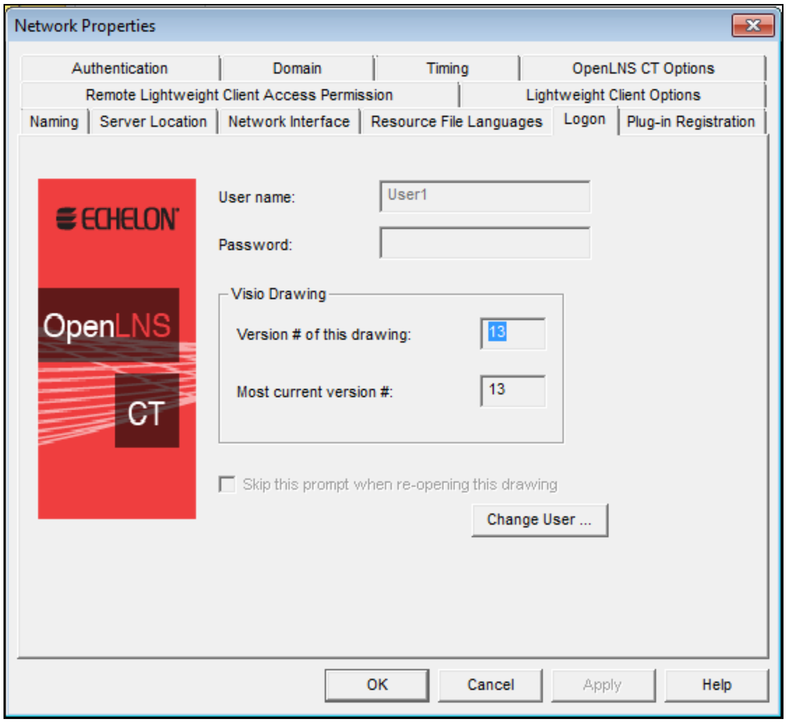
1. In the OpenLNS CT drawing, click **Add-ins**, point to **OpenLNS CT**, and then select **User Profiles**. The **User Profiles** dialog box opens.
2. Select a user profile to modify from the **Name** list.
3. To configure the user profile, change the options for the user interface, access and/or privileges, and then click **Apply**. OpenLNS CT saves the changes for the selected user. If the user is currently logged onto the network drawing from another computer, they must close and re-open OpenLNS CT for the changes to take effect.

   To delete the user profile, click **Delete**. A confirmation prompt opens. Click **Yes** to delete the selected user.

Changing User Profiles in an OpenLNS CT Drawing

You can switch user profiles in an OpenLNS CT drawing. To change users, follow these steps:

1. In the OpenLNS CT drawing, click **Add-ins**, point to **LonWorks Network**, and then select **Network Properties**. The **Network Properties** dialog box opens.
2. Click the **Logon** tab.



User name: User1

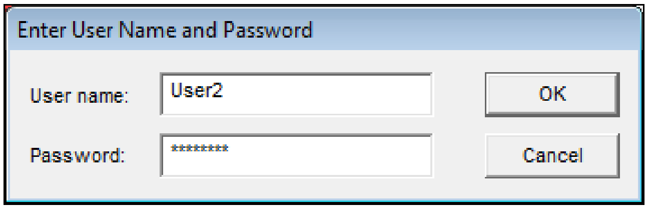
Password:

Version # of this drawing: 13

Most current version #: 13

□ Skip this prompt when re-opening this drawing

3. Click **Change User**. The **Enter User Name and Password** dialog box opens.



User name: User2

Password: ********

4. Enter the user name and password of the user you want to logon as.
5. Click **OK**. You will be logged on as the selected user.

Using OpenLNS CT Remotely with User Profiles

When you open an OpenLNS CT network on a remote client, you will be prompted for your user name and password (provided that the administrator password has been set for the network). After logging on to the network, you will have the same privileges as you do when working on a local client.

4

Designing Networks

This chapter describes how to design a network using OpenLNS CT. It covers how to create the following objects in an OpenLNS CT drawing: application devices, functional blocks, network variables, routers, channels, and subsystems. It explains how to connect network variables. It explains working with OpenLNS CT SmartShapes and layers, customizing the OpenLNS CT user interface, and using OpenLNS CT with AutoCAD drawings.

Creating a LonWorks Network

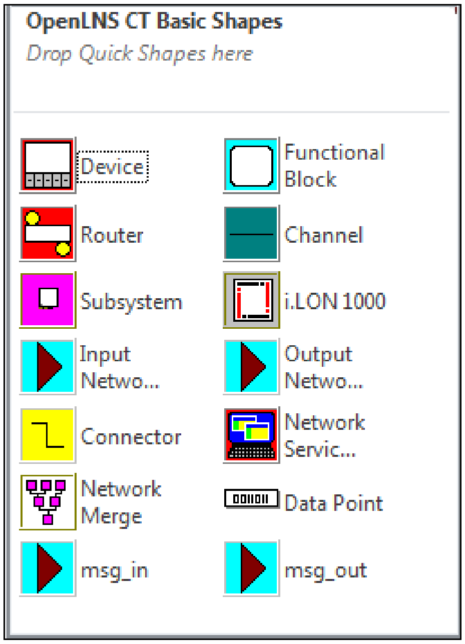
You can design a LonWorks network and then install it using OpenLNS CT. To design a LonWorks network with OpenLNS CT, you create an OpenLNS CT drawing. This entails adding OpenLNS CT SmartShapes for channels, routers, devices, and functional blocks; making logical connections between devices so they can send and receive data to and from each other; and organizing the network into subsystems. You can add OpenLNS CT SmartShapes at any time under any installation scenario—you can add OpenLNS CT SmartShapes while you are offsite designing the network, or you can add them when you are onsite commissioning devices and performing network maintenance.

To install the network, you commission the application devices and routers in it. Commissioning associates the device SmartShapes created in the OpenLNS CT drawing with the physical devices on the network.

This chapter describes how to create an OpenLNS CT drawing. Commissioning devices is covered in Chapter 5, *Installing Networks*.

Creating an OpenLNS CT Drawing

To create an OpenLNS CT drawing, you add application devices, routers, functional blocks, and other OpenLNS CT objects to the drawing page. To add an OpenLNS CT object, you drag its OpenLNS CT SmartShape from a stencil onto the drawing page. If required, the appropriate wizard will then open and guide you through the process of creating that SmartShape. The following sections describe how to create the objects in the **OpenLNS CT Basic Shapes** stencil.



Notes:

- Several of the SmartShapes in the **OpenLNS CT Basic Shapes** stencil are discussed elsewhere in this user's guide. For creating Network Service Device SmartShapes, see *Using Network Service Device SmartShapes* in Chapter 3, *Getting Started*; for creating Data Point SmartShapes, see *Using Data Point SmartShapes* in Chapter 6, *Monitoring and Controlling Networks*; and for using the Network Merge SmartShape, see *Merging OpenLNS CT Networks* in Chapter 7, *Maintaining Networks*.

- If you have licensed OpenLNS CT as part of the NodeBuilder tool, the **OpenLNS CT Basic Shapes** stencil is replaced with a **NodeBuilder Basic Shapes** stencil that includes Release Target Device and Development Target Device shapes.

Creating Application Devices

An *application device* consists of hardware and software that runs an application and communicates with other devices using the ISO/IEC 14908-1 Control Network Protocol. An application device in an OpenLNS CT drawing corresponds to a physical device on the network.

To add an application device to the network, you choose the device SmartShape and name, as well as the template, location, channel, and optional description for the device. You can add a device to without OpenLNS CT being attached to the network.

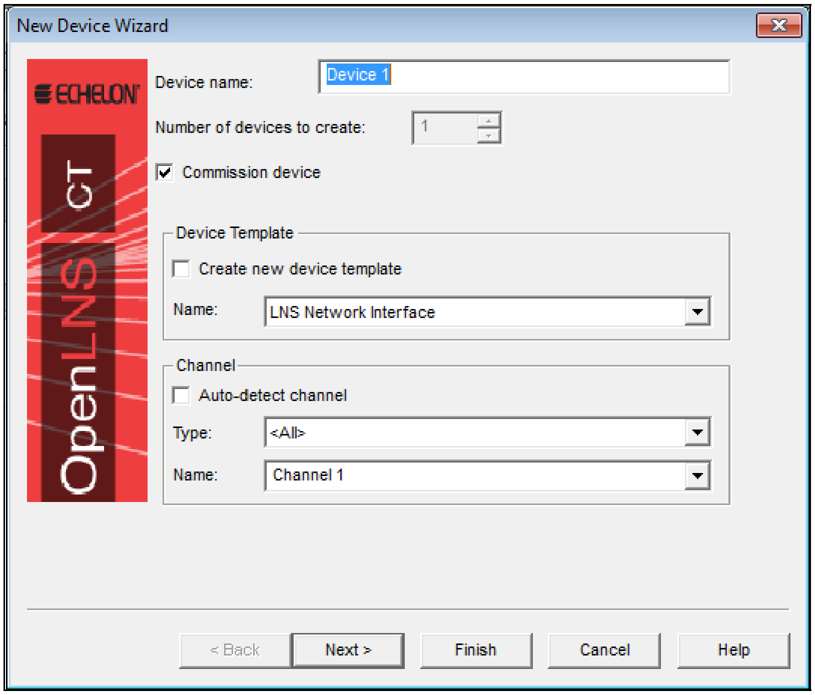
When you add a device to your OpenLNS CT drawing, you must define it, but you do not have to commission the device until you are ready to install it. This is how you design an engineered system. You define and configure devices offsite; bring the network database onsite; and then commission the devices. Under the ad-hoc installation scenario, you define and commission the devices in one step while onsite.

You can create an application device in one step. To do this, you enable automatic channel selection and then drag a custom device master SmartShape to your OpenLNS CT drawing, near the channel on which the device will be attached. Otherwise, the New Device Wizard opens when you add a device SmartShape to your OpenLNS CT drawing. See *Using Automatic Channel Selection for Devices* in this section for how to enable this feature. See Chapter 12, *Creating and Using OpenLNS CT SmartShapes and Stencils*, for how to create and use custom device master SmartShapes.

The first page of the New Device Wizard includes a **Commission Device** check box for commissioning the device. Commissioning a device associates the physical device on the network with the device SmartShape that you added to the OpenLNS CT drawing. To commission the device immediately after defining it, select this check box.

To add a device to an OpenLNS CT drawing, follow these steps:

1. Drag a Device SmartShape (▭) from the **OpenLNS CT Basic Shapes** stencil or other stencil to the drawing page. The New Device Wizard opens unless you enabled automatic channel selection and dropped a custom device master SmartShape near a channel SmartShape.



New Device Wizard

Device name: Device 1

Number of devices to create: 1

☑ Commission device

Device Template

□ Create new device template

Name: LNS Network Interface

Channel

□ Auto-detect channel

Type: <All>

Name: Channel 1

2. Enter the following information for the device.

Device Name

Enter the name of the device. This name must be unique within the currently displayed subsystem. The device name may be up to 85 characters and it may include embedded spaces; however, the name may not include period, backslash, colon, forward slash, or double quote characters.

The default device name is the abbreviated device SmartShape name followed by a dash, a space, and the instance number “1”. The instance number in the default device name is incremented one for each subsequent device of that type you create. For example, the default device name of the first analog input device you create is “AI- 1”, the second is “AI- 2”, and so on.

You can change this default naming convention by clicking **Add-ins**, clicking **OpenLNS CT**, clicking **OpenLNS CT Options**, and then selecting **Naming** from the **Options Category** list. See *Naming Options* in Appendix A for information on how to use the options on this page to change the default naming convention.

Number of Devices to Create

Select the number of devices to be created. If you create more than one device, OpenLNS CT will automatically increment the instance number in the default names of the additional devices and place them on the drawing adjacent to the first SmartShape. If the device name does not contain an instance number at the end, OpenLNS CT will append the instance number 1 to the second device (for example, “Device” for the first device becomes “Device 1” for the second device), and it will increment the instance number for additional devices.

Commission Device

Select this check box to commission the device after defining it. This option is disabled if you are creating more than one device.

Device Template

Create a New Device Template

Select this check box to create a new device template for the device. If you select this check box, the Specify Device Template window opens after you click **Next.** In this window, you can define a new template by uploading the device interface definition from the device or importing an existing device interface (XIF) file from your computer.

This check box is unavailable for device SmartShapes with pre-defined device templates such as the LonPoint device SmartShapes.

Name

Displays the OpenLNS device template that defines the device's interface. If the device SmartShape has a pre-defined device template (such as the LonPoint device SmartShapes), that template will appear in the box. Otherwise, this box will be blank.

If you are creating a new device template, enter a name for the new device template. If you are using an existing device template, select one from the list.

Channel

Auto-detect Channel

Select this check box to automatically detect the channel on which the device is attached. For this option to function, OpenLNS CT must be attached to the network, and all routers between the OpenLNS CT drawing network interface and the device must be installed, commissioned, and online. Otherwise, an error will be returned during commissioning.

Selecting this check box disables the **Type** and **Name** boxes below.

Do not select this check box if you are using routers that are configured as repeaters or bridges.

If this check box is cleared, you must explicitly select which channel the device is on from the **Name** box.

Type

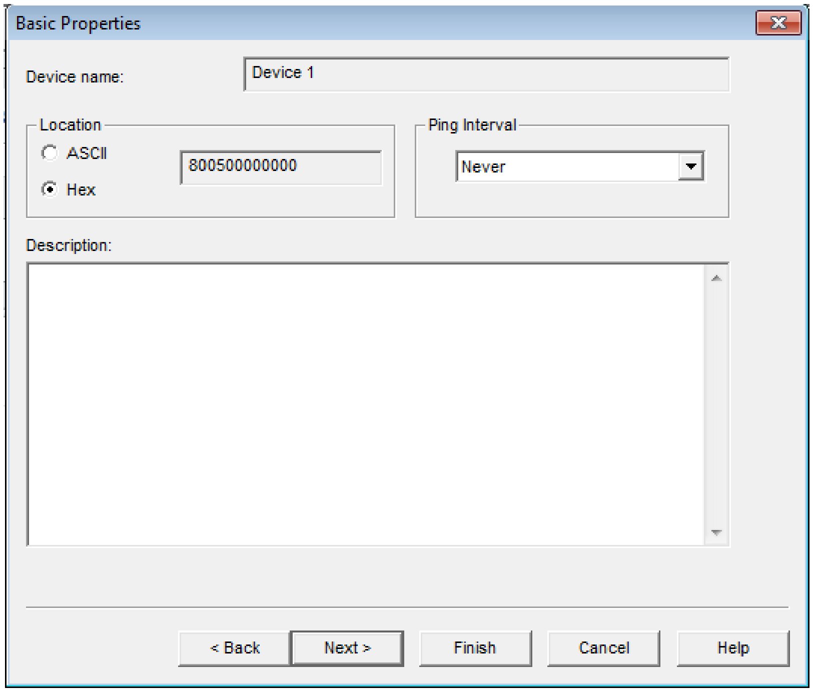
Select the channel type for the device. By default, this box displays the channel type of the channel currently displayed in the **Name** list. You can use it to filter the existing channels in the **Name** list to those with a specific channel type.

You can limit the channel types displayed by specifying the channel type in a custom device SmartShape.

Name

Select the channel that you will attach to the device. You can create a new channel SmartShape by entering a new channel name. The channel name must be unique within the network and be no more than 85 characters long. If you are creating a new channel, specify the channel type in the **Type** box.

3. Click **Next**. If you are creating a NodeBuilder device, a window opens allowing you to select the NodeBuilder device template; otherwise skip to step 4. After you have created and/or selected the NodeBuilder device template, click **Next** (see the *NodeBuilder FX User's Guide* for more information). The Device Properties window opens.



Basic Properties

Device name: Device 1

Location
○ ASCII
● Hex
800500000000

Ping Interval
Never

Description:

4. Enter the following information:

Location

Specifies a 6-byte location string that documents the device's location within the network.

To store the device's subsystem ID in this field, follow these steps:

1. Click **Add-ins,** point to **OpenLNS CT**, and then click **OpenLNS Options**. Select **Recovery** from the options category list and then select the **Set Location Property to Subsystem ID** check box under **Subsystem Recovery Options**.

2. Resynchronize the network. When you resynchronize the network, select the **Update Location Path and ID in Devices** check box in the **Synchronize OpenLNS CT Drawing** dialog. The device's current subsystem ID will be stored in this field. The OpenLNS CT tool can then use this information to place devices in their correct subsystems in the event you need to use the OpenLNS database Recovery Wizard to recover a lost network database.

   See *Recovering an OpenLNS CT Network* in Chapter 7, *Maintaining Networks*, for more information on the network recovery process.

If the **Set Location Property to Subsystem ID** check box is cleared, you can enter an optional location description for the device. You can enter up to 6 ASCII text characters, or up to 12 hex digits.

If you do not specify a value and the **Set Location Property to Subsystem ID** check box in the **Network Properties: Recovery** options category is cleared, the current value in the device will be uploaded into this field and stored in the OpenLNS database when the device is commissioned.

Ping Interval

Enables the OpenLNS Server to ping the device periodically. Pinging verifies that a device is still operating and communicating with the network.

Set the ping interval based on the expected movement of the device. Select **Never** to disable pinging. To enable pinging, change the ping interval to **15 minutes** if you expect the device will seldom move or if the device is on a power line channel, **2 minutes** if you expect it will move somewhat frequently, and **1 minute** if you expect it will move frequently.

You must set a ping value for OpenLNS CT to display device and functional block errors on their respective SmartShapes. See *Using OpenLNS CT Styles* for more information.

Description Provides an optional description of the device. This description has no effect on network operation, but you can use it to provide additional documentation for as-built reports.

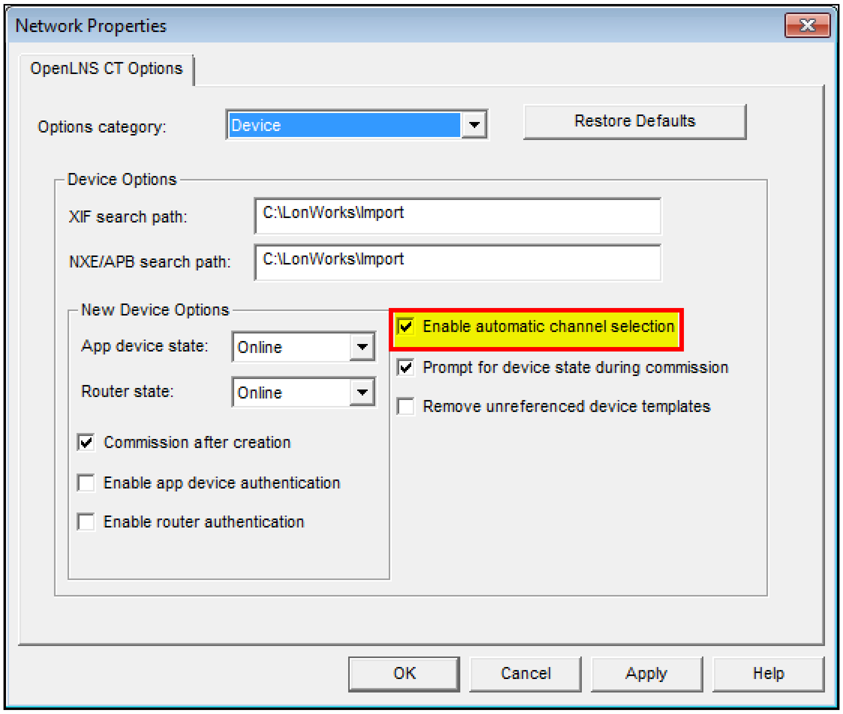
5. If you selected the **Commission Device** check box on the first page of the wizard, the first window for commissioning the device opens. See *Commissioning a Device* in Chapter 5, *Installing Networks*, for instructions on commissioning your device.

   If you did not select the **Commission Device** check box, click **Finish** to complete the device definition process and add the device SmartShape to your OpenLNS CT drawing.

Using Automatic Channel Selection for Devices

You can use automatic channel selection to enable OpenLNS CT to automatically select the appropriate channel for a new application device, skipping the New Device Wizard. This feature lets you create a new device with a single click and drag. To enable automatic channel selection in your OpenLNS CT drawing, follow these steps:

1. In the OpenLNS CT drawing, click **Add-ins**, point to **OpenLNS CT**, and then select **OpenLNS CT Options**. The **Network Properties** dialog box opens.
2. Select **Device** from the **Options Category** list.
3. Select the **Enable Automatic Channel Selection** check box.
4. Click **OK**.



Options category: Device

Device Options

XIF search path: C:\LonWorks\Import

NXE/APB search path: C:\LonWorks\Import

New Device Options

App device state: Online

Router state: Online

☑ Commission after creation

☐ Enable app device authentication

☐ Enable router authentication

☑ Enable automatic channel selection

☑ Prompt for device state during commission

☐ Remove unreferenced device templates

You can also have automatic channel selection enabled by default in each OpenLNS CT drawing you create or open. In the OpenLNS CT Design Manager, click the OpenLNS CT **Default Options** tab, select **Device** from the **Options Category** list, and then select the **Enable Automatic Channel Selection** check box.

The order of the criteria used by OpenLNS CT to select the appropriate channel is as follows: (1) closest qualified channel to the device SmartShape, in which the channel SmartShape is within 1 ½ times the width of the device SmartShape; (2) most recently used or created channel; and (3) other available defined channels. If no channel can be selected using the preceding criteria, the New Device Wizard will appear.

Automatic channel selection only applies to those device SmartShapes that include a device template definition and channel type information such as master device SmartShapes and copies of existing device SmartShapes. The New Device Wizard will therefore always appear when you drag the generic Device SmartShape from the **OpenLNS CT Basic Shapes** stencil to your OpenLNS CT drawing.

Changing the Channel of an Application Device

You can move an application device to a different channel, preserving the device's configuration and all of its connections. You can do this to assign a device to a channel different than the one that was automatically selected by OpenLNS CT, or to physically move a device to a different compatible channel. To move a device to a different channel, you can either right-click the device and then click **Move Device** on the shortcut menu, or you can use the **Connector** tool on the Visio **Standard** toolbar to graphically move the device.

To use the **Move Device** command, right-click the device SmartShape and click **Move Device** on the shortcut menu. See *Moving OpenLNS CT SmartShapes* in this chapter for how to move a device using the **Move Device** command.

To use the Visio **Connector** tool to move the device, follow these steps:

1. Select the **Connector** tool () on the Visio Standard toolbar.
2. Position the **Connector** tool over the "**x**" connection point of the device so that a red box opens over the "**x**".

3. Click and hold the mouse button.
4. Drag the **Connector** tool to the center of the SmartShape of the desired channel so that a red box opens around the channel SmartShape.
5. Release the mouse button.
6. Select a different tool on the Visio toolbar if you will not be making additional connections or reassignments (if you continue to use the **Connector** tool to add new devices, routers, or functional blocks SmartShapes to your drawing, Visio will attempt to connect the SmartShapes, which may not be desired or could generate errors.)

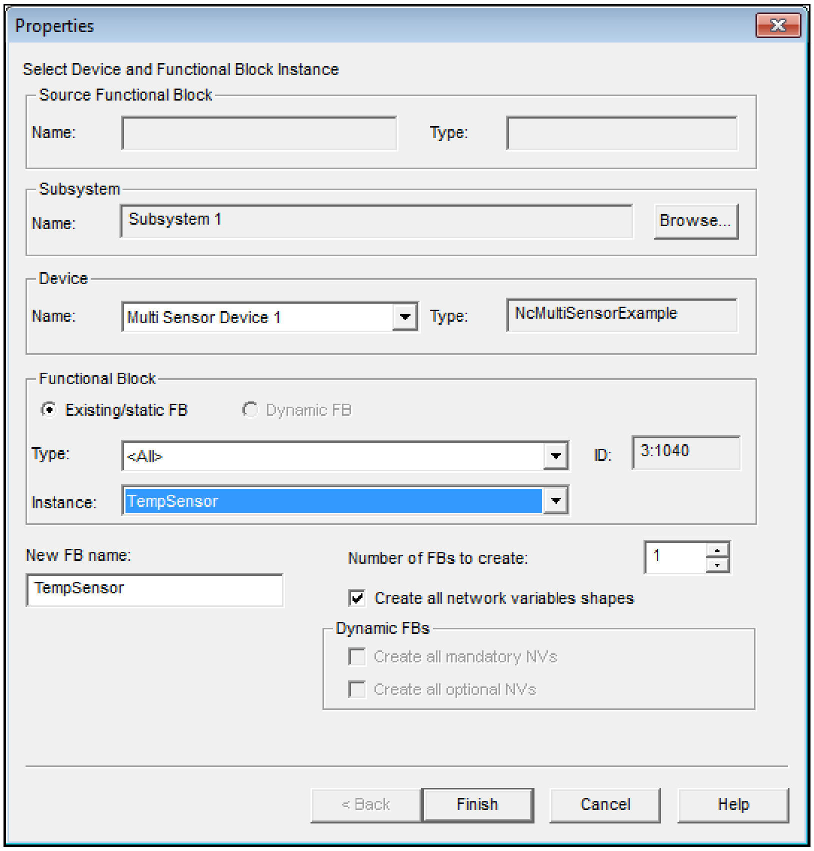
Creating Functional Blocks

A *functional block* encapsulates a set of network variables and configuration properties that perform a specific device function. For example, a four-port digital input device could have functional blocks for each of its four digital inputs. Each functional block would contain an output network variable representing the state or setting of a digital input. In addition, each functional block could contain configuration properties that control how frequently the digital input data is transmitted to other functional blocks. Ultimately, the task that the functional block performs in this example is transmitting the digital input data to other functional blocks (another functional block receiving the digital input data could then use it to turn a lamp on or off).

You can create a functional block in one step. To do this, you enable automatic device selection and then drag a custom functional block master SmartShape to your OpenLNS CT drawing near the device to which you want to assign the functional block. Otherwise, the New Functional Block Wizard opens when you add a functional block SmartShape to your OpenLNS CT drawing. See *Using Automatic Device Selection* in this section for how to enable this feature. See Chapter 12, *Creating and Using OpenLNS CT SmartShapes and Stencil*, for how to create and use custom functional block master SmartShapes.

To add a functional block to an OpenLNS CT drawing, follow these steps:

1. Drag a Functional Block SmartShape (▢) from the **OpenLNS CT Basic Shapes** stencil to your OpenLNS CT drawing. The Functional Block Wizard opens unless you enabled automatic device selection and dropped a custom functional block master SmartShape near a device SmartShape.



Properties

Select Device and Functional Block Instance

Source Functional Block

Name:

Type:

Subsystem

Name: Subsystem 1

Device

Name: Multi Sensor Device 1

Type: NcMultiSensorExample

Functional Block

● Existing/static FB ○ Dynamic FB

Type: <All>

ID: 3:1040

Instance: TempSensor

New FB name: TempSensor

Number of FBs to create: 1

☑ Create all network variables shapes

Dynamic FBs

□ Create all mandatory NVs

□ Create all optional NVs

2. Enter the following information:

Source Functional Block

Name Displays the name of the source functional block (a master SmartShape or an existing functional block) from which the new functional block was created.

This box will be empty if the functional block was created by dragging the generic functional block master SmartShape to the OpenLNS CT drawing.

Type Displays the functional profile represented by the source functional block (open loop sensor object, for example).

This box will be empty if the functional block was created by dragging the generic functional block master SmartShape to the OpenLNS CT drawing.

Subsystem

Name Specify the subsystem containing the functional block's device. The default is the subsystem containing the functional block.

Browse Click to browse the subsystem hierarchy and assign the functional block to a device in a different subsystem.

Device

Name Select a device from a list of all the devices in the selected subsystem that supports the functional block you are creating. If you are creating a generic functional block SmartShape, all devices in the selected subsystem with available functional blocks will be listed.

Type

Specifies the device template associated with the functional block.

Functional Block

Select whether the functional block you are creating is static or dynamic.

- **Static/Existing**. Select a functional block that is defined in the device interface.
- **Dynamic FB**. This option is only available for devices that support dynamic functional blocks (for example, a controller such as the SmartServer).

Type / FPT File

If you selected **Static/Existing**, select the functional profile for the functional block. The default is the functional profile for the functional block currently listed in the **Functional Block Name** box.

If you selected **Dynamic FB**, select the functional profile type file from the **FPT File** list.

Instance / FPT Type

If you selected **Static/Existing**, select which instance of the functional profile in the **Type** box to be created.

If you selected **Dynamic FB**, select the standard functional profile template (SFPT) or user-defined functional profile template (UFPT) to be assigned to the dynamic functional block.

Name

Select the functional block.

If you are creating a functional block from a custom SmartShape that specifies a specific functional profile, the number of functional blocks listed corresponds with the number of available functional blocks matching the functional profile on the associated device.

For example, if you create a functional block for a 4-port digital input device that has four digital inputs, four digital input functional blocks (one for each input) will initially be listed. If you then create a second functional block for the same device, only three functional blocks will be available in the list.

If you are creating a generic functional block SmartShape, all possible functional blocks for the selected device will initially be listed.

ID

Displays the scope and functional profile number of the functional block using the form <*Scope*>:<*Number*>.

If the functional block is a LonMark standard functional profile then the "<*Scope*>:" is omitted because it is 0.

New FB Name

Enter the name of the functional block. This name must be unique within the device and should be descriptive of the task to be performed by the functional block. The name may be up to 85 characters, but it may not include the period, backslash, colon, forward slash, or double quote characters.

If the functional block was created from a master SmartShape or a copy of an existing functional block, the default functional block name is the name of the source functional block. If there is a naming conflict, a separator character and an instance number (starting at 1) are appended to the functional block name. For each subsequent functional block with the same name for the same device, the instance number is incremented by one.

For example, the default name of a new functional block created from a digital output functional block master SmartShape named “LED” would be “LED” if it was the only one named as such for the assigned device. If a copy of the LED functional block was created and assigned to the same device, the default name would be “LED 1”. A second copy would be named “LED 2”, and so on.

If a functional block is created by dragging the generic functional block master SmartShape to the OpenLNS CT drawing, the default functional block name is the programmatic name of the functional block.

OpenLNS CT by default inserts the name of the associated device and a period before the functional block name in the OpenLNS CT drawing. Therefore, if the first digital output functional block named “LED” in the previous example was created for a device named “DO- 1”, the default functional block name would actually be “DO- 1.LED”. The second functional block named “LED” assigned to that same device would be “DO- 1.LED- 1”, and so on. To enable or disable this feature for all new OpenLNS CT network designs or for an existing OpenLNS CT network design, set or clear the **Include Device Name in Default FB Name** check box in the **Network Properties: Functional Block** options category as described in Appendix A.

To change the default naming convention for all new OpenLNS CT network designs or for an existing OpenLNS CT network, set options in the **Network Properties: Naming** options category as described in Appendix A.

Number of FBs Create

Enter the number of functional blocks to be created. When more than one functional block is created, the additional functional blocks will retain the same name as the first, and will be assigned to other devices in the same subsystem. For example, if the first functional block is assigned to **Device 1**, the second will be assigned to **Device 2**. If a matching device does not exist, you will be prompted to select the device.

Create all Network Variables Shapes

Select this check box to automatically create SmartShapes for all network variables defined in the functional block.

You can add unassigned network variables after the functional block has been created if this check box is cleared.

Dynamic FBs

Create all Mandatory NVs

Select this check box to automatically create all mandatory network variables defined in the functional profile for a dynamic functional block. The network variables are created in the OpenLNS network database.

To create the SmartShapes in your OpenLNS CT drawing, select the **Create all Network Variables Shapes** check box.

Create all Optional NVs

Select this check box to automatically create all optional network variables defined the functional profile for a dynamic functional block. The network variables are created in the OpenLNS network database.

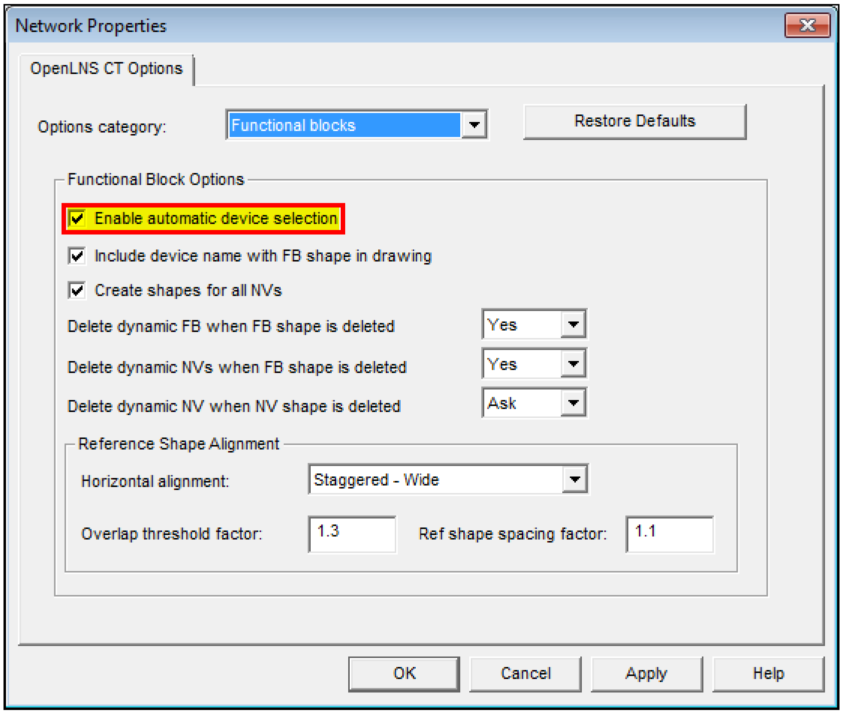
To create the SmartShapes in your OpenLNS CT drawing, select the **Create all Network Variables Shapes** check box.

3. Click **Finish** to add the functional block SmartShape to the OpenLNS CT drawing.

Using Automatic Device Selection

You can use automatic device selection to enable OpenLNS CT to automatically select the appropriate device for a new functional block, skipping the Functional Block Wizard. This feature lets you create a new functional block with a single click and drag. To enable automatic device selection in your OpenLNS CT drawing, follow these steps:

1. In the OpenLNS CT drawing, click **Add-ins**, point to **OpenLNS CT**, and then select **OpenLNS CT Options**. The **Network Properties** dialog box opens.
2. Select **Functional Blocks** from the **Options Category** list.
3. Select the **Enable Automatic Device Selection** check box.
4. Click **OK**.


Network Properties

OpenLNS CT Options

Options category: Functional blocks

Functional Block Options

☑ Enable automatic device selection

☑ Include device name with FB shape in drawing

☑ Create shapes for all NVs

Delete dynamic FB when FB shape is deleted: Yes

Delete dynamic NVs when FB shape is deleted: Yes

Delete dynamic NV when NV shape is deleted: Ask

Reference Shape Alignment

Horizontal alignment: Staggered - Wide

Overlap threshold factor: 1.3

Ref shape spacing factor: 1.1

The order of the criteria used by OpenLNS CT to select the appropriate device is as follows: (1) closest qualified device SmartShape to the functional block SmartShape where the device SmartShape is within 1 ½ times the width of the functional block SmartShape, and then (2) most recently created or assigned device. If no device can be selected using the preceding criteria, the Functional Block Wizard will appear.

Reassigning a Functional Block

You can move a functional block to a different device, preserving all of the functional block's configurations and connections. You can do this to assign a functional block to a device different than the one that was automatically selected by OpenLNS CT, or to reassign any functional block to a different compatible device. To move a functional block to a different device, you can either right-click the functional block and use the **Move FB** command, or you can use the **Connector** tool on the Visio **Standard** toolbar to graphically reassign the functional block.

To use the **Move FB** command to reassign a functional block to a different device, follow these steps:

1. Right-click the functional block you want to move, point to **Move FB**, and click **Change Assigned Device** to reassign the functional block to a device on the current subsystem. The Functional Block Wizard opens
2. Under **Device**, select the device to which you want to reassign the functional block. Then, under **Functional Block**, select the instance of the functional block to associate with the reassigned device. Click **Finish**.

To use the connector tool to reassign a functional block to a different device, follow these steps:

1. Select the **Connector** tool () on the Visio **Standard** toolbar, point to the functional block until a red box opens around the SmartShape, and hold down the mouse button.
2. Point to the device SmartShape and then release the mouse button. Alternatively, you can select the desired device first and then move the **Connector** tool to the functional block.
3. Select a different tool on the Visio **Standard** toolbar if you will not be making additional connections or reassignments (if you continue to use the **Connector** tool to add new devices, routers, or functional blocks SmartShapes to your drawing, Visio will attempt to connect the SmartShapes, which may not be the desired action or may be an action that generates errors).

Configuring a Functional Block

You can configure a functional block by right-clicking the functional block SmartShape and then clicking **Configure** on the shortcut menu. If the functional block has a registered configuration plug-in, that plug-in opens. If the functional block does not have a configuration plug-in, the OpenLNS CT Browser opens.

Copying a Functional Block Configuration

Once you have configured a functional block, you can create new functional blocks that use the same configuration or update existing functional blocks with the configured settings.

Creating a New Functional Block from a Configured Functional Block

You can create a new functional block from a configured functional block in two ways:

- Hold down CTRL and drag the functional block to a new location within the same network. The Functional Block Wizard opens, unless you enabled automatic device selection. Make any desired changes to the functional block information. Click **OK**. The new functional block SmartShape will appear on your OpenLNS CT drawing.
- Create a new custom functional block master SmartShape. Drag the configured functional block to an OpenLNS CT stencil to which you have write access. The master SmartShape for the configured functional block will include all the settings for any configuration properties that you modified. You can then re-use the configured functional block by dragging it from the OpenLNS CT stencil on which it is saved to the same or any other OpenLNS CT drawing.

Functional blocks created using either of these methods will initially have the same configuration property settings as the source functional block. Any changes you make to a functional block afterwards will not affect other functional blocks of the same type.

Updating a Functional Block from a Configured Functional Block

You can update existing functional blocks with the settings of a configured functional block in two ways: use the **Copy Config** and **Paste Config** commands on the shortcut menu, or use the **Replicate Settings** commands on the functional block shortcut menu.

To use the **Copy Config** and **Paste Config** commands on the shortcut menu, follow these steps:

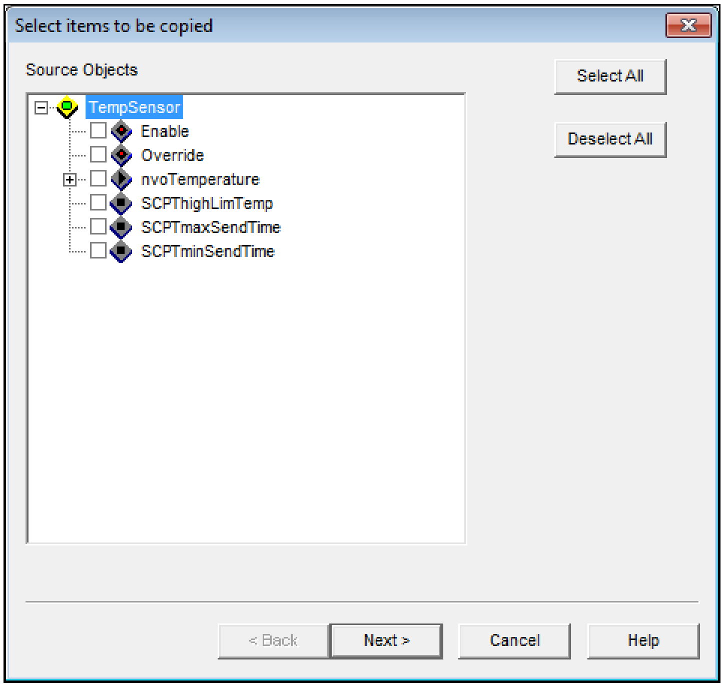
1. Right-click the configured functional block, point to **Configuration Properties** on the shortcut menu, and then click **Copy Config**.

2. Right-click the functional block you want to update with the configured settings, point to **Configuration Properties** on the shortcut menu, and then click **Paste Config**.

   Tip: You can also paste the configuration properties to a text file to get a summary of the configuration property settings. Right-click the configured functional block, point to **Copy Config** from the shortcut menu, open a text editor or word processing application, and then paste the configuration properties.

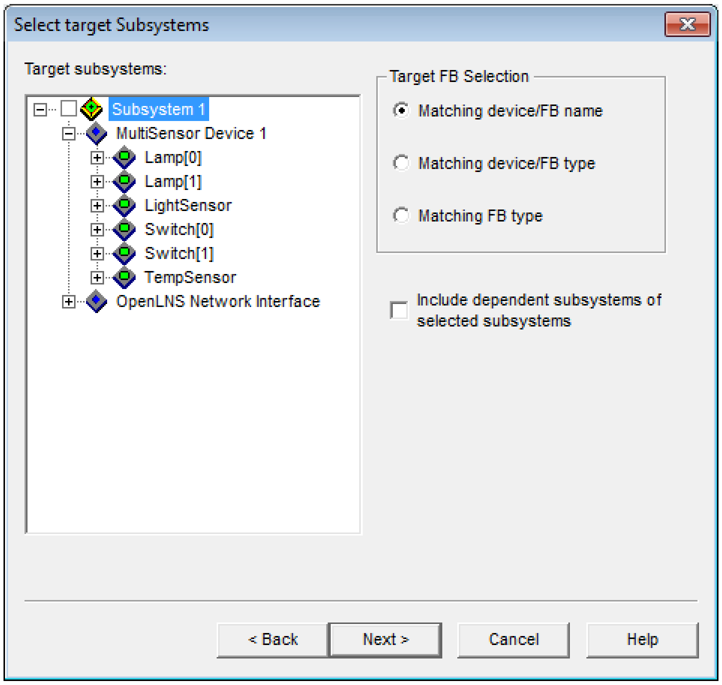
To use the **Replicate Settings** command on the functional block shortcut menu, follow these steps.

1. Right-click the configured functional block, point to **Configuration Properties** on the shortcut menu, and then click **Replicate Settings**. The **Select Items to be Copied** dialog box opens.



2. Select the configuration properties in the functional block to be copied. To select all the configuration properties in the functional block, click **Select All**.

3. Click **Next**. The **Select Target Subsystems** dialog box opens.



Select target Subsystems

Target subsystems:

- □ Subsystem 1
  - MultiSensor Device 1
    - Lamp[0]
    - Lamp[1]
    - LightSensor
    - Switch[0]
    - Switch[1]
    - TempSensor
  - OpenLNS Network Interface

Target FB Selection

● Matching device/FB name

○ Matching device/FB type

○ Matching FB type

□ Include dependent subsystems of selected subsystems

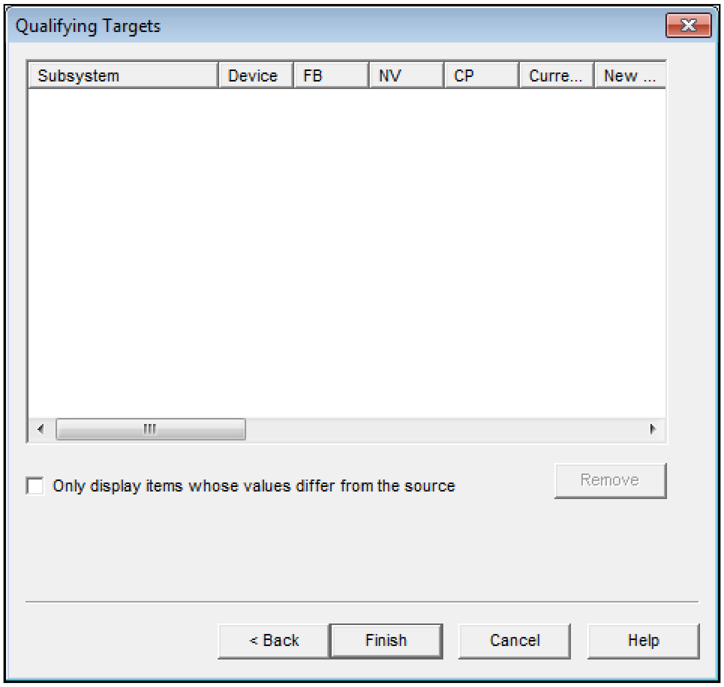
4. Select the target subsystems into which you want the configuration properties to be copied. You can expand a subsystem to view any nested subsystems.

5. Select the criteria used to determine which functional blocks will be updated with the configured settings. You have three choices:

   - **Matching Device/FB name**. Both the device and functional block names must match for the functional block to be configured
   - **Matching Device/FB Type**. Both the device and functional block types must match
   - **Matching FB Type**. Only the functional block type must match).

   Select the **Include Dependent Subsystems** check box if you want to replicate the configuration property settings in applicable functional blocks in the nested subsystems within the selected subsystems.

6. Click **Next**. The **Qualifying Targets** dialog box opens. This dialog lists all the functional blocks meeting the criteria selected in Step 5.



7. You can remove a functional block target from the list by selecting the target's subsystem name and then clicking **Remove**. If you only want to view those functional blocks in which the value of the configuration property in the target functional block differs from the source, click the check box at the bottom. You can sort the entries in a column by clicking the column header. Click the header again to toggle the order (ascending or descending).
8. Click **Finish**. The functional block targets will be updated.

Functional blocks updated using either of these methods will initially have the same configuration property settings as the source functional block. Any changes you make to a functional block afterwards will not affect other functional blocks of the same type.

Creating a Functional Block Master SmartShape

You can create a custom functional block master SmartShape to re-use a functional block that you have configured. When you create a new functional block using your custom functional block master SmartShape, the Functional Block Wizard will be skipped (provided that automatic device selection is selected) and the functional block will be created with the same network variables and configuration property settings as the original functional block. To create a custom functional block master SmartShape, drag the desired functional block SmartShape onto an OpenLNS CT stencil to which you have write access.

Manufacturers may also supply custom master SmartShapes for their functional blocks. For example, Echelon provides LonPoint functional block master SmartShapes that can be used with the LonPoint devices.

See Chapter 12, *Creating and Using Custom OpenLNS CT SmartShapes and Stencils*, for more information on creating and using OpenLNS CT master SmartShapes.

Creating Dynamic Functional Blocks

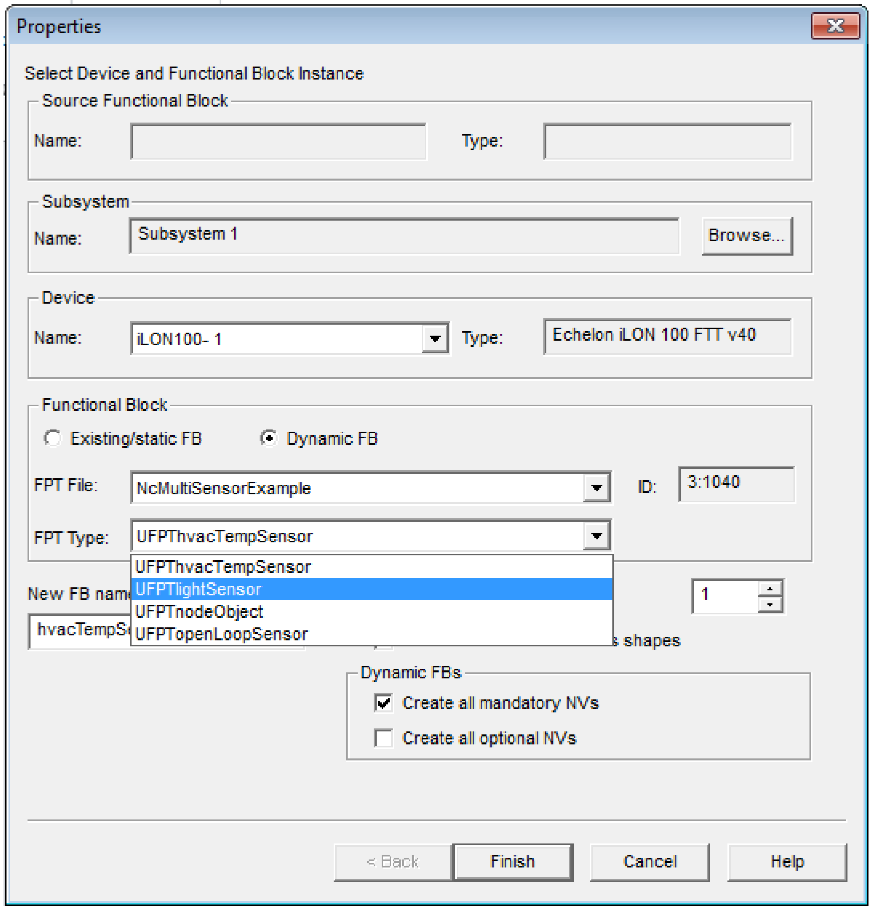
A dynamic functional block is a functional block that is not pre-loaded on a device. Most devices use static functional blocks, which are statically defined by the device application. You can view the static functional blocks that are available on a device by right-clicking the device SmartShape, selecting **Properties** on the shortcut menu, and then clicking the **Functional Blocks** tab.

Dynamic functional blocks are typically used by controllers that do not have a static interface. These controllers typically require the integrator to run a plug-in or other configuration software that defines the control algorithm and configures the controller. The controller manufacturer can use dynamic functional blocks to enable an OpenLNS CT user to add functional blocks to the controller using OpenLNS CT.

Dynamic functional blocks are created in the same manner as static functional blocks. To create a dynamic functional block, follow these steps:

1. Drag a functional block SmartShape from a stencil to the OpenLNS CT drawing, or copy an existing functional block SmartShape. The functional block SmartShape you drag or copy can be either the generic functional block SmartShape, or a custom functional block master SmartShape that specifies the device types and functional profiles for which it is valid. The New Functional Block Wizard opens.

2. When the New Functional Block Wizard opens, the **Type** and **Instance** boxes change to **FPT File** and **FPT Type** if the device supports dynamic functional blocks. Select the functional profile type file from the **FPT File** list, and then select the standard functional profile template (SFPT) or user-defined functional profile template (UFPT) to be assigned to this dynamic functional block from the **FPT Type** list.



Source Functional Block Name:
Source Functional Block Type:
Subsystem Name: Subsystem 1
Device Name: iLON100- 1
Device Type: Echelon iLON 100 FTT v40
○ Existing/static FB ● Dynamic FB
FPT File: NcMultiSensorExample
ID: 3:1040
FPT Type: UFPThvacTempSensor
UFPThvacTempSensor
UFPTlightSensor
UFPTnodeObject
UFPTopenLoopSensor
New FB nam hvacTempS
1
s shapes
☑ Create all mandatory NVs
☐ Create all optional NVs

3. Enter the name of the dynamic functional block in the **New FB Name** box. If the source functional block was the generic functional block SmartShape, the default name will be the name of the selected functional profile template minus its SFPT or UFPT pre-fix.

4. If you want the network variables defined in the dynamic functional block to be represented with SmartShapes, select the **Create all Network Variables Shapes** check box.

5. If the selected functional profile defines mandatory and/or optional network variables, the corresponding check boxes under **Dynamic FBs** for creating those network variables in the dynamic functional block will be enabled. By default, the **Create all Mandatory NVs** check box will be selected and the **Create all Optional NVs** check box will be cleared. Select the check boxes for the types of network variables for which you want OpenLNS CT to create.

   You can still add or remove network variables after the dynamic functional block has been created.

6. Click **Finish** to add the dynamic functional block SmartShape to the OpenLNS CT drawing.

You cannot change the functional profile for a dynamic functional block after you create the functional block SmartShape. Instead, you need to delete the dynamic functional block and then create a new dynamic functional block that implements the desired functional profile.

Deleting Dynamic FBs

You can only remove a dynamic functional block from a device when you delete the dynamic functional block SmartShape from the OpenLNS CT drawing. When you attempt to delete a dynamic functional block SmartShape from the OpenLNS CT drawing, the effect on the dynamic functional block on the device depends on the settings in the **Delete Dynamic FBs when FB Shape is Deleted** check box in the **Network Properties: Functional Blocks** options category.

- If **Yes** is set, the dynamic functional block will be removed from the device when its SmartShape is deleted.

- If **No** is set, the dynamic functional block will remain on the device. To delete the dynamic functional block, drag a SmartShape for that functional block to your OpenLNS CT drawing and then delete it. This is the same process to use if you want to delete a dynamic functional block created with an OpenLNS application other than OpenLNS CT.

- If **Ask** is set, you will be prompted to select whether you want to delete the dynamic functional block. Select **Yes**.

To access the **Network Properties: Functional Blocks** options category, click **Add-ins**, click **OpenLNS CT**, click **OpenLNS CT Options**, and then select **Functional Blocks** from the **Options Category** list.

Creating Dynamic Functional Block Master SmartShapes

You can create master SmartShapes for dynamic functional blocks in the same manner as static functional blocks. The dynamic functional block master SmartShape will contain user cells indicating that the functional block is dynamic, the functional profile used to create the dynamic functional block, and whether to create mandatory and optional network variables (see *Editing Master SmartShape User-Defined Cells* in Chapter 12, *Creating and Using Custom OpenLNS CT SmartShapes and Stencils* for more information). When you drag the dynamic functional block master SmartShape to your OpenLNS CT drawing, the values set in the user-defined cells will be the defaults. The values will also determine whether the New Functional Block Wizard opens when you drag the SmartShape to your OpenLNS CT drawing.

Dynamic network variables included on the functional block master SmartShape will be created regardless of the settings in the check boxes under **Dynamic FBs**.

Creating a Virtual Functional Block

A *virtual functional block* is a static functional block that that contains the network inputs and outputs for a device that are not part of other functional blocks on the device. Network inputs and outputs consist of network variables, configuration properties, and message tags.

To add a virtual functional block to your OpenLNS CT drawing, follow these steps:

1. Drag a functional block SmartShape from the **OpenLNS CT Basic Shapes** stencil to your OpenLNS CT drawing. The **Functional Block Wizard** opens.

2. Under **Device**, select the device to which you want to assign the virtual functional block from the **Name** list

3. Under **Functional Block**, select **Virtual Functional Block** from the **Name** list. This option only opens in the list if the device supports a virtual functional block

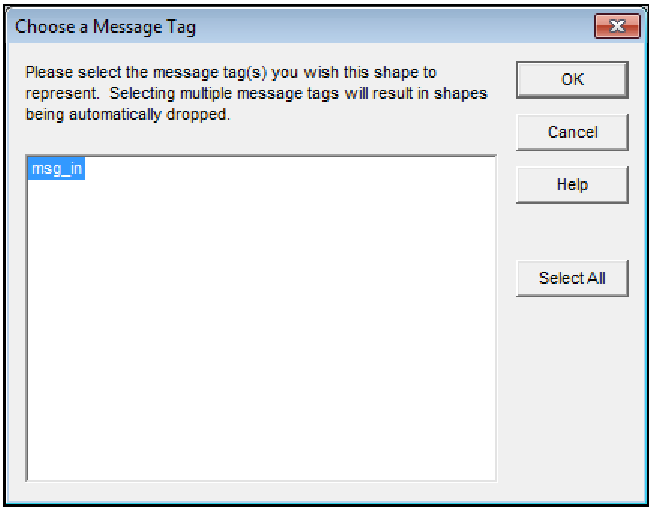
4. Click **Finish**.

Adding a Message Tag SmartShape to a Virtual Functional Block

You can add a message tag SmartShape to the virtual functional block of a device that supports message tags. Message tags are similar to network variables except that they do not have pre-defined types. They are used for standard data transfer mechanisms such as the LonWorks file transfer protocol and the LonWorks data log transfer protocol. In addition, they are used as a proprietary interface to devices such as factory calibration interfaces. There are no restrictions in regards to connecting message tags such as having compatible types as there are for network variables.

To add a message tag SmartShape to a virtual functional block, follow these steps:

1. Drag an input or output message tag SmartShape (▶) from the **OpenLNS CT Basic Shapes** stencil onto the virtual functional block. The **Choose a Message Tag** dialog box opens.



Choose a Message Tag

Please select the message tag(s) you wish this shape to represent. Selecting multiple message tags will result in shapes being automatically dropped.

msg_in

2. Select one or more message tag SmartShapes to add to the virtual functional block from the list of available message tags. If you select multiple message tags, additional SmartShapes will automatically be added to the virtual functional block. To select all of the listed message tags, click **Select All**.

3. Click **OK**. The selected message tag SmartShapes appear on the virtual functional block SmartShape in the appropriate position.

Tip: You can change the name or position of a message tag SmartShape in the same manner as a network variable SmartShape. See *Changing Network Variable Name* and *Changing Network Variable Position* for how to do this.

Deleting Message Tag SmartShapes

You can remove message tag SmartShapes from one or more virtual functional blocks in the same manner as network variable SmartShapes. See *Deleting a Network Variable SmartShape* in the next section for how to do this.

Creating Network Variables

Network variables allow a device to send and receive data over the network to and from other devices. Network variables are data items (such as temperature, the state of a switch, or actuator position setting) encapsulated within functional blocks that a particular device application expects to receive from other physical devices (an *input network variable*) or expects to make available to other physical devices (an *output network variable*).

There are two types of network variables: static and dynamic. A *static network variable* is defined by the device application and is always available in the device. Most LONWORKS devices have functional blocks that include static network variables. A *dynamic network variable* is added to a functional block on a device after the device has been commissioned. Some LONWORKS devices support dynamic network variables. Other types of devices that support dynamic network variables include controllers and gateways with dynamic interfaces.

Creating a Network Variable SmartShape

You can create a network variable SmartShape and then use it to either create a new network variable master SmartShape with an associated type and direction, or add a dynamic network variable with a specified type and direction to a functional block that supports dynamic network variables. To create a new network variable SmartShape, follow these steps:

1. Select a functional block that contains a network variable with the desired network variable type that you want to use for the new network variable SmartShape, and then select the network variable.
2. Right-click the network variable that you selected in step 1 and then select **Create Input NV Template** or **Create Output NV Template** on the shortcut menu to create an input or output network variable SmartShape, respectively. OpenLNS CT creates a new network variable SmartShape on the current drawing page.

   The default name of this new network variable SmartShape will be the name of the source network variable followed by its direction in brackets. For example, if you create an output network variable from a network variable named "SWITCH", the default name of the new NV SmartShape will be "SWITCH [Out]".

   If the source NV contains a direction in its name such as "nvo" or "nvi" and that direction does not match that of the new NV SmartShape, the name of the direction will be reversed in the default name of the NV SmartShape. Using the previous example, if the source NV is named "nvi SWITCH", the default name of the new NV SmartShape will be "nvo SWITCH [Out]".

Creating a Network Variable Master SmartShape

Using a network variable SmartShape, you can create a network variable master SmartShape. You can then use the network variable master SmartShape to add a dynamic network variable with a specified network variable type and direction to functional blocks that support dynamic network variables. To create a network variable master SmartShape, follow these steps:

1. Create a network variable SmartShape following the steps in *Creating a Network Variable SmartShape*.
2. Drag the network variable SmartShape to an OpenLNS CT stencil to which you have write access. If the stencil is read-only, you will be prompted to either change the stencil to be writeable or cancel the operation. A new master SmartShape opens on the stencil. The default name of the network variable master SmartShape is the name of the network variable SmartShape without its direction suffix.
3. Save the stencil

Adding Network Variable SmartShapes to a Functional Block

You can automatically add all available static and/or dynamic network variables to a functional block when you create a new functional block SmartShape, or you can selectively add them after you create the functional block.

To automatically add all available network variables to a functional block, select the **Create Shapes for all Network Variables** check box in the New Functional Block Wizard when you add a functional block SmartShape to your drawing.

To selectively add network variables, clear the **Create Shapes for all Network Variables** check box in the New Functional Block Wizard. This creates a functional block with no network variable SmartShapes. You then add one or more network variable SmartShapes to the functional block using one of the following three methods:

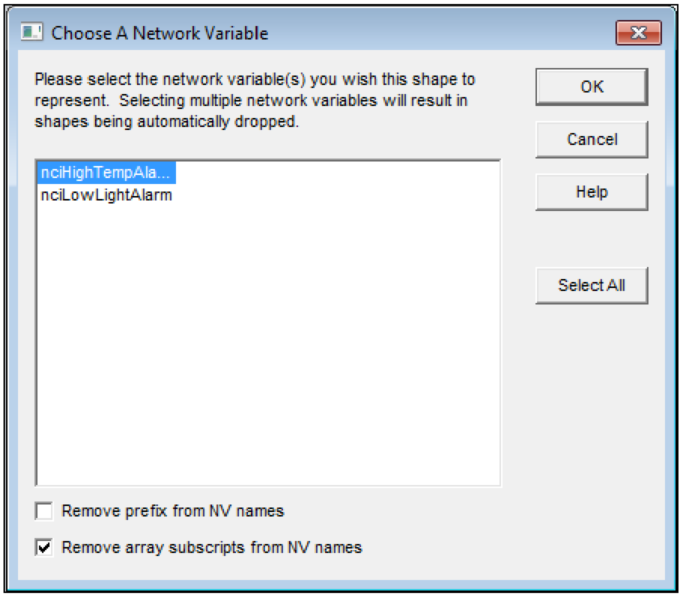
- Drag a generic network variable SmartShape onto the functional block. This method can be used for adding both static and dynamic network variables to a functional block.
- Drag a network variable SmartShape from the OpenLNS CT NV SmartShapes stencil onto the functional block. This method can only be used for adding dynamic network variables to a functional block.
- Drag a network variable master SmartShape onto the functional block. This method can only be used for adding dynamic network variables to a functional block.

Tip: You can quickly add a dynamic network variable to a functional block using any of last two methods.

Creating Network Variables Using Generic Network Variable SmartShapes

Using generic network variable SmartShapes, you can add an existing static network variable or create and add a dynamic variable to a functional block following these steps:

1. Drag an input or output network variable SmartShape () from the **OpenLNS CT Basic Shapes** stencil onto the functional block. The **Choose a Network Variable** dialog box opens.



Please select the network variable(s) you wish this shape to represent. Selecting multiple network variables will result in shapes being automatically dropped.

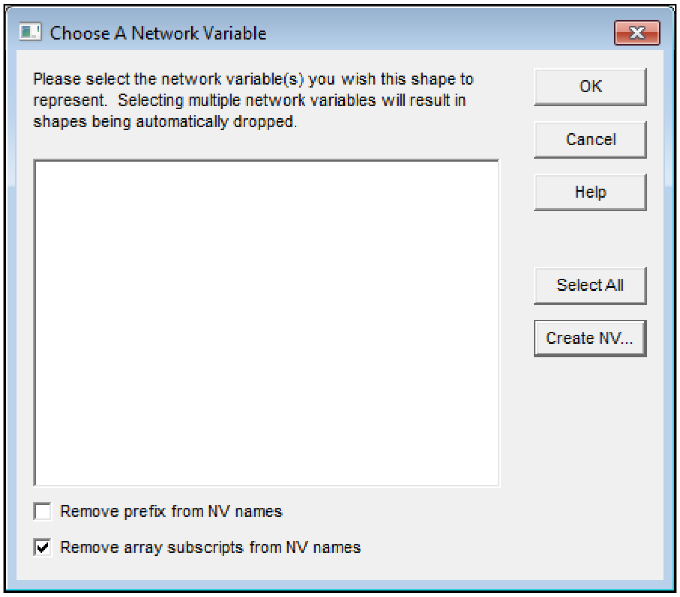
nciHighTempAla...
nciLowLightAlarm

□ Remove prefix from NV names

☑ Remove array subscripts from NV names

Note: If the device supports dynamic network variables, this dialog will also have a **Create NV** button.

2. To add a static network variable to the functional block, skip to step 8. To create and add a dynamic network variable, click **Create NV**.



Choose A Network Variable

Please select the network variable(s) you wish this shape to represent. Selecting multiple network variables will result in shapes being automatically dropped.

□ Remove prefix from NV names

☑ Remove array subscripts from NV names

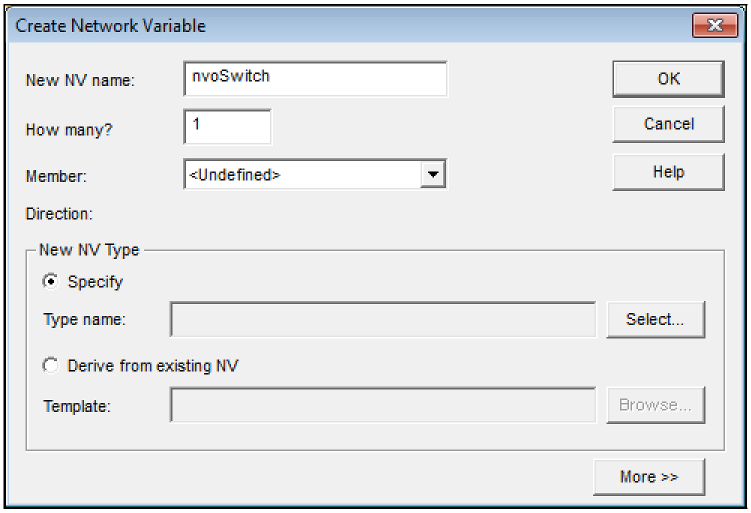
3. The **Create Network Variable** dialog box opens. Enter the name, number, and type of the dynamic network variable(s) you are creating.

   a. Enter the name of the dynamic network variable. This name must be unique within the functional block. It may be up to 85 characters long and may include embedded spaces, but it may not include period, backslash, colon, forward slash, or double quote characters.

   b. Enter the number of dynamic network variables you are creating.

   c. If you are creating more than one dynamic network variable, OpenLNS CT will automatically append an index, starting at 1, to the second dynamic network variable. OpenLNS CT will then increment the index for each additional dynamic network variable you are creating. For example, if the dynamic network variable name is “mynv” and you are creating three network variables, the names will be “mynv”, “mynv1”, and “mynv2”.

      If you are creating 10 or more network variables, you should append a “01” to the dynamic network name (for example, mynv01). If you are creating three network variables, the names will be “mynv01”, “mynv02”, and “mynv03”. This ensures that the dynamic network variable names will be listed in numerical sequence in various OpenLNS CT menus and reports.



Create Network Variable

New NV name: nvoSwitch

How many? 1

Member: <Undefined>

Direction:

New NV Type

● Specify

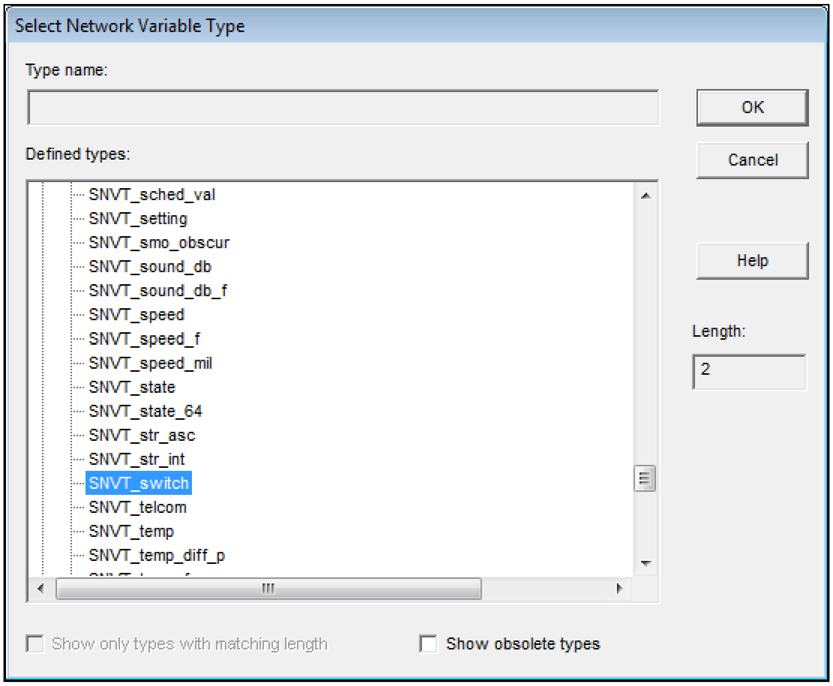
Type name:

○ Derive from existing NV

Template:

d. Select the type of the dynamic network variable based on a defined type, or the type of an existing network variable in your OpenLNS CT drawing.

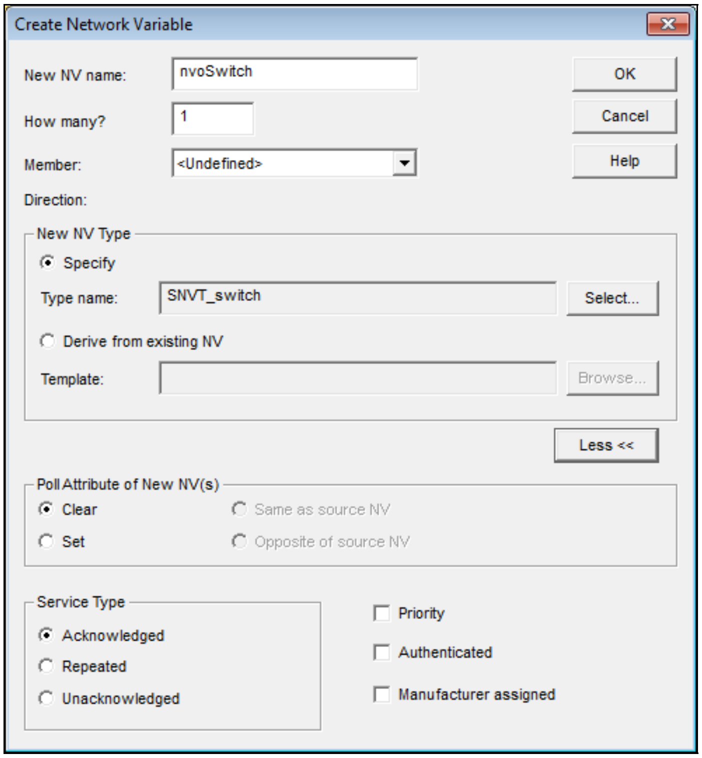
- To select a defined type, select **Specify** under the **New NV Type** box and then click **Select**. The **Select Network Variable Type** dialog box opens. This dialog box lists built-in data types and resource files. Expand the items listed to view the available built-in data types, SNVTs, and UNVTs. Then click the type you want the dynamic network variable to use. Click **OK**.



Length: 2

- To create a dynamic variable from the template of an existing network variable in your OpenLNS CT drawing, select **Derive from Existing NV** under the **New NV Type** box and then click **Browse**. The Select Object window opens. Expand the listed objects to view the network variables in your OpenLNS CT drawing. Click the network variable you want to use as the source for the dynamic network variable. Click **OK**.

4. Click **More** to set the polling attributes and the messaging service for the dynamic network variable(s) you are creating.



New NV name: nvoSwitch

How many? 1

Member: <Undefined>

Direction:

New NV Type

● Specify

Type name: SNVT_switch

○ Derive from existing NV

Template:

Poll Attribute of New NV(s)

● Clear ○ Same as source NV

○ Set ○ Opposite of source NV

Service Type

● Acknowledged

○ Repeated

○ Unacknowledged

□ Priority

□ Authenticated

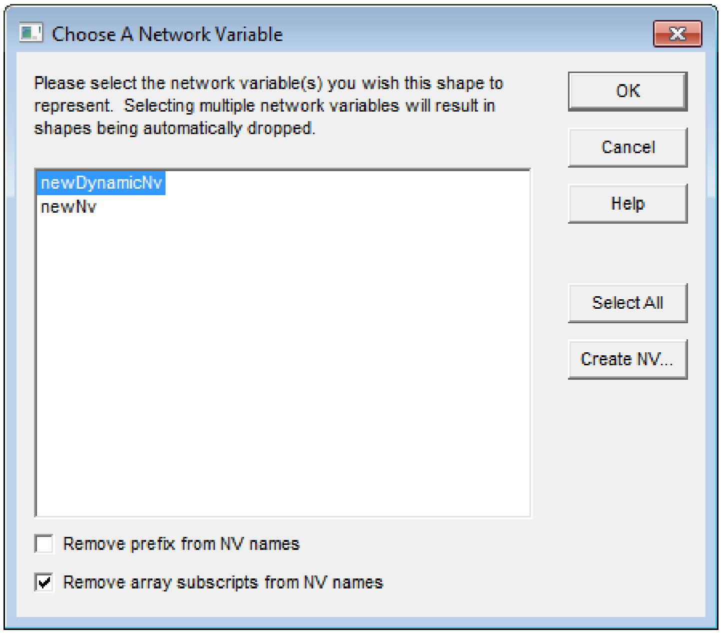
□ Manufacturer assigned

5. Under **Poll Attributes of New NV(s)**, select whether the dynamic network variable(s) will by default use polled monitoring. You have four choices:

   - Select **Clear** to disable polling.
   - Select **Set** to enable polling.
   - Select **Same as Source NV** to use the same poll attribute of the source network variable (enabled or disabled). This option is only available if the dynamic network variable you are creating is derived from an existing network variable.
   - Select **Opposite of Source NV** to use the opposite poll attribute (if the source network variable has polling enabled, the dynamic network variable will be non-polling). This option is only available if the dynamic network variable you are creating is derived from an existing network variable.

6. Under **Service Type**, select whether connections with the dynamic network variable use **Acknowledged**, **Repeated**, or **Unacknowledged** messaging service if the dynamic network variable is based on a defined type. You can also enable priority and/or authenticated messaging on the connections. See *Connection Description Properties* later in this chapter for more information on messaging service types and options.

7. Click **OK** to return to the **Choose a Network Variable** dialog box. The new dynamic network variable is listed in the dialog.



Please select the network variable(s) you wish this shape to represent. Selecting multiple network variables will result in shapes being automatically dropped.

newDynamicNv
newNv

☐ Remove prefix from NV names

☑ Remove array subscripts from NV names

8. Select one or more static and/or dynamic network variables SmartShapes to add to the functional block from the list of network variables available for the selected direction. If you select multiple network variables, additional SmartShapes will automatically added to the functional block. To select all of the listed network variables, click **Select All**.

9. To remove any “**nvo**” or “**nvi**” prefixes, select the **Remove Prefix from NV Names** check box. To remove any indexes appended to NV names, select the **Remove Array Subscripts from NV Names** check box. OpenLNS CT will not remove indexes from network variable names if there is any chance of a naming conflict.

10. Click **OK**. The selected network variable SmartShapes will appear on the functional block SmartShape. Input network variable SmartShapes are positioned on the left side of the functional block; output network variables are located on the right side.

Creating Dynamic Network Variables Using Network Variables SmartShapes from the OpenLNS CT NV SmartShapes Stencil

To add a dynamic network variable to a functional block using a network variable SmartShape in the OpenLNS CT NV SmartShapes stencil, follow these steps:

1. Drag a network variable SmartShape from the OpenLNS CT NV SmartShapes stencil onto the desired functional block. This functional block must support dynamic network variables. The **Create Network Variable** dialog box opens.

2. Following steps 3–6 in *Creating Dynamic Network Variables Using Network Variable SmartShapes*, change the default name, if desired, enter the number of new dynamic network variable(s) you are creating, and set the poll attributes and the message service type and options.

3. Click **OK** to add the dynamic network variable to the functional block.

Creating Dynamic Network Variables Using Network Variable Master SmartShapes

To add a dynamic network variable to a functional block using a network variable master SmartShape, follow these steps:

1. Drag a network variable master SmartShape variable SmartShape onto the desired functional block. This functional block must support dynamic network variables. The **Create Network Variable** dialog box opens.

2. Following steps 3–6 in *Creating Dynamic Network Variables Using Network Variable SmartShapes*, change the default name, if desired, enter the number of new dynamic network variable(s) you are creating, and set the poll attributes and the message service type and options.

3. Click **OK** to add the dynamic network variable to the functional block.

Changing a Network Variable Name

You can change the network variable name on a functional block. OpenLNS CT will record the change in the OpenLNS CT drawing and in the OpenLNS network database. To change the name of a network variable, click its functional block and then click the network variable SmartShape. Right-click the network variable SmartShape, click **Properties** on the shortcut menu, and then enter the new name in the **Name** box. OpenLNS CT only requires that network variable names be unique within a functional block; therefore, if you list all the network variables in a device, some may have duplicate names.

Changing Network Variable Position

Once a functional block has a network variable SmartShape placed on it, the network variable SmartShape becomes part of the functional block SmartShape. If you move the functional block within the drawing or from one subsystem to another, the network variable will remain in the same position on the functional block.

You can change where a network variable SmartShape is positioned on a functional block. To change the location of a network variable SmartShape, click its functional block, click the network variable SmartShape, right-click the network variable SmartShape, and then clear **Automatic Positioning** if it is currently selected. You can then move the selected network variable SmartShape up or down on the functional block.

Deleting a Network Variable SmartShape

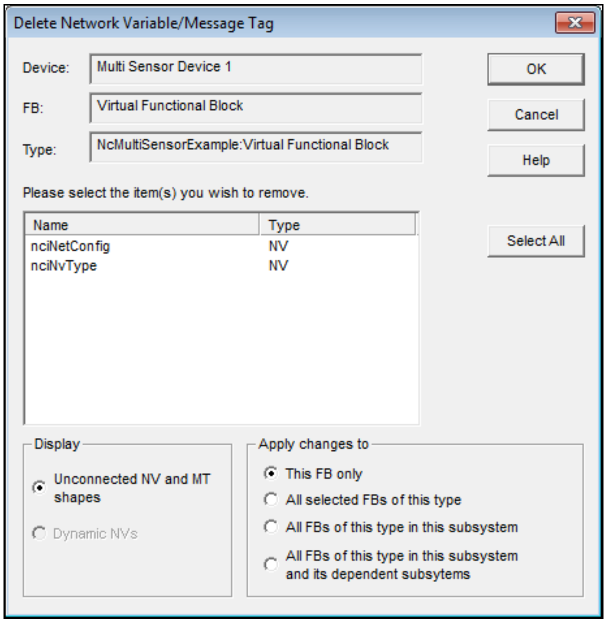
You can remove unconnected network variable SmartShapes from one or more functional blocks.

To delete a network variable SmartShape from a single functional block, follow these steps:

1. Click the functional block with the network variable you want to remove.
2. Click the network variable SmartShape.
3. Right-click the network variable SmartShape.
4. Click **Delete** on the shortcut menu.

To delete a network variable SmartShape from multiple functional blocks, follow these steps:

1. Delete any connections to the network variable on all the functional blocks. You cannot remove a connected network variable from a functional block.
2. Right-click the functional block.
3. Click **Delete NVs and MTs** on the shortcut menu. The **Delete Network Variable/Message Tag** dialog box opens.



Device: Multi Sensor Device 1

FB: Virtual Functional Block

Type: NcMultiSensorExample:Virtual Functional Block

Please select the item(s) you wish to remove.

Name	Type
nciNetConfig	NV
nciNvType	NV

Display

● Unconnected NV and MT shapes

○ Dynamic NVs

Apply changes to

● This FB only

○ All selected FBs of this type

○ All FBs of this type in this subsystem

○ All FBs of this type in this subsystem and its dependent subsytems

4. Select one or more network variable SmartShapes to delete. To select multiple network variable SmartShapes, hold down CTRL and click the SmartShapes you want to delete. To select all of the listed network variables, click **Select All**.

5. Set the following options:

Display — Select which network variable and message tag SmartShapes are displayed.

- Select **Unconnected NV and MT Shapes** to view all the unconnected network variable and message tag SmartShapes on the functional block.

- Select **Dynamic NVs** to view all the unconnected dynamic network variable SmartShapes on the functional block.

Apply Changes to — Select the scope of the removal of the selected network variable and message tags SmartShapes. You have four choices:

- **This FB Only**. The selected network variable and message tag SmartShapes will be deleted from the selected functional block only.

- **All selected FBs of this Type**. The selected network variable and message tag SmartShapes will be deleted from all functional blocks in the OpenLNS CT drawing with the same type of the selected functional block.

- **All FBs of this Type in this Subsystem**. The selected network variable and message tag SmartShapes will be deleted from all functional blocks in the current subsystem with the same type of the selected functional block.

- **All FBs of this Type in this Subsystem and its Dependent Subsystems**. The selected network variable and message tag SmartShapes will be deleted from all functional blocks in the current subsystem and from any of its nested subsystems with the same type of the selected functional block.

6. Click **OK**.

Deleting Dynamic Network Variables

You can delete a dynamic network variable from a functional block. When you delete a dynamic network variable, you can either delete the network variable SmartShape, or you can delete both the network variable SmartShape and the implementation. You can also delete a dynamic network variable by deleting the functional block containing it.

To choose the action that occurs when you delete a dynamic network variable or a functional block containing a dynamic network variable, select the **Delete Dynamic NVs when FB Shape is Deleted** and **Delete Dynamic NVs when NV Shape is Selected** check boxes under **Functional Blocks Options** in the **Network Properties: Functional Blocks** options category as described in Appendix A. Select one of the following options:

- **Yes** to remove both the dynamic network variable SmartShape and implementation from the functional block.
- **No** to remove only the dynamic network variable SmartShape. The implementation remains on the functional block.
- **Ask** to be prompted each time a dynamic network variable is to be deleted

To access the **Network Properties: Functional Blocks** options category, click **Add-ins**, point to **OpenLNS CT**, and then click **OpenLNS CT Options**. Select **Functional Blocks** from the **Options Category** list.

Creating a Router

A router enables application devices on separate channels to communicate. You can use a single router to connect two channels or use multiple *redundant routers* between the same pair of channels.

You can use an MPR-50 Multi-Port Router to connect up to four TP/FT-10 free topology twisted-pair channels, or to connect one or more TP/FT-10 channels to a TP/XF-1250 high-speed backbone. See the *MPR-50 Multi-Port Router User's Guide* for more information on installing and using the MPR-50 Multi-Port Router.

You can use routers to enable OpenLNS CT access to LONWORKS networks over IP-852 channels and to integrate multiple native LONWORKS networks into one large network that uses a high-speed IP-852 channel as a backbone. IP-852 routers include the SmartServer (with IP-852 routing option), i.LON 100 Internet Server (with IP-852 routing option), and i.LON 600 IP-852 Router.

Other compatible routers include LONWORKS routers based on the Router 5000 chips or RTR-10 Router Core, and ISO/IEC 14908-1 (ANSI/CEA-709.1 or EN 14908) and ISO/IEC 14908-4 compliant routers. For more information on IP-852 routing using the SmartServer, i.LON 100 Internet Server, or i.LON 600 IP-852 Router, see the *SmartServer 2.0 User's Guide*, *i.LON 100 e3 Internet Server User's Guide*, and *i.LON 600 LonWorks/IP Server User's Guide*, respectively.

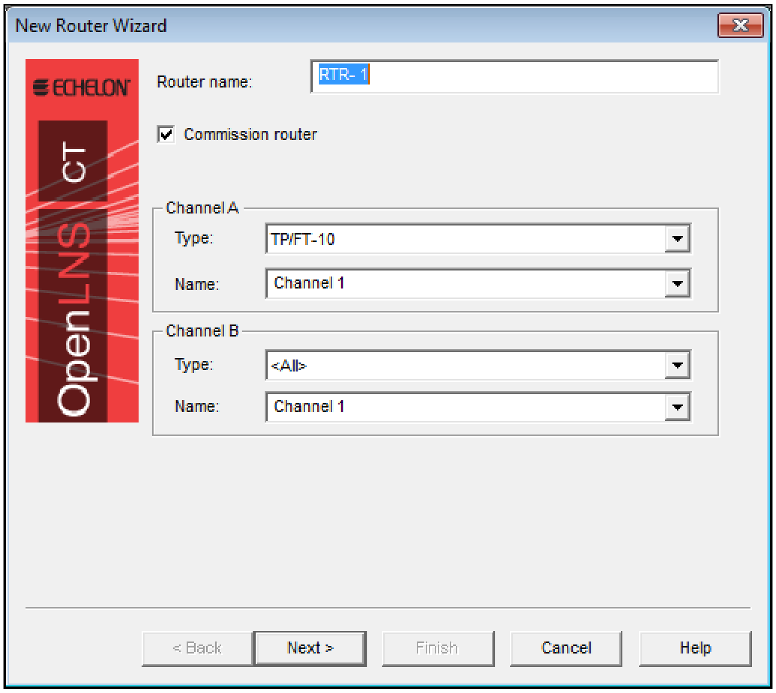
To add a router to the network, drag a router SmartShape to your OpenLNS CT drawing and then specify the router name, select the two channels it will connect, and specify the router properties.

The first page of the New Router Wizard includes a **Commission Router** check box for commissioning the router. Commissioning associates the physical router on the network with the router shape. To commission the router after adding it to the OpenLNS drawing, select this check box. The wizard will prompt you to identify the physical router before adding the router SmartShape to your network drawing. Otherwise, the wizard will add the router SmartShape to your network drawing after you have finished defining it.

You can create and commission router in one step. To do this, you enable automatic channel selection, drag a router SmartShape to your OpenLNS CT drawing near the two channels the router will connect, and then press the service pin on the router. If automatic channel selection is disabled, the New Router Wizard opens each time you add a router SmartShape to your OpenLNS CT drawing. See *Using Automatic Channel Selection for Routers* in this section for how to enable this feature.

To add a router to an OpenLNS CT drawing, follow these steps:

1. Drag a Router SmartShape () from the OpenLNS CT stencil to the drawing page. The New Router Wizard opens unless you enabled automatic channel selection.



New Router Wizard

Router name: RTR- 1

☑ Commission router

Channel A
Type: TP/FT-10
Name: Channel 1

Channel B
Type: <All>
Name: Channel 1

2. Enter the following information:

Router Name

Enter the name of the router. This name must be unique within the network. The router name may be up to 85 characters and include embedded spaces, but it may not include period, backslash, colon, forward slash, or double quote characters.

The default router name is “RTR- 1”. The instance number in the default router name is incremented for each subsequent router you create.

You can change this default naming convention by clicking **Add-ins**, clicking **OpenLNS CT**, clicking **OpenLNS CT Options**, and then selecting the **Naming** option from the **Options Category** list.

See *Naming Options* in Appendix A for information on how to use the options on this page to change the default naming convention.

Commission Router

Select this check box to commission the router after defining it.

Type

Select the channel type for each of the router’s two channels. By default, each of these boxes displays the channel type of the channel currently displayed in the **Name** list. You can use the **Type** list to filter the existing channels in the **Name** list to those with a specific channel type.

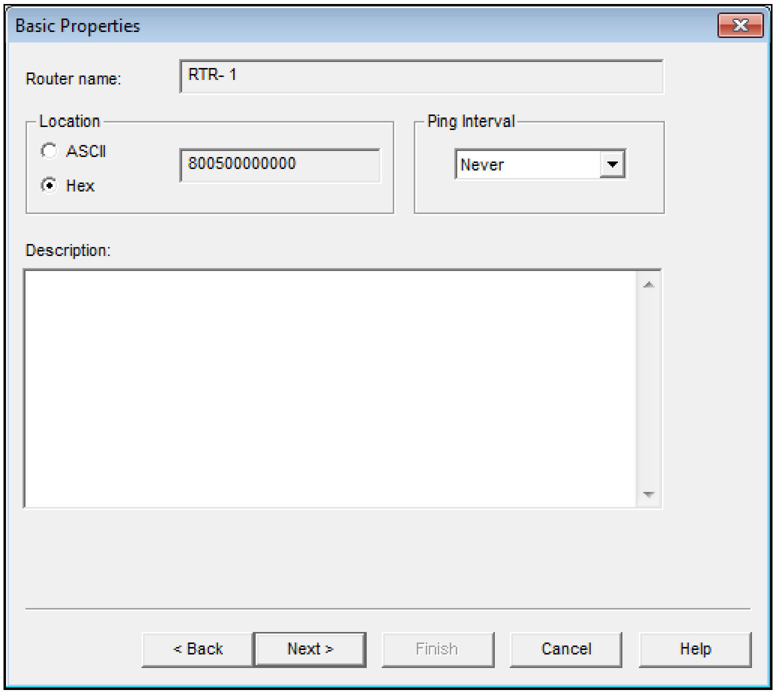
You can limit the channel types displayed by specifying the channel types in a custom router SmartShape.

Name

Select the two channels that you will attach to the router. You can enter a new channel name for one or both of the channels to add new channel SmartShapes to your OpenLNS CT drawing.

The channel names must be unique within the network and be no more than 85 characters long. If you are creating one or two new channels, specify the channel type for each channel in the **Type** box.

3. Click **Next**. The Router Properties window opens.



Basic Properties

Router name: RTR- 1

Location

○ ASCII

● Hex

800500000000

Ping Interval: Never

Description:

4. Enter the following information:

Location

Specifies the 6-byte location string for the router, which documents the router's location within the network.

To store the router's subsystem ID in this field, follow these steps:

3. Click **Add-ins,** point to **OpenLNS CT**, and then click **OpenLNS Options**. Select **Recovery** from the options category list and then select the **Set Location Property to Subsystem ID** check box under **Subsystem Recovery Options**.

4. Resynchronize the network. When you resynchronize the network, select the **Update Location Path and ID in Devices** check box in the **Synchronize OpenLNS CT Drawing** dialog. The router's current subsystem ID will be stored in this field. The OpenLNS CT tool can then use this information to place devices in their correct subsystems in the event you need to use the OpenLNS database Recovery Wizard to recover a lost network database.

   See *Recovering an OpenLNS CT Network* in Chapter 7, *Maintaining Networks*, for more information on the network recovery process.

If the **Set Location Property to Subsystem ID** check box is cleared, you can enter a location description for the router. You can enter up to 6 ASCII text characters, or up to 12 hex digits.

If you do not specify a value and the **Set Location Property to Subsystem ID** check box in the **Network Properties: Recovery** options category is cleared, the current value in the router will be written to this field and stored in the OpenLNS network database when the router is commissioned.

Ping Interval

Enables the OpenLNS Server to ping the router periodically. Pinging verifies that a router is still operating and communicating with the network.

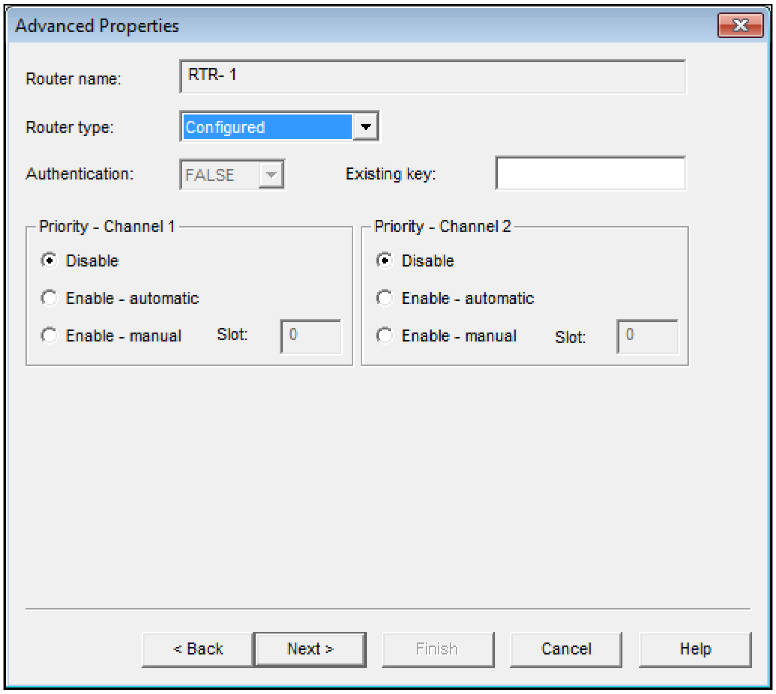
Set the ping interval based on the expected movement of the router. Select **Never** to disable pinging. To enable pinging, change the ping interval to **15 minutes** if you expect the router will seldom move or if the router is on a power line channel, **2 minutes** if you expect it will move somewhat frequently, and **1 minute** if you expect it will move frequently.

You must set a ping value for OpenLNS CT to display errors on a router's SmartShape. See *OpenLNS CT Device Styles* in Chapter 8, *Managing Networks*, for more information.

Description

Provides an optional description of the router. This description has no effect on network operation, but you can use it to provide additional documentation for as-built reports.

5. Click **Next**. The Advanced Router Properties window opens.



Advanced Properties

Router name: RTR- 1

Router type: Configured

Authentication: FALSE

Existing key:

Priority - Channel 1

● Disable

○ Enable - automatic

○ Enable - manual Slot: 0

Priority - Channel 2

● Disable

○ Enable - automatic

○ Enable - manual Slot: 0

6. Enter the following information:

Router Name

Displays the name of the router.

Router Type

Specify one of the following router types: **Configured**, **Repeater**, **Permanent Repeater**, **Bridge**, or **Permanent Bridge**.

The default mode is **Configured**, which means that the router determines which packets to forward based on internal routing tables that are configured by OpenLNS CT. This is the recommended type because it allows OpenLNS CT to optimize network traffic and determine automatically the channel to which each device is attached.

You can select **Repeater** to forward all valid packets received on one channel to the other channel. A **Permanent Repeater** behaves similarly, but its type cannot be changed in the network after the router is created.

You can select **Bridge** to forward all valid packets that match the network domain. A **Permanent Bridge** behaves similarly, but its type cannot be changed in the network after the router is created.

OpenLNS CT allows subnets to span permanent repeaters and bridges, but not non-permanent types. You can use permanent repeaters and bridges to preserve subnet IDs, and non-permanent types to maintain flexibility in order to change the router type later.

Authentication

Select whether authentication is enabled for communication with OpenLNS CT. Authentication can only be enabled for a router if authentication is enabled for the OpenLNS CT drawing. To check whether authentication is enabled for the OpenLNS CT drawing, click **Add-ins**, click **LonWorks Network**, click **Network Properties**, and then click the **Authentication** tab.

Existing Key

Specify the authentication key if the router has already been installed on another network with an authentication key. The authentication key can be up to 12 hexadecimal digits for 48-bit authentication or 24 hexadecimal digits for 96-bit authentication. If you specify a key of less than 12 digits, OpenLNS CT adds leading zeroes to create a 12-digit key.

Note: OpenLNS supports only a single authentication key for all authenticated devices; therefore, if you use 96-bit authentication keys, all devices that are authenticated must support 96-bit authentication.

Priority

Indicate whether priority messaging is enabled on one or both sides of the router. Priority messaging consumes network resources and bandwidth, and can degrade network performance. Select priority messaging when critical devices that use priority messaging are on one of the channels attached to the router and need to forward priority messages through the router. If you enable priority messaging, select whether the priority slot will be determined automatically or manually. If you select **Manual**, enter the priority slot in the **Slot** box.

7. If you selected the **Commission Device** check box on the first page of the wizard, the first window for commissioning the router opens. See *Commissioning a Device* in Chapter 5, *Installing Networks*, for instructions on commissioning the router.

   If you did not select the **Commission Device** check box on the first page of the wizard, click **Finish** to add the router SmartShape to your OpenLNS CT drawing.

Using Automatic Channel Selection for Routers

You can use automatic channel selection to enable OpenLNS CT to automatically select the appropriate channels for a new router, skipping the New Router Wizard. This feature lets you create a new router with a single click and drag. To enable automatic channel selection in your OpenLNS CT drawing, see *Using Automatic Channel Selection for Devices* earlier in this chapter.

Changing a Router Channel

You can change one or both of the channels assigned to a router SmartShape. You can do this to assign a router to a channel different from the one that was automatically selected by OpenLNS CT, or to physically move a router to a different compatible channel. To move a router to a different channel, you can either right-click the router and then click **Move Device** on the shortcut menu, or you can use the **Connector** tool on the Visio **Standard** toolbar to graphically move the router.

To use the **Move Device** command, right-click the router SmartShape and click **Move Device** on the shortcut menu. See *Moving OpenLNS CT SmartShapes* in this chapter for moving a device using the **Move Device** command.

To use the Visio **Connector** tool to move the device, follow these steps:

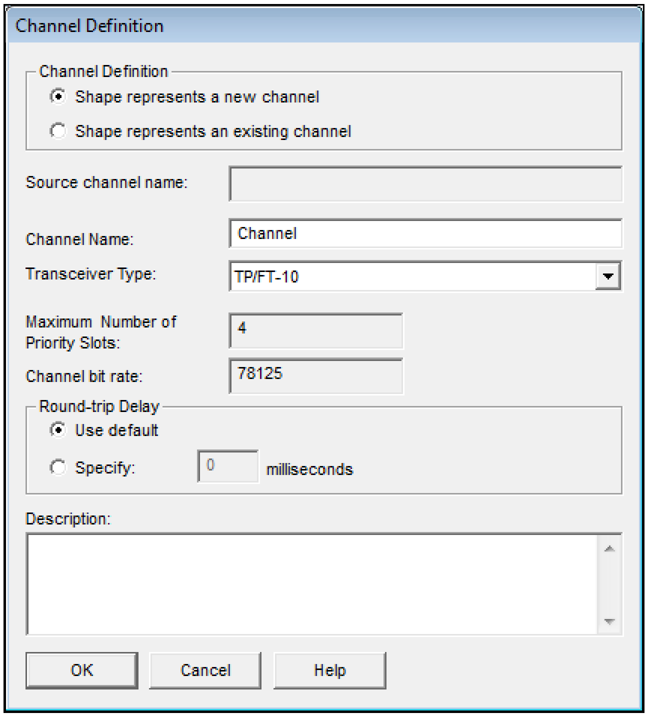
1. Select the **Connector** tool (┌┘) on the Visio Standard toolbar.
2. Position the Connector tool over the “**x**” connection point of the router so that a red box opens over the “**x**”.
3. Click and hold the mouse button.
4. Drag the **Connector** tool to the center of the SmartShape of the desired channel so that a red box opens around the channel SmartShape.
5. Release the mouse button.
6. Select a different tool on the Visio **Standard** toolbar if you will not be making additional connections or reassignments (if you continue to use the **Connector** tool to add new devices, routers, or functional blocks SmartShapes to your drawing, Visio will attempt to connect the SmartShapes, which may not be the desired action or could be an action that generate errors).

Creating a Channel

A *channel* is the physical media upon which devices communicate between each other. When you initially create an OpenLNS CT drawing, it contains one channel named **Channel 1** by default. You can add additional channels to your OpenLNS CT drawing. Doing so allows you to use different media within the same network, isolate network traffic for performance, isolate devices for reliability, and increase the number of devices beyond the limit of a specific transceiver.

To add a channel to an OpenLNS CT drawing, follow these steps:

1. Drag the Channel SmartShape (⊟) from the **OpenLNS CT Basic Shapes** stencil to your OpenLNS CT drawing. The **Channel Definition** dialog box opens.



Channel Definition

Channel Definition
● Shape represents a new channel
○ Shape represents an existing channel

Source channel name:

Channel Name: Channel

Transceiver Type: TP/FT-10

Maximum Number of Priority Slots: 4

Channel bit rate: 78125

Round-trip Delay
● Use default
○ Specify: 0 milliseconds

Description:

2. Enter the following information:

Channel Definition

Select whether the SmartShape represents a new or an existing channel.

- Select **Shape Represents a New Channel** if you are adding a new channel to your OpenLNS CT drawing.
- Select **Shape Represents an Existing Channel** to place devices in different subsystems on the same channel. Each subsystem may have only one channel SmartShape for a given channel, but the same channel may be referenced by channel SmartShapes in many subsystems.

Source Channel Name

If you are creating a channel from a copy (a custom master SmartShape that includes a channel or an existing channel), this field displays the name of the source channel.

Channel Name

If you are creating a new channel, enter the channel name. The name must be unique within the network and may be up to 85 characters long. The name may include embedded spaces, but it may not include periods, backslashes, colons, forward slashes, or double quotes.

If you are adding an existing channel to the current subsystem, select which channel you want to add from the **Channel Name** list.

Transceiver Type

If you are creating a new channel, select the channel type from the list.

If you are adding an existing channel to the current subsystem, the channel type will be displayed in a read-only box.

Maximum Number of Priority Slots

Displays the maximum number of priority slots available on the selected channel. This is a read-only box unless **Transceiver Type** is set to **Custom**.

Channel Bit Rate

Specifies the typical performance (packets per second) of the selected channel type.

Round-Trip Delay

Select whether to use the default round-trip delay or to specify a time for the channel in milliseconds.

- Select **Use Default** to use the default of two packet cycles based on the average packet size.
- Select **Specify** to set the value to the expected longest round-trip time of a message (for example, message and response). This option allows expected traffic patterns to be input into the system so that the timer calculations can be affected accordingly.

Description

If you are creating a new channel, you can enter an optional description of the channel. This description has no effect on network operation, but you can use it to provide additional documentation for as-built reports.

3. Click **Finish** to add the channel SmartShape to your OpenLNS CT drawing.

Creating a Subsystem

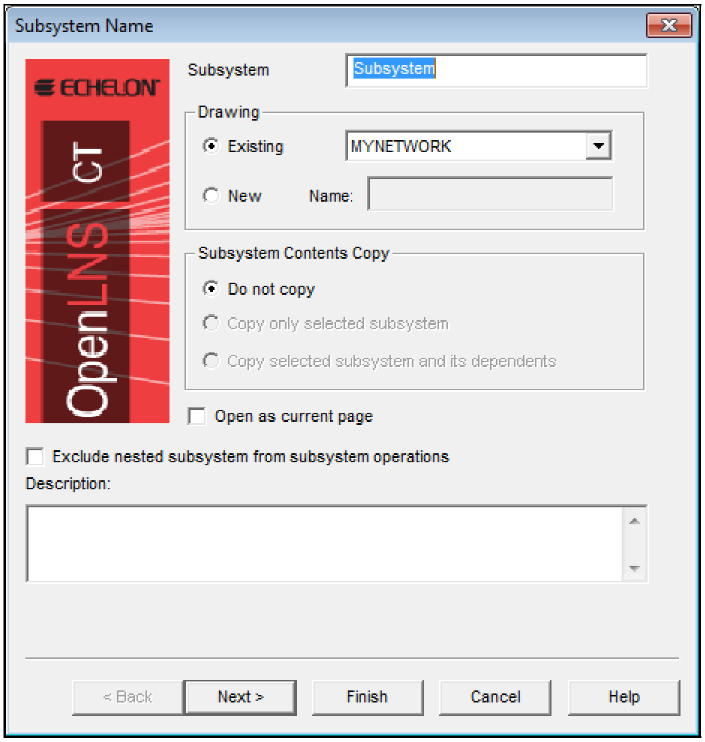
Subsystems allow you to break a large network into more manageable parts. In a building control network, for example, one subsystem could contain lighting controls and another temperature controls.

Each page of an OpenLNS CT drawing represents a subsystem. Subsystems may be nested like directories on your computer, so the temperature control subsystem could contain other subsystems representing each floor of the building, for example.

You can also create subsystems in separate OpenLNS CT drawings. Using this feature, you can create large networks while maximizing Visio’s performance by limiting the number of pages in a single drawing. You can also create separate subsystems for large networks with multiple installers in which each installer works on their respective subsystem.

To add a subsystem to an OpenLNS CT drawing, follow these steps:

1. Drag the Subsystem SmartShape (▣) from the **OpenLNS CT Basic Shapes** stencil to your OpenLNS CT drawing. The New Subsystem Wizard opens.



Subsystem: Subsystem

Drawing

● Existing MYNETWORK

○ New Name:

Subsystem Contents Copy

● Do not copy

○ Copy only selected subsystem

○ Copy selected subsystem and its dependents

□ Open as current page

□ Exclude nested subsystem from subsystem operations

Description:

2. Enter the following information:

Subsystem

Enter the name of the subsystem. The name must be unique within the current subsystem and may be up to 85 characters long. The name may include embedded spaces, but it may not include periods, backslashes, colons, forward slashes, or double quotes. Subsystem names are case sensitive.

If you create a subsystem from a copy (a custom subsystem master SmartShape or an existing subsystem SmartShape), OpenLNS CT will automatically append a number, starting at 1, to the subsystem name in the page tab. The number will be incremented by one each time a subsystem with same name is created. The subsystem name in the page tab, though, may not match that in the title block at the top-center of the drawing.

Subsystems with same name can exist within different levels in the subsystem hierarchy of the same network.

Drawing

Select whether to create the new subsystem in an existing subsystem or in a new OpenLNS CT drawing page.

- Select **Existing** to create the subsystem in an existing subsystem within the current OpenLNS CT network. Select the existing subsystem in which the new subsystem will be placed. The current subsystem is listed by default.
- Select **New** to create the subsystem in a new drawing page within the current OpenLNS CT network. Enter the name of the new drawing page in the **Name** box. The default name is <*Network Name*> **Subsystem** [*Index*]. An index will be appended to the name of the drawing page if a naming conflict exists.

Subsystem Contents

Select which objects to copy in the subsystem and any nested

Copy

subsystems. You have three choices:

- **Do not Copy**. Only the subsystem SmartShape will be copied. None of the contents of the original subsystem will be copied into the new subsystem. This is the only option available when you create a new subsystem by dragging a Subsystem SmartShape from the **OpenLNS CT Basic Shapes** stencil to the drawing (all of the options are available when you create a new subsystem by copying an existing one).
- **Copy only Selected Subsystem**. Only the contents of the original subsystem will be copied to the new subsystem. Nested subsystems and their contents will not be copied.
- **Copy Selected**. All contents of the original subsystem, including nested subsystems and their contents, will be copied to the new subsystem.

Open as Current Page

Open the new subsystem after it has been created. If this check box is cleared, OpenLNS CT will create the new subsystem but remain in the current subsystem. This check box is cleared by default.

Exclude Nested Subsystems from Subsystem Operations

Applies subsystem operations to only the devices in the parent subsystem. This check box is cleared by default, which means that subsystem operations are applied to devices in the parent subsystem and all of its nested subsystems.

Description

Provides an optional description of the subsystem. This description has no effect on network operation, but you can use it to provide additional documentation for as-built reports.

3. If you are creating this subsystem in a network that uses user profiles and passwords, click **Next** to set *Guest Privileges* for this subsystem (see *User Profiles* in Chapter 3, *Getting Started* for more information). Otherwise, click **Finish** to add the subsystem to your OpenLNS CT drawing.

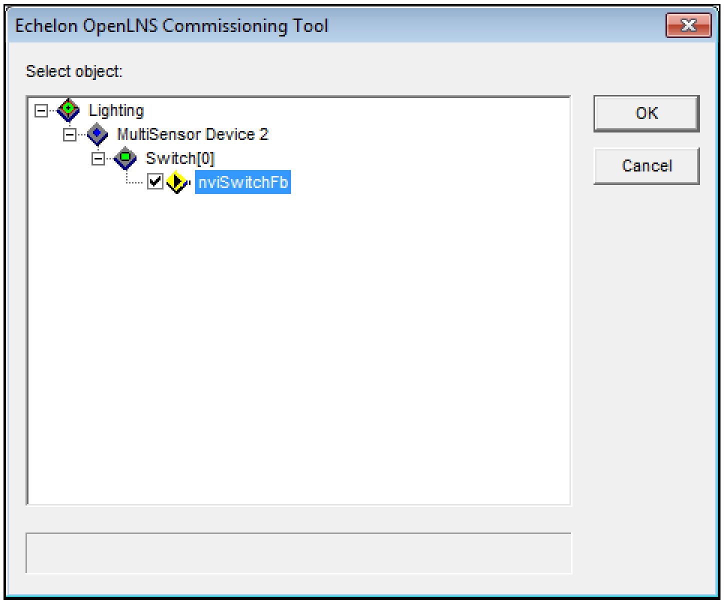
Creating a Supernode

A supernode is a subsystem with its own network variable interface. Using a supernode allows you to provide a simple interface to an arbitrarily complex subsystem. For example, a supernode that contains lighting controls may have a network variable interface that allows the devices contained within it to be put into emergency override. This allows the integrator to identify easily the critical interfaces into the subsystem while ignoring those that are typically only used internally by the subsystem.

Any network variable on any functional block in a subsystem may be exported one or more levels up the subsystem hierarchy to the subsystem SmartShape and be used as a supernode.

To create a supernode, follow these steps:

1. Create a subsystem, and then create one or more devices and one or more functional blocks in that subsystem.
2. Go back to the parent subsystem and drag a network variable SmartShape onto the subsystem SmartShape you just created.
3. A dialog opens in which you select a network variable of the appropriate direction in that subsystem.



4. Right-click the subsystem and click **Expand** from the network navigator. All devices, functional blocks, and available network variables in the network hierarchy will be displayed. Select one or more network variables to be supernodes. To select all the available network variables, click **Select All** from the network navigator.

5. Click **OK**. SmartShapes for the selected network variables appear on the subsystem SmartShape.

Tips: You can connect supernodes just as you would use network variables on a functional block. See *Creating Connections* in this chapter for more information on connecting network variables.

For optimal performance, limit the number of network variables on a single supernode to 30. More than 30 network variables on a single supernode will degrade your network's performance.

Renaming and Deleting a Supernode Network Variable

You can change the name or delete an unconnected supernode network variable SmartShape. Changing the name of or deleting a supernode network variable SmartShape will not affect the naming or presence of the source network variable.

To change the name of a supernode network variable, click the subsystem SmartShape and click the network variable SmartShape you want to rename. Right-click the selected network variable SmartShape, select **Properties** from the shortcut menu, and enter a new name in **Exported Name** box in the **Network Variables Properties** dialog box.

To delete a network variable from a supernode, click the subsystem SmartShape and click the network variable SmartShape you want to delete. Right-click the selected network variable SmartShape and select **Delete** from the shortcut menu.

Copying a Subsystem or Supernode

You can copy a subsystem or supernode to create a new subsystem or supernode with the same devices, functional blocks, connections, and configuration as the original, including any nested subsystems and supernodes.

To copy a subsystem or supernode, follow these steps:

1. Right-click the subsystem or supernode SmartShape to be copied and click **Copy** on the shortcut menu.

2. Right-click anywhere in the OpenLNS CT drawing, in the same or in a different subsystem, and click **Paste** on the shortcut menu. The New Subsystem Wizard opens.

   Instead of performing steps 1 and 2, you can hold down CTRL and drag the subsystem or supernode SmartShape to the desired location.

3. Enter the information for the new subsystem following the instructions in *Creating a Subsystem*.
4. Click **Finish**. OpenLNS CT copies the subsystem or supernode. If only a single end of a reference connection is in a copied subsystem, the network variable will be connected to the existing connection specified by the reference. The copy operation does not require any user input, with the following exceptions:

   - OpenLNS CT prompts you to determine whether copied channel SmartShapes represent an existing channel or a new channel.
   - If the copy causes functional blocks to be created that are not associated with any device, OpenLNS CT prompts you to determine on which devices the new functional blocks reside.

Creating Connections

Creating connections enables devices to exchange data. You can create *unicast* connections in which an output network variable is connected to a compatible input network variable, and you can create *multicast* connections in which a single output or input network variable is connected to multiple compatible input or output network variables, respectively. Network variables must have the same type to be compatible. Once you create a connection, the input network variables will receive all updates from the output network variables in the connection. You can also connect input and output message tags, and the input message tags will receive all updates from the output message tags in the connection. This process of connecting network variables and message tags is called *binding*, and the logical connections are thought of as *virtual wires*.

You can connect a network variable on any functional block or supernode in a network with a compatible network variable on any other functional block or supernode—even if the network variables are in different subsystems. You can also create a connection between input and output network variables on the same device. This is called a *turnaround connection*. A turnaround connection may be appropriate for a device that receives the data it transmits.

Every connection has one *hub* network variable and one or more *target* network variables. The end of the hub network variable is marked with an **x**; the target network variables are each marked with a **+**. When creating a unicast connection, it does not matter which end of the connection is the hub and which is the target. However, when creating large multicast *fan-in* connections (multiple output network variables connected to a single input network variable) or *fan-out* connections (a single output network variable connected to multiple input network variables), make the common network variable the hub. This will reduce the time required to make the connection. For example, in a fan-in connection, the single input network variable should be the hub; in a fan-out connection, the single output network variable should be the hub.

You can connect functional blocks in three ways:

- Use the Connector SmartShape (┐) on the **OpenLNS CT Basic Shapes** stencil.
- Use the **Connector** tool on the Visio **Standard** toolbar. This is the easiest way to create unicast connections.
- Use the **Network Variable Connection** dialog box. This is the easiest way to create multicast connections.

The following subsections explain how to use each of these three methods.

Creating a Connection with the Connector SmartShape

You can create a connection with the Connector SmartShape (┐) in the **OpenLNS CT Basic Shapes** stencil. You can only use this method to connect functional blocks or supernodes that are in the same subsystem. You can connect network variables or message tags on functional blocks or supernodes whether they have network variable or message tag SmartShapes or not.

To connect functional blocks or supernodes with network variable or message tag SmartShapes, drag a Connector SmartShape from the **OpenLNS CT Basic Shapes** stencil to your OpenLNS CT drawing, placing the left end of the Connector SmartShape on the connection point on the end of the output network variable or output message tag you want to connect (a red square will appear around the '**x**' connection point). Then drag the right end of the Connector SmartShape to the connection point of the input network variable or input message tag. Connector SmartShapes that have been properly connected on both ends will be black; otherwise, they will be red. To create a fan-out or fan-in connection, repeat the process with additional Connector SmartShapes.

Note: You must connect the connector shape to the output network variable before connecting it to the input network variable, and you must make the connection as you add the connector shape to the drawing. If you do not connect the output network variable first or you drop the connector shape in the drawing and then connect the network variables, OpenLNS CT may not use a smart connection description for the connection (for example, repeated service for a fan-out connection). In addition, you must make the connection as you add the connector shape to the drawing (you cannot add the connector shape to the drawing and then connect the network variables).

To attach a Connector SmartShape to a functional block that does not have a network variable or message tag SmartShape, place one end of the connector SmartShape directly onto one of the four connection points around the functional block, and connect the other end to a network variable or message tag SmartShape or onto one of the four connection points around another functional block. Once a functional block is connected via one of its connection points, the **Network Variable Connection** dialog box opens. Follow the steps in *Creating a Connection with the Network Variable Connection Dialog Box* to finish creating the connection.

If you create a connection with multiple message tags, connect the output message tags together (instead of using the **msg_in** tag). Select one of the output message tags as the hub and connect it to all the other output message tags (do not daisy chain the connection). The resulting connection will be the same regardless of which message tag you select as the hub.

See *Connection Master SmartShapes* in Chapter 12, *Creating and Using Custom OpenLNS CT SmartShapes and Stencils*, for information on creating multiple connections in a single operation.

Creating a Connection with the Connector Tool

You can create a connection using the **Connector** tool. This is the quickest way to create unicast connections. To create a connection with the **Connector** tool, follow these steps:

1. Select the **Connector** tool (⌐) on the Visio **Standard** toolbar.
2. Point to the output network variable or message tag SmartShape of the functional block or supernode to be connected until a red square opens around the SmartShape.
3. Hold down the mouse button and then point to the input network variable or message tag SmartShape of the functional block or supernode to be connected.
4. Release the mouse button.
5. Select a different tool on the Visio toolbar if you will not be making additional connections or reassignments. If you continue to use the **Connector** tool to add new devices, routers, or functional blocks SmartShapes to your drawing, Visio will attempt to connect the SmartShapes, which may not be desired or could generate errors.

Creating a Connection with the Network Variable Connection Dialog Box

You can create a connection using the **Network Variable Connection** dialog box. This dialog box provides a powerful and flexible method for creating connections. It provides a hierarchal view of the devices, functional blocks, and network variables in your OpenLNS CT network. You use this view to select the network variable hub and one or more targets to connect. It is the quickest method for creating fan-in or fan-out connections because you can select multiple network variable targets to connect to a hub network variable with a few mouse clicks. You can change the properties of the connection while you are creating it (you can also change the properties after you create the connection).

You must use this dialog if you want to connect network variables or message tags that are located in different subsystems, or if you are connecting a functional block without network variable or message tag SmartShapes.

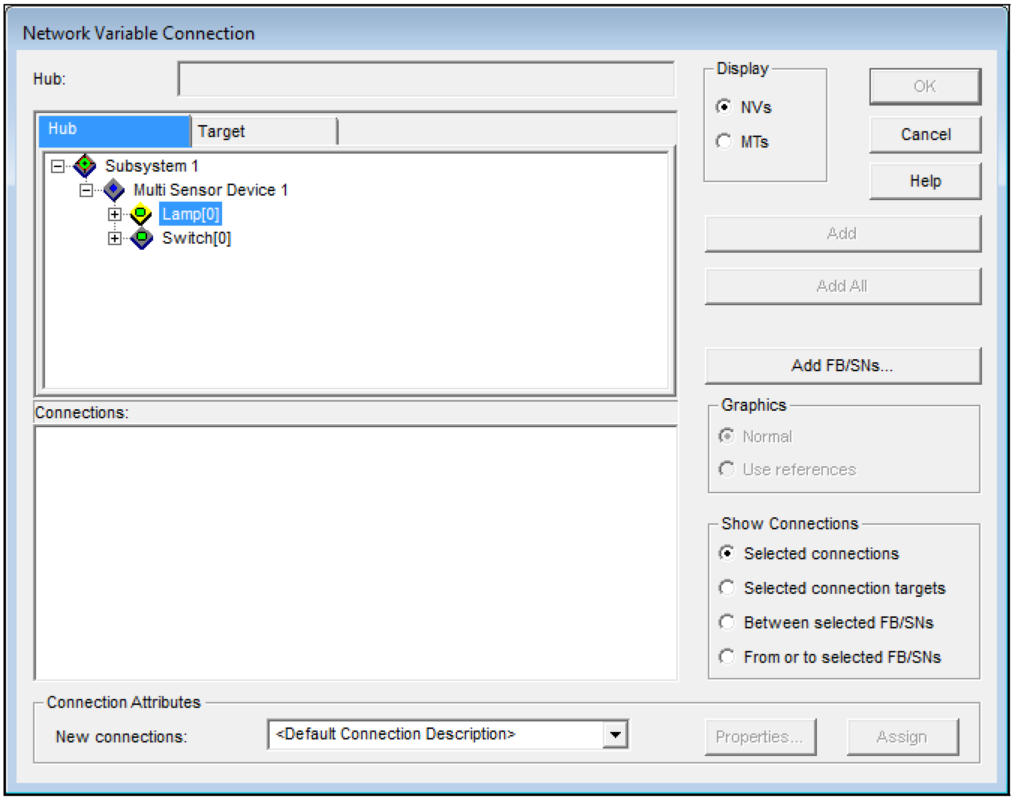
To create a connection using the **Network Variable Connection** dialog box, follow these steps:

1. To connect two or more functional blocks or supernodes in the same subsystem, hold down CTRL and click the functional blocks and supernodes to be connected. The selected functional blocks and supernodes will be highlighted pink, and a green box will appear around them. Right-click one of the selected functional blocks or supernodes and then click **Connect** on the shortcut menu.

    - To connect functional blocks in different subsystems or to select network variables to be connected, right-click an empty space in your OpenLNS CT drawing, and then click **Connect** on the shortcut menu. The Select Objects window opens. Use the network navigator to select the functional blocks and/or supernodes you want to connect and then click **OK**.

    - To create a turnaround connection, right-click a functional block or supernode and then click **Connect** on the shortcut menu. The Select Objects window opens. Use the network navigator to select the functional block or supernode to connect and then click **OK**.

2. The **Network Variable Connection** dialog box opens.



Display: ● NVs ○ MTs

Graphics: ● Normal ○ Use references

Show Connections: ● Selected connections ○ Selected connection targets ○ Between selected FB/SNs ○ From or to selected FB/SNs

New connections: <Default Connection Description>

3. Create the network variable connection by selecting the following options:

Hub: (box) Displays the currently selected hub network variable. This is useful when you are selecting the target network variables.

Hub (tab) Displays a hierarchal view of the network, consisting of the subsystems, devices, functional blocks, and all available network variables that can be selected as the hub. Select the hub network variable on this tab before clicking the **Target** tab.

Target Displays a hierarchal view of the network, consisting of the subsystems, devices, functional blocks, and all compatible network variables that can be selected as targets. Select a target and then click **Add**.

Connections Lists the connections that are scheduled to be created.

Connection Attributes

New Connections Displays the connection description that will be assigned to all new connections created. Select **<Default Connection Description>** to use a smart connection description as described in *Using Automatic Connection Description Selection*.

Properties Opens the **Connection Details** dialog, which specifies the attributes of the selected connection description. You can also create new connection descriptions in this dialog.

Assign Assigns the currently selected connection description to all connections in the **Connections** box.

Display Select which types of connection points are displayed.

- **NV**. Displays available or compatible network variables.
- **MT**. Displays available message tags. This option is only available if you are connecting virtual functional blocks.

Add Adds a connection with the currently selected network variable target to the Connections pane.

Add All Adds connections with all compatible network variable targets to the Connection pane.

Add FB/SNs Opens the **Select Object** dialog box, which you can use to add functional blocks or supernodes to the hub and target tabs. This includes functional blocks or supernodes in different subsystems and those in the same subsystem that were not initially selected. Right-click any object in this dialog and use the network navigator to expand, select, and deselect functional blocks or supernodes. Click **OK** when finished.

Graphics Select the type of connector SmartShape to be used in the OpenLNS CT drawing for the connection.

- **Normal**. A connection SmartShape will be used to represent the new connection.
- **Use References**. A reference connection SmartShape will be used to represent the new connection. This option will automatically be selected if you are connecting functional blocks or supernodes in different subsystems.

Show Connections Select which connections are displayed in the Connections box. You have four choices:

- **Selected Connections**. Displays the connections that you selected in the OpenLNS CT drawing when you opened the **Network Variable Connections** dialog box.
- **Selected Connection Targets**. Displays all connections to the selected targets.
- **Between Selected FB/SNs**. Displays connections between any of the selected functional blocks and supernodes and displays connection between any pairs.
- **From or To Selected FB/SNs**. Displays all connections from or to all selected functional blocks and supernodes.

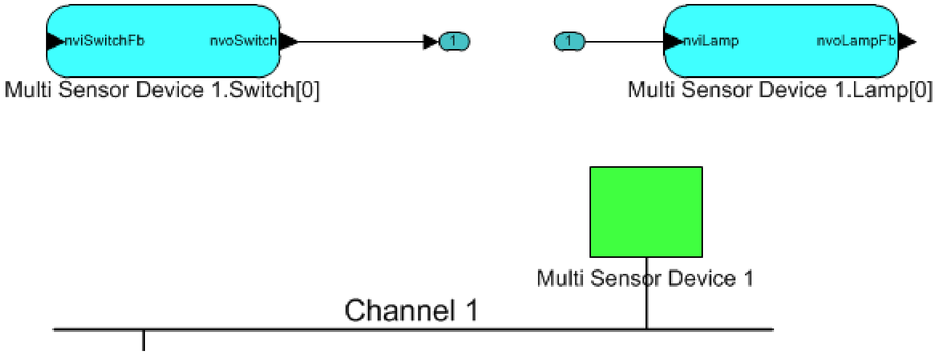
4. Under the **Hub** tab, select the hub network variable for the connection. The icon for the selected NV will be highlighted yellow. To connect a network variable on a functional block or supernode not listed in the **Hub** tab, including those in a different subsystem, click **Add FB/SNs** and add the desired functional block or supernode.
5. Click the **Target** tab. Select the desired target network variable for the connection. The icon for the selected target network variable will be highlighted yellow. To connect a network variable on a functional block or supernode that is not listed in the **Target** tab, including those in a different subsystem, click **Add FB/SNs** to add the desired functional block or supernode.
6. Click **Add**. The connection will be added to the **Connections** box below. Repeat steps 3 and 4 to create additional connections. Optionally, you can select one target network variable and then **Add All** to create connections to the selected hub with all compatible target network variables with the same name and type of the selected target network variable. This is useful for quickly creating large fan-in or fan-out connections.

7. In the **New Connections** box, select a connection description to assign to all the new connections listed in the **Connections** box. You can click **Properties** to view and modify the properties of the connection description displayed in the **New Connections** box. See *Using Connection Descriptions* in this chapter for more information on using and creating connection descriptions.
8. Click **OK**. The connections listed in the **Connections** box will be added to your OpenLNS CT drawing.
9. If OpenLNS CT is attached to the network and OnNet, and the devices participating in the connection have been commissioned, the connection information in the physical devices is also updated.

Using Reference Connection SmartShapes

You can use reference connection SmartShapes to simplify an OpenLNS CT drawing and connect functional blocks and supernodes in different subsystems. A reference connection replaces the single connector SmartShape between a set of connected functional blocks or supernodes with a pair of identically named reference connection SmartShapes that are attached to both ends of the connection. The reference connection functions in the same manner as a normal connection— the only difference is the graphics used to represent the connection.

Reference connection SmartShapes by default are numbered, starting at 1. The numbering for each subsequent set of reference connector SmartShapes used is incremented by one. You can rename the reference connector SmartShapes to something meaningful; however, the name must be unique within the network. If a name is not unique, OpenLNS CT will suggest an alternative name that is unique (it will suggest a name with an index [starting at 1] appended to the name, or a name with the index incremented). If you change the name on one end of the connection, the name on the other end will automatically be updated. You cannot renumber reference connection SmartShapes.



Reference connections are automatically used for connections between functional blocks or supernodes in different subsystems. You can also use reference connections to simplify your OpenLNS CT drawing by reducing the number of crisscrossing connector SmartShapes that can clutter an OpenLNS CT drawing.

Note: Moving a reference connection in a remote lightweight client does not move it on the local client.

To use reference connection SmartShapes, you can create new connections that are represented by reference connection SmartShapes (see *Creating a Connection with the Network Variable Connection Dialog Box*), or you can right-click an existing connection represented by a connector SmartShape and then click **Use Reference** on the shortcut menu to change the connector SmartShape to a reference connection SmartShape.

Aligning Reference Connection SmartShapes

You can automatically align and keep aligned all the reference connection SmartShapes in a subsystem or the reference connection SmartShapes attached to a specific functional block. This feature eliminates the need to manually move functional block, connector, and connector reference SmartShapes to keep your OpenLNS CT drawing organized.

To align all reference connection SmartShapes in a subsystem, right-click an empty space in the subsystem, point to **Display Options**, and then click **Align All Reference Shapes** on the shortcut menu. This will set the **Align Reference Shapes** option on all the functional blocks in the subsystem and align all the reference connection SmartShapes in the subsystem based on the settings in the **Network Properties: Functional Block** options category.

To enable automatic reference connection SmartShape alignment for one or more functional blocks, right-click the functional block (or select multiple functional blocks and right-click one of them), point to **Display Options**, and then click **Align Reference Shapes** if it is not set. The reference connection SmartShapes attached to the selected functional block(s) will then be aligned based on the settings in the **Network Properties: Functional Block** options category.

Once the automatic alignment feature is enabled on a functional block, all attached reference connection SmartShapes will follow the functional block when it is moved. In addition, reference connection SmartShapes will automatically be re-aligned when you make changes your OpenLNS CT drawing (for example, if you add or remove network variables or reference connection SmartShapes).

When this feature is disabled, the attached reference connection SmartShapes are no longer locked to the location of the functional block SmartShape. The attached reference connection SmartShapes will initially still move with the functional block SmartShape, but they can also be moved independently. However, once you move a reference connection SmartShape, it will no longer follow the functional block SmartShape.

See *Functional Block Options* in Appendix A for customizing the automatic reference connection SmartShape alignment feature.

Using Connection Descriptions

You can set the properties for a connection by either manually selecting a connection description or having OpenLNS CT automatically select a connection description.

Connection Description Properties

A connection description is a set of properties, namely the message service type and addressing mode, that apply to a network variable connection. The message service type determines how network variable updates (messages) are delivered and verified. The addressing mode determines how a message packet travels over the network. There are three main message service types: acknowledged, repeated, and unacknowledged. Addressing modes include subnet/node ID for unicast connections, and alias, broadcast, or group for multicast (fan-out) connections.

Message Service Type

You can specify one of the following message service types for each connection description. The message service types vary in reliability and resources consumed.

- *Acknowledged.* The sending device expects to receive confirmation from the receiving device or devices that a network variable update was delivered. The sending application is notified when an update fails, but it is up to the developer of the sending device to handle the notification in the device application. While acknowledged service is very reliable, it can create excessive message traffic, especially for large fan-out or polled fan-in connections. When acknowledged messaging is used, every receiving device has to return an acknowledgment. Acknowledged messaging can be used with up to 63 receiving devices, but an acknowledged message to 63 devices generates at least 63 acknowledgements—more if any retries are required due to lost acknowledgements.

- *Repeated.* The sending device sends out a series of network variable updates, but does not expect any confirmation from the receiving device. Repeated service with three repeats has a 99.999% success rate in delivering messages. Repeated service provides the same probability of message delivery as acknowledged messaging with the same number of retries, with significantly lower network overhead for large multicast fan-out connections. For example, a repeated message with three retries to 64 devices generates 4 packets on the network, whereas an acknowledged message requires at least 65 packets

- *Unacknowledged*. The sending device sends out the network variable update only once and does not expect any confirmation from the receiving device. This message service type consumes the least amount of resources, but is the least reliable.

You can enable authentication for any acknowledged connection. With *authentication*, the receiving device challenges the sending device for verification in between receiving the network variable update and sending an acknowledgement to the sending device. This results in significantly more traffic as acknowledged service.

You can enable priority messaging for any connection. With *priority messaging*, the device with the highest priority sends its packet before any other devices can send theirs. This is accomplished by assigning each priority device a time slot where it can transmit before all other lower priority and non-priority devices. These time slots consume network bandwidth; therefore, priority messaging should only be used for critical devices and data.

Addressing Mode

You can specify the addressing mode for each connection description. The addressing mode specifies the type of destination addressed that is used for packets within the connection. Addressing modes include subnet/node ID for unicast connections, and broadcast, group, or alias for multicast (fan-out) connections. Each of these addressing modes is described as follows:

- *Subnet/Node ID addressing*. Message packets travel from the sending device to the destination device using the 2-byte logical address of the destination device.
- *Broadcast addressing*. Message packets are sent to all devices within a domain (all devices in a network) or a subnet (a set of devices on one channel or a set of channels connected by permanent repeaters).
- *Group addressing*. Message packets are sent to devices assigned to a group, which requires a 1-byte address shared by all members of the group.
- *Alias addressing*. Message packets are sent to devices using network variable aliases and unicast addressing, if possible. Aliases behave like shadow copies of network variables and can be used to overcome connection limitations of non-alias network variables.

Subnet/node ID and broadcast addressing do not require an address table entry on the destination device or devices, whereas group addressing does. Alias addressing requires an address table entry on the destination device when used with group addressing.

Viewing and Creating Connection Descriptions

OpenLNS CT provides fifteen pre-defined connection descriptions, which vary in the message service type and addressing mode used. The following table details each of these pre-defined connection descriptions.

Connection Description	**Smart Connection Description (SCD)***	**Messaging Service**	**Multicast Addressing Mode**
Acknowledged		Acknowledged	Group
Acknowledged with Alias Multicast		Acknowledged	Alias
ConnectDescTemplate_0**		N/A	N/A
Repeated***		Repeated	Group

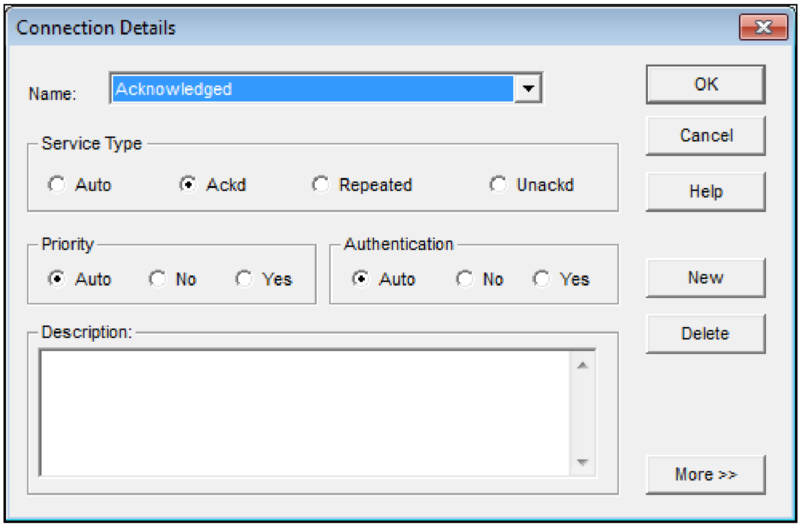
Repeated with Alias Multicast		Repeated	Alias
Repeated with Broadcast Multicast		Repeated	Broadcast
Repeated with Group Multicast		Repeated	Group
SCD_Alias	●	Acknowledged	Alias
SCD_Default	●	Repeated	Group (if available); otherwise, domain broadcast
SCD_SubnetBcast	●	Repeated	Subnet broadcast
SCD_Unicast	●	Acknowledged	Group
Unacknowledged***		Unacknowledged	Group
Unacknowledged with Alias Multicast		Unacknowledged	Alias
Unacknowledged with Broadcast Multicast		Unacknowledged	Broadcast
Unacknowledged with Group Multicast Unacknowledged**		Unacknowledged	Group

***Smart Connection Descriptions (SCDs)** are pre-defined connection descriptions that OpenLNS CT automatically assigns to a connection being created or changed based on the number of target network variables, whether the target network variables are located on the same or different subnet, and the availability of network variable aliases. See *Using Automatic Connection Description Selection* for more information.

****ConnectDescTemplate_0** is the default connection description. It enables OpenLNS CT to automatically select a smart connection description that uses the most appropriate message service type and addressing mode for the connection. It cannot be modified or deleted.

*****Repeated** and **Repeated with Group Multicast** are identical, as are **Unacknowledged** and **Unacknowledged with Group Multicast**. **Repeated** and **Unacknowledged** support connections using these connection descriptions in OpenLNS CT networks that were created with previous releases of the LonMaker tool.

You can view the properties of these pre-defined connections and create new connection descriptions in the **Connection Details** dialog box. You can also use this dialog to change the properties of a pre-defined connection. To access this dialog box, right-click a connector SmartShape, select **Properties** from the shortcut menu, and click **Properties** on the **Network Variable Connection** dialog box. Optionally, you can click **Add-ins**, click **OpenLNS CT**, click **OpenLNS CT Options**, select **Smart Connection Descriptions** from the options category list, and then click **Details**.



Name: Acknowledged

Service Type: ○ Auto ● Ackd ○ Repeated ○ Unackd

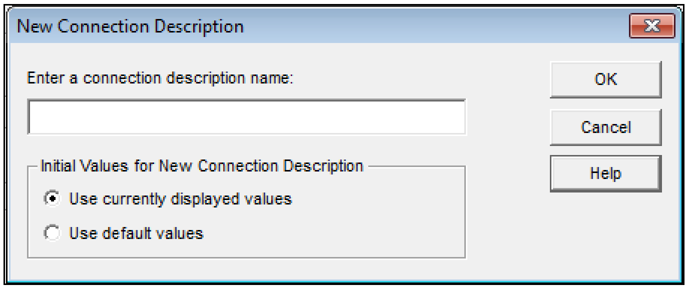
Priority: ● Auto ○ No ○ Yes

Authentication: ● Auto ○ No ○ Yes

Description:

To view the properties of a connection description, select the connection description from the **Name** list. If you do make changes, click **OK** and OpenLNS CT then updates all network variable members of the connection with the modified connection description

To create a new connection description, click **New**. The **New Connection Description** dialog box opens.


New Connection Description
Enter a connection description name:
OK
Cancel
Help
Initial Values for New Connection Description
● Use currently displayed values
○ Use default values

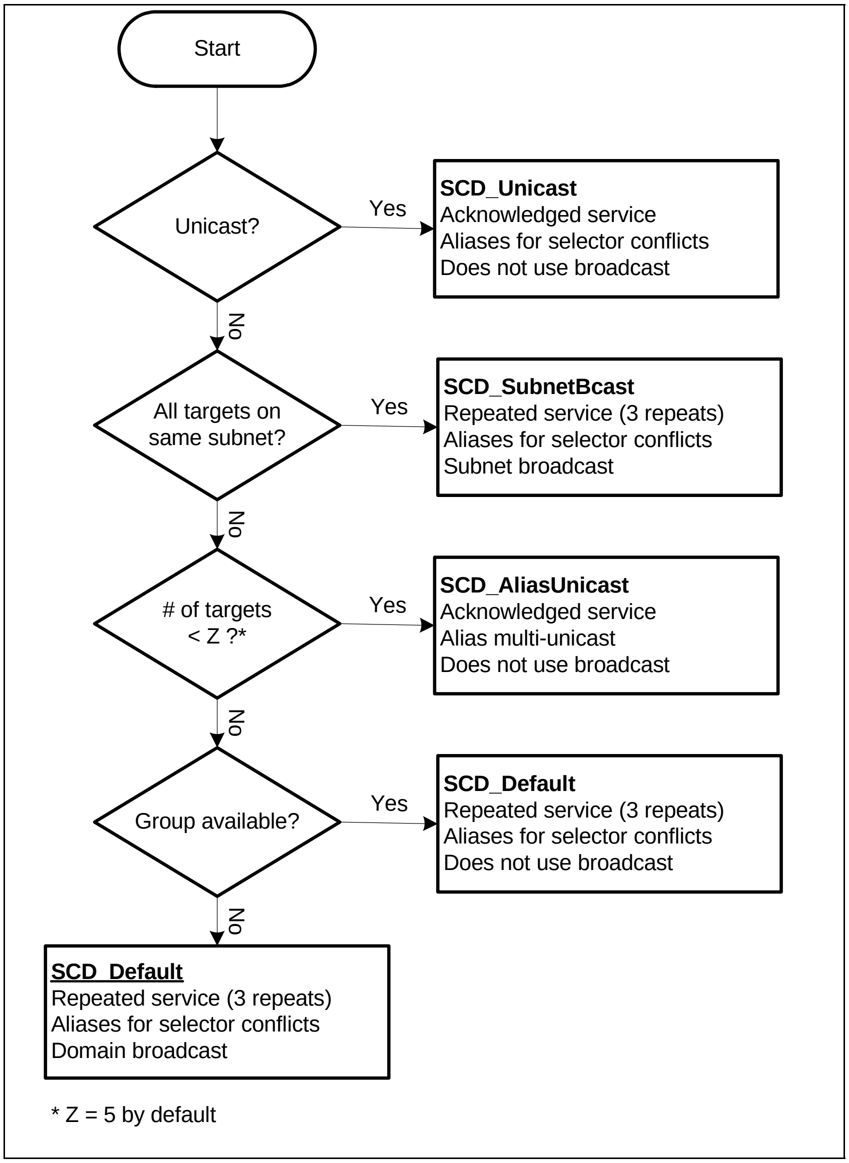
Enter the name of the new connection description and select whether the currently displayed or default attributes are used, and then click **OK**. Make any desired changes to the attributes of the new connection description and then click **OK**. The new connection description will appear in the **Name** list in the **Network Variable Connection** dialog box.

Using Automatic Connection Description Selection

You can use OpenLNS CT to automatically select the connection description for network variable connections. OpenLNS CT will select the connection description that optimizes network performance, while providing reliable message delivery and minimizing network resource usage.

To support automatic connection description selection, the pre-defined connection descriptions include *smart connection descriptions* (SCDs) . When a smart connection description is assigned to a connection, OpenLNS CT automatically selects the appropriate SCD when a connection is created or changed. OpenLNS CT determines the appropriate SCD based on the number of target network variables, whether the target network variables are located on the same or different subnet, and the availability of network variable aliases.

The following flow chart demonstrates the process OpenLNS CT uses to select the appropriate SCD.

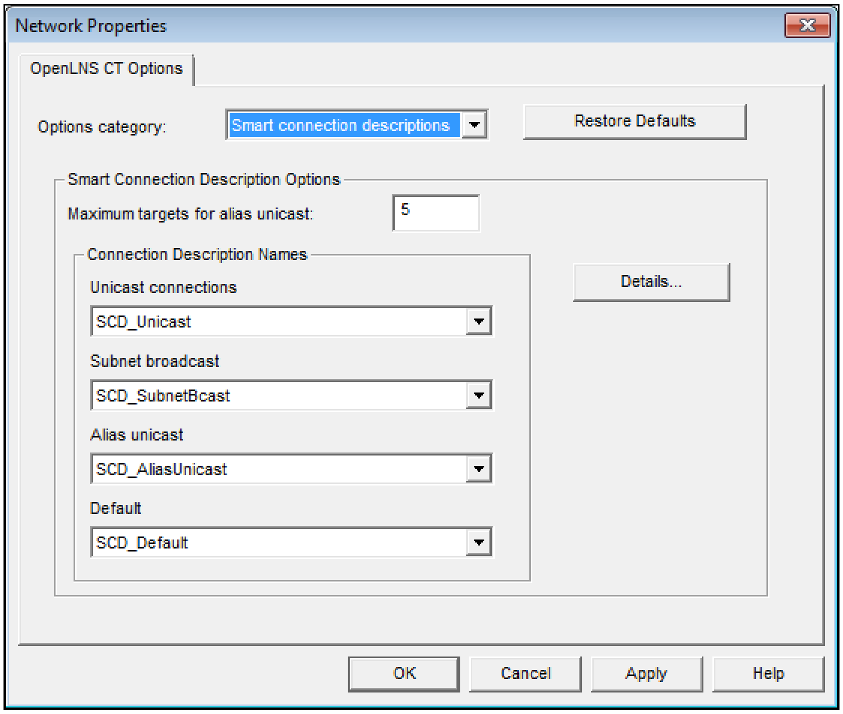


These automatic SCD selection rules do not apply to message tag connections, fan-in connections (where the hub is an input network variable), or connections in which you specify a non-SCD for the connection description.

To modify the automatic SCD selection rules, follow these steps:

1. In the OpenLNS CT drawing, click **Add-ins**, point to **OpenLNS CT**, and then select **OpenLNS CT Options**. The **Network Properties** dialog box opens.

2. Select **Smart Connection Descriptions** from the **Options Category** list.


Network Properties

OpenLNS CT Options

Options category: Smart connection descriptions

Smart Connection Description Options

Maximum targets for alias unicast: 5

Connection Description Names

Unicast connections: SCD_Unicast

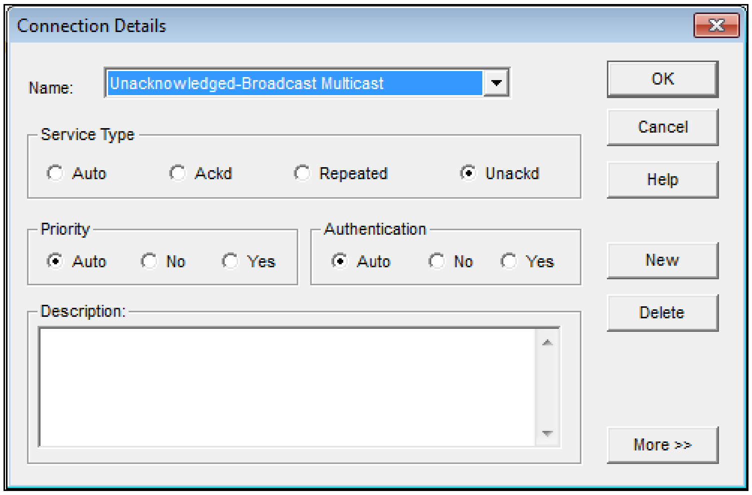
Subnet broadcast: SCD_SubnetBcast

Alias unicast: SCD_AliasUnicast

Default: SCD_Default

3. You can change or disable the connection description used for each condition in the automatic connection description selection process. If you disable the connection description for a given condition, OpenLNS CT will select the next defined connection description. You can also change the maximum number of target network variables used to select between the Alias Unicast and Default conditions. The default value of **5** provides a good balance between total transaction time and network resource usage.

4. Clicking **Details** opens the **Connection Details** dialog, in which you can modify pre-defined connection descriptions or create new ones.


Connection Details

Name: Unacknowledged-Broadcast Multicast

Service Type: ○ Auto ○ Ackd ○ Repeated ● Unackd

Priority: ● Auto ○ No ○ Yes

Authentication: ● Auto ○ No ○ Yes

Description:

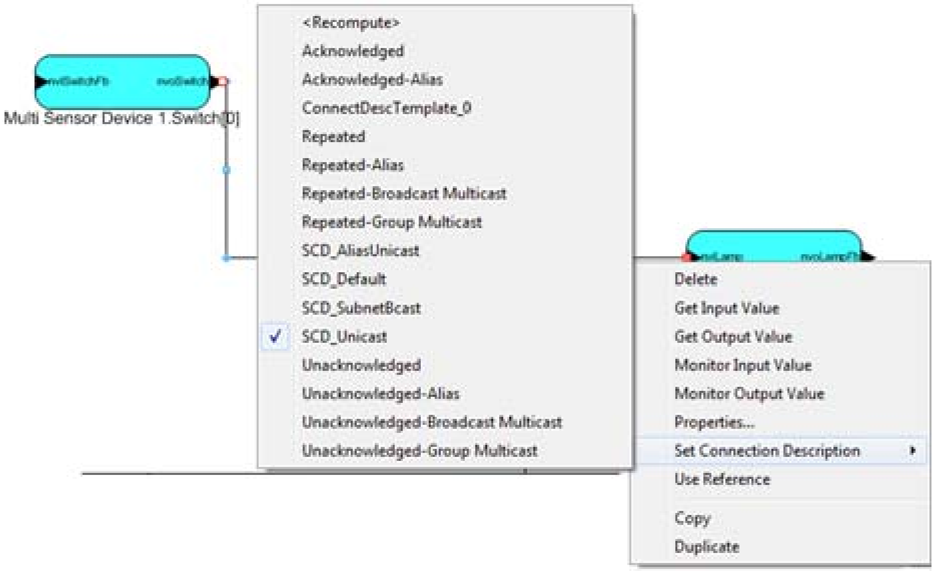
5. Click **OK** to return to the **Smart Connection Descriptions** options.

6. Click **OK**.

Changing the Connection Description for a Connection

You can quickly change a connection description used for a connection by manually selecting a new one. To manually select a new connection description, right click the connection, point to **Set**

Connection Description, and then click the desired connection description on the shortcut menu. To disable automatic connection description selection, select a connection description that does not start with "**SCD_**". To enable automatic connection description selection, select **<Recompute>**.



Hiding and Showing Connector SmartShapes

You can hide connector SmartShapes to simplify an OpenLNS CT drawing, or you can selectively show connector SmartShapes attached to specific functional blocks or supernodes to examine their connections.

You can hide or show all of the connector SmartShapes in a subsystem; hide/show connector SmartShapes attached to specific functional blocks or supernodes; or control the creation and hide/show connector SmartShapes attached to a functional block using the functional block properties **Options** tab.

Hiding and Showing All Connector SmartShapes in a Subsystem

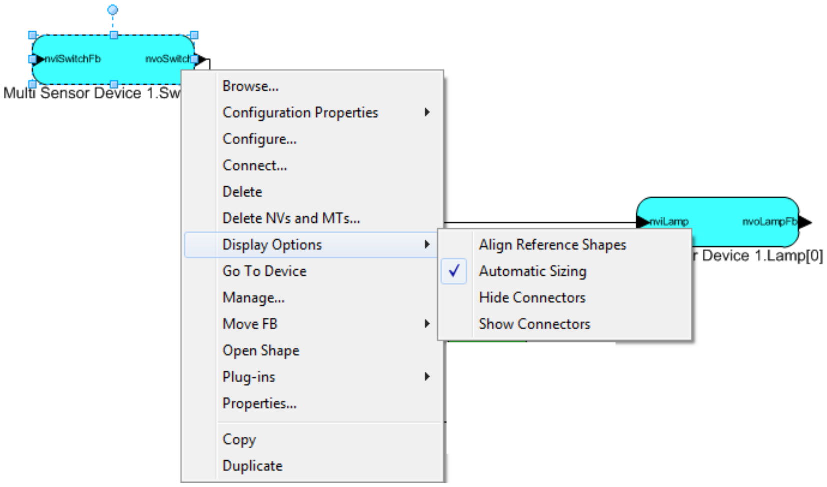
You can hide or show all of the connector SmartShapes in a subsystem. To do this, right-click an empty space in the drawing, point to **Display Options**, and then click **Hide All Connections** or **Show All Connections**.

Monitored connections cannot be hidden using the **Hide All Connections** command. See *Using Monitored Connections* in Chapter 6, *Monitoring and Controlling Networks*, for more information on enabling and disabling connection monitoring.

Hiding and Showing Selected Connector SmartShapes Attached to Functional Blocks or Supernodes

You can hide or show the connector SmartShapes attached to specific functional blocks or supernodes. To do this, right click the functional block or supernode, point to **Display Options**, and then click **Hide Connectors** or **Show Connectors**. The connector SmartShape will be hidden if **Hide Connectors** was selected on the functional block at the other end of the connection, and neither end of the connection is being monitored (monitored connections cannot be hidden using the **Hide Connectors** command).

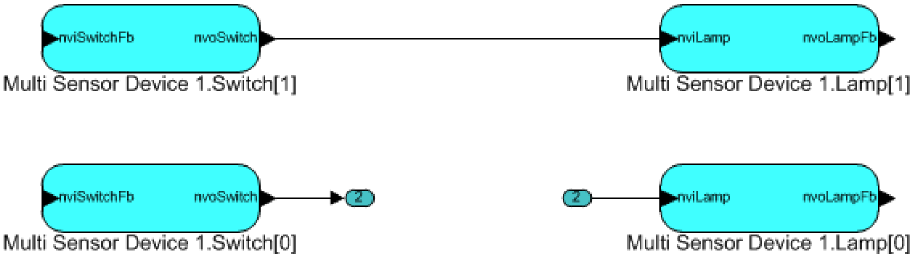
You can also hide or show connectors for multiple functional blocks or supernodes at one time. Drag a selection net around the functional blocks or supernodes with connectors you want to hide or show, or click one functional block, hold down CTRL, click all other functional blocks and supernodes, and then right-click one of the selected functional blocks; point to **Display Options**; and then click **Hide Connectors** or **Show Connectors**.



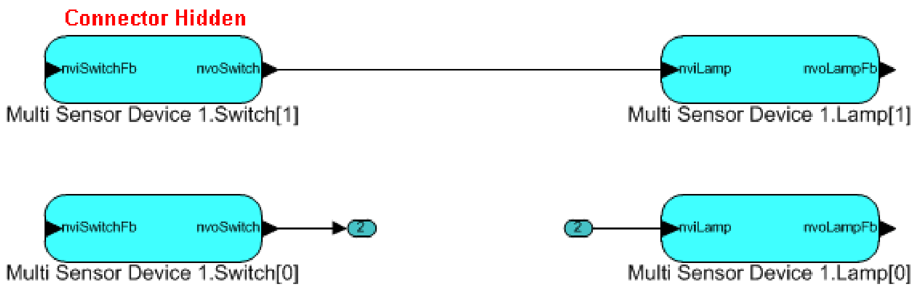
To hide a connector SmartShape directly attached to two functional block SmartShapes (not attached using a reference SmartShape), select **Hide Connectors** on both functional blocks. The connector SmartShape will be hidden at both ends. To show the connector SmartShape, you only need to select **Show Connectors** on one of the attached functional blocks.

Unlike connector SmartShapes attached directly to functional blocks, a connector SmartShape attached via reference can be hidden on one end while being shown on the other. To hide a connector SmartShape attached via a reference SmartShape, select **Hide Connectors** on one of the attached functional blocks. The connector SmartShape and its reference SmartShape will both be hidden on that end. Repeat this step on the other attached functional block to hide the corresponding SmartShapes on the other end of the connection.

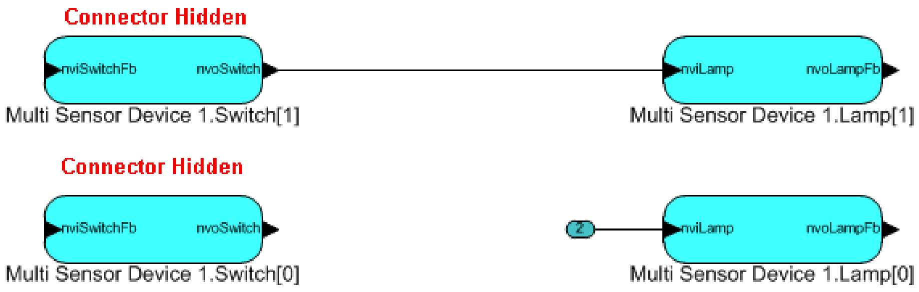
The following series of figures demonstrates how you can hide and show connections. This example uses two analog input functional blocks connected to an analog functional block. One connector SmartShape is attached directly to both functional blocks; the other is attached via a reference SmartShape. **Show Connectors** is set by default on all three functional blocks.


nviSwitchFb
nvoSwitch
Multi Sensor Device 1.Switch[1]
nviLamp
nvoLampFb
Multi Sensor Device 1.Lamp[1]
nviSwitchFb
nvoSwitch
2
Multi Sensor Device 1.Switch[0]
2
nviLamp
nvoLampFb
Multi Sensor Device 1.Lamp[0]

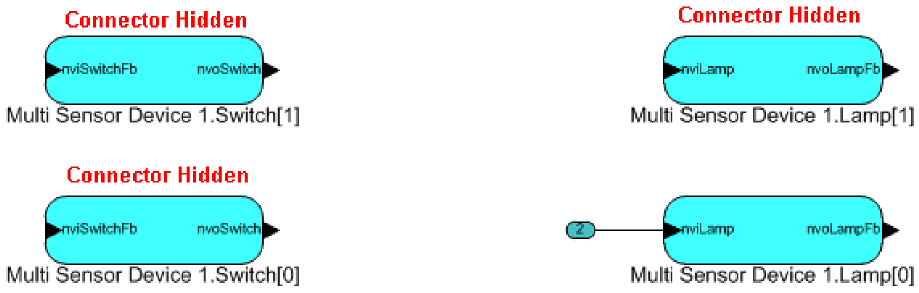
If you right click the **Switch[1]** functional block, point to **Display Options**, and then click **Hide Connectors**. There is no change because **Show Connectors** is still set on the **Lamp[1]**functional block.


Connector Hidden
nviSwitchFb
nvoSwitch
nviLamp
nvoLampFb
Multi Sensor Device 1.Switch[1]
Multi Sensor Device 1.Lamp[1]
nviSwitchFb
nvoSwitch
2
2
nviLamp
nvoLampFb
Multi Sensor Device 1.Switch[0]
Multi Sensor Device 1.Lamp[0]

If you right click the **Switch[0]** functional block, point to **Display Options**, and then click **Hide Connectors**, the connector and reference SmartShapes attached to this functional block are hidden. There is no change to the connector and reference SmartShapes attached to the **Lamp[0]** functional block because connector SmartShapes attached via reference are hidden/shown independently on each end.


Connector Hidden
nviSwitchFb
nvoSwitch
nviLamp
nvoLampFb
Multi Sensor Device 1.Switch[1]
Multi Sensor Device 1.Lamp[1]
Connector Hidden
nviSwitchFb
nvoSwitch
2
nviLamp
nvoLampFb
Multi Sensor Device 1.Switch[0]
Multi Sensor Device 1.Lamp[0]

If you right click the **Lamp[1]** functional block, point to **Display Options**, and then click **Hide Connectors**, the connector SmartShape between the **Switch[1]** and **Lamp[1]** functional blocks is now hidden because **Hide Connectors** is selected on both attached functional blocks.

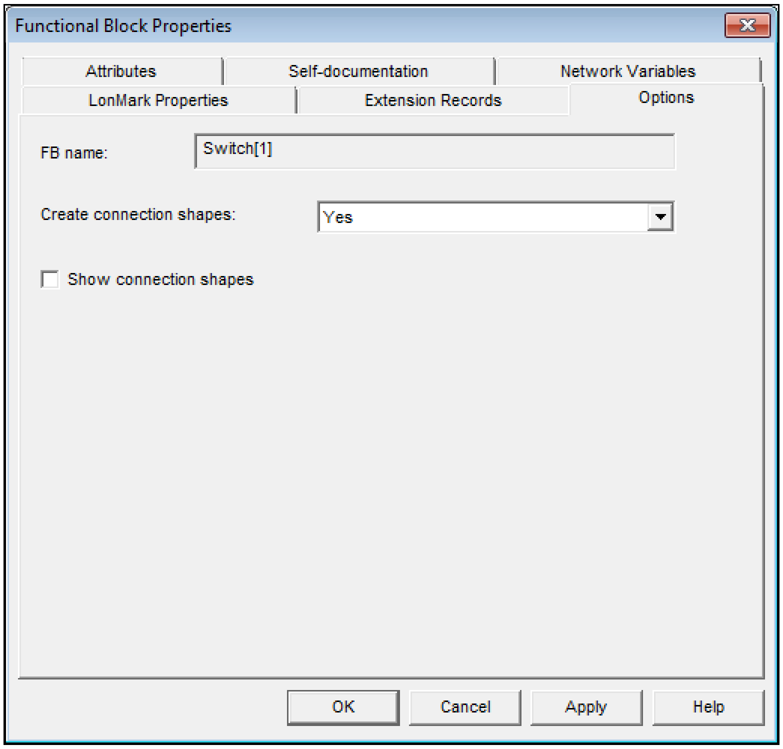


Hiding and Showing Connector SmartShapes Attached to a Functional Block

You can control the creation of connector SmartShapes and control whether they are hidden or displayed from the **Options** tab in the **Functional Block Properties** dialog. The settings in the dialog apply only to functional blocks (not to supernode). To use this dialog to control functional block connections, follow these steps:

1. Right-click a functional block, click **Properties** on the shortcut menu, and then click the **Options** tab.

   To select multiple functional blocks, click one functional block, hold down CTRL, click all other functional blocks with connectors you want to hide or show, right-click one of the selected functional blocks, click **Properties** on the shortcut menu, and then click the **Options** tab.



Functional Block Properties

FB name: Switch[1]

Create connection shapes: Yes

☐ Show connection shapes

2. Select the following options for creating and displaying connection SmartShapes for the selected functional block or functional blocks.

Create Connection Shapes Select whether SmartShapes are created and displayed in the OpenLNS CT drawing for connections with the selected functional block(s). You have three choices:

- **Yes**. Creates and displays SmartShapes in the OpenLNS CT drawing for connections with this functional block if **Yes** or **If Other FB has Connection Shapes** is set in the functional block on the other end of the connection. This is the default.

- **No**. Does not create or display SmartShapes in the OpenLNS CT drawing for connections with this functional block, regardless of the setting in the functional block on the other end of the connection. In addition, automatically clears and disables the **Show Connection Shapes** check box, which in turn hides all existing connections attached to the selected functional block(s). Connections involving this functional block will still be created in the network database, but they will not be represented by a connection SmartShape.

- **If Other FB has Connection Shapes**. Creates and displays SmartShapes in the OpenLNS CT drawing for connections with this functional block only if **Yes** is set in the functional block on the other end of the connection. A connection SmartShape is not created if **No** or **If Other FB has Connection Shapes** is set on in the functional block on the other end of the connection.

Show Connection Shapes

Select whether connection SmartShapes attached to the selected functional block(s) are displayed. This check box is selected by default, but it is cleared and unavailable if **No** is selected in the **Create Connection Shapes** list.

If this check box is selected, connections with the functional block(s) will be represented with a connection SmartShape provided that the following is true: **Yes** is selected in the **Create Connection Shapes** list for one of the functional blocks in the connection, and **Yes** or **If Other FB has Connection Shapes** is selected in the functional block on the other end of the connection. If these conditions are true, a connection SmartShape will be shown regardless if this option is cleared for the functional block on the other end of the connection.

If this check box is cleared, a connection SmartShape will be hidden if this check box is cleared in the functional block on the other end of the connection.

Viewing and Navigating OpenLNS CT Network Design

OpenLNS CT provides several methods for viewing and navigating your OpenLNS CT network design. You can navigate through the network from within a drawing page, and you can use the Network Explorer to navigate, view, and modify the subsystems, devices, and device templates in your network design. The following describes how to view and navigate your network using the drawing page, and how to navigate, view, and modify your network design with the Network Explorer.

Navigating the Network within the Drawing Page

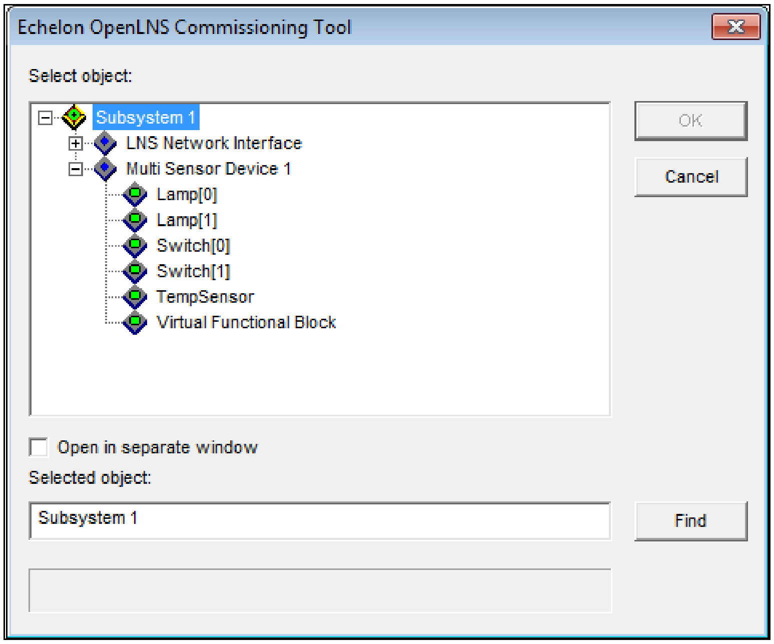
You can navigate through your network starting from any page of your network design. Each page in your network design represents a subsystem in the network. The initial page is called the *top-level (root) subsystem*. The top-level subsystem can have multiple subsystems nested below it, and those nested subsystems can have multiple subsystems nested below them, and so on (the subsystem directly above another is referred to as the *parent* subsystem). The result is that your drawing file represents a network hierarchy consisting of the top-level subsystem and potentially multiple levels of subsystems nested below it.

From the drawing page, you can open any subsystem in the network and go to any object in the subsystem, or you can move up one level in the network hierarchy and open the parent subsystem

Opening any Subsystem

You can open any subsystem of a network from any page within the network design. To open a subsystem in the network from the drawing page, follow these steps:

1. Right-click an empty spot in the drawing page and then click **Go to Subsystem** on the shortcut menu. The Select Object window opens.



□ Open in separate window

Selected object: Subsystem 1

2. The window displays a hierarchal view of the subsystems in your network, with the nested subsystems and devices in the top-level subsystem and the current subsystem shown. You can expand an object to view any others further down the network hierarchy such as nested subsystems, devices, and functional blocks. You can also right-click an object and use the Network Navigator to expand an object. Using the Network Navigator and expanding the top-level subsystem will cause the complete hierarchy of all subsystems in the network to be displayed. See *Using Network Navigators* for more information on using this tool.

3. Select a subsystem or an object in the subsystem.

4. To open the selected subsystem in a new window, select the **Open in Separate Window** check box.

5. Click **OK**. The subsystem or the subsystem containing the object you selected will open.

Opening a Parent Subsystem

You can move up one level in the network hierarchy and open the parent subsystem using either of the following two methods:

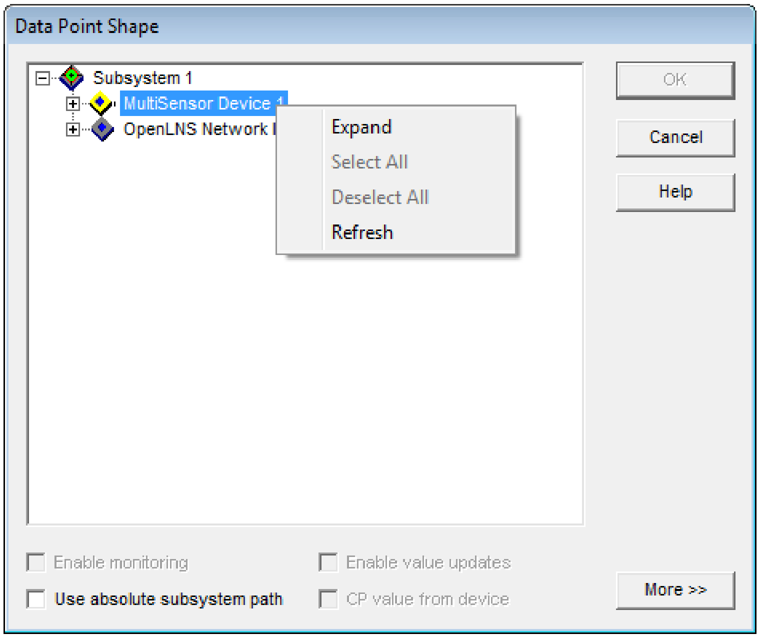
- Double-click the title block. The title block is located at the top-center of the drawing page and contains the name of the subsystem.

- Right-click an empty spot in a subsystem and select **Go to Parent Subsystem** from the shortcut menu.

  With either method, the parent subsystem opens in the same window if it is located in the same drawing. If the parent subsystem is located in a different drawing, it opens in a new window.

Using Network Navigators

You can use a network navigator to expand, select, and deselect items when viewing or configuring an OpenLNS CT network. The network navigator displays a hierarchal view of the subsystems in your network. It is available in several scenarios including the following: selecting a subsystem to open, adding functional blocks or supernodes to a connection, selecting network variables to assign to a subsystem, and choosing the subsystem into which to replicate a connection.



☐ Enable monitoring

☐ Enable value updates

☐ Use absolute subsystem path

☐ CP value from device

To use the network navigator, right-click an object that can be expanded and then click one of the following commands in the shortcut menu:

Expand — Displays all objects below the selected item. Selecting expand on the top-level subsystem will display all objects in the network hierarchy.

Select All — Expands the item and selects the check boxes of all the objects below it. Click **Select All** on the top-level subsystem to select the check boxes of all the objects in the network hierarchy. This option is only available in scenarios in which multiple objects can be selected.

Deselect All — Expands the item and clears the check boxes of all the objects below it. Click **Deselect All** on the top-level subsystem to clear the check boxes of all the objects in the network hierarchy. This option is only available in scenarios in which multiple objects can be selected.

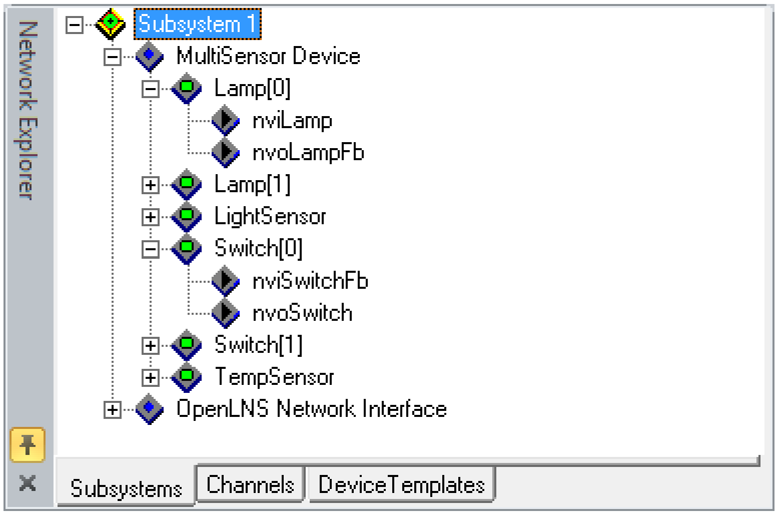
Refresh — Collapses and/or clears the check boxes of all the objects below the selected item.

Navigating and Managing a Network with the Network Explorer

You can navigate and manage your network using the Network Explorer. The Network Explorer is an auto-hide window that displays a hierarchical view of your OpenLNS CT network and supports all of the commands available within the drawing pages. You can use the Network Explorer as a navigational tool or a network management tool. As a navigational tool, the Network Explorer simplifies and accelerates the task of navigating through a large OpenLNS CT network. It does this by displaying the network as a single tree with the devices, routers, functional blocks, and network variables in the network listed in hierarchal order below their respective subsystems, channels, and device templates. As a network management tool, the Network Explorer lets you configure and modify the objects as you would in the drawing page. You just right-click an object in the Network Explorer and click commands on the object's shortcut menu.

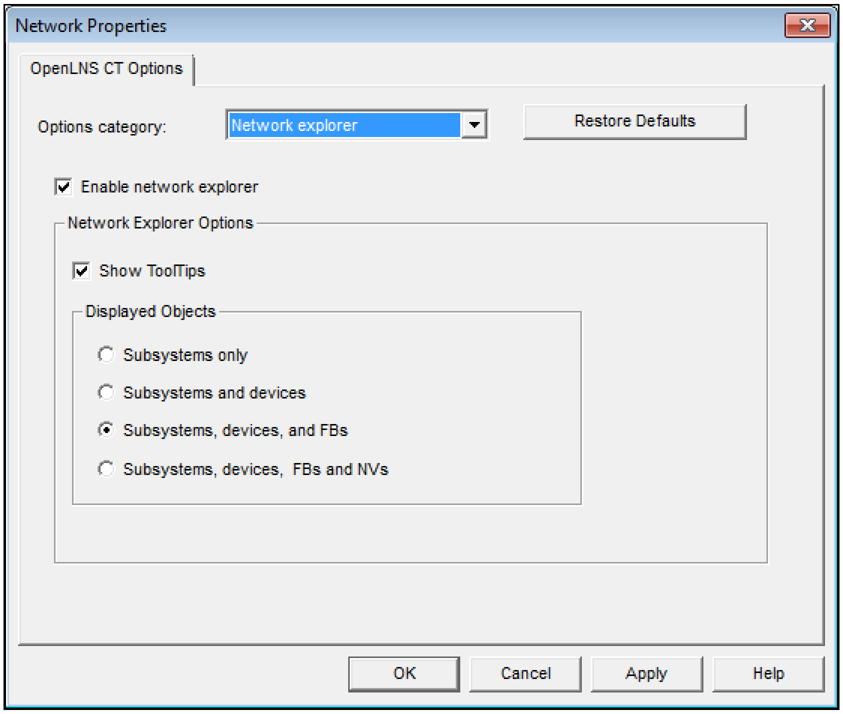
To display or hide the Network Explorer, open the **View** menu and then select **Network Explorer**. The objects displayed in the Network Explorer depend on the settings in the **Network Properties:**

Network Explorer options category. By default subsystems, devices, and functional blocks are displayed.



To expand or reduce the scope of the objects displayed in the Network Explorer, follow these steps:

1. In the OpenLNS CT drawing, click **Add-ins**, point to **OpenLNS CT**, and then select **OpenLNS CT Options**. The **Network Properties** dialog box opens.
2. Select **Network Explorer** from the **Options Category** list.
3. Select the desired scope under the **Displayed Objects** box. See *Network Explorer Options* in Appendix A for more information.
4. Click **OK**.



Options category: Network explorer

☑ Enable network explorer

Network Explorer Options

☑ Show ToolTips

Displayed Objects

○ Subsystems only
○ Subsystems and devices
● Subsystems, devices, and FBs
○ Subsystems, devices, FBs and NVs

The Network Explorer provides separate subsystem, channel, and device template views, each on its own tab. To change views, click the appropriate tab.

- In the subsystem view, the top-level subsystem and all the subsystems it contains are displayed, and all the other objects in your OpenLNS CT drawing are listed below their respective subsystems in a hierarchal structure (in the order of nested subsystems, devices, functional blocks, and then network variables).
- In the channel view, the channels in your OpenLNS CT drawing are listed alphabetically, and the objects in the OpenLNS CT drawing are listed in a hierarchal structure under the channels on which they are connected (in the order of devices, functional blocks, and then network variables).
- The device template view is similar to the channel view except that the device templates are listed alphabetically and the devices (and their functional blocks and network variables) are listed under the device templates they use. From the device template view, you access the **Device Template** menu and the **Device Template Properties** dialog box. See *Device Template Options* in this chapter for more information on updating and setting validation options for device templates.

Navigating a Network with the Network Explorer

You can navigate the subsystems, devices, device templates, and other objects in your network with the Network Explorer. Expand an object to display the objects below it in the network hierarchy; collapse an object to hide the objects beneath it. Double-click an object to open the drawing page containing the object. The object you double-clicked in the Network Explorer will be selected when the drawing page opens (if you double-clicked a subsystem, no object will be selected when the drawing page opens).

For example, in the subsystem view, the Network Explorer initially displays the nested subsystems and devices in the top-level and current subsystems. You can expand one of the devices to reveal its functional blocks and then expand one of the functional blocks to reveal its network variables. When you double-click a functional block of a device in the top-level subsystem, the drawing page for the top-level subsystem will open and the functional block will be selected.

Managing a Network with the Network Explorer

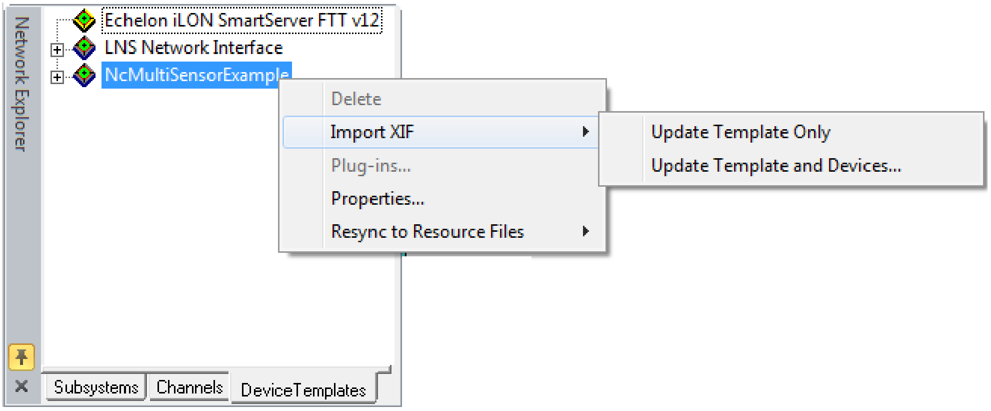
To use the Network Explorer as a network management tool, right-click a subsystem, device, router, functional block, or network variable SmartShape and select a command from the shortcut menu. The shortcut menu that opens is the same shortcut menu you can access in the drawing page. See the sections on subsystems, devices, routers, functional blocks, and network variables in this document for a description of the commands.

Managing and Editing Device Templates with the Network Explorer

You can edit, update, and delete a device template with the Network Explorer. A *device template* contains all the attributes of a given device type, such as its functional blocks, network variables, and configuration properties. You can create a device template by importing a device interface (XIF) file supplied by the device manufacturer, or by uploading the interface definition from the physical device. A device template is identified by its name and its program ID. Both must be unique within a network—you cannot have two device templates with the same name or the same program ID in a single network.

Managing Device Templates

You can manage device templates from the **Device Templates** tab. To edit, update, or delete a device template, click the **Device Templates** tab and then right-click the device template. The Device Template shortcut menu opens.



The Device Template shortcut menu contains the following options:

Delete — Deletes the selected device template. This option is unavailable if there are any devices currently using the device template, or if there is a device that has been discovered by the OpenLNS Server but does not yet appear in the OpenLNS CT drawing.

Import XIF — Updates the device template (and optionally devices) with the data from the XIF that you select.

Plug-ins — Displays the list of OpenLNS plug-ins registered for the selected device template. This option is unavailable if the device does not have any defined plug-ins.

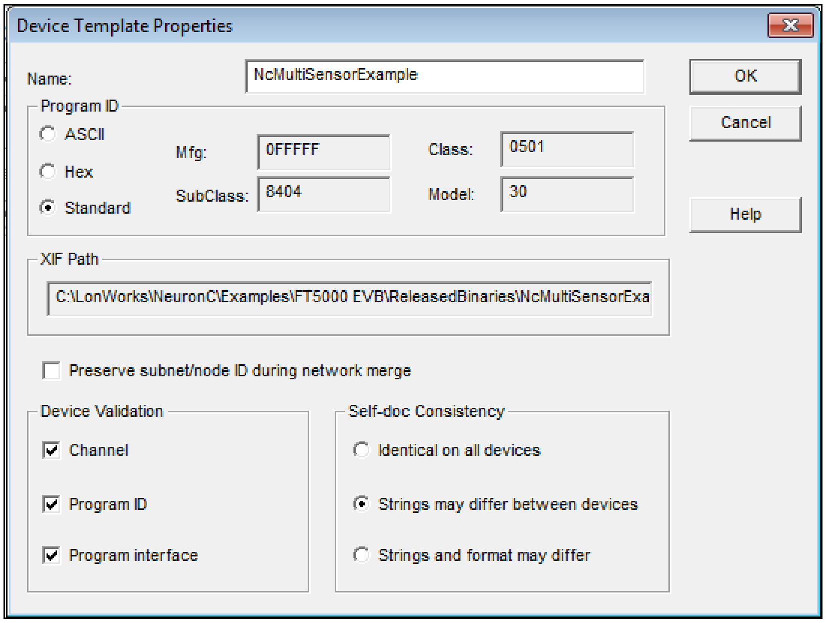
If you remove a template with an associated plug-in and you later re-import the template, you must re-register the plug-in.

Properties — Opens the **Device Template Properties** dialog box. See *Device Template Properties* for more information on the options in this dialog box.

Resync to Resource Files — Updates the device template (and optionally devices) with the current data from the device resource files.

Editing Device Template Properties

You can edit a device template. To edit a device template, right-click the template on the **Device Templates** tab in the Network Explorer, and then click **Properties** on the shortcut menu. The **Device Template Properties** dialog box opens, showing the attributes of the selected device template.



Device Template Properties

Name: NcMultiSensorExample

Program ID

○ ASCII

○ Hex

● Standard

Mfg: 0FFFFF

Class: 0501

SubClass: 8404

Model: 30

XIF Path

C:\LonWorks\NeuronC\Examples\FT5000 EVB\ReleasedBinaries\NcMultiSensorExa

□ Preserve subnet/node ID during network merge

Device Validation

☑ Channel

☑ Program ID

☑ Program interface

Self-doc Consistency

○ Identical on all devices

● Strings may differ between devices

○ Strings and format may differ

The **Device Template Properties** dialog box contains the following options:

Name

Specifies the name of the device template. You can change the device template name to any other valid, unique name.

Program ID

Displays the program ID of the device template. You can select one of the following options for displaying the program ID (this selection does not change the program ID):

- **ASCII**. Displays the program ID as an ASCII string. Non-printable characters are displayed as "/x*HH*", where *HH* is the hex value of the unprintable byte.
- **HEX**. Displays the program ID as hexadecimal numbers
- **Standard**. Displays the program ID as standard program ID fields. If the program ID is not a standard program ID (the leading nibble value is not 8 or 9), this display option is unavailable.

XIF Path

Displays the name of the XIF file used to create the device template (if known). The path specified is relative to the LONWORKS Import folder on the OpenLNS server computer.

Preserve Subnet/Node during Network Merge

Enables the device to keep its subnet/node ID during a network merge. This option is disabled by default.

Typically, a network merge changes a device's ID subnet/node ID, which may require reprogramming of the subnet/node IDs of the devices in the networks being merged.

If there are devices in the two networks being merged that have the same subnet ID and node ID, you will need to manually assign a unique subnet/node for one of the devices. See *Merging OpenLNS CT Networks* in Chapter 7 for more information on network merges.

Device Validation

Selects which validation steps are performed by the OpenLNS Server when you commission devices that use the selected device template.

When you commission a device, the OpenLNS Server by default will validate that the physical device is on the correct assigned **Channel** and that the device template has the same **Program ID** and **Program Interface** (XIF) defined in the network database.

Clearing these check boxes may reduce the time required to commission a device. This may be useful when you are commissioning a large number of devices on a slow channel such as power line, or when commissioning time is very expensive such as a remote OpenLNS Server communicating over a dial-up connection.

If you clear the check boxes, make sure that all devices using the selected device template contain the correct program information and are installed on the correct channel before commissioning them. Doing so will help prevent the devices from being improperly configured or having communication problems.

Self-doc Consistency Selects the required level of consistency for the self-documentation data (strings and format) of the devices using the selected device template.

Choose from three levels of consistency (listed from least to most flexible): **Identical on All Devices**, **Strings may Differ Between Devices**, or **Strings and Format may Differ**.

You can use the **Strings may Differ Between Devices** or **Strings and Format may Differ** options to install devices from a device manufacturer that have the same program ID but slightly different self-documentation strings. This typically happens when a device manufacturer makes changes to the descriptive part of a device's self-documentation data.

You can use the **Strings and Format may Differ** option to install devices from a device manufacturer that have the same program ID but use a different format for the self-documentation data. For example, if the manufacturer changed from the version 1 to version 2 format. **Note**: This option will not work for some devices. See the OpenLNS CT ReadMe file for more information.

Working with OpenLNS CT Layers

You can use OpenLNS CT layers to hide and show parts of your drawing. OpenLNS CT layers are separate views for the physical, logical, and data components of your OpenLNS CT network. The physical layer includes the devices, routers, channels; the logical layer contains the functional blocks, network variables, and unmonitored connections; and the data layer consists of monitored connections. Subsystems and supernodes belong to both the physical and logical layers. To hide subsystems and supernodes, hide both the physical and logical layers. Hiding only the logical layer hides the supernode network variables.

You can show or hide an OpenLNS CT layer using either of the following two methods:

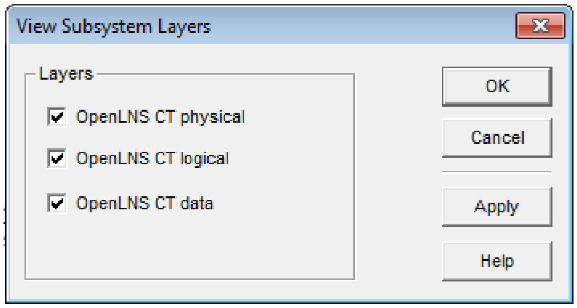
- Click the icon of the corresponding layer in OpenLNS CTbar.

Physical Layer Shows/hides all supernodes, subsystems, devices, routers, and channels.

Logical Layer Shows/hides all supernodes, subsystems, functional blocks, message tags, network variables, and unmonitored connections.

	Data Layer	Shows/hides all monitored connections.

- Right-click an empty space in your OpenLNS CT drawing, point to **Display Options** on the shortcut menu, and select **View Subsystems Layers** from the shortcut menu. The **View Subsystem Layers** dialog box opens. Select or clear the OpenLNS CT Layers you want displayed or hidden. You can click **Apply** if you want to see how your OpenLNS CT drawing opens with the current settings. When you are done, click **OK**.


View Subsystem Layers

Layers

☑ OpenLNS CT physical

☑ OpenLNS CT logical

☑ OpenLNS CT data

Working with OpenLNS CT SmartShapes

You can view, edit, move, copy, and delete OpenLNS CT SmartShapes. An OpenLNS CT *SmartShape* is a SmartShape within an OpenLNS CT drawing that represents a physical component in a LONWORKS network such as device, channel, network variable, or connection. This section describes how to work with OpenLNS CT SmartShapes.

Viewing and Setting OpenLNS CT SmartShape Properties

You can view and set the properties of an OpenLNS CT SmartShape. If you modify the properties with OpenLNS CT attached to the network and OnNet, the changes will be sent to the physical devices immediately. Otherwise, the changes will be stored in the OpenLNS database and then propagated to the physical devices when OpenLNS CT is attached and placed OnNet.

To view the properties of an OpenLNS CT SmartShape, right-click the SmartShape and select **Properties** from the shortcut menu. A **Properties** dialog box opens. For a description of the properties that you can view and modify, click **Help** in the dialog box.

Moving an OpenLNS CT SmartShape

You can move an OpenLNS CT SmartShape within an OpenLNS CT drawing. You can reposition it within the current subsystem; move it into a different subsystem; or change the channel in on which it is located, which can be in the same or in a different subsystem. You cannot move an OpenLNS CT SmartShape using Visio’s cut and paste commands.

Repositioning an OpenLNS CT SmartShape

You can re-position any OpenLNS CT SmartShape within a subsystem by dragging it to a different location on the same drawing page.

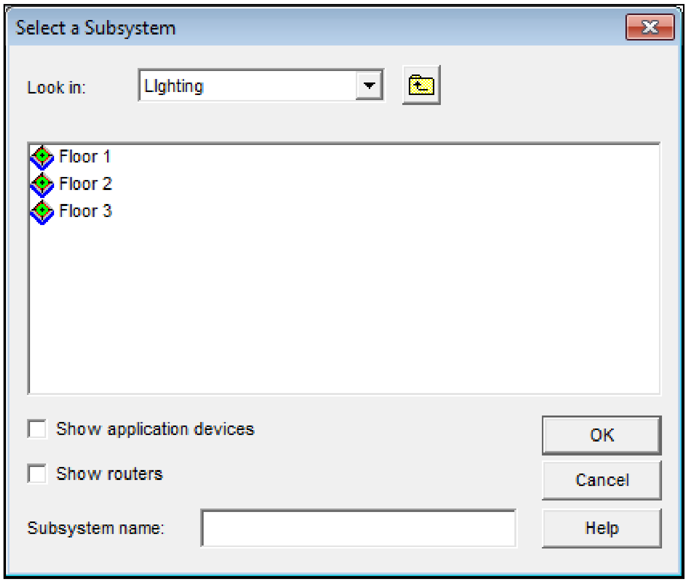
Changing the Subsystem for an OpenLNS CT SmartShape

You can move a device, router, functional block, and subsystem SmartShape into another subsystem. To do this, follow these steps:

1. Right-click the SmartShape. You can select multiple SmartShapes by holding down CTRL and clicking the SmartShapes, or clicking an empty space in the drawing page and dragging a selection net around the SmartShapes.

2. Point to **Move Device** (for device, router, and subsystem SmartShapes) or **Move FB** (for functional block SmartShapes) on the shortcut menu.

3. Click **Change Subsystem** (for device, router, and functional block SmartShapes) or **Move** (for subsystem SmartShapes). The **Select a Subsystem** dialog box opens. This dialog lists any subsystems nested within the current subsystem.



Select a Subsystem

Look in: Lighting

Floor 1
Floor 2
Floor 3

□ Show application devices

□ Show routers

Subsystem name:

4. Click the **Up** button () to move up one level in the network hierarchy; double-click a subsystem to move down one level. Select the new subsystem.

5. Click **OK**.

6. If you move a functional block with any connector SmartShapes, you will be prompted to have OpenLNS CT convert the connector SmartShapes to reference connection SmartShapes. Click **OK**.

Changing the Channel for an OpenLNS CT SmartShape

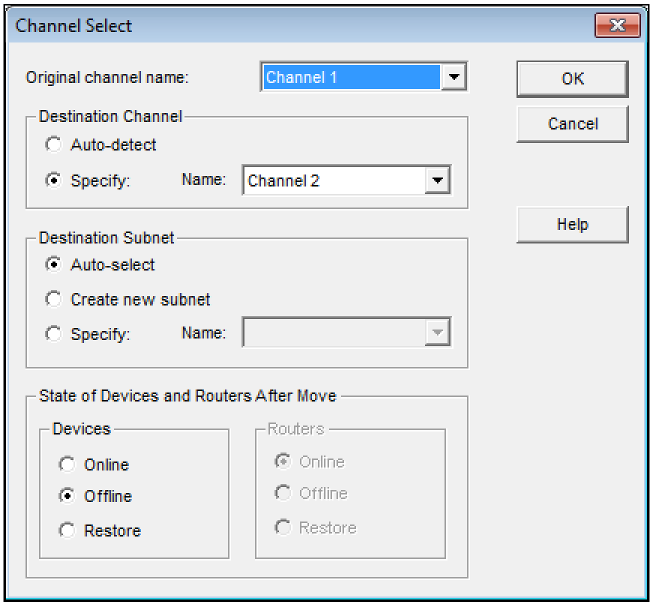
You can move a device, router, or OpenLNS computer onto a different channel. OpenLNS computers include OpenLNS CT computers, the OpenLNS Server, and other OpenLNS clients.

Moving a Device or Router to a Different Channel

To change the channel on which devices and routers are located, follow these steps:

1. Right-click the device or router SmartShape, or a subsystem containing the device or router, point to **Move Device**, and then click **Change Channel** on the shortcut menu.

   You can move multiple device and router SmartShapes attached to the same channel by holding down CTRL and clicking the SmartShapes, or clicking an empty space in the drawing page and dragging a selection net around the SmartShapes. The **Channel Select** dialog box opens.


Channel Select

Original channel name: Channel 1

Destination Channel
○ Auto-detect
● Specify: Name: Channel 2

Destination Subnet
● Auto-select
○ Create new subnet
○ Specify: Name:

State of Devices and Routers After Move

Devices
○ Online
● Offline
○ Restore

Routers
● Online
○ Offline
○ Restore

2. Set the following options:

Original Channel Name

Specifies the current channel on which the devices or routers are attached.

If you are moving devices or routers from a subsystem SmartShape that has devices or router SmartShapes connected on two or more channels, select the channel with the devices or routers you want to move. Only those devices or routers located on the selected channel will be moved.

Select the **Exclude Nested Subsystems from Subsystem Operations** check box in the **Subsystem Properties** to limit the scope of the move to the current subsystem. To select or clear this check box, right-click a Subsystem SmartShape and select **Properties** from the shortcut menu, or right-click an empty space in the drawing page and select **Subsystem Properties** from the shortcut menu.

Destination Channel

Specifies the new channel.

- Select **Auto Detect** to have OpenLNS CT automatically discover the new channel. The OpenLNS CT computer must be attached to the network to use this option. You cannot move routers using this option.
- Select **Specify** to manually select the channel from the **Name** list.

Destination Subnet

Specifies the new subnet (routers cannot have their subnet changed).

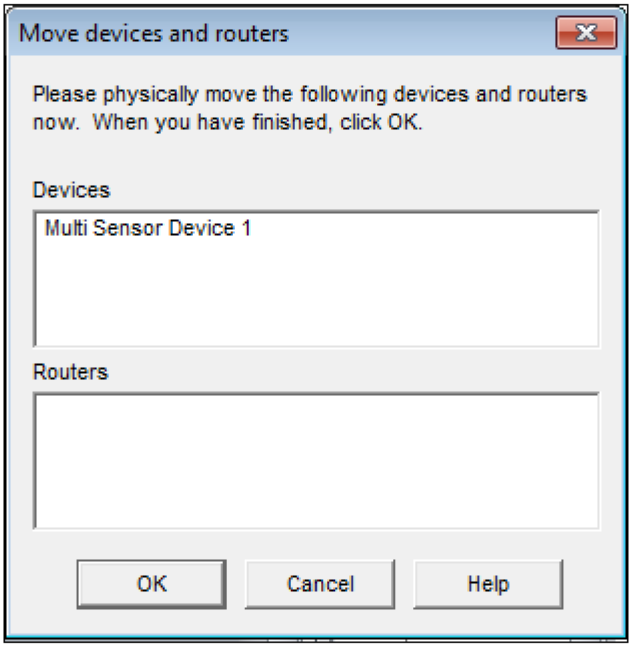
- Select **Auto-Select** to have OpenLNS CT choose an appropriate subnet.
- Select **Create New Subnet** to create a new subnet and place the device on it.
- Select **Specify** to manually choose an existing subnet from the **Name** list.

State After Move

Specifies the desired state of the devices or routers after they have been moved.

- Select **Online** to put the devices or routers in the online state. This is the default for routers. The behavior in the online state depends on the device. A Neuron-hosted device, for example, will run its application after it has been moved.
- Select **Offline** to put the devices or routers in the offline state. This is the default for devices. The behavior in the offline state depends on the device. A Neuron-hosted device, for example, will not run its application after it has been moved. Devices in the offline state still respond to Online, Wink, and Test commands. The devices also receive network variable updates; however, their applications will not process these values, and the devices will not send network variable updates.
- Select **Restore** to return the devices or routers to their states before they were moved.

3. Click **OK**. If OpenLNS CT is OnNet and any of the devices or routers have been commissioned, the **Move Devices and Routers** dialog box opens.


Move devices and routers

Please physically move the following devices and routers now. When you have finished, click OK.

Devices

Multi Sensor Device 1

Routers

4. Physically move the device or router onto the channel you selected.
5. Click **OK**. OpenLNS CT completes the channel change.
6. If the channel change fails, a **Re-commission Prompt** will open in which you can recommission the devices that you tried to move. Click **Yes** to recommission the devices. Re-commissioned routers will be put in the online state; devices will be placed offline. Click **No** to leave the devices deconfigured.

Moving an OpenLNS Computer to a Different Channel

You can move an OpenLNS computer (OpenLNS CT computer, OpenLNS Server, or other OpenLNS client) to a different channel following these steps:

1. Right-click the OpenLNS Network Interface SmartShape for the OpenLNS CT computer, or the NSD SmartShape for another OpenLNS computer, and then click **Change Channel** from the shortcut menu.

2. Change the channel on which the OpenLNS computer is attached following step 2 in *Moving a Device or Router to a Different Channel*.
3. If you are moving an OpenLNS CT computer and you will be using a different network interface, click **Add-ins**, point to **LonWorks Network**, and then click **Network Properties**. Click the **Network Interface** tab, select the name of the new network interface, and then click **OK**.
4. Click **OK**.

Copying an OpenLNS CT SmartShape

You can copy an OpenLNS CT SmartShape in your OpenLNS CT drawing to create a new SmartShape with the same properties of the original. For example, if you copy a device SmartShape, OpenLNS CT creates a new device SmartShape with the same device template, configuration properties, and device properties as the SmartShape from which it was copied. *Extension records* (data records created by OpenLNS tools such as OpenLNS CT to store tool-specific information) that indicate they can be copied will also be copied.

You can copy an OpenLNS CT SmartShape in two ways:

- Right-click the OpenLNS CT SmartShape, select **Copy** from the shortcut menu, right-click an empty space in your OpenLNS CT drawing, and then click **Paste**.
- Select the OpenLNS CT SmartShape, hold down CTRL, and then drag the SmartShape to a new location within the current drawing page.

You can copy multiple SmartShapes at the same time by holding down CTRL and clicking the SmartShapes you want to copy, or clicking an empty space in the drawing page and dragging a selection net around the SmartShapes.

If you copy a device SmartShape, the copy represents a different physical device than the original. You must assign a new Neuron ID to the new device SmartShape when it is commissioned. You cannot use this method to copy a network variable or message tag SmartShape. See *Creating a Network Variable SmartShape* for instructions on how to copy a network variable SmartShape. You can only copy a connection reference SmartShape in conjunction with the attached Connector SmartShape.

Tip: To copy an OpenLNS CT SmartShape from one to network to another, create a custom master SmartShape in the current OpenLNS CT drawing and then drag the master SmartShape to the destination drawing. See Chapter 12, *Creating and Using a Custom OpenLNS CT Master SmartShape, for more details on creating master SmartShapes.*

Deleting an OpenLNS CT SmartShape

You can delete an OpenLNS CT SmartShape. To do this, either right-click the SmartShape and then click **Delete** on the shortcut menu, or click the SmartShape and then press DELETE. If you selected the **Confirm Deletes** check box in the **Network Properties: General** options category (see Appendix A), click **Yes** to confirm the change.

If you remove a SmartShape that requires deleting other SmartShapes associated with it (a device with one or more functional blocks assigned to it, for example), a message opens warning you that the associated SmartShapes will also be deleted. Click **Yes** to delete the SmartShape and the additional SmartShapes; click **No** to cancel the operation.

When you delete an OpenLNS CT SmartShape, the physical devices on the network may be affected. In this case, OpenLNS CT and OpenLNS Server change the network configuration of the devices when OpenLNS CT is OnNet and the OpenLNS server can communicate with the devices. The following table summarizes how physical devices are affected when you delete various OpenLNS CT SmartShapes.

Deleted OpenLNS CT Shape	**Affect on Physical Device**
Commissioned Application Device	Application device is decommissioned and no longer has a network address.
Commissioned Router	Router is decommissioned and no longer has a network address.
OpenLNS Network Interface or NSD	OpenLNS CT attempts to decommission the NSD. If the NSD is using a Layer 2 interface, it cannot be decommissioned. Disconnects any remote full clients using the NSD.
Functional block (static)	No change.
Dynamic functional block	No change; however, a device plug-in may remove the functional block from the application device.
Network variable (static)	No change.
Dynamic network variable	Dynamic network variable is removed from application device.
Connection	Connection information is deleted from application device.
Channel	No change to devices attached to the channel. You cannot delete a channel SmartShape that has one or more devices attached to it on the same page.

Customizing the User Interface

You can customize the OpenLNS CT user interface. You can change the title block that opens on each page of an OpenLNS CT drawing, and the commands that appear on the OpenLNS CT menus.

Editing the Title Block

You can change the title block that opens on each page of an OpenLNS CT drawing. The default OpenLNS CT templates create a title block at the bottom right corner of each page. The title block provides the following information about the drawing: drawing title, page number, name of the drawing's creator, date the drawing was created, full file name of the drawing, and a brief description of the drawing or network.

To edit the title block, open the **Edit** menu, point to **Go To,** and select **Background**. Select the part of the title block you want to edit (no cursor opens) and then enter the information you want displayed. To return to the main drawing, open to the **Edit** menu, point to **Go To**, and select **Foreground**. The changes you made to the title block are displayed.

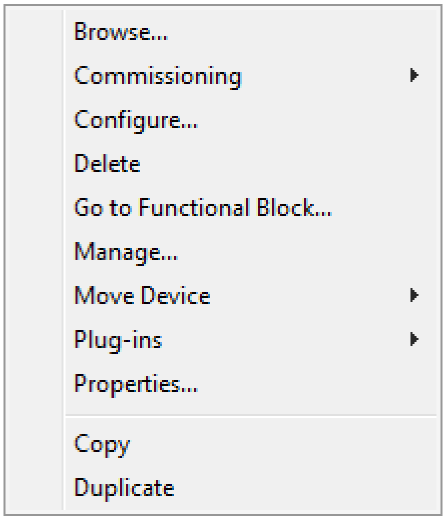
Tip: You can use the same method to add your company logo to your OpenLNS CT drawing.

Using OpenLNS CT SmartShape Menus

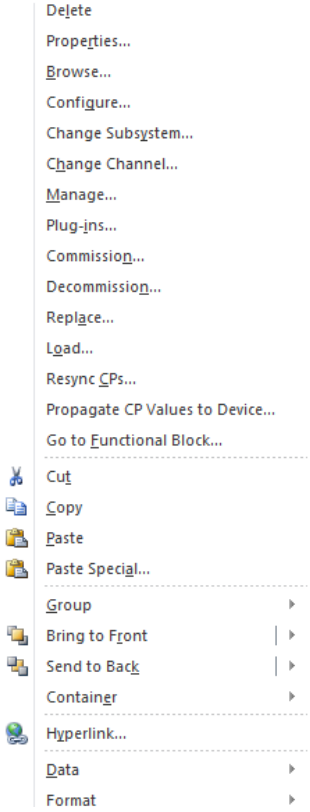
You can change the appearance of the OpenLNS CT shortcut menus. You can select either a hierarchical menu that removes Visio commands not typically used with OpenLNS CT SmartShapes, or you can select a flat menu that includes all Visio SmartShape commands. The hierarchical menus provide a compact top-level shortcut menu for each OpenLNS CT SmartShape with hierarchical sub-menus. For example, the device SmartShape menu has sub-menus for commissioning, moving, managing, and plug-ins, and the commissioning sub-menu contains the commands for commissioning, decommissioning, loading, and replacing devices; propagating configuration property values; and resynchronizing configuration properties. The hierarchal OpenLNS CT shortcut menus are set by default. To enable or disable hierarchal OpenLNS CT shortcut menus, click **Add-ins**, point to **OpenLNS CT**, and then select or clear the **Use OpenLNS CT Shape Menus** option.

The following figures illustrate the difference of a device SmartShape menu with **OpenLNS CT Shape Menus** selected and cleared.

Hierarchal
Device Shortcut Menu



Flat
Device Shortcut Menu



Using AutoCAD Drawings

You can import AutoCAD drawings into your OpenLNS CT drawing and export your OpenLNS CT drawing as an AutoCAD drawing.

Importing an AutoCAD Drawing

You can import an AutoCAD drawing. You can do this to use a drawing created by a mechanical system designer as a background for your OpenLNS CT drawing. This allows you to show the physical placement of devices within the mechanical system.

To import an AutoCAD drawing into the background of an OpenLNS CT drawing, follow these steps:

1. Open the **Insert** menu and select **CAD Drawing**.
2. Select the AutoCAD drawing (.**dwg** or .**dxf** extension).
3. In the **AutoCAD Drawing Layer Status** dialog, specify which layers to convert from the AutoCAD file. Because OpenLNS CT only supports importing an AutoCAD drawing as a background, you can choose to convert layers as Displayed Layers only. Layer 0 indicates unlayered data.
4. Specify the drawing page size and scale for the AutoCAD drawing in the Page Size/Scale dialog. These settings do not affect the size of OpenLNS CT SmartShapes in the drawing.
5. Open the **Edit** menu and select **Go To Foreground**.

6. The AutoCAD drawing opens in the OpenLNS CT drawing as a background.

Exporting an AutoCAD Drawing

You can export an OpenLNS CT drawing as an AutoCAD drawing. This lets you create an as-built report in the same format as your other system blueprints. An OpenLNS CT drawing saved in AutoCAD drawing format can be marked up by AutoCAD tools, but not manipulated as an AutoCAD object.

To export an OpenLNS CT drawing as an AutoCAD drawing, follow these steps:

1. Save the OpenLNS CT drawing as an AutoCAD drawing (**.dwg** format).
2. In the **Visio Drawing Layer Status** dialog, select the layers to export. Click **Options** to select a different AutoCAD output file version or OpenLNS CT drawing mapping scheme.
3. OpenLNS CT saves the drawing in AutoCAD format, and you can open the drawing using AutoCAD software. If your network contains multiple OpenLNS CT drawings, you must export each drawing as a separate AutoCAD drawing to export the entire network.

5

Installing Networks

This chapter describes how to install devices using OpenLNS CT, including how to load applications into them, set the initial state of their applications, set the source of their configuration properties, and select how they manage device-specific configuration properties. It also explains the different methods for acquiring device Neuron IDs and how to alternatively use the device discovery method to install a network.

Network Installation Overview

You can install the devices that you added to your OpenLNS CT network design in *Designing Networks*. To install devices, you commission the devices and routers in your OpenLNS CT network design. Commissioning associates the device SmartShape created in the OpenLNS CT drawing with a physical device on the network and updates the physical device to match the configuration in your network design.

You can commission devices after you design the network (engineered system installation) or as you add them to the network design (ad-hoc installation). You can also partially design a network and commission devices, and then incrementally add and commission more devices to the network.

You can add and commission a device in one step by selecting the **Commission Device** check box in the New Device or New Router Wizard. You can set this check box to be selected by default by selecting the **Commission after Creation** check box in the **Network Properties: Device** options category (see Appendix A).

To commission a device, you specify its Neuron ID. OpenLNS CT then updates the network address of the device, loads the device application (if necessary), sets the initial state of the device application (running or not), and updates the device's configuration properties. You can specify the Neuron ID by pressing the service pin on a device, manually entering (typing or scanning) it, or manually discovering the device (see *Commissioning using Device Discovery* for more information on using this method). A device is automatically commissioned when you load a new application into it or install it as a replacement for another commissioned device on the network.

When you commission a device, the network design is updated to include the Neuron ID of the device and its network configuration. The physical device may also be updated depending on whether OpenLNS CT is attached to the network and whether it is OnNet or OffNet.

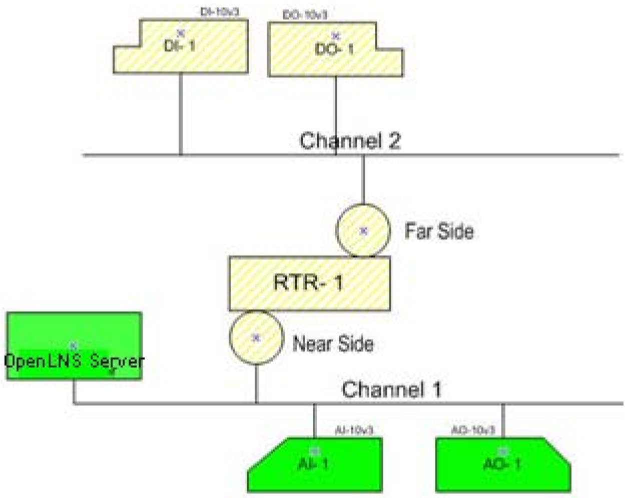
- If OpenLNS CT is attached to the network and OnNet, commissioning updates the physical device. Specifically, OpenLNS CT associates the Neuron ID with the device and downloads all network variable connections and configuration properties.
- If OpenLNS CT is unattached to the network or attached but OffNet, commissioning associates the Neuron ID with the device, but the physical device is not updated. To update the physical device, you need to attach OpenLNS CT to the network, set OpenLNS CT OnNet, and then recommission the device. Recommissioning is the same as commissioning except that you do not have to specify the Neuron ID because it is already stored in the network database. If OpenLNS CT is unattached, you can only manually enter the Neuron ID (you cannot press the service pin or use device discovery to specify the Neuron ID).

Commissioning a Device

To commission a device, you do the following:

1. Select the device SmartShape. You can select multiple device SmartShapes to commission more than one device at a time.
2. Load the application image, if necessary (application devices only).
3. Set the initial state of the application and select the source of configuration property values (application devices only).
4. Set the Neuron ID.

Note: You must commission a router before commissioning any devices on its far side (the side furthest away from the OpenLNS Server). The following figure demonstrates a network in which the devices on the far side of the router cannot be commissioned because the router has not yet been commissioned.



Selecting Devices for Commissioning

To select the device or devices to be commissioned, follow these steps:

1. Right-click the device SmartShape of the device you want to commission, point to **Commissioning**, and then click **Commission** on the shortcut menu. To select multiple devices, do one of the following:

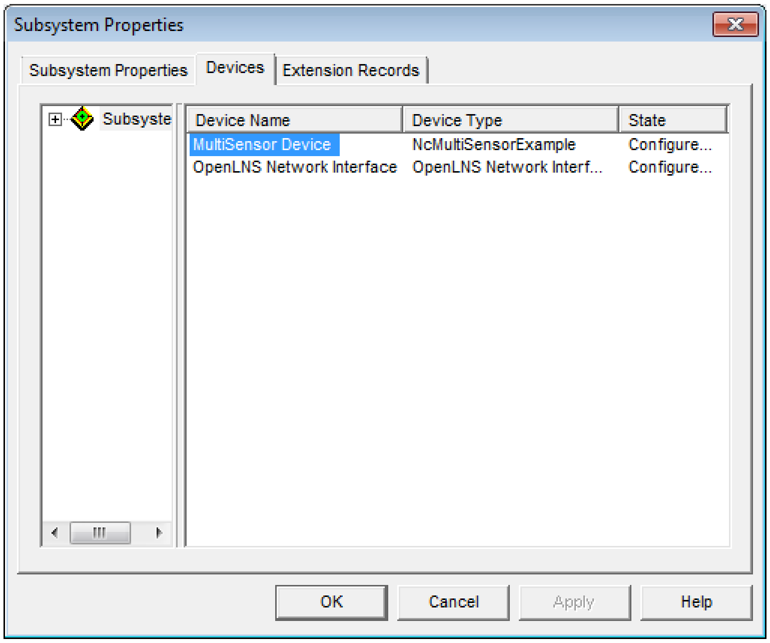
   - To select multiple devices within a subsystem, hold down CTRL and click the device SmartShapes, or click an empty space in the drawing page and drag a selection net around them. Then right-click one of the selected device SmartShapes, point to **Commissioning**, and click **Commission** on the shortcut menu.

   - To select all the devices within a subsystem, select the **Exclude Nested Subsystems from Subsystem Operations** check box (to access this option, right-click the Subsystem SmartShape, or an empty space in the network if you are in the top-level subsystem, and select **Subsystem Properties** from the shortcut menu). Then right-click the Subsystem SmartShape, or an empty space in the drawing page if you are in the top-level subsystem, point to **Commissioning**, and then click **Commission** on the shortcut menu.

   - To select all the devices within a subsystem and all of its nested subsystems, clear the **Exclude Nested Subsystems from Subsystem Operations** check box, right-click the Subsystem SmartShape, or an empty space in the drawing page if you are in the top-level subsystem, point to **Commissioning**, and then click **Commission** on the shortcut menu.

   - To select all the devices within a network, clear the **Exclude Nested Subsystems from Subsystem Operations** check box, right-click an empty space in the drawing page of the top-level subsystem, point to **Commissioning**, and then click **Commission** on the shortcut menu.

     Tip: To view the state of the devices in a subsystem (configured and online or offline, or not commissioned), right-click a subsystem SmartShape or empty space in the drawing page, select **Subsystem Properties** from the shortcut menu, and click the **Devices** tab. If subsystems are nested within the selected subsystem, the **Devices** tab will have an additional pane displaying the subsystem hierarchy. From this pane, select the subsystem with the devices you want to view.



Device Name	Device Type	State
MultiSensor Device	NcMultiSensorExample	Configure...
OpenLNS Network Interface	OpenLNS Network Interf...	Configure...

2. The Commission Device Wizard opens. Where you start in the wizard depends on whether you are commissioning an application device or router, and whether OpenLNS CT is attached to the network and OnNet or OffNet.

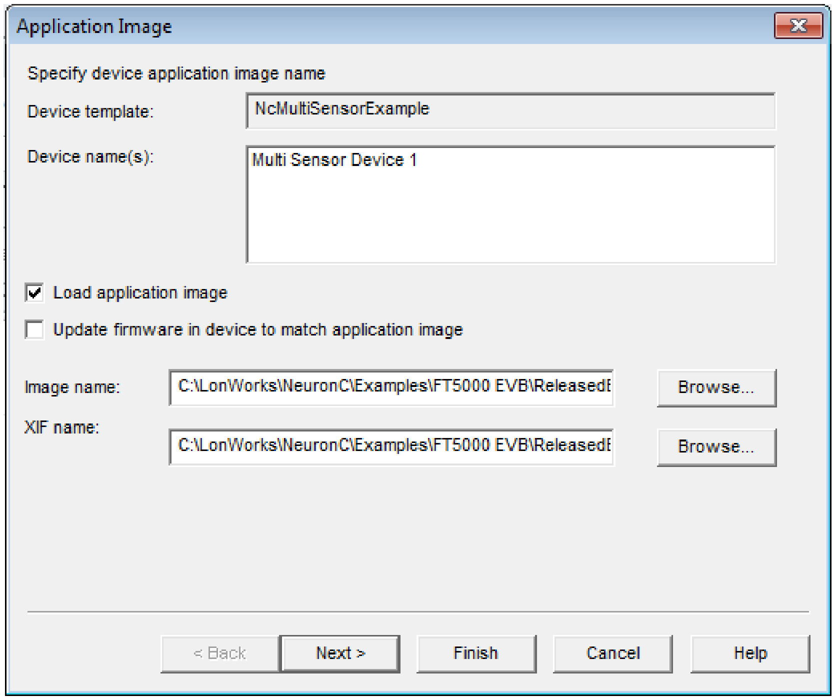
   - If OpenLNS CT is unattached to the network or attached but OffNet, the wizard will skip the Device Application Image and Initial State windows and begin with the Device Identification window.

   - If OpenLNS CT is attached to the network and OnNet, and you are commissioning at least one application device, the wizard will begin with Device Application Image window. If you are only commissioning routers, the wizard will begin with the Initial State window.

Loading a Device Application Image

You can load an application image into an application device that supports application loading (some application devices have a fixed application image that cannot be changed). For application devices that support application loading, the device manufacturer may load the application into the application device when it is manufactured, or they may give you the option to either load the initial application, a different application, or an upgraded application.

When you load an application image into an application device, you can also load a new Neuron firmware image into the device. OpenLNS CT supports devices that have application images based on different Neuron firmware images.



Application Image

Specify device application image name

Device template: NcMultiSensorExample

Device name(s): Multi Sensor Device 1

☑ Load application image

□ Update firmware in device to match application image

Image name: C:\LonWorks\NeuronC\Examples\FT5000 EVB\ReleasedE

XIF name: C:\LonWorks\NeuronC\Examples\FT5000 EVB\ReleasedE

1. Specify the device application by setting the following options:

Load Application Image Select this check box to load an application image into a Neuron-hosted application device that has writeable application memory and comes with an application file (**.ndl**, **.nxe** or **.apb** extensions).

Update Firmware in Device to Match Application Image Select this check box to upgrade or downgrade the Neuron firmware in the application device to make the firmware compatible with the application image being loaded into the device.

Each application image requires a specific version of the Neuron firmware. If the application image requires a different version of the Neuron firmware than the one already loaded in the device, you can load the correct version of the Neuron firmware by selecting this check box.

For a device to support firmware updates, it must be hosted on a Neuron 5000 core or on a Neuron 3150 core with at least 32 KB of nonvolatile memory starting at address 0.

If this check box is cleared, and the folder containing the selected application file contains a version of the file that matches the current device Neuron firmware with a file name of <*appImage*>fw<*n*>.apb, where n is the firmware version, OpenLNS CT will automatically load that application file into the device. Otherwise, OpenLNS CT will load the selected application file into the device.

Note: A device may become inoperable if there is a power failure during a firmware update.

Image Name Specify the pathname of the application file. If you specified a device interface (XIF) file in the **Device Template** dialog (see *Creating Application Devices* in Chapter 4) and you are creating and commissioning the device at the same time, you cannot specify a different XIF file. If an application image file is found in the same directory as the XIF file of the device template, that directory is the default.

XIF Name — If you are loading an application file with an .nxe extension, select the pathname of the XIF file.

OpenLNS CT only requires the binary application file (**.apb** extension) and the binary device interface file (**.xfb** extension); however, it automatically creates binary files from existing text files (**.nxe** and **.xif** extensions) and saves them in the application file folder that contains the corresponding .nxe file.

If you are working on a remote client, OpenLNS CT looks for the application and device interface files on the OpenLNS Server. You must specify the full pathname (including drive) of the application and device interface files on the OpenLNS Server. You cannot load an application file directly from a remote client computer onto a device.

2. Click **Next** to continue. If you are commissioning multiple devices with different device templates, additional windows for each device template and the associated devices will open. Repeat this process for each device template. When there are no more device templates, the Setting the Device Initial State window opens.

When you load an application image, verify that the Device Name field contains the name of the device to be loaded to ensure that the application is being loaded into the correct device. If the device already has an application, OpenLNS CT will compare the program ID of the new application with the program ID of the application in the device. If both applications use a type 8 or 9 program ID, a warning message is displayed if the device Class or Subclass are different. A warning message is also displayed if a program ID that is not a type 8 or 9 program ID will be overwritten with a type 8 or 9 program ID, or vice versa. If you see this warning, verify that you have the correct device and application. If you do have the correct device and application, you can safely ignore the warning message.

Setting the Initial Device State and Source of Configuration Property Values

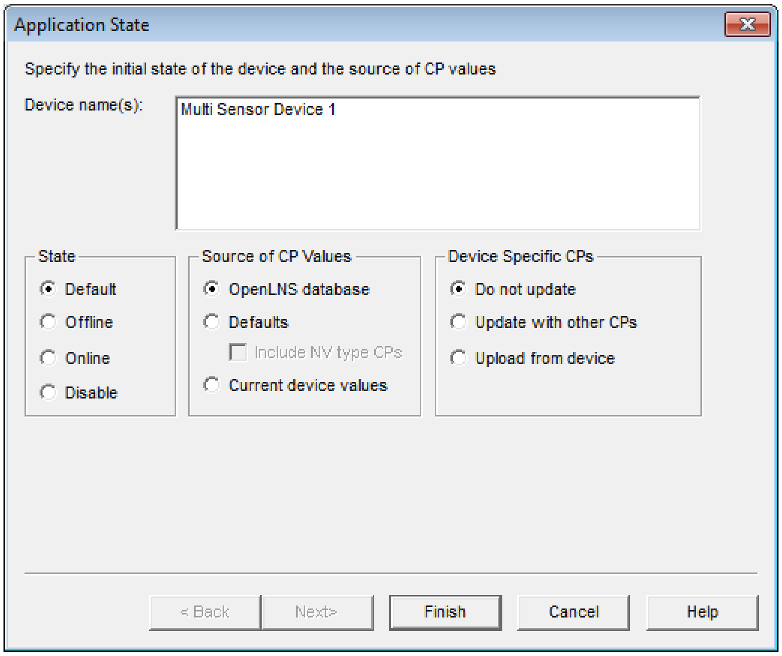
You can set the initial device state and the source of configuration property values (application devices only) when you commission a device. You can also specify how device-specific configuration properties are managed. Unlike other configuration properties, device-specific configuration properties have values that can be modified by tools not based on OpenLNS or by the device itself.

Alternatively, you can have OpenLNS CT skip the Setting the Device Initial State window and use the default settings in the **Network Properties: OpenLNS CT Options** dialog. You can set the default initial state of devices under the **New Device State** box in the **Network Properties: Device** options category. You can set the default source of the configuration property values under the **CP Value Source** box in the **Network Properties: Configuration Properties** options category.

When commissioning a router, this window contains only the options to set the initial state of the router application, as routers do not have configuration properties. When you are commissioning application devices and routers together, the initial state window will appear for the devices first and then for the routers.

Setting the Initial Application Device State and Source of Configuration Property Values

When the Device Initial State window opens, follow these steps:



Application State

Specify the initial state of the device and the source of CP values

Device name(s): Multi Sensor Device 1

State

● Default
○ Offline
○ Online
○ Disable

Source of CP Values

● OpenLNS database
○ Defaults
□ Include NV type CPs
○ Current device values

Device Specific CPs

● Do not update
○ Update with other CPs
○ Upload from device

1. Specify the initial state of the device applications, source of the configuration property values, and management of device-specific configuration properties by setting the following options:

State — Set the state of the device application after it has been commissioned. You have four choices:

- **Default**. The state of the device application is determined by the current setting under the **New Device Options** box in the **Network Properties: Device** options category. If the device was previously commissioned and is able to communicate, the default state is the state of the device before the current operation.

- **Online**. Places the device in the online state. The behavior in the online state depends on the device. A Neuron-hosted device, for example, will run its application after it has been commissioned.

- **Offline**. Places the device in the offline state. The behavior in the offline state depends on the device. A Neuron-hosted device, for example, will not run its application after it has been commissioned. This is the default and can be used to bring up a system incrementally.

  When a device is offline, portions of its application may still function and it will still respond to Online, Wink, and Test commands. An offline device still receives network variable updates, but it typically does not process updated network variable values or send network variable updates.

  Resetting an offline device makes it go online, unless the device is in the hard offline state, in which case it will remain offline after a reset.

- **Disable**. Keeps the device application disabled even if the device is reset. For a device to be disabled, it must contain a Node Object functional block. If **Disable** is selected and the device does not contain a Node Object, the device is placed offline. As with offline operation, portions of a disabled device's application may continue to function.

Source of CP Values

Specify the source of initial values for the device's configuration properties.

- **OpenLNS Database**. The current configuration property values stored in the OpenLNS network database will be written to the device.
- **Default**. The default configuration property values stored in the device template will be written to the device. This option requires a device template based on a version 4 or higher XIF file with specified default values. See the *LONMARK Device Interface File Developer's Guide* on the LonMark Web site for more information (*www.lonmark.org/technical_resources/guidelines/developer*). This is the default.

  If you select **Default**, select whether network variable type configuration properties (**SCPTnvType**) are restored to their default values when the device is commissioned. If you want them to be restored, select the **Include NV Type CPs** check box. This check box is cleared by default. This operation will not work if any of the affected network variables are connected.
- **Current Device Values**. The current configuration property values set in the device will be written to the OpenLNS network database.

  This option is only available if you cleared the check box to load an application image into the device in the Device Application Image window.
- **Application Image File**. The current configuration property values in the selected application image file will be written to the device.

  This option is only available if you selected the check box to load an application image into the device in the Device Application Image window.

Device Specific CPs

Select how device-specific configuration properties are managed when a device is commissioned.

- **Do Not Update**. Device-specific configuration properties will not be modified in the device or the OpenLNS network database. This is the default.
- **Update with Other CPs**. The device-specific configuration property values will be set based on the option selected in the **Source of CP Values** box. For example, if you select **OpenLNS Database** in the **Source of CP Values** box, the device-specific configuration property values stored in the OpenLNS network database will be written to the device.
- **Upload from Device**. The current device-specific configuration property values stored in the device will be written to the OpenLNS network database.

  This option is only available if you cleared the check box to load an application image into the device in the Device Application

Image window.

- **Upload from New App Image**. The current device-specific configuration property values in the selected application image file will be written to the device.

  This option is only available if you selected the check box to load an application image into the device in the Device Application Image window.

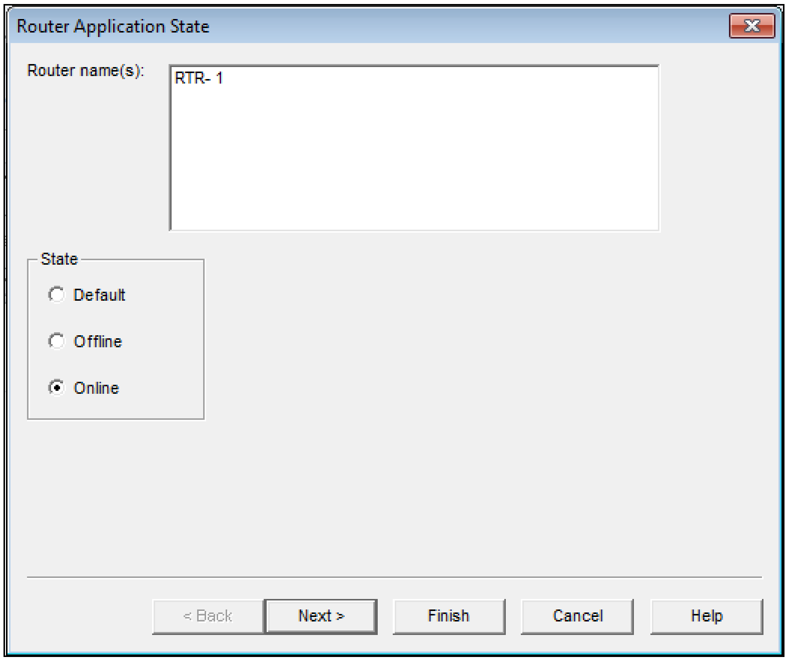
- **Preserve Device Values**. The current device-specific configuration property values set in the device will be written to the OpenLNS network database before the new application image is loaded into the device. Once the new application image has been installed, the device-specific configuration property values in the OpenLNS network database are written to the new device.

  This option is only available if you selected the check box to load an application image into the device in the Device Application Image window.

2. Click **Next**. If you are commissioning routers, the Initial Router State window opens. Otherwise, the Device Identification Method window opens (in this case, see *Selecting the Device Identification Method* to continue installing the devices).

Setting the Initial Router State

When the Router Initial State window opens, follow these steps:


Router Application State

Router name(s): RTR- 1

State

○ Default

○ Offline

● Online

1. Specify the initial state of the router by setting the following option:

State — Set the initial state of the router. You have three choices:

- **Default**. The state of the router is determined by the current setting in the **New Device Options** box located in the **Network Properties: Device** options category. If the router was previously commissioned and is able to communicate, the default state is the state of the router before the current operation.

- **Offline**. The router will not forward packets.
- **Online**. The router will forward packets between the channels connected to it. This is the default. If the network will have devices installed on the far side of a router (the side furthest away from the OpenLNS Server), you must commission the router before any of the application devices connected on its far side.

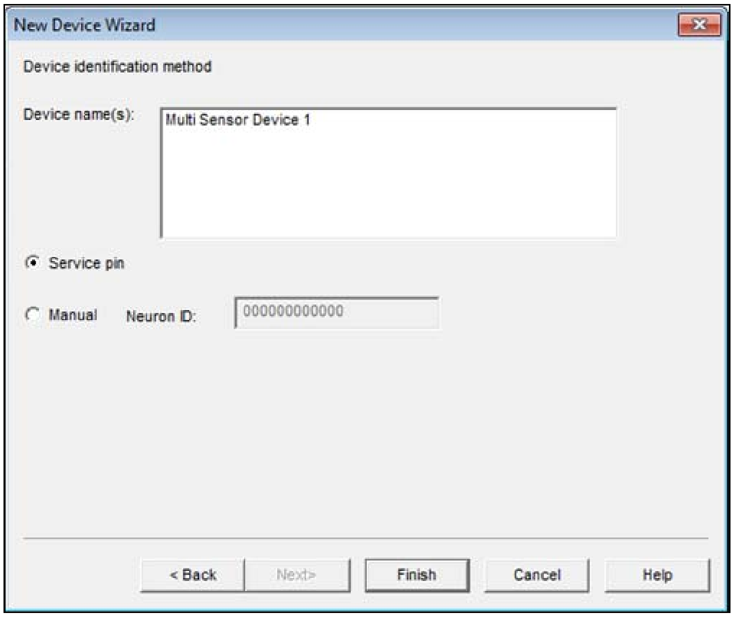
2. Click **Next** to continue. The Device Identification Method window opens. See *Installing Devices and Routers* to continue installing the router.

Setting the Neuron ID

You must set the Neuron ID when you commission a device. You can have OpenLNS CT acquire the Neuron ID by pressing a service pin on the device, or you can manually enter the Neuron ID. You can manually enter a Neuron ID by typing its 12-digit hex string or scanning a bar code on the device, if available. If OpenLNS CT is not attached to the network, you can only enter the Neuron ID manually.

Selecting the Device Identification Method

When the Device Identification Window opens, follow these steps:



Device name(s): Multi Sensor Device 1

● Service pin

○ Manual   Neuron ID: 000000000000

1. Select the method for acquiring the Neuron ID by setting the following option.

Device Identification Method

Specify the method for acquiring the Neuron ID of the devices. You have two choices:

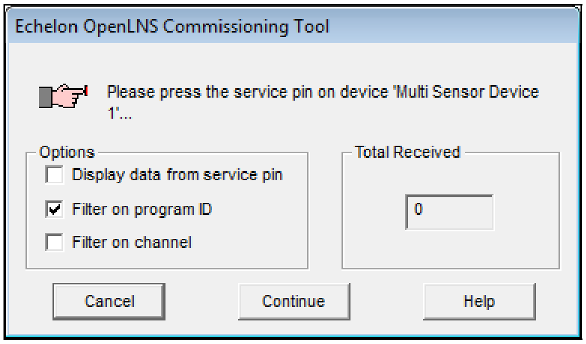
- **Service Pin**. The Neuron ID is acquired by pressing the service pin on the device. See *Using the Service Pin Method.*
- **Manual**.   If you are commissioning one device or router, type or scan the Neuron ID into the **Neuron ID** box. If you are commissioning multiple devices or routers, you will enter the Neuron IDs in the subsequent dialog box. See Using the *Manual Entry Method*.

  Manual entry is typically used if the device or its service pin are difficult to access.

2. Click **Finish**. If you selected the **Service Pin** method, you are prompted to press the service pin as described in *Using the Service Pin Method*. Once the Neuron ID is specified or acquired, OpenLNS CT commissions the device, and changes the color of the device SmartShape to green to indicate that the device has been commissioned.

Using the Service Pin Method

If you selected **Service Pin** as the device identification method, a prompt opens instructing you to press the service pin of the devices or routers you are installing.



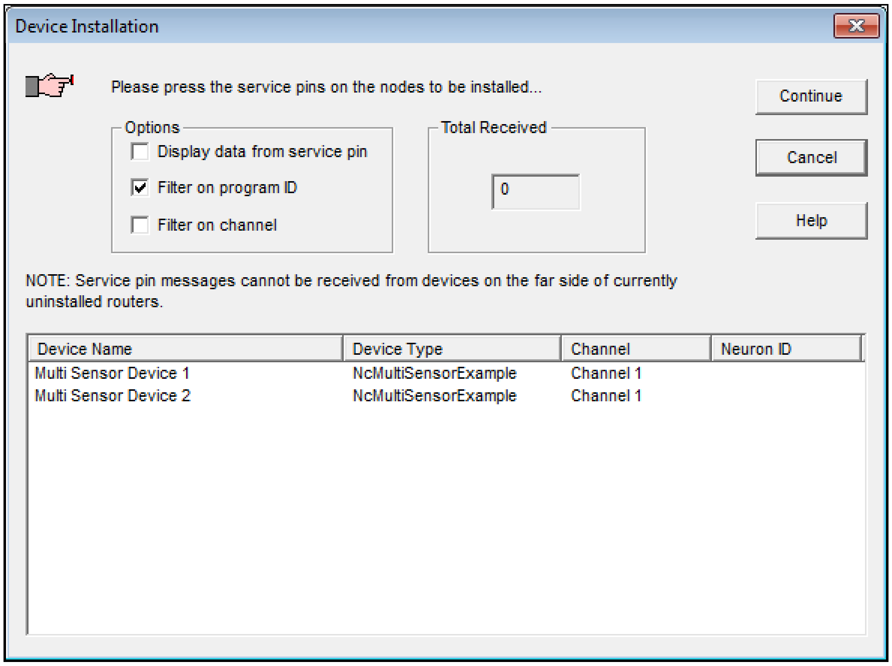
Echelon OpenLNS Commissioning Tool

Please press the service pin on device 'Multi Sensor Device 1'...

Options
- ☐ Display data from service pin
- ☑ Filter on program ID
- ☐ Filter on channel

Total Received: 0

If you are installing multiple devices, the prompt will also include a pane listing the names, types, channels, and Neuron IDs (as you push service pins and acquire them) of the devices you are installing.



Device Installation

Please press the service pins on the nodes to be installed...

Options
- ☐ Display data from service pin
- ☑ Filter on program ID
- ☐ Filter on channel

Total Received: 0

NOTE: Service pin messages cannot be received from devices on the far side of currently uninstalled routers.

Device Name	Device Type	Channel	Neuron ID
Multi Sensor Device 1	NcMultiSensorExample	Channel 1	
Multi Sensor Device 2	NcMultiSensorExample	Channel 1	

To acquire the Neuron IDs and finish installing the devices, follow these steps:

1. Select any of the following service pin options

   Options

   Display Data from Service Pin Select this check box to display a dialog that contains the information returned in the service pin message. This check box is cleared by default.

   Filter on Program ID Select this check box to ignore service pin messages that do not include the same program ID as the device being commissioned. This option is selected by default. This option is cleared if the application image will be loaded during commissioning.

   Filter on Channel Select this check box to ignore service pin messages that do not originate from the same channel as the device being installed. This check box is cleared by default.

   On networks that use repeaters, OpenLNS CT cannot always determine the source channel of the service pin message. For these networks, it may be necessary to clear this check box in order to commission the device.

   The default value for this check box is set in the **Network Properties: Service Pin** options category.

   Total Received Indicates the total number of service pin messages received by OpenLNS CT (including those ignored due to mismatched program ID or channel).

   If this number is incrementing when you press the service pin, but OpenLNS CT is ignoring the service pin message, clear both filter options and select the **Display Data from Service Pin** check box. You may then compare the data received against the expected data to analyze the problem.

2. Press the service pins of the devices you are commissioning.

3. If you are installing one device, OpenLNS CT will begin commissioning the device. The total number of service pin messages received from the network will appear in the **Total Received** box. When OpenLNS CT is done commissioning the device, it will return to the OpenLNS CT drawing. The device SmartShape will be solid green (online) or crosshatched green (soft offline), indicating that the physical device has been commissioned.

4. If you are installing multiple devices, OpenLNS CT will attempt to determine which device you are commissioning based on the service pin message. If multiple devices qualify, OpenLNS CT will select the first one that does not already have an assigned Neuron ID.

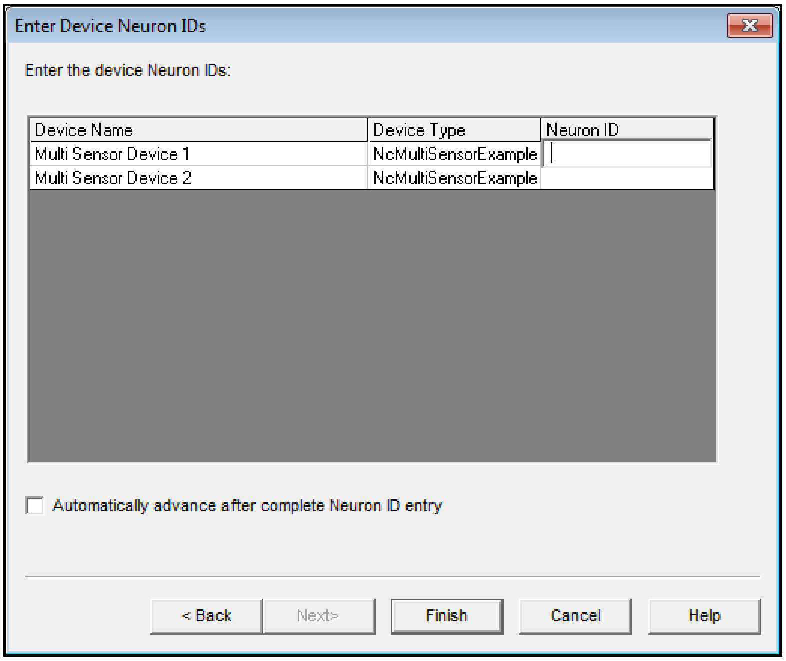
   - You can specify which device or router you are commissioning by clicking its name under the device name column and then pressing its service pin. If a service pin message is filtered or OpenLNS CT is unable to identify the device, the number in the **Total Received** box will be incremented, but no new Neuron ID will appear in the device list.

   - As you press service pins, the total number of service pin messages received from the devices you are commissioning will appear in the **Total Received** box. When all the Neuron IDs have been acquired, OpenLNS CT will commission the devices. When OpenLNS CT is done commissioning, it will return to the OpenLNS CT drawing. The device SmartShapes will be will be solid green (soft online) or crosshatched green (offline), indicating that the physical device has been commissioned.

5. Click **Continue** to quit pressing service pins and to commission the devices that have already been acquired. Click **Cancel** to quit pressing service pins and not commission any devices.

Using the Manual Entry Method

If you selected **Manual** as the device identification method or if OpenLNS CT is not attached to the network, and you are commissioning multiple devices, a prompt opens instructing you to enter the Neuron IDs of the devices you are installing.



Enter Device Neuron IDs

Enter the device Neuron IDs:

Device Name	Device Type	Neuron ID
Multi Sensor Device 1	NcMultiSensorExample	
Multi Sensor Device 2	NcMultiSensorExample	

□ Automatically advance after complete Neuron ID entry

The device Neuron ID is provided by the manufacturer as a 12-digit hex string or as a bar code label on the device that can be scanned.

Entering a Neuron ID Manually

To enter a device's Neuron ID follow these steps:

1. Click within the **Neuron ID** box of the desired device.
2. Type or scan in the device's Neuron ID.
3. After all the Neuron IDs have been entered, click **Finish** to commission the devices. When OpenLNS CT is done commissioning, the device SmartShapes will be solid green (online) or crosshatched green (soft offline), indicating that the physical devices have been commissioned.

You can select the **Automatically Advance** check box to automatically move to the next listed device after you enter a Neuron ID.

Entering a Neuron ID Using a Bar Code Scanner

Some device manufacturers provide a bar code label on their devices that has the Neuron ID. You can use several methods to scan the bar code and store the device Neuron ID in the OpenLNS database.

A *wedge* interface scanner provides the easiest method of transmitting bar code information from the bar code label to OpenLNS CT. This type of scanner is connected to your computer by a wedge inserted between the keyboard and the system unit. The computer interprets the scanned bar code the same as if you entered the Neuron ID on the keyboard. Most wedge interface scanner vendors supply all necessary cables and connectors along with the scanner itself.

You can choose from three other types of scanners: a gun-style laser scanner, a gun-style LED scanner, and pen-style scanner. The price of these scanners correlates with their effectiveness. A gun-style laser scanner is the most expensive, but provides the best accuracy and reliability. Gun-style LED scanners are moderately priced and fairly reliable. Pen-style scanners are the lowest priced, but are typically not reliable on the fine-line bar codes typically used for Neuron IDs. Check with your scanner vendor for prices on the different scanner types. If you are using a LonPoint device, or a device with a similar bar code label, it is recommended that you use a gun-style laser or LED scanner. One scanner that you can use with OpenLNS CT devices is the Symbol Technologies LS 3000, which is made available by Barcode Logics. See your scanner documentation for instructions on programming the scanner.

Commissioning an IP-852 Router

To commission an IP-852 router such as a SmartServer (with IP-852 routing option), i.LON 100 Internet Server (with IP-852 routing), or i.LON 600 IP-852 router, follow these steps:

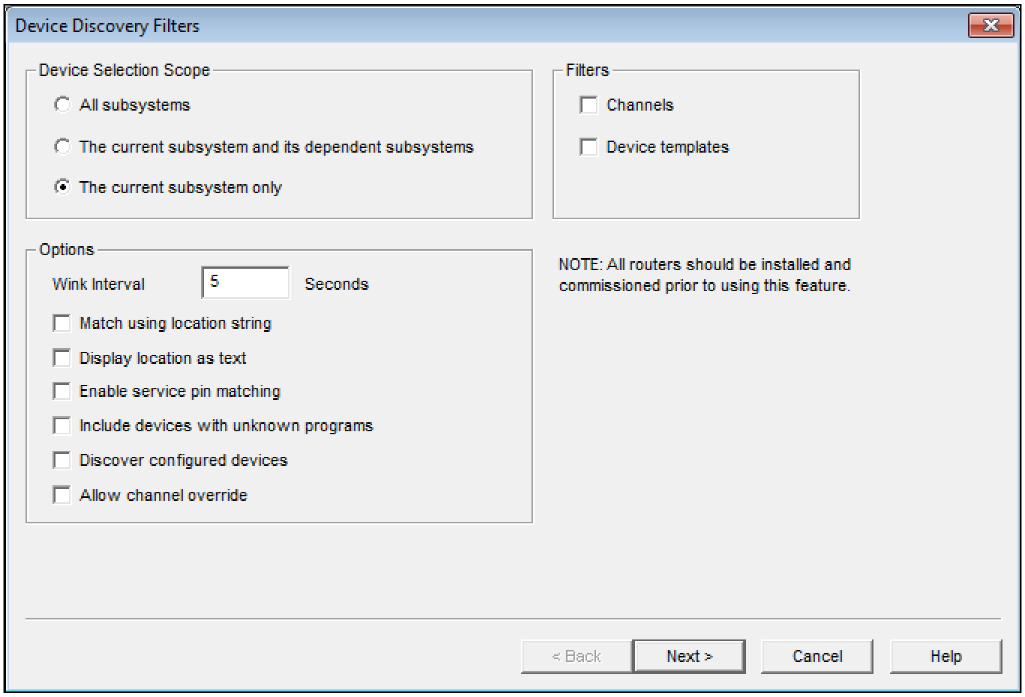
1. Use the IP-852 Configuration Server to add the IP-852 router to the IP-852 channel as described under *Using an IP 852 Network Interface* in Chapter 3. Leave the IP-852 Configuration Server open and return to OpenLNS CT.
2. Right-click the i.LON IP-852 router's SmartShape and then select **Commission** from the shortcut menu. The Commission Device Wizard opens.
3. If you are commissioning a SmartServer or i.LON 100 Internet Server, select the initial state and source of the configuration property values for the Web server.
4. Select the initial state of the IP-852 router. Click **Next**.
5. Select the device identification method. Click **Next**.
6. Press the service pin of the IP-852 router, or manually enter its Neuron ID and click **Finish**. The IP-852 router will be commissioned.
7. Repeat steps 2–6 for additional IP-852 routers. Close the IP-852 Configuration Server.

Commissioning Using Device Discovery

You can install a network by automatically discovering the devices to be commissioned. With the device discovery method, OpenLNS CT searches the physical network for uncommissioned devices that match device SmartShapes defined in your OpenLNS CT drawing. Devices are matched based on their program ID and channel. If there is no unique match, you can select the appropriate device by winking it or by pressing a service pin. After discovered devices are matched with defined SmartShapes, you can have OpenLNS CT commission the devices. You can use the device discovery method to quickly install networks that have been designed with the engineered system installation scenario.

To install a network using the device discovery method, follow these steps:

1. Click **Add-ins**, point to **LonWorks Network**, and then click **Discover Devices**. The **Device Discovery Filters** dialog box opens.


Device Discovery Filters

Device Selection Scope

○ All subsystems

○ The current subsystem and its dependent subsystems

● The current subsystem only

Filters

□ Channels

□ Device templates

Options

Wink Interval 5 Seconds

□ Match using location string

□ Display location as text

□ Enable service pin matching

□ Include devices with unknown programs

□ Discover configured devices

□ Allow channel override

NOTE: All routers should be installed and commissioned prior to using this feature.

2. Select the following options:

Device Selection Scope

Selects which subsystems will be searched for uncommissioned device SmartShapes. You have three choices:

- **All Subsystems**. The entire OpenLNS CT drawing will be searched for uncommissioned device SmartShapes.
- **The Current Subsystem and Dependent Subsystems**. The current subsystem and any subsystems nested in it will be searched for uncommissioned device SmartShapes.
- **The Current Subsystem only**. Only the current subsystem will be searched for uncommissioned device SmartShapes.

Options

Wink Interval

Specifies the number of seconds to wait between winking devices when more than one is selected.

Match using Location String

Uses location string in addition to channel and program ID to match devices. This option is useful for networks that have a unique location string for each device, or if you have a handheld tool that assigns a unique location string to each device.

Display Location as Text

Interprets location string as ASCII text. The default is to display the location string as hex digits.

Enable Service Pin Matching

Enables matching to be done by selecting a device SmartShape in the **Defined Devices** list and pressing the service pin of a physical device. The physical device does not need to be in the list of discovered devices; therefore, this method can be used to commission configured devices. However, the physical device may not already be assigned to a device SmartShape. You can only select one device in the **Defined Devices** list when you press the service pin.

Include Devices with Unknown Programs

Adds physical devices to the discovered device list that are applicationless or have an unrecognized application. Such physical devices can then be matched to any device SmartShape with a matching channel. However, after matching, you must click **Details** and specify an application image to load into the device.

Discover Configured Devices

Discovers devices that have previously been configured. If you select this option, a dialog opens and you must indicate the domain of the configured devices to be discovered.

Allow Channel Override

Allows matching of physical devices and device SmartShapes with different channels. If you select this option, you must make sure that the device SmartShape and physical device are located on the same channel in order to commission the device.

This option is useful if your network has routers configured as repeaters, in which case a physical device could be discovered on the wrong channel. This option allows you to match the device in this scenario.

Filters

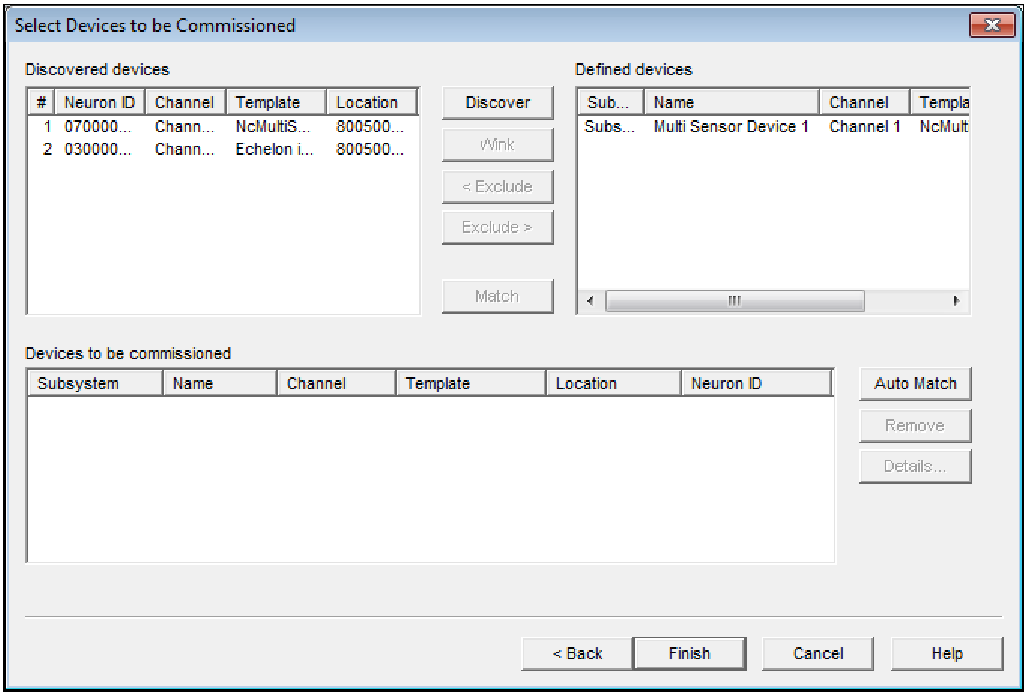
Channels

Limits the devices discovered based on the channel on which they are attached.

Device Templates

Limits the type of devices discovered based on the device template used.

3. Click **Next**. If you selected the discover configured devices option or the channel or device filtering options, complete steps 4–6, as needed. Otherwise, skip these steps and go to step 7 to select which devices to match and commission.
4. If you selected the channel filtering option, select the channel that OpenLNS CT will search for devices. To select multiple channels, hold down CTRL and click the desired channels. Click **Next**.
5. If you selected the device template filtering option, select the type of device that OpenLNS CT will attempt to discover. To select multiple device templates, hold down CTRL and click the desired channels. Click **Next**.
6. If you selected the **Discover Configured Devices** option, select the domain ID of the devices to be re-installed in the current network. These devices were self-installed or previously installed on another network. If you need to discover configured devices on multiple domains, select the first domain, click **Next**, click **Back**, and then select the next domain ID.
7. If there is a router between the OpenLNS Server and the configured device(s) to be discovered, this router must be configured with the same domain as the device(s), unless the router is configured as a repeater. If the network has configured routers, you can temporarily change them to repeaters, complete the device discovery process, and then change them back. Click **Next**.
8. The **Select Devices to be Commissioned** dialog box opens.



Discovered devices

#	Neuron ID	Channel	Template	Location
1	070000...	Chann...	NcMultiS...	800500...
2	030000...	Chann...	Echelon i...	800500...

Defined devices

Sub...	Name	Channel	Templa
Subs...	Multi Sensor Device 1	Channel 1	NcMult

Devices to be commissioned

Subsystem	Name	Channel	Template	Location	Neuron ID

This dialog box allows you to manually or automatically match the discovered devices on the network with the defined device SmartShapes in the OpenLNS CT drawing. To select a device from any of the lists, click the entry in the leftmost column (the **#** column in the **Discovered Devices** list, and the **Subsystem** column in the **Defined Devices** and **Devices to be Commissioned** lists). After selecting a device, its field in the leftmost column will be shaded blue. Once you click a device in the other list, the field will be shaded gray. When the devices in each column are compatible, the match command will become available.

If you selected the **Include Devices with Unknown Programs** option and such devices are discovered, they will be listed under Discovered devices. However, the **Template** field will be empty if the device is applicationless, or it will display the device's program ID in hex format if there is no corresponding device template.

9. Use the following commands to match device SmartShapes in your OpenLNS CT drawing to physical devices on the network:

Discovered and Defined Device Lists

Discover

Searches for devices. If one or more physical devices were not discovered, you can force their discovery by pressing their service pins and then pressing the **Discover** button. Devices that were removed from the **Discovered Devices** list using the **Exclude** button are not rediscovered.

Wink

Winks the selected device(s) in the **Discovered Devices** list. The action performed by the device is application dependent. For example, some devices may blink their power LED when winked.

This command is only available if one or more devices are selected in the **Discovered Devices** list.

Exclude

Removes the selected device(s) from the **Discovered Devices** or **Defined Devices** list. Once a device has been excluded, it can only be re-listed by restarting the device discovery process.

This command is only available if a device is selected in the appropriate device list.

Match Matches the device selected in the **Defined Devices** list is to the device selected in the **Discovered Devices** list. The devices are removed from their respective lists and the matched device is added to the **Devices to be Commissioned** list. This command is only available if one device is selected from each list and the selected devices are compatible.

Compatible devices have the same program ID and channel unless the **Allow Channel Override** or **Include Devices with Unknown Programs options** have been selected. Discovered devices with an unknown program can be matched to any defined device. However, after the device is matched, you must click **Details** and specify an application image to be loaded into the device before you click the **Finish** button.

If the **Enable Service Pin Matching** option was selected, you can also match devices by selecting a device from the **Defined Devices** list and then pressing the service pin of a compatible device to which it should be matched.

Devices to be Commissioned

Auto Match Matches all discovered and defined devices that can be uniquely matched. The devices are removed from their respective lists and the matched devices are added to the **Devices to be Commissioned** list.

Remove Removes the selected devices from the **Devices to be Commissioned** list. The devices are returned to the **Discovered Devices** and **Defined Devices** lists.

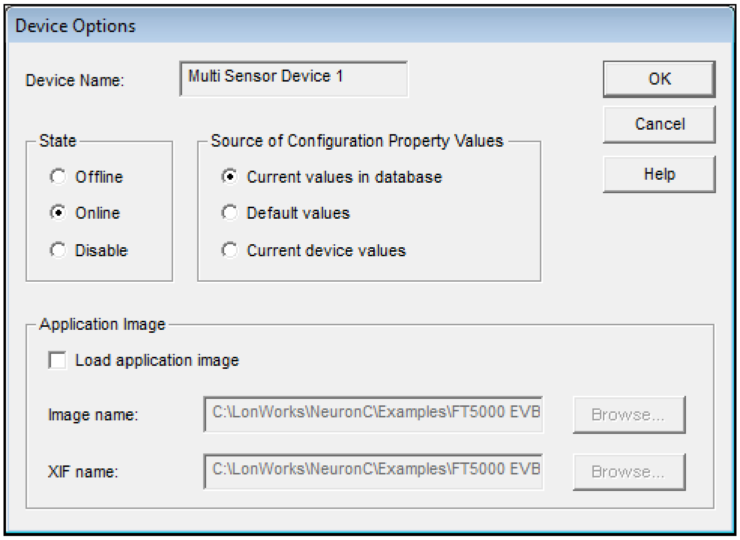
This command is only available if one or more devices are selected in the **Devices to be Commissioned** list.

Details Opens the **Device Options** dialog box.

Click **Details** to set the initial application state and the source of the configuration property values for the selected matched devices. By default, the devices will be set in a soft offline state.

You must click **Details** to specify an application image to load into an applicationless device that has been discovered and subsequently matched.

10. Click **Details** to set the initial application state and the source of the configuration property values of a matched device, or to specify an application image to load into a matched device that is applicationless. The **Device Options** dialog box opens.



Device Name: Multi Sensor Device 1

State

○ Offline

● Online

○ Disable

Source of Configuration Property Values

● Current values in database

○ Default values

○ Current device values

Application Image

□ Load application image

Image name: C:\LonWorks\NeuronC\Examples\FT5000 EVB

XIF name: C:\LonWorks\NeuronC\Examples\FT5000 EVB

11. Select an application image and set the options for the initial application state and source of configuration property values. See *Loading a Device Application Image* and *Setting the Initial Application Device State and Source of Configuration Property Values* for information on setting these options. Click **OK**.
12. Click **Finish**. OpenLNS CT commissions the devices in the **Devices to be Commissioned** list and sets the state of their applications. When OpenLNS CT is done commissioning, it returns to the OpenLNS CT drawing. The device applications will be in a soft offline state, unless you specified the application state in the **Device Options** dialog box. The device SmartShapes will be green indicating that the devices have been commissioned, or crosshatched green if an application state was not specified or the device application was set offline.

6

Monitoring and Controlling Networks

This chapter describes how you can monitor and control the devices in your network with OpenLNS CT. It describes the three methods you can use to read and/or write network variables and configuration properties: using monitored connections, browsing with the OpenLNS CT Browser, and using Data Point SmartShapes. This chapter covers how to bind network variables to the host in order to receive event-driven updates. It also describes how to change the types and formats of network variables and configuration properties. It explains how to create simple HMIs in your OpenLNS CT drawing with Data Point and Visio SmartShapes.

Monitoring and Controlling Overview

You can monitor and control the physical devices in a network by reading and writing the network variables on the devices. You can also read configuration property values. You can use any of the following three methods to read and/or write network variables and configuration properties:

- *Monitored connections*. Monitored connections provide a simple means to view network variable values within your network. Using monitored connections, you can view the values of network variables on your OpenLNS CT drawing. This capability is useful for monitoring and debugging your installation, as monitored connections provide an easy way to visualize the flow of data through your functional blocks. You can also display a network variable value without monitoring a connection.
- *OpenLNS CT Browser*. The OpenLNS CT Browser is a plug-in that can be used to read and write network variables and configuration properties on any functional block on any device. This is useful for getting a detailed view of the operation of any functional block, and for setting configuration properties on functional blocks that do not have *plug-ins*.
- *Data Point SmartShape*. The Data Point SmartShape is an OpenLNS CT SmartShape that you can use to monitor and control the value of any network variable, configuration property, or functional block state (enabled or in override) in your network. You can also create simple human-machine interfaces (HMIs) within your OpenLNS CT drawing using the Data Point SmartShape.

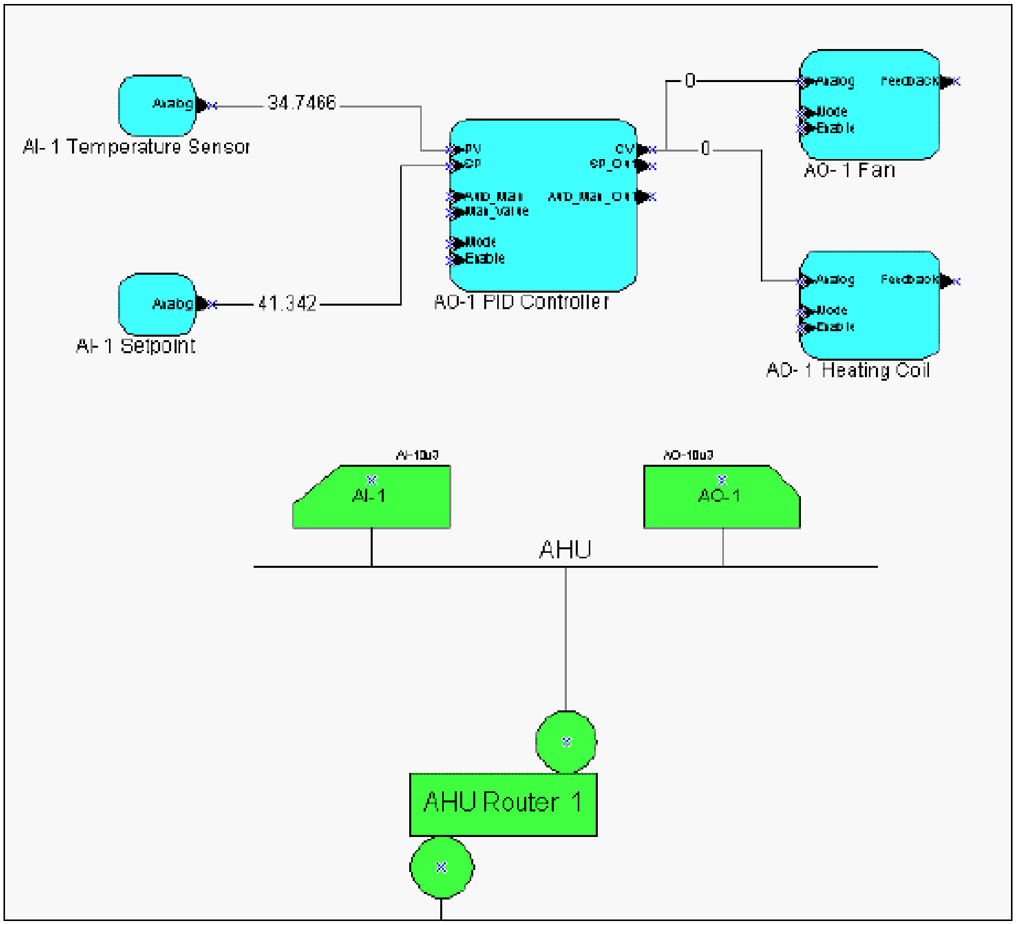
Using Monitored Connections

You can use a monitored connection to monitor the value of a network variable that is attached to a connector or reference connection SmartShape. You can monitor one end of a connection or both ends at the same time.

To monitor a connection, follow these steps:

1. Right-click an empty space in the drawing page and set **Enable Monitoring** on the shortcut menu.
2. To enable monitoring for a connector SmartShape, use one of the following two methods:
   - Double-click the connector SmartShape. The input network variable value opens on the connector SmartShape in black text. Double-click the connector SmartShape again to display both the output and input network variable values. Double-click again to display just the output network variable. Double-click again to cycle through the same choices.
   - Right-click the connector SmartShape and select either **Monitor Input Value** or **Monitor Output Value**. The network variable value on the side of the connection you selected opens on the connector SmartShape. To monitor both ends of the connection, right-click the connector SmartShape again and set monitoring on the other end of the connection.
3. To enable monitoring for a reference connection SmartShape, use one of the following two methods:
   - Double-click the reference connection SmartShape. The network variable value on the side of the connection you selected opens. To monitor both ends of the connection, double-click the reference connection SmartShape on the other end of the connection.
   - Right-click the reference connector SmartShape and then click **Monitor Value** on the shortcut menu. The network variable value opens on the reference connector SmartShape. To monitor both ends of the connection, right-click the reference connection SmartShape on the other end of the connection and click **Monitor Value**.

The following figure illustrates monitored connections on a PID controller. The values shown on the connection SmartShapes represent the actual network variable values within the operational system.



When you enable monitoring on a connection, the network variable values are continuously polled. The default poll interval is **1** second. To change the default poll interval, click **Add-ins**, point to **LonWorks Network**, and then click **Network Properties**. Click the **Timing** tab, set the default poll rate, and then click **OK**. Excessive polling can create heavy network traffic and consequently degrade network performance.

Monitoring connections can interfere with the successful completion of network installation and maintenance operations such as commissioning devices; therefore, OpenLNS CT disables monitoring during these types of operations.

For more information on monitoring network variable values, see *Monitoring Network Variables* in this chapter.

Displaying a Network Variable Value

You can display the value of a network variable without monitoring a connection as described in the previous section. The network variable must be attached to a connector or reference connection SmartShape, but the connector or reference connection SmartShape does not have to be connected to any other network variables. To display the value of a network variable with a connector SmartShape, right-click the connector SmartShape and then click **Get Input Value** or **Get Output Value** on the shortcut menu. For a reference connection SmartShape, right-click the reference connection SmartShape and then click **Get Value** on the shortcut menu. The network variable value on the side of the connection you selected opens on the connector SmartShape.

By default, the network variable value is displayed for **15** seconds. You can change the display duration by clicking **Add-ins**, clicking **OpenLNS CT**, clicking **OpenLNS CT Options**, and then selecting **NV Browser/ Monitoring** from the **Options Category** list.

Using the OpenLNS CT Browser

You can use the OpenLNS CT Browser to monitor and control the network variables and configuration properties in your network. The OpenLNS CT Browser provides a table view of the selected network variables and configuration properties. You can add all the network variables and configuration properties for a functional block or device to the Browser, and you can delete selected network variables and configuration properties from it.

The OpenLNS CT Browser is an OpenLNS plug-in included with OpenLNS CT. To use the OpenLNS CT Browser, you must register it when you open a network as described in *Creating a New OpenLNS CT Network Design* in Chapter 3, *Getting Started*, or you can register it after you open the network, as described in *Re-registering Plug-ins* in Chapter 9, *Using Plug-ins*. You only have to register the OpenLNS CT Browser once per computer. This is because once the OpenLNS CT Browser is registered on one computer, it is available to all other networks on the same computer.

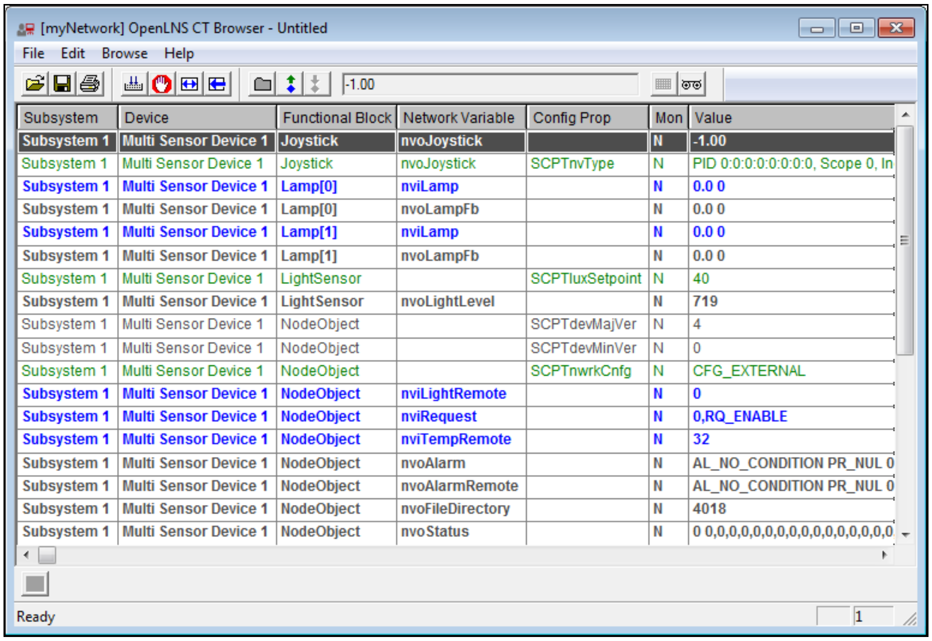
Many functional blocks may come with their own device plug-ins that provide a custom interface for their network variables and configuration properties. A device plug-in provides an application-specific view of a functional block or device. The OpenLNS CT Browser, though, is a generic plug-in that you can use with any functional block or device.

Starting the OpenLNS CT Browser

To start the OpenLNS CT Browser, select one or more functional block and device SmartShapes, right-click one of the selected SmartShapes, and then click **Browse** on the shortcut menu. When the OpenLNS CT Browser opens, it lists all the network variables and configuration properties for the selected functional blocks, and it lists all the network variables and configuration properties on all the functional blocks for the selected devices.

After the OpenLNS CT Browser is open, you can invoke the **Browse** command on other functional blocks and devices. The network variables and configuration properties are added to the bottom of the existing OpenLNS CT Browser window.

Once a browser window is opened, it remains open until you close it. Even if you close the OpenLNS CT drawing, the browser continues to run until you close it. If you open a Browser window, close the OpenLNS CT drawing, then open another browser, both Browser windows run concurrently.



Subsystem	Device	Functional Block	Network Variable	Config Prop	Mon	Value
Subsystem 1	Multi Sensor Device 1	Joystick	nvoJoystick		N	-1.00
Subsystem 1	Multi Sensor Device 1	Joystick	nvoJoystick	SCPTnvType	N	PID 0:0:0:0:0:0:0:0, Scope 0, In
Subsystem 1	Multi Sensor Device 1	Lamp[0]	nviLamp		N	0.0 0
Subsystem 1	Multi Sensor Device 1	Lamp[0]	nvoLampFb		N	0.0 0
Subsystem 1	Multi Sensor Device 1	Lamp[1]	nviLamp		N	0.0 0
Subsystem 1	Multi Sensor Device 1	Lamp[1]	nvoLampFb		N	0.0 0
Subsystem 1	Multi Sensor Device 1	LightSensor		SCPTluxSetpoint	N	40
Subsystem 1	Multi Sensor Device 1	LightSensor	nvoLightLevel		N	719
Subsystem 1	Multi Sensor Device 1	NodeObject		SCPTdevMajVer	N	4
Subsystem 1	Multi Sensor Device 1	NodeObject		SCPTdevMinVer	N	0
Subsystem 1	Multi Sensor Device 1	NodeObject		SCPTnwrkCnfg	N	CFG_EXTERNAL
Subsystem 1	Multi Sensor Device 1	NodeObject	nviLightRemote		N	0
Subsystem 1	Multi Sensor Device 1	NodeObject	nviRequest		N	0,RQ_ENABLE
Subsystem 1	Multi Sensor Device 1	NodeObject	nviTempRemote		N	32
Subsystem 1	Multi Sensor Device 1	NodeObject	nvoAlarm		N	AL_NO_CONDITION PR_NUL 0
Subsystem 1	Multi Sensor Device 1	NodeObject	nvoAlarmRemote		N	AL_NO_CONDITION PR_NUL 0
Subsystem 1	Multi Sensor Device 1	NodeObject	nvoFileDirectory		N	4018
Subsystem 1	Multi Sensor Device 1	NodeObject	nvoStatus		N	0 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

The OpenLNS CT browser color codes the configuration properties and network variables in the following manner:

Blue Input network variables. You can write values to these network variables. See *Updating Network Variable and Configuration Property Values.*

Green Writeable configuration properties. You can write values to these configuration properties. See *Updating Network Variable and Configuration Property Values*.

Grey Output network variables and read-only configuration properties. You cannot write to these objects.

Each row in the browser contains the following information about a network variable or configuration property:

Subsystem Displays the name of the subsystem in which the functional block containing the network variable or configuration property is located.

Device Displays the name of the device associated with the functional block containing the network variable or configuration property.

Functional Block Displays the name of the functional block containing the network variable or configuration property.

Network Variable Displays the name of the network variable if described in the row. If the row describes a configuration property associated with a specific network variable, this column shows that network variable. Otherwise, this column is blank.

Config Prop Displays the standard or user-defined type of the configuration property if the row describes a configuration property. (see *types.lonmark.org*) If a row describes a network variable, this column is blank.

Mon Indicates whether the network variable is monitored. A "**Y**" indicates that monitoring is on; a status indicator follows the "**Y**", indicating when a new value is received by the browser. Configuration properties cannot be monitored.

Value Contains the value of the network variable or configuration property. For monitored network variables, the value is updated through network variable polling or bound updates as described under *Monitoring Network Variables*.

Note: If a plug-in other than the OpenLNS CT Browser opens when you click **Browse**, the other plug-in was registered after the OpenLNS CT Browser, and it is currently specified as the default browser. To change the status of the plug-in as the default browser, follow these steps:

1. Right-click a functional block, point to **Plug-ins**, then click **Plug-ins** on the shortcut menu. The **OpenLNS CT Plug-ins** dialog opens.
2. Locate and select the current default browser and then click **Info**. The **OpenLNS CT Information** dialog opens.
3. Under the **Options** box, clear the **Default Plug-in** check box, and then click **OK**. The selected plug-in is no longer the current default browser.

The OpenLNS CT Browser Toolbar

You can use the OpenLNS CT browser toolbar to perform many browser functions quickly. The toolbar contains the following buttons:

Open Opens an existing browser file. See *Customizing the Browser*.

Save Saves changes to a browser file. If the browser file is new, the Save As dialog opens. Name the file and choose a folder in which to save the file. See *Customizing the Browser* for more information.

Print Prints the data in the Browser window.

Monitor All On Turns monitoring on for all network variables displayed in the current Browser window. See *Monitoring Network Variables* for more information.

Monitor All Off Turns monitoring off for all network variables in the current Browser window. See *Monitoring Network Variables* for more information.

Adjust Columns Automatically sizes each column displayed in the Browser window to fit the longest string in that column.

Refresh All Updates the Browser window with all current network variable and configuration property values. If any names of the displayed objects have been changed using another tool, the displayed names are updated.

Properties Allows you to view and set properties for the currently selected network variable or configuration property object. See *Updating Network Variable and Configuration Property Properties* for more information.

Get Value Refreshes the value for the currently selected network variable or configuration property. You can only update network variable values if you are attached to the network.

Set Value Changes the value of the currently selected input network variable or writeable configuration property to the value indicated in the **Value** box located right of this button. For enumerated values, click the down arrow to select from the possible values. Network variable values can only be modified if you are attached to the network.

Table Opens a dialog that displays the selected configuration property values in a table (only for configuration properties that contain a list or table of values). See *Updating Network Variable and Configuration Property Values* for more information.

Details Displays the details of the selected network variable or configuration property. See *Updating Network Variable and Configuration Property Values* for more information.

Display Messages Displays a log of error messages received while the browser is active. The Display Messages button is located at the bottom left corner of the browser. See *Displaying Error Messages* for more information.

Customizing the Browser

You can hide rows and columns in the browser to make it easier to view just the data you want to monitor and control. For example, you may be viewing data from devices in a single subsystem;

therefore, you would hide the subsystem column. In another example, you may want to monitor and control only a few of the network variables listed in the browser; therefore, you would hide the rows of all the other network variables that you do not want to monitor and control.

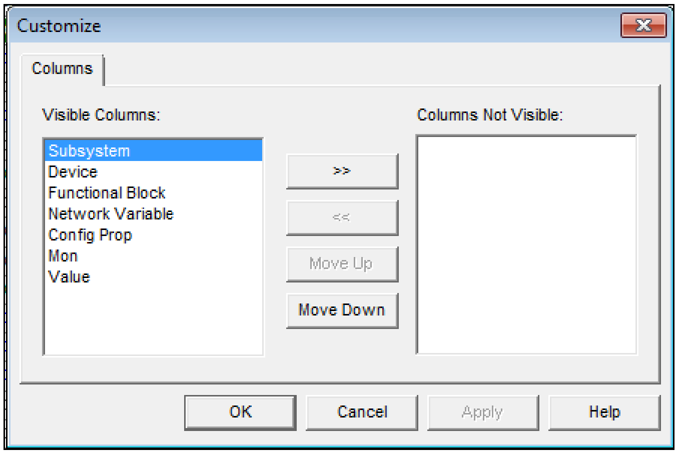
Customizing Browser Columns

You can hide, change the order, or adjust the width of the columns in the OpenLNS CT Browser.

Hiding or Changing the Order of Browser Columns

To hide or change the order of the browser columns, follow these steps:

1. Click **Browse** and then select **Customize**. The **Customize** dialog box opens.



2. To hide columns, select the columns in the **Visible Columns** list you want to hide and then move them to the **Columns Not Visible** list. You cannot hide the **Value** column. The OpenLNS CT Browser displays only the columns you did not hide.

3. To change the order of the columns in the browser, select the column you want to move and click **Move Up** or **Move Down** based on the desired order.

4. Click **OK**. The OpenLNS CT Browser will reflect the changes you made.

Adjusting the Width of Browser Columns

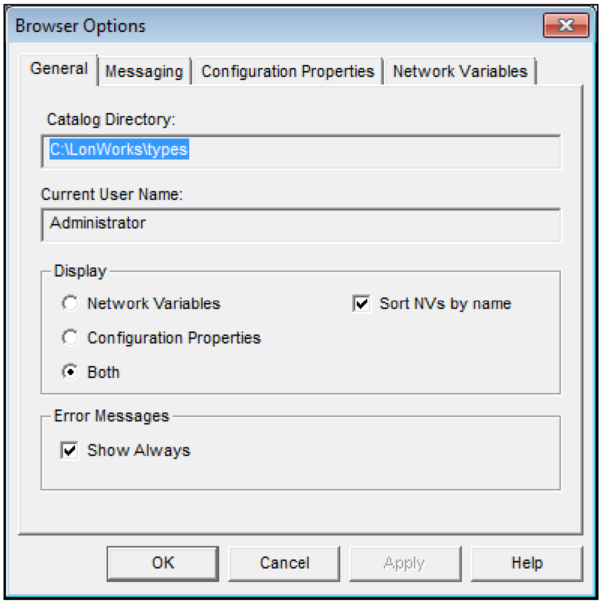
You can adjust the width of the columns in the browser to display the information efficiently. To adjust the column widths, click **Browse** and then click **Adjust Columns** or click the **Adjust Columns** button (↔) in the browser toolbar. The browser automatically resizes the columns to display the longest text string in each column.

Customizing Browser Rows

You can customize the OpenLNS CT Browser so that it only displays rows containing network variables or configuration properties, or you can hide specific network variables or configuration properties by deleting the rows on which they appear.

Selecting Browser Rows to be Displayed

To select which objects are displayed in the OpenLNS CT Browser (network variables and/or configuration properties), click **Browse** and then click **Options**. The **Browser Options** dialog opens with the **General** tab selected.



Catalog Directory: C:\LonWorks\types

Current User Name: Administrator

Display

○ Network Variables

○ Configuration Properties

● Both

☑ Sort NVs by name

Error Messages

☑ Show Always

Under the display box, select which objects you want displayed in the browser. You have three choices:

- **Network Variables**. Only network variables are displayed in the browser.
- **Configuration Properties**. Only configuration properties are displayed in the browser.
- **Both**. Network variables and configuration properties are displayed in the browser.

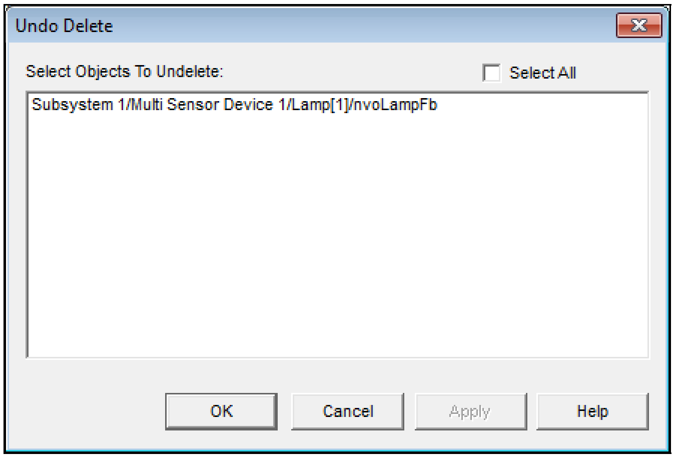
Select the **Sort NVs by Name** check box to list the network variables in the browser alphabetically. If this option is cleared, they are listed by network variable index. Changes to this option do not affect the current browser window. To update the display, close and restart the browser.

Hiding Browser Rows

To hide a network variable or configuration property in the OpenLNS CT Browser, click anywhere in the row containing the object you want to hide, click **Edit**, and then click **Delete Row**. Alternatively, you can click a row and then press DELETE. The browser hides the selected row. This selection does not affect the network variable or the configuration property itself.

To hide all configuration properties and network variables, click **Edit**, click **Delete All Rows**, and then click **OK** on the warning dialog that opens.

To re-display a network variable or configuration property row that is currently hidden, click **Edit**, and then click **Undelete Row**. The **Undo Delete** dialog opens.



Select Objects To Undelete:

□ Select All

Subsystem 1/Multi Sensor Device 1/Lamp[1]/nvoLampFb

This dialog lists all the hidden objects in the browser. Select the object or objects you want to show, or select the **Select All** check box to select all the objects, and then click **OK**. The selected objects appear in the browser again.

Saving Browser Customization

You can save and later restore the rows and columns that you customize in the OpenLNS CT Browser to a browser (.brw extension) file. To save your rows and columns to a browser file, open the **File** menu and select **Save** or **Save As**. Browse to the folder in which browser file is to be stored and enter the name of the browser file.

To re-open a browser file, open the OpenLNS CT Browser from an OpenLNS CT drawing (right-click a functional block or device and then click **Browse** on the shortcut menu) or open the OpenLNS CT Browser as a standalone application (click **Start** on the taskbar, point to **Programs**, point to the **Echelon OpenLNS CT** folder, and then select **OpenLNS CT Browser)**. In the OpenLNS CT Browser, open the **File** menu, click **Open**, browse to the folder in which the browser file is stored, and then select the browser file. Alternatively, you can locate the desired browser file in Windows Explorer and then double-click the file.

To print a browser file, click the **Print** button (🖨) on the browser toolbar, or open the **File** menu and select **Print**. To preview the file as it will be printed, open the **File** menu and select **Print Preview**. To return to the browser window from print preview, click the **Close** button.

Monitoring Network Variables

You can use the OpenLNS CT Browser to monitor network variables.

Enabling Network Variable Monitoring

To enable the OpenLNS CT Browser to monitor a network variable, right-click the network variable row you want to monitor and then click **Monitor** on the shortcut menu. When a selected network variable is being monitored, this option is set in the shortcut menu. To monitor all network variables, right-click any network variable and then click **Monitor All On**, on the shortcut menu, or click the **Monitor All On** button (⇊) on the browser toolbar.

The **Mon** column for monitored network variables changes to **'Y'** and the **Value** column displays the current network variable value. The browser blinks a status indicator next to the **'Y'** when a new value for the monitored network variable opens.

The OpenLNS CT Browser polls network variables at the interval specified on the Network Variables tab of the **Browser Options** dialog. The default poll interval is **1** second. To change how frequently network variables are polled, open the **Browse** menu, click **Options**, click the **Network Variables** tab, and enter the desired poll interval in the **Poll Interval** box. The browser does not poll monitored output network variables if they are connected to the browser computer as described in *Using Bound Updates*.

Disabling Network Variable Monitoring

To stop the OpenLNS CT Browser from monitoring a network variable, right-click the network variable row and then clear the **Monitor** option on the shortcut menu. To disable monitoring for all network variables, right-click any network variable and then click **Monitor All Off** on the shortcut menu, or click the **Monitor All Off** button (✋) on the browser toolbar.

Getting Network Variable Values

You can get the value of a network variable manually without using polling. Right-click a network variable row and then click **Get Value** on the shortcut menu, or click anywhere in the network variable row and click the **Get Value** button (⇅) on the browser toolbar.

You can update all network variable values by clicking the **Refresh All** button (⇦) on the browser toolbar or opening the **Browse** menu and selecting **Refresh All**.

Using Bound Updates

You can use bound updates with the OpenLNS CT Browser to reduce the network overhead of polling many network variables. Bound updates use a connection between the network variable being monitored and the browser. To enable bound updates for a network variable, right-click a network variable row, click **Properties** on the shortcut menu, click the **Monitoring** tab and then select the **Bind this Network Variable to the Browser** check box. This creates an *implicit connection* to the browser. An implicit connection is a network variable connection that is automatically created by the browser. You can also explicitly create bound connections to the OpenLNS CT browser which are in turn automatically used by the browser instead of polling as described in *Binding Network Variables to the Host*.

Each implicit connection requires a network variable on the OpenLNS CT computer, and consumes network variable binding resources on the sending device or devices. This creates unnecessary network variables and consumes unnecessary resources if you are using bound monitoring with many network variables of the same type. To reduce the number of network variables on your computer and reduce network variable binding resource consumption, you can use one of the following methods:

- Use polled monitoring by clearing the **Bind this Network Variable to the Browser** check box.
- Explicitly bind the network variable to the host as described in *Binding Network Variables to the Host.*
- Use aliases on the devices that are bound to the fan-in connection.
- Define more aliases on the device containing the network variable with the fan-in connection.

Binding Network Variables to the Host

You can bind network variables to the OpenLNS CT host to receive event-driven updates for monitoring. You can use event-driven updates to minimize the monitoring network overhead associated with inputs that change infrequently such as digital sensors. Event-driven updates may generate excessive network traffic if the updates are frequent.

You can set up event-driven updates using the OpenLNS CT Browser, as described in *Using Bound Updates*. The method described there implicitly creates a new network variable on the OpenLNS CT host for each network variable to be monitored using bound monitoring. If a network variable or connection cannot be created, bound connections are not used and no error message is displayed

To optimize host network variable usage, you can bind many network variables into a single host network variable. This is called a *fan-in connection*. For example, in a system containing several thousand devices, each of which has an output network variable that indicates its alarm state, separate connections between each of these network variables and the host would consume a significant amount of system resources. It is more efficient to create just one network variable on the host and bind all of the output network variables to it. Network variables bound to the host may be monitored by an OpenLNS application such as the LNS DDE Server. OpenLNS allows the individual network variables to be monitored, even though they are received on the same network variable input on the host. Therefore, there is no disadvantage to using a fan-in connection. To differentiate between the different inputs, OpenLNS applications should monitor the network variables on the remote devices instead of the network variables created on the OpenLNS CT computer. OpenLNS automatically uses event-driven updates if a connection to the OpenLNS CT computer exists.

To create a network variable on the host, follow these steps:

1. Add a virtual functional block to your OpenLNS CT drawing.

    a. Drag the generic functional block SmartShape from the **OpenLNS CT Basic Shapes** stencil to the OpenLNS CT drawing. The Functional Block Wizard opens

    b. Under the **Device** box, select **OpenLNS Network Interface** from the **Name** list.

    c. Under the **Functional Block** box, select **Virtual Functional Block** from the **Name** list.

d. Enter the name of the Virtual Functional Block in the **New FB Name** box.

2. Drag an Input Network Variable SmartShape from the OpenLNS CT Basic Stencil to the virtual functional block. The **Choose a Network Variable** dialog opens.

3. Click **Create NV**. Create a dynamic network variable, as described in A*dding Network Variables to a Functional Block* in Chapter 4, *Designing Networks*. Select a network variable type that is compatible with the network variable you are binding to your compute.

4. Connect the input dynamic network variable you added to the virtual functional block to the output network variable that you want your computer to monitor.

Updating Network Variable and Configuration Property Values

You can control the values of input network variables and writeable configuration properties for the devices in your network. You cannot change output network variable values and read-only configuration properties.

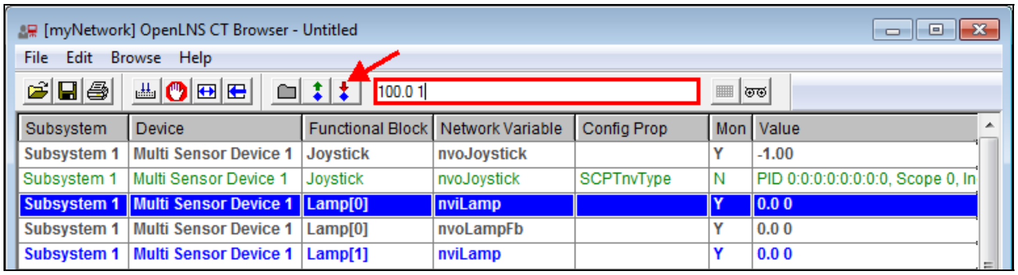
Setting Values

To set the value of an input network variable or writeable configuration property, follow these steps:

1. Click anywhere in the row containing the input network variable (colored blue) or writeable configuration property (green).

2. Enter a valid value in the **Value** box on the Browser toolbar, or choose a value from the drop-down list if you are changing an enumerated value.

3. Click the **Set Value** button (↕) on the Browser toolbar, or right-click the object you are changing and then click **Set Value** on the shortcut menu. The value is updated in the **Value** column. If you have entered an invalid value, an error message is displayed in a **Display Message** dialog.



100.0 1

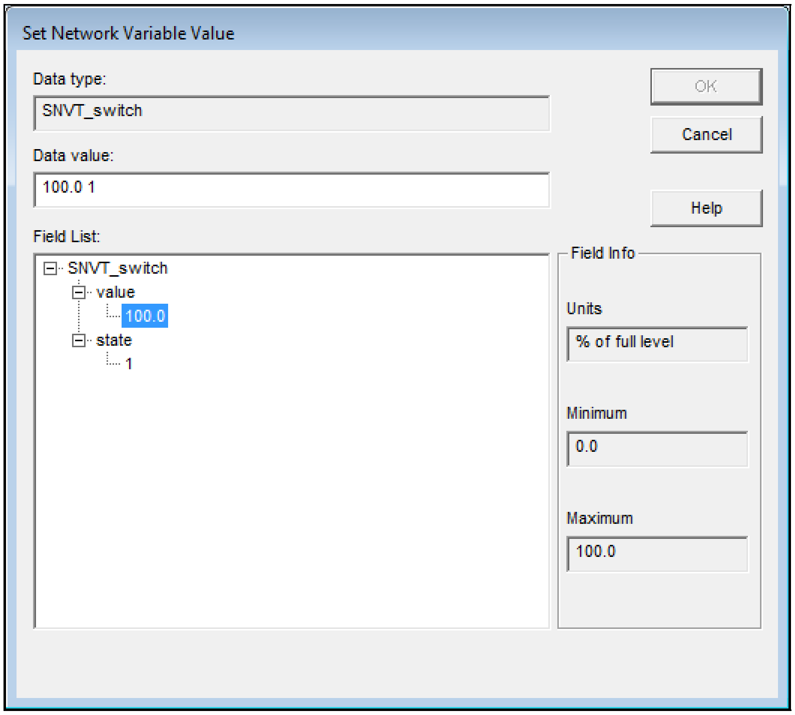
Subsystem	Device	Functional Block	Network Variable	Config Prop	Mon	Value
Subsystem 1	Multi Sensor Device 1	Joystick	nvoJoystick		Y	-1.00
Subsystem 1	Multi Sensor Device 1	Joystick	nvoJoystick	SCPTnvType	N	PID 0:0:0:0:0:0:0:0, Scope 0, In
Subsystem 1	Multi Sensor Device 1	Lamp[0]	nviLamp		Y	0.0 0
Subsystem 1	Multi Sensor Device 1	Lamp[0]	nvoLampFb		Y	0.0 0
Subsystem 1	Multi Sensor Device 1	Lamp[1]	nviLamp		Y	0.0 0

Setting Values for Structured Objects

You can use the **Set Network Variable/Set Configuration Property** dialog to set the value of one or more of the fields of a structured input network variable or writeable configuration property (structured means that the object has multiple fields).

To open the **Set Network Variable/Set Configuration Property** dialog, right-click the object and click **Details** on the shortcut menu.

This dialog displays the name, units, and maximum and minimum values for each of the fields. For example, a network variable with a **SNVT_switch** data type opens in the **Value** field in the Browser with a value of **100.0 1**. If you are not familiar with this network variable and try to set it in the **Value** box, you may enter an invalid value or enter a value that causes a device to behave in an undesired manner. If you open the **Set Network Variable Property** dialog box, you can see that this network variable has **value** and **state** fields and view the minimum and maximum values for each of these fields.



Set Network Variable Value

Data type: SNVT_switch

Data value: 100.0 1

Field List:

```
SNVT_switch
  value
    100.0
  state
    1
```

Field Info

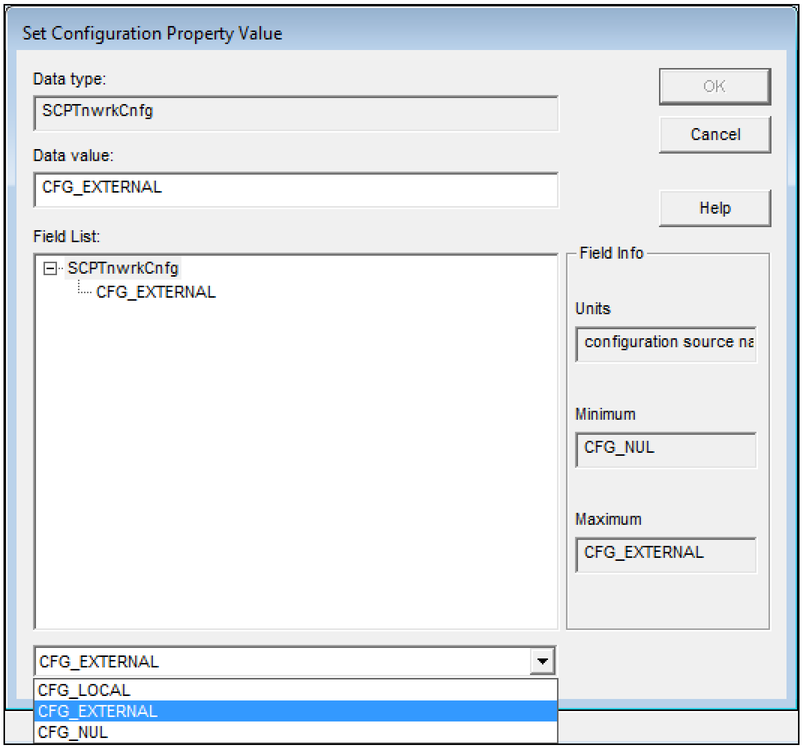
Units: % of full level

Minimum: 0.0

Maximum: 100.0

All the fields for the selected object are listed in structural order in the **Field List** box. To set the value for a field, expand that field and then click twice slowly on the value that opens below it. Enter a valid value for the field (the minimum and maximum values appear in the **Field Info** box to the right). Click **OK** to apply the changes.

If a network variable or configuration property has an enumerated type such as **SNVT_hvac_mode** or **SCPTnwrkCnfg**, a list of all the possible enumerations for the selected object opens the bottom of the dialog. Select the desired enumeration from the list and then click **OK**. Alternatively, you can click twice slowly on the value that opens below the object in the **Field List** box and enter a valid enumeration for the object.



Data type: SCPTnwrkCnfg

Data value: CFG_EXTERNAL

Field List:

- SCPTnwrkCnfg
  - CFG_EXTERNAL

Units: configuration source na

Minimum: CFG_NUL

Maximum: CFG_EXTERNAL

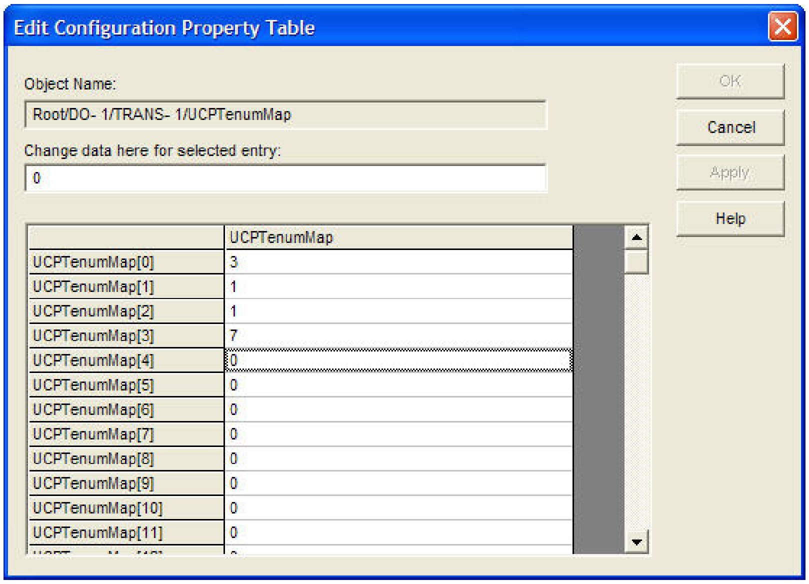
CFG_EXTERNAL

- CFG_LOCAL
- CFG_EXTERNAL
- CFG_NUL

Setting Values for Configuration Property Arrays

You can view and set values for a configuration property array using a table instead of using the **Value** box or the **Set Configuration Properties** dialog. To view a configuration property array, right-click a configuration property array and then click **Table** on the shortcut menu. The **Edit Configuration Property Table** dialog opens.

This dialog contains a table listing the values for each element in the configuration property array. To change a value, double-click the value of the element you want to change, enter the new value in the **Change Data Here for Selected Entry** box, or for enumerated types, select the desired value from the **Change Data Here for Selected Entry** list, and then click **Apply** to make the change or click **OK** to make the change and exit the dialog.



Object Name: Root/DO- 1/TRANS- 1/UCPTenumMap

Change data here for selected entry: 0

	UCPTenumMap
UCPTenumMap[0]	3
UCPTenumMap[1]	1
UCPTenumMap[2]	1
UCPTenumMap[3]	7
UCPTenumMap[4]	0
UCPTenumMap[5]	0
UCPTenumMap[6]	0
UCPTenumMap[7]	0
UCPTenumMap[8]	0
UCPTenumMap[9]	0
UCPTenumMap[10]	0
UCPTenumMap[11]	0

Clearing Values

You can clear the displayed value of a network variable or configuration property by right-clicking the object you want to clear and then clicking **Clear Value** on the shortcut menu.

To clear all values in the browser display, open the **Browse** menu and select **Clear All Values**. Clearing the values in the browser display does not affect the values in the functional block.

Getting Values

You can get the current value of a network variable or configuration property. To do this, right-click the network variable or configuration property and then click **Get Value** on the shortcut menu, or click the object and then click the **Get Value** button (↕) on the browser toolbar. The new value opens in the **Value** column.

To get the current values for all the network variables and configuration properties in the browser window, open the **Browse** menu and select **Refresh All**, or click the **Refresh All** button (⬅) on the browser toolbar. The new values are displayed in the **Value** column.

Changing a Network Variable or Configuration Property Type

You can change the type of a network variable or configuration property. The network variable or configuration property type specifies the units and structure of the data contained within the network variable or configuration property. A network variable or configuration property type is typically fixed for most devices; however, devices may support network variables and configuration properties with changeable types as described by the *LONMARK Application Layer Interoperability Guidelines.*

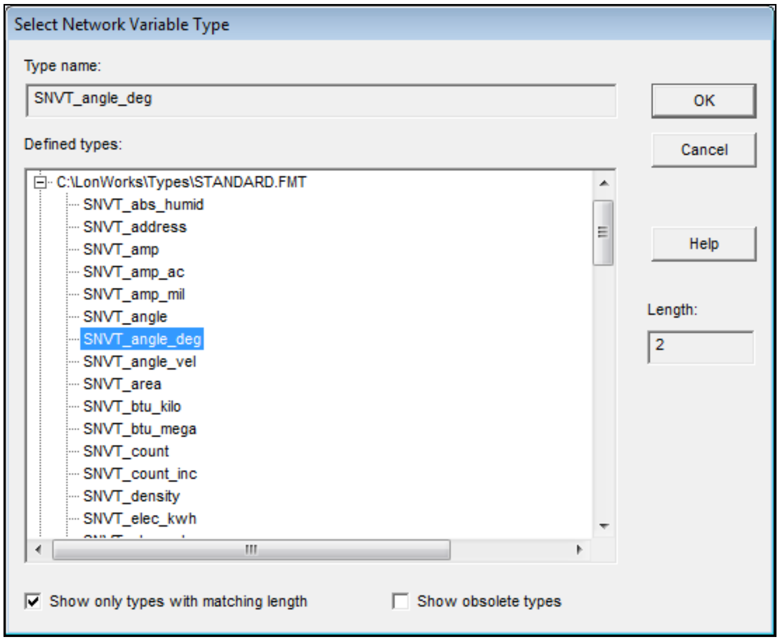
You can use the OpenLNS CT Browser to change a network variable type. When you change the network variable type, the browser updates the type in the OpenLNS network database, updates the type in the physical device if OpenLNS CT is OnNet, and automatically chooses a format for displaying the network variable data that matches the type, provided one is available. When you change a network variable type, the Browser also changes the type of any configuration properties that inherit their type from the network variable. For example, an analog input device may be connected to a variety of input devices, such as temperature sensors, pressure sensors, or dials. If the input device measures temperature, a **SNVT_temp_f** type may be used for its output network variable to define the value as a temperature. If the device measures air pressure, a **SNVT_press_f** may be used for its output network variable to define the value as a pressure. The same device may have **SCPTlowLimit1** and **SCPThighLimit1** configuration properties that inherit their types from the output network variable. The browser will change the configuration property type for these SCPTs when you change the output network variable type.

Not all network variables have changeable types, so check the documentation of the device and functional block you are browsing to determine whether the Browser will allow you to change the type of a specific network variable.

Network variables with a changeable type will always have an associated **SCPTnvType** configuration property (in the case of some Echelon devices, a **UCPTnvTypeInfo** configuration property defined in the Echelon resource file set) that keep track of their type. When you change a network variable type using the OpenLNS CT Browser, OpenLNS CT updates this configuration property.

To change a network variable type in the OpenLNS CT Browser, right-click the network variable and then click **Change Type** on the shortcut menu. This menu item is only available for network variables with changeable types. The **Select Network Variable Type** dialog opens.

You cannot change a network variable type if the network variable is connected or being monitored by OpenLNS CT. Trying to change the type of a network variable that is connected or being monitored by OpenLNS CT generates an error in the Browser.


Select Network Variable Type

Type name: SNVT_angle_deg

Defined types:

C:\LonWorks\Types\STANDARD.FMT
- SNVT_abs_humid
- SNVT_address
- SNVT_amp
- SNVT_amp_ac
- SNVT_amp_mil
- SNVT_angle
- SNVT_angle_deg
- SNVT_angle_vel
- SNVT_area
- SNVT_btu_kilo
- SNVT_btu_mega
- SNVT_count
- SNVT_count_inc
- SNVT_density
- SNVT_elec_kwh

Length: 2

☑ Show only types with matching length

□ Show obsolete types

This dialog lists built-in data types, and resource files containing defined network variable types (SNVTs and UNVTs). Expand the items listed to view the available data types. Click the type for the network variable and then click **OK**.

- Select the **Show Only Types with Matching Length** check box to display only those types with the same size as the current network variable type.
- Select the **Show Obsolete Types** check box to display network variable types that are marked obsolete in the current standard resource file set.
- The **Length** box displays the size (in bytes) of the currently selected type. You can customize the sizes for the **RAW**, **RAW_HEX**, **RAW_HEX_PACKED**, and **STRING** built-in data types and for the **SNVT_scene_cfg** configuration property.

Notes:

- Resource files for both the LONMARK standard types and Echelon user-defined types are included with OpenLNS CT. Device manufactures may also supply their own resource files for their user-defined types and formats. These are typically installed with the manufacturer's plug-in.
- For more information on specific standard network variable types (SNVTs) and standard configuration property types (SCPTs), see *types.lonmark.org*.
- For more information on user network variable types (UNVTs) and user configuration property types (UCPTs), see the documentation included with the functional blocks containing the network variable or configuration property. For information on Echelon's UNVTs and UCPTs, see *types.echelon.com*. For information on creating and using user types, see the *NodeBuilder Resource Editor User's Guide*.

Changing a Network Variable or Configuration Property Format

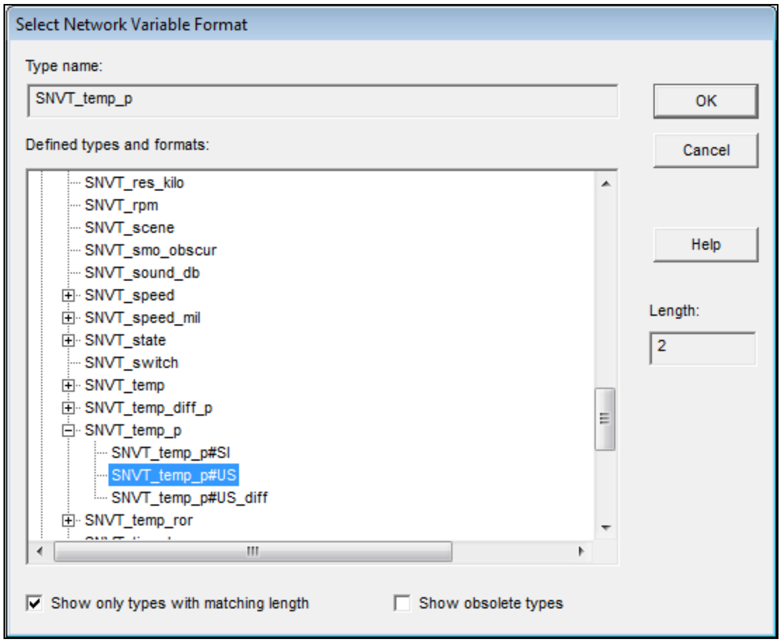
You can change the format of a network variable or configuration property. A network variable or configuration property format specifies how the raw data contained within a network variable or configuration property is translated for display or for use by an application. The format also specifies how data entered by you or an application is translated to the raw data to be transmitted on a LONWORKS network.

The format may include scaling and offset values to convert one type of data to another such as Celsius to Fahrenheit or kilograms to pounds, localized formatting specifications, or text strings that are to be displayed as part of the value. For example, a temperature sensor may report a temperature value with a type of **SNVT_temp_f**. The **SNVT_temp_f** type is defined as a 32-bit signed floating-point value representing a Celsius temperature. You can change the format of **SNVT_temp_f** type to Fahrenheit or differential Fahrenheit (the Fahrenheit format has an offset of 32; differential Fahrenheit has an offset of 0). You can modify network variable and configuration property formats or create your own using the NodeBuilder Resource Editor.

If you change a network variable type, the format is automatically changed to the default format for the new type. You can change the format to a format associated with a different type than the actual network variable type. For example, a network variable may have a fixed type of **SNVT_count_f**, but you may set its format to **SNVT_temp_f#SI** to interpret the count value as a Celsius temperature.

A network variable or configuration property format does not affect how the corresponding value is encoded for transmission on the network. For example, a **SNVT_temp_f** value is always transmitted in Celsius format, but it can appear in Celsius, Fahrenheit, or differential Fahrenheit formats.

You can use the OpenLNS CT Browser to change the format of network variables and configuration properties. To do this, right-click the network variable or configuration property in the browser window and then click **Change Format** on the shortcut menu. The **Select Network Variable Format** dialog opens.



Select Network Variable Format

Type name: SNVT_temp_p

Defined types and formats:

SNVT_temp_p#US

Length: 2

☑ Show only types with matching length

□ Show obsolete types

This dialog lists available types and formats. Each format is associated with a type. If a type supports multiple formats, you can expand the type to see all the associated formats. By default, the dialog opens to the current type with its formats listed underneath it (if it has more than one) and the current format highlighted.

To change the format, click the desired format and then click **OK**. In addition to the formats contained in the standard format file and other format files included in this dialog, you may choose to format the data in one of five built-in data types: **RAW**, **RAW_HEX**, **RAW_PACKED**, **STRING**, or **UNIT_TEXT**. To choose one of these formats, expand the **Built-in Data Types** header and select the desired format.

- Select the **Show Only Types with Matching Length** check box to list only those network variable types with the same size as the current SNVT, UNVT, or built-in data type.

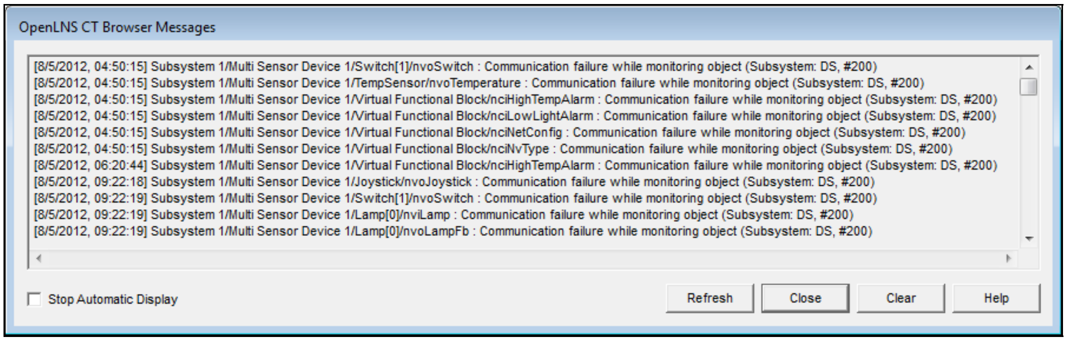
- Select the **Show Obsolete Types** check box to list those network variable types that are marked obsolete in the current standard resource file set.
- The **Length** box displays the size (in bytes) of the currently selected type.

Tip: Many standard network variable types include formats with "**#US**" and "**#SI**" format specifiers. You can use these specifiers to identify default formats when the appropriate default is determined by the regional settings on the computer running OpenLNS CT. OpenLNS CT uses the **#US** formats when your Windows regional settings specify a US measurement system, otherwise OpenLNS CT uses the #SI formats. To view the measurement system used on your computer, follow these steps:

1. Open the Windows Control Panel.
2. Double-click **Regional and Language Settings**. The **Regional and Language Settings** dialog opens. Click **Customize**. The **Customize Regional Settings** dialog opens.
3. The **Measurement System** box displays the units that will be used (US or Metric).

Displaying Error Messages

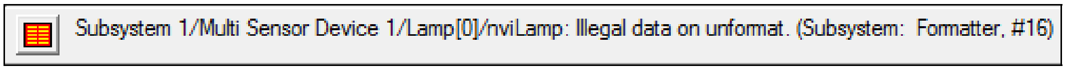
You can display error messages when monitoring and updating network variable and configuration property values in the OpenLNS CT Browser. By default, the **OpenLNS CT Browser Messages** dialog automatically opens when an error occurs.



```
[8/5/2012, 04:50:15] Subsystem 1/Multi Sensor Device 1/Switch[1]/nvoSwitch : Communication failure while monitoring object (Subsystem: DS, #200)
[8/5/2012, 04:50:15] Subsystem 1/Multi Sensor Device 1/TempSensor/nvoTemperature : Communication failure while monitoring object (Subsystem: DS, #200)
[8/5/2012, 04:50:15] Subsystem 1/Multi Sensor Device 1/Virtual Functional Block/nciHighTempAlarm : Communication failure while monitoring object (Subsystem: DS, #200)
[8/5/2012, 04:50:15] Subsystem 1/Multi Sensor Device 1/Virtual Functional Block/nciLowLightAlarm : Communication failure while monitoring object (Subsystem: DS, #200)
[8/5/2012, 04:50:15] Subsystem 1/Multi Sensor Device 1/Virtual Functional Block/nciNetConfig : Communication failure while monitoring object (Subsystem: DS, #200)
[8/5/2012, 04:50:15] Subsystem 1/Multi Sensor Device 1/Virtual Functional Block/nciNvType : Communication failure while monitoring object (Subsystem: DS, #200)
[8/5/2012, 06:20:44] Subsystem 1/Multi Sensor Device 1/Virtual Functional Block/nciHighTempAlarm : Communication failure while monitoring object (Subsystem: DS, #200)
[8/5/2012, 09:22:18] Subsystem 1/Multi Sensor Device 1/Joystick/nvoJoystick : Communication failure while monitoring object (Subsystem: DS, #200)
[8/5/2012, 09:22:19] Subsystem 1/Multi Sensor Device 1/Switch[1]/nvoSwitch : Communication failure while monitoring object (Subsystem: DS, #200)
[8/5/2012, 09:22:19] Subsystem 1/Multi Sensor Device 1/Lamp[0]/nviLamp : Communication failure while monitoring object (Subsystem: DS, #200)
[8/5/2012, 09:22:19] Subsystem 1/Multi Sensor Device 1/Lamp[0]/nvoLampFb : Communication failure while monitoring object (Subsystem: DS, #200)
```

□ Stop Automatic Display

This dialog displays all browser errors. Select the **Stop Automatic Display** check box and click **Close** to stop automatically displaying this dialog. If you have disabled automatic display and an error occurs, the **Display Messages** button at the bottom left corner of the Browser window turns red and a description of the error opens to the right of the button.



```
Subsystem 1/Multi Sensor Device 1/Lamp[0]/nviLamp: Illegal data on unfomat. (Subsystem:  Formatter, #16)
```

To delete all the current messages listed in the dialog, click **Clear**. All error messages are cleared from the window and the Display Message button is unavailable if you close the cleared dialog.

You can disable or enable automatic display of the OpenLNS CT Browser Messages dialog by opening the **Browse** menu, clicking **Options**, and under the **General** tab, clearing or selecting the **Error Messages: Show Always** check box.

If there are no error or warning messages, the **Displays Messages** command in the **Browse** menu and the **Display Messages** button are unavailable.

Managing Functional Blocks

You can disable, override, and test a functional block from the OpenLNS CT Browser. To do this, open the **Browse** menu and then click **Manage**. See *Managing Functional Blocks* in Chapter 8, *Managing Networks*, for more information.

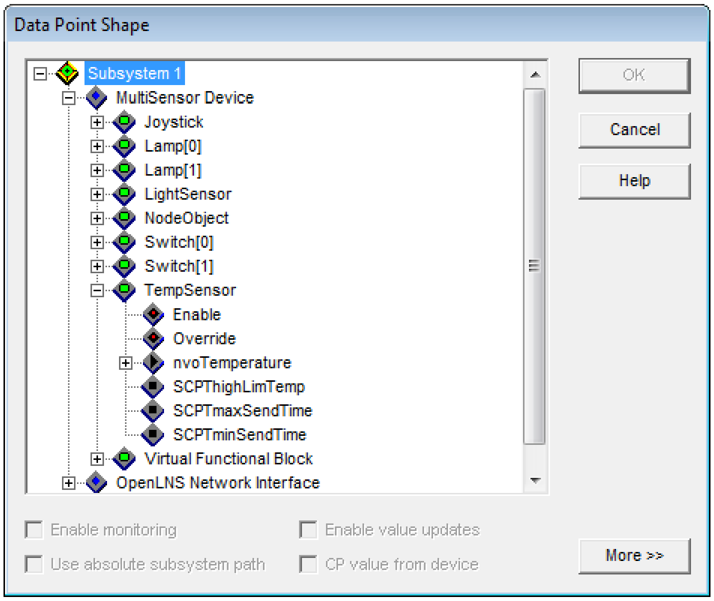
Using Data Point SmartShapes

You can use a Data Point SmartShape in an OpenLNS CT drawing to monitor the value of any network variable configuration property, or functional block state (enabled or in override). You can also use a Data Point SmartShape to update the value of an input network variable or configuration property. You can use Data Point SmartShapes to create simple human-machine interface (HMI) applications within your OpenLNS CT drawing.

Adding and Monitoring a Data Point SmartShape

You can add a Data Point SmartShape to your OpenLNS CT drawing, and use it to monitor a network variable configuration property, or functional block state. To add a Data Point SmartShape to your OpenLNS CT drawing, follow these steps:

1. Drag a Data Point SmartShape (▭) from the **OpenLNS CT Basic Shapes** stencil to your OpenLNS CT drawing. The **Data Point Shape** dialog box opens.



   This dialog provides a hierarchal view of your OpenLNS CT network, listing the objects in your network in the order of subsystems, devices, functional blocks, functional block states, network variables, and then configuration properties.

2. Select the network variable, configuration property, or functional block state to monitor by browsing through the network hierarchy. You can also right-click a subsystem and click **Expand** from the network navigator to display all the objects in that subsystem, including any nested subsystems and their objects.

3. Optionally, you can select the following basic attributes for the selected data point.

   Enable Monitoring

   Enables monitoring for the selected data point. Creates a monitor set and monitor point for a network variable data point if one is selected. All network variable Data Point SmartShapes and monitored network variable connections in the same subsystem are placed in one OpenLNS monitor set.

   This check box is cleared by default.

   Use Absolute Subsystem Path

   Controls how OpenLNS CT interprets a new Data Point SmartShape created from a copy (an existing Data Point SmartShape or a master SmartShape). This check box is cleared by default.

If this check box is selected, the same data point will be monitored in a new Data Point SmartShape.

If this check box is cleared, OpenLNS CT will attempt to locate the appropriate data point using the relative *subsystem\device\functional block* path of the source. In addition, you can create a Data Point SmartShape that you can copy to a different subsystem and have it refer to a data point in that subsystem.

Enable Value Updates

Allows you to enter a new value for the selected data point, which will then be updated in the OpenLNS network database. This check box is cleared by default.

If the selected data point is an input network variable, a device-specific configuration property, or a functional block state, the device will be updated in addition to the OpenLNS network database. If the selected data point is a non–device-specific configuration property, only the OpenLNS network database will be updated. This option is unavailable for output network variables.

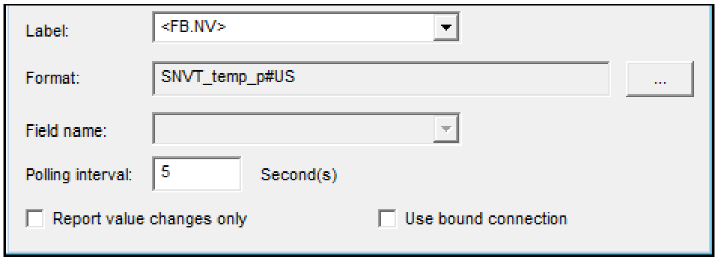
If this check box is cleared, you can double-click the Data Point SmartShape to update it with the current value of the selected data point.

CP Value from Device

Enables configuration property values to be updated and queried directly to and from the device. This option is only available if the selected data point is a configuration property.

This check box is selected by default for device-specific configuration properties, and it is cleared for non–device-specific configuration properties.

4. To set advanced options, click **More**. The dialog expands to include the following advanced properties for the selected data point.



Label: <FB.NV>

Format: SNVT_temp_p#US

Field name:

Polling interval: 5 Second(s)

□ Report value changes only

□ Use bound connection

Label

Selects the text displayed underneath the Data Point SmartShape. The options available depend on the selected data point. You can also enter your own label for the Data Point SmartShape. If you enter your own label and later create an instance of the Data Point SmartShape (a master SmartShape or a copy), that label will be selected by default in this list.

If the selected data point is a network variable, the following options are available:

- **None** (no label).
- **NV** (the network variable name).
- **FB.NV** (the functional block name, network variable name,

and field name [if defined]). This is the default.

- **Device.FB.NV** (the device name, functional block name, network variable name, and field name [if defined]).

If the selected data point is a functional block state, the following options are available:

- **None** (no label).
- **Status** (the status name [enable/override]).
- **FB.Status** (the functional block name and status name). This is the default.
- **Device.FB.Status** (the device name, functional block name, and status name).

If the selected data point is a functional block configuration property, the following options are available:

- **None** (no label).
- **CP** (the configuration property name).
- **FB.CP** (the functional block name and configuration property name). This is the default.
- **Device.FB.CP** (the device name, functional block name, and configuration property name).

If the selected data point is a functional block network variable, the following options are available:

- **None** (no label).
- **CP** (the configuration property name).
- **NV.CP** (the network variable name, configuration property name, and field name [if defined]).
- **FB.NV.CP** (the functional block name, network variable name, configuration property name, and field name [if defined]). This is the default.
- **Device.FB.NV.CP** (the device name, functional block name, network variable name, configuration property name, and field name [if defined]).

Format

Specifies the network variable or configuration property format for the value displayed in the Data Point SmartShape. By default, the OpenLNS default format is used. You can change the format by clicking the button to the right of the format field.

This option is unavailable if the selected data point is a functional block state.

Field Name

Specifies whether the Data Point SmartShape displays the entire structure of a network variable or configuration property value or just one or more fields within that structure.

For example, **SNVT_switch** has a value and a state field. If you select an input network variable with a **SNVT_switch** type as the data point, the switch is off, and you select **<all>** (the default) from the field name list, the Data Point SmartShape displays the entire structure, which would be "**0.0 0**". If you select **value**, the value displayed in the Data Point SmartShape will be "**0.0**", and if you select **state**, the value displayed will be "**0**".

This option is unavailable if the network variable or configuration property type does not have a structure, or the selected data point is a functional block state.

Polling Interval

Specifies how frequently (in seconds) the data point is polled. The default is **5** seconds for polled network variables and **15** seconds for monitored configuration properties and functional block states.

If you change the polling interval, the value you enter becomes the default for new Data Point SmartShapes of that object type until you close the drawing and then re-open it from the OpenLNS CT Design Manager. For example, if you change the polling interval for a network variable to 8 seconds, all new network variable data points that you create will be polled every 8 seconds. The default monitoring rate for functional block states and configuration properties, though, would not be affected.

Report Value Changes Only

Select whether a network variable Data Point SmartShape is updated anytime a poll (or update) occurs, or is only updated when the network variable value has changed. This check box is cleared by default, which means that the network variable Data Point SmartShape is updated anytime the network variable is polled (or updated).

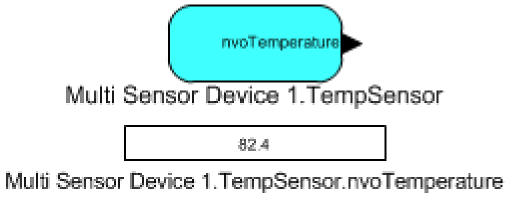
This option is only available if the selected data point is a network variable.

Use Bound Connection

Selects whether the OpenLNS server monitors a network variable data point by creating a connection from the OpenLNS Network Service Device (NSD) to the network variable, or by polling. This check box is cleared by default, which means the OpenLNS server uses polling.

This option is only available if the selected data point is an output network variable. If an input network variable is selected, the OpenLNS server automatically uses polling.

5. Click **OK**. The Data Point SmartShape is added to your OpenLNS CT drawing. The data point value opens in the Data Point SmartShape if monitoring is enabled in the OpenLNS CT drawing and in the Data Point SmartShape.


nvoTemperature
Multi Sensor Device 1.TempSensor
82.4
Multi Sensor Device 1.TempSensor.nvoTemperature

6. If no data point value opens in the Data Point SmartShape, verify that monitoring is enabled in both the network and in the Data Point SmartShape. Right-click the Data Point SmartShape and select **Monitor Value** on the shortcut menu if it is not already selected. If the data point value opens in brackets, right-click a clear space in the drawing page and select **Enable Monitoring** on the shortcut menu.

Updating a Data Point

You can update the value of a Data Point SmartShape that represents an input network variable, configuration property, or functional block state. A Data Point SmartShape may represent a scalar value, a structured value, or an enumerated value.

- A *scalar value* is a single number such as a temperature or a pressure.

- A *structured value* is a value that contains multiple fields such as a switch value with state and level fields.

- An *enumerated value* is a value with multiple states each represented by an enumeration string, such as an occupancy state with occupied and unoccupied values or an HVAC mode. A functional block state is also an enumerated value.

Updating a Scalar Data Point

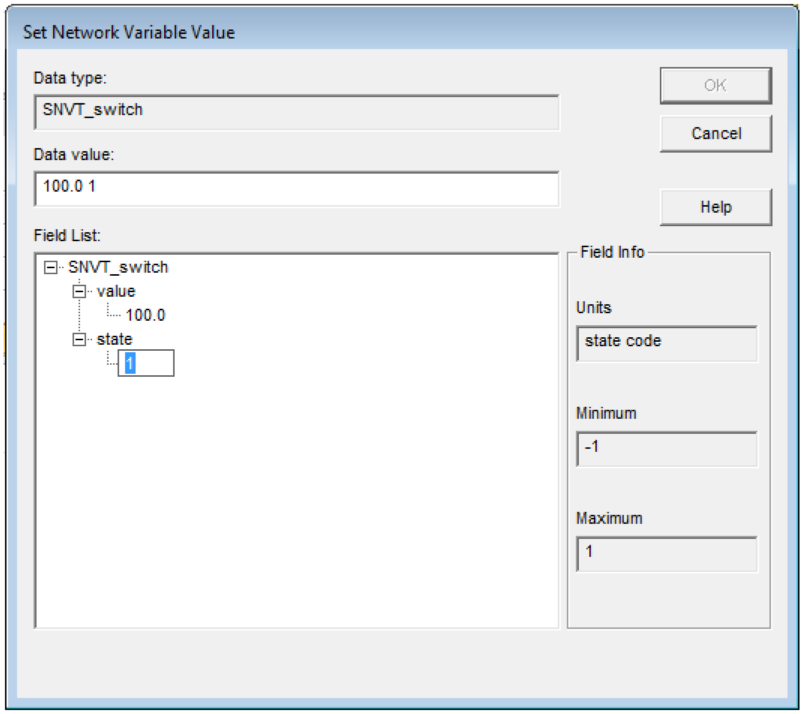
To set the value of a scalar data point, follow these steps:

1. If updates are not enabled, right-click the Data Point SmartShape, select **Properties** on the shortcut menu, and then select the **Enable Value Updates** check box in the **Data Point Shape** dialog. Click **OK**.
2. Double-click the Data Point SmartShape and then enter the desired value.

Updating a Structured Data Point

To set the value of an input network variable or configuration property data point that has a structure (a type with multiple fields), follow these steps:

1. If updates are not enabled, right-click the Data Point SmartShape, select **Properties** on the shortcut menu, and then select the **Enable Value Updates** check box in the **Data Point Shape** dialog. Click **OK**.
2. Double-click the Data Point SmartShape and then enter the desired value for one or more of the data point fields. Alternatively, you can use the **Set Network Variable Value** dialog to set the values for the fields. The benefit of using this dialog is that it displays the name, unit type, and maximum and minimum values for each of the fields, which makes it easier to enter a valid value. To use the **Set Network Variable Value** dialog, follow steps 3–5.
3. Right-click the Data Point SmartShape and click **Value Details** on the shortcut menu. The **Set Network Variable Value** dialog box opens.



Data type: SNVT_switch

Data value: 100.0 1

```
SNVT_switch
  value
    100.0
  state
    1
```

Units: state code

Minimum: -1

Maximum: 1

4. All the fields for the selected data point are listed in structural order in the **Field List** box. You can set the value for a structured data point in two ways:

- Expand a field in the **Field List** box and then click twice slowly on the value that opens below the field. Enter a valid value for the field (the minimum and maximum values appear in the **Field Info** box to the right). Repeat for any other fields for which you want to set the value.
- Enter a valid value in the **Data Value** box for one or more of the fields in the structured data point.

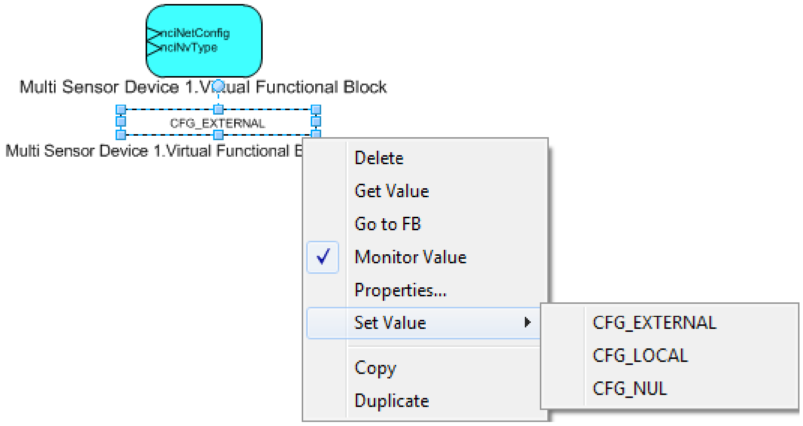
5. Click **OK** to apply the changes

Note: If you update an input network variable that is connected to an output network variable on another device, updates from the other device to the output network variable may overwrite your changes.

Updating an Enumerated Data Point

To set the value of an enumerated data point, follow these steps:

1. If OpenLNS CT SmartShape menus are not enabled, click **Add-ins**, click **OpenLNS CT**, and then select **Use OpenLNS CT Shape Menus**.
2. If updates are not enabled, right-click the Data Point SmartShape, select **Properties** on the shortcut menu, and then select the **Enable Value Updates** check box in the **Data Point Shape** dialog. Click **OK**.
3. Right-click the enumerated data point and point to **Set Value** on the shortcut menu. A list of all the possible enumerations for the data point opens.
4. Select the desired enumerated value.



Getting a Data Point Value

You can get the value of any data point by double-clicking the Data Point SmartShape or right-clicking the Data Point SmartShape and clicking **Get Value** on the shortcut menu. The Data Point SmartShape value is updated with the current value of the data point. OpenLNS CT must be attached to the network to use this option.

If monitoring on the Data Point SmartShape is disabled, the value displayed in the Data Point SmartShape will be cleared in *n* seconds, where *n* is the value set in the **'Get Value' Display Duration** box in the **Network Properties: NV Browser/Monitoring** options category. By default, the value is **15** seconds. You can disable the clearing of the Data Point SmartShape by setting this value to **0**. To set this value in the OpenLNS drawing, click **Add-ins**, click **OpenLNS CT**, click **OpenLNS CT Options**, then select **NV Browser/Monitoring** from the **Options Category** list. See *NV Browser/Monitoring Options* in Appendix A for more information on setting this value.

Creating and Using a Custom Data Point Master SmartShape

You can create a custom data point master SmartShape from an existing Data Point SmartShape, and you can include a Data Point SmartShape with other SmartShapes in a custom OpenLNS CT master SmartShape. All the properties of the Data Point SmartShape—including the selected data point, label, fonts, and colors—are saved in the custom master SmartShape. You can then re-use the custom data point master SmartShape to add new Data Point SmartShapes to your OpenLNS CT drawing.

Creating a Custom Data Point Master SmartShape

To create a custom functional block master SmartShape, drag the desired Data Point SmartShape or a set of SmartShapes that includes a Data Point SmartShape onto an OpenLNS CT stencil to which you have write access.

If the **Use Subsystem Absolute Path** check box in the **Data Point Shape** dialog box is selected when you create the custom data point master SmartShape, the monitored data point will be the same as the one in the original Data Point SmartShape when you drag the master SmartShape to your drawing.

If this option is cleared, OpenLNS CT will store a relative reference in the custom data point master SmartShape of the subsystem path from the location of the original (source) Data Point SmartShape to that of the monitored object. OpenLNS CT will then use this information to determine the data point to be monitored when you create a new Data Point SmartShape from the custom master SmartShape.

Using a Custom Data Point Master SmartShape

You can create a custom Data Point SmartShape by dragging a custom Data Point master SmartShape from an OpenLNS CT stencil to your drawing. The custom Data Point SmartShape has the same properties as the original Data Point SmartShape that you used to create the master SmartShape. To create a new Data Point SmartShape from a master SmartShape, follow these steps:

1. Drag the custom data point master SmartShape from its stencil to your OpenLNS CT drawing. The **Data Point Shape** dialog box opens with a default data point selected.
2. The default data point depends on whether the **Use Subsystem Absolute Path** option was selected in the original Data Point SmartShape. If this option was selected, the default data point is the same as the one in the original Data Point SmartShape. If this option was cleared, OpenLNS CT will try to identify the appropriate default data point based on the subsystem in which the new Data Point SmartShape was created and the relative path to the original Data Point SmartShape.
3. Accept the default data point or select a different data point to be monitored in the new Data Point SmartShape.
4. Click **OK**. The Data Point SmartShape is added to your OpenLNS CT drawing.

If you drag a custom OpenLNS CT master SmartShape that includes a Data Point SmartShape to your OpenLNS CT drawing, the new Data Point SmartShape will be created without the **Data Point Shape** dialog opening.

Note: If the **Use Subsystem Absolute Path** option was cleared, and you change the name of any of the objects (subsystem, device, functional block, or the data point object) referenced by the master SmartShape, and you drag the master SmartShape to a different subsystem, OpenLNS CT may not be able to determine the appropriate data point for the new Data Point SmartShape. If OpenLNS CT cannot determine the data point, the new Data Point SmartShape will be outlined red (the default style for Connection Unbound).

Example: A subsystem named “Subsystem 1,Building1.Floor1.Room1” contains a Data Point SmartShape, and the **Use Subsystem Absolute Path** check box for the Data Point SmartShape is cleared. The Data Point SmartShape references a data point on a functional block in the “Subsystem 1.Building1.Floor2.Room10” subsystem. The subsystem path stored in the Data Point SmartShape is therefore “^.^.Floor2.Room10”. This indicates that when the custom Data Point master SmartShape is added to a drawing, OpenLNS CT is to attempt to locate the monitored object by moving two subsystem levels up the network hierarchy and then down to the “Floor2.Room10” subsystem in the network.

A custom Data Point master SmartShape is created by dragging the original SmartShape to an OpenLNS CT stencil. If the custom Data Point master SmartShape is dragged to a “Subsystem 1.Building2.Floor1.Room1” subsystem, it will reference a data point in the “Subsystem 1.Building2.Floor2.Room10” subsystem.

Copying and Creating a Data Point SmartShape

You can create a new Data Point SmartShape by copying an existing one following these steps:

1. Copy an existing Data Point SmartShape. You can do this three different ways:
   - Right-click a Data Point SmartShape, click **Copy** on the shortcut menu, right-click an empty space in the desired subsystem, and click **Paste** on the shortcut menu.
   - Right-click a Data Point SmartShape and then click **Duplicate** on the shortcut menu.
   - Select a Data Point SmartShape, hold down CTRL, drag the Data Point SmartShape to any location in the subsystem, and then release the mouse.
2. The **Data Point Shape** dialog box opens with a default data point selected.
3. Accept the default data point or select a different data point to be monitored in the new Data Point SmartShape.
4. Click **OK**. The Data Point SmartShape is added to your OpenLNS CT drawing.

Note: If you change the name of the original data point and paste a new Data Point SmartShape in a different subsystem, OpenLNS CT will be able to determine the appropriate data point for the new Data Point SmartShape. This is because OpenLNS CT first checks the handles of the original data point when creating a new Data Point SmartShape from a copy.

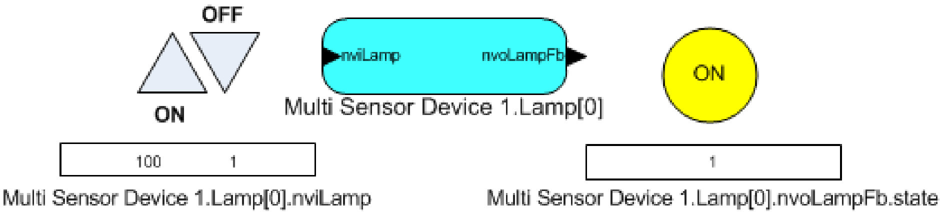
Deleting a Data Point SmartShape

You can delete a Data Point SmartShape from an OpenLNS CT drawing. To do this, follow these steps

1. Right-click the Data Point and then click **Delete** on the shortcut menu. If confirmed deletes are enabled, you will see a dialog asking you to confirm the deletion.
2. If prompted, click **Yes** to delete the SmartShape.

Creating an HMI with Data Point SmartShapes

You can use Data Point SmartShapes and Visio shapes to design simple human machine interfaces (HMIs) in your OpenLNS CT drawing. An *HMI* provides a graphical interface for monitoring and controlling a network. For example, you can create an HMI for a lighting system. The HMI could display the current state of a lamp, and it could include override switches that let you manually turn the lamp on and off.


OFF
ON
nviLamp
nvoLampFb
Multi Sensor Device 1.Lamp[0]
ON
100
1
1
Multi Sensor Device 1.Lamp[0].nviLamp
Multi Sensor Device 1.Lamp[0].nvoLampFb.state

You can use high-end HMI tools, such as Wonderware’s InTouch or Intellution FIX, to represent more complex types of network interactions. These tools are developed with a scripting language tuned to specifically address HMI tasks. In addition, these tools offer components that provide reporting and analysis, history, alarm logging, event handling, and Internet-enabling.

To create a simple HMI with Data Point and Visio SmartShapes, you do the following:

1. Add Data Point SmartShapes to your drawing as described in *Adding and Monitoring a Data Point SmartShape*. You use the Data Point SmartShapes to monitor and control data points.
2. Add Visio shapes to your drawing (you can use any Visio shape—the previous lighting HMI example uses shapes in the Visio Basic SmartShapes stencil). You use the Visio shapes to read and write to the Data Point SmartShapes.
3. Customize the ShapeSheets of the Data Point and Visio shapes so that they can interact with each other. You insert functions in the ShapeSheets that are executed when a supported event occurs (for example, when a Visio shape is clicked or the value in a Data Point SmartShape changes). You can easily create functions using Add-Ons, Macros, or both. For more information on customizing ShapeSheets, search for ShapeSheets in the online Visio Help or go to the Visio 2010 ShapeSheet Reference at *msdn.microsoft.com/en-us/library/office/ff768297.aspxh*.

The following two sections describe how to customize the ShapeSheets of the Data Point and Visio shapes using the previous lighting HMI example. The first section explains how to make a Visio shape write a value to a Data Point SmartShape; the second explains how to make a Visio shape read a value from the Data Point SmartShape.

Writing Data Point SmartShape Values

You can use a Visio shape to write a value to a Data Point SmartShape. You can write to a Data Point SmartShape using an Add-On or a Macro.

Using an Add-On to Write Values

To use an Add-On to have a Visio shape write a value to a Data Point SmartShape, follow these steps:

1. Enable the Visio **Developer** tab. To do this, click **File** and then click **Options**. In the **Visio Options** dialog, click the **Customize Ribbon** section. On the right-hand side, enable the **Developer** check box, and then click **OK**.
2. Click the Visio shape that will write a value to an input network variable.
3. Click the **Developer** tab, point to **Show ShapeSheet**, and then click **Shape** (alternatively, you can right-click the shape, and click **Show ShapeSheet** on the shortcut menu). The ShapeSheet for the selected Visio shape opens.
4. Click the **EventDblClick** cell in the **Events** section and enter a function in the **Formulas** box above the drawing diagram. The formula you use depends on whether you are writing to a data point or a field within a structured data point.

   - **Data Point Formula Syntax**

     RUNADDONWARGS("LMW Set Value","/SelOverride/ShapeOverride=<*Data Point Shape ID***> /Value= <***n***>")**

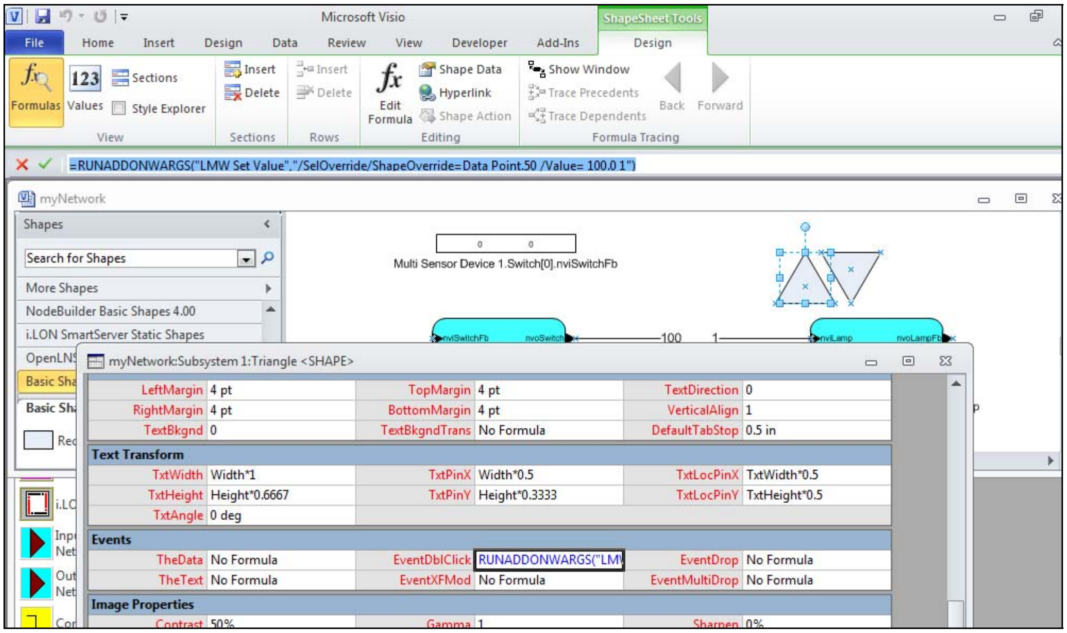
     - To get the ID of a Data Point SmartShape, click the Data Point SmartShape, click the **Developer** tab, and then click **Shape Name**. An index is appended to the Shape ID when you add a second SmartShape of a given type to your drawing. For example, the ID of the first Data Point SmartShape added in the lighting HMI example is **Data Point**; the ID of the second data point is **Data Point.50**.

       If the SmartShape has an index, you can use the following naming convention to enter the Data Point Shape ID: **Sheet.**<*index*>.
     - *n* is a valid value to be written to the Data Point SmartShape.

- **Example**

  The following example demonstrates a function that writes an "**ON**" value to a **SNVT_swtich** data point:

  RUNADDONWARGS("LMW Set Value","/SelOverride/ShapeOverride=DataPoint.50 /Value= 100.0 1")



=RUNADDONWARGS("LMW Set Value","/SelOverride/ShapeOverride=Data Point.50 /Value= 100.0 1")

LeftMargin	4 pt	TopMargin	4 pt	TextDirection	0
RightMargin	4 pt	BottomMargin	4 pt	VerticalAlign	1
TextBkgnd	0	TextBkgndTrans	No Formula	DefaultTabStop	0.5 in
Text Transform					
TxtWidth	Width*1	TxtPinX	Width*0.5	TxtLocPinX	TxtWidth*0.5
TxtHeight	Height*0.6667	TxtPinY	Height*0.3333	TxtLocPinY	TxtHeight*0.5
TxtAngle	0 deg				
Events					
TheData	No Formula	EventDblClick	RUNADDONWARGS("LM	EventDrop	No Formula
TheText	No Formula	EventXFMod	No Formula	EventMultiDrop	No Formula
Image Properties					
Contrast	50%	Gamma	1	Sharpen	0%

- **Data Point Field Formula Syntax**

  RUNADDONWARGS("LMW Set Field Value","/SelOverride/ShapeOverride=<*Data Point SmartShape ID***> /Value= <***n***>")**

- **Example**

  The following example demonstrates a function that writes to the **state** field of a **SNVT_swtich** data point:

  =RUNADDONWARGS("LMW Set Field Value","/SelOverride/ShapeOverride=DataPoint.50 /Value= 0")

5. Click the accept button (✓) in the **Formulas** box.
6. To have some logic performed on the data point value instead of specifying a value, replace *n* with **&Scratch.X**. The function is as follows: **RUNADDONWARGS("LMW Set [Field] Value", "/SelOverride /ShapeOverride=<***Data Point SmartShape ID***> /Value=" &Scratch.X)**. You can use this type of formula to incrementally change the value of a data point when you double-click the Visio shape.

   In the **X** cell of the **Scratch** section, enter a formula. For example, to create a formula that brightens a lamp when you double-click the Visio shape, enter the following: **=FORMAT(MODULUS("<***Data Point SmartShape ID***>!User.MonValue"+10,110), "0")**.

Using a Macro to Write Values

To write to a value to a Data Point SmartShape using a Macro, follow these steps:

1. Click the **Developer** tab, and then click **Macros**. The **Macros** dialog box opens.

2. In the **Macros Name:** dialog box, enter a name for the Macro. Click **Create**. The **Network Passwords** dialog opens if this is the first time you are creating a Macro or entering VBA code in your OpenLNS CT drawing.
3. Enter **enumerate** in the **Password** box.
4. The Visio VBA editor opens. The left pane of the editor displays the OpenLNS CT drawing; the right pane displays the VBA code associated with the Macro you are creating.
5. To insert a Macro to write to a scalar data point, enter the following VBA code:

   ```
   Set myaddon = Application.Addons("LMW Set Value")
   myaddon.Run ("/doc=<document ID> /page=<page ID> /SelOverride
   /SmartShapeOverride=<Data Point SmartShape ID> /Value= n")
   ```

   - *Document ID is* the document in which the Visio shape resides.
   - *Page ID* is the page in which the Visio shape resides.
6. To insert a Macro to write to a field in a structured data point, enter the following VBA code:

   ```
   Set myaddon = Application.Addons("LMW Set Field Value")
   myaddon.Run ("/doc=<document ID> /page=<page ID> /SelOverride
   /SmartShapeOverride=<Data Point Shape ID> /Value= n")
   ```
7. Enter additional code for any other functions to be performed when you double-click the Visio shape.
8. In the OpenLNS CT drawing, right-click the Visio shape, and then click **Show ShapeSheet** on the shortcut menu. The ShapeSheet of the selected Visio shape opens.
9. In the **EventDblClick** cell of the **Events** section, enter the following function: **RUNMACRO("<***Macro name***>")**, where *Macro Name* is the name of the Macro you entered in step 2.

Reading Data Point SmartShape Values

You can use a Visio shape to read a value in a Data Point SmartShape. You can read a Data Point SmartShape using an Add-On or a Macro. If you want to perform a function based on the data point value, create a Macro; otherwise, use an Add-On.

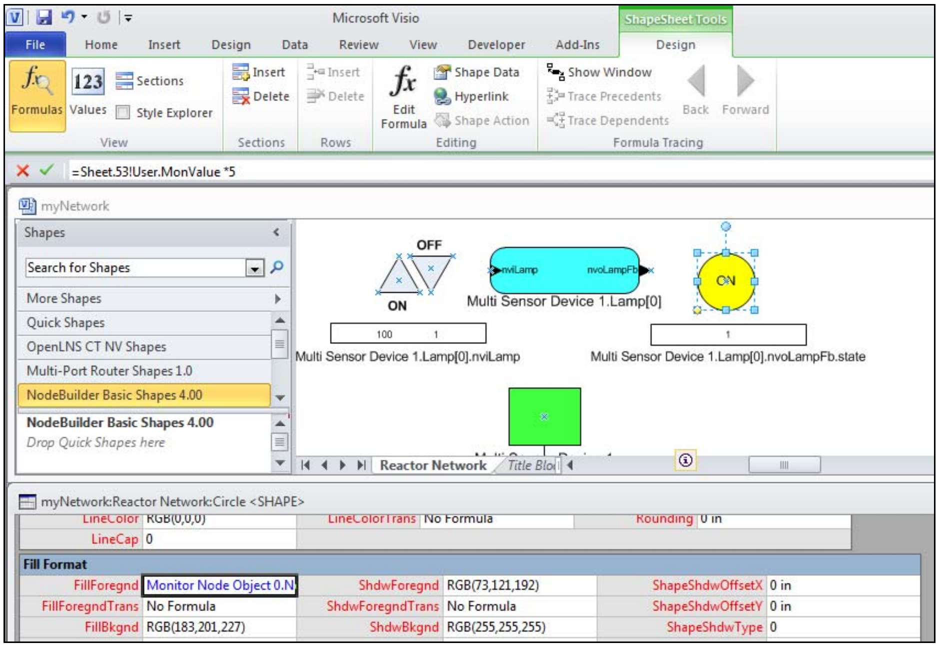
Using an Add-On to Read Values

To use an Add-On to have a Visio shape read from a value in a Data Point SmartShape, follow these steps:

1. Click a Visio shape that is reading a Data Point SmartShape monitoring an input or output network variable.
2. Right-click the Visio shape, and then click **Show ShapeSheet** on the shortcut menu. The ShapeSheet of the selected Visio shape opens.
3. Enter the following function the **FillForegnd** cell of the **FillFormat** section: **Sheet.<***Data Point SmartShape ID index***>!User.MonValue**. For a Visio shape that is reading the state of a lamp, the color of the Visio SmartShape will toggle between yellow (on) and white (off).

   Data Point SmartShape ID index refers to the numeric digits at the end of the Data Point SmartShape name. For example, if the name of Data Point SmartShape is **Data Point.53**, you will enter **Sheet.53!User.MonValue** (essentially, replace “Data Point” with “Sheet”).

   - To change the color for the “on” state, you can perform some arithmetic function on the Data Point SmartShape value. For example, to change the color of the Visio SmartShape to yellow when the lamp is on, enter the following function: **Sheet.<***Data Point SmartShape ID index***>!User.MonValue *5**.



=Sheet.53!User.MonValue *5

myNetwork:Reactor Network:Circle <SHAPE>

LineColor	RGB(0,0,0)	LineColorTrans	No Formula	Rounding	0 in
LineCap	0				
Fill Format					
FillForegnd	Monitor Node Object 0.N	ShdwForegnd	RGB(73,121,192)	ShapeShdwOffsetX	0 in
FillForegndTrans	No Formula	ShdwForegndTrans	No Formula	ShapeShdwOffsetY	0 in
FillBkgnd	RGB(183,201,227)	ShdwBkgnd	RGB(255,255,255)	ShapeShdwType	0

- If you are monitoring the lamp value, you can change the color of the Visio shape based on the brightness of the lamp. To do this, enter the following function: **Sheet.<***Data Point SmartShape ID index***>!User.MonValue/10**.

4. Click the accept button (✓) in the **Formulas** box.

Using a Macro to Read Values

To read from a Data Point SmartShape value using a Macro, follow these steps:

1. Click the **Developer** tab, and then click **Macros**. The **Macros** dialog box opens.

2. In the **Macros Name:** dialog box, enter a name for the Macro. Click **Create**. The **Network Passwords** dialog opens if this is the first time you are creating a Macro or entering VBA code in your OpenLNS CT drawing.

3. Enter **enumerate** in the **Password** box.

4. The Visio VBA editor opens.

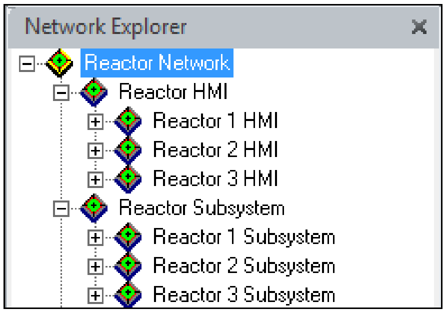
5. In the right pane, enter VBA code for the Visio shape to read from a data point. The following code is used to change the text in the Visio shape to "**ON**" and the color to yellow when the state of the lamp is on (1). If the state of the lamp is off (0), the text in the Visio shape changes to "**OFF**" and the color changes to black.

```
Sub LED
       Dim vsoPage As Visio.Page
       Dim myShape As Shape
       Dim myShape2 As Shape
       Set myShape = ActivePage.Shapes("Circle")
       Set myShape2 = ActivePage.Shapes("Data Point.50")

       If myShape2.Cells("User.MonValue") = 1 Then
              myShape.Cells("FillForegnd") = 5
              myShape.Text = "ON"
       Else
              myShape.Cells("FillForegnd") = 0
              myShape.Text = "OFF"
       End If

End Sub
```

6. In the OpenLNS CT drawing, right-click the Visio shape used to monitor the Data Point SmartShape, and then click **Show ShapeSheet** on the shortcut menu. The ShapeSheet of the selected Visio shape opens.
7. Enter the following function the **FillForegnd** cell of the **FillFormat** section: **RUNMACRO("<*Macro name*>")**, where *Macro Name* is the name of the Macro you entered in step 2.
8. Click the accept button (✓) in the **Formulas** box.

Organizing HMIs

When you create an HMI, place all the controls comprising the HMI in a single subsystem that is separate from the subsystem containing the devices and functional blocks it represents. If you have multiple subsystems that each have HMIs, establish a naming convention to correlate each HMI with the subsystem it monitor and controls. This makes it easier for you to use, modify, and maintain the network as you add more HMIs and subsystems to the network. For example, a reactor network could contain multiple reactor subsystems that each have their own HMIs. You would place the reactor subsystems in one subsystem, and the HMI subsystems in another. The following figure demonstrates how you can organize the reactor network using this naming convention.



7

Maintaining Networks

This chapter provides an overview of network maintenance tasks that you can perform with OpenLNS CT. It describes loading, replacing, and decommissioning devices. It explains how to resynchronize and propagate configuration properties values. It explains how to back up and restore an OpenLNS CT network; create an OpenLNS CT network by recovering information from the physical network; and resynchronize the network database, OpenLNS CT drawing, and physical devices. It explains how to merge two OpenLNS CT networks.

Maintaining Networks Overview

You can perform routine maintenance to update, repair, and optimize an installed network. You can maintain the network and its components, the OpenLNS CT drawing, and the OpenLNS network database. For example, if a device or router fails, you can replace it on the physical network and then logically replace it in the OpenLNS CT network, preserving the device's configuration and all of its connections.

This chapter describes the following device and network maintenance tasks:

- Loading devices.
- Replacing devices.
- Decommissioning devices.
- Resynchronizing device configuration properties.
- Propagating device configuration property values by using OpenLNS CT as a passive configuration tool.
- Backing up and restoring the network.
- Recovering the network.
- Synchronizing the network.
- Merging networks.

Tip: You can perform device maintenance tasks on multiple devices at one time. The manner in which you select multiple devices depends on the desired network scope of the task.

- To select multiple devices within a subsystem, hold down CTRL and click the devices, or click an empty space in the drawing page and drag a selection net around them. Then right-click one of the selected devices, point to **Commissioning** on the shortcut menu, and select the desired maintenance task.
- To select all the devices within a subsystem, select or clear the **Exclude Nested Subsystems from Subsystem Operations** check box to exclude or include, respectively, the subsystems nested within the selected subsystem (to access this option, right-click the Subsystem SmartShape, or an empty space in the network if you are in the top-level subsystem, and select **Subsystem Properties** from the shortcut menu). Then right-click the Subsystem SmartShape, or an empty space in the drawing page if you are in the top-level subsystem, point to **Commissioning** on the shortcut menu, and select the desired maintenance task.
- To select all the devices within a network, right-click an empty space in the drawing page of the top-level subsystem, point to **Commissioning** on the shortcut menu, and then select the desired maintenance task.

Loading Devices

You can load a new application image, and optionally a new Neuron firmware image, into an application device.

You can load an application with a device template that is the same as or different from the original device. When you load an application, you can create a new device template from a device interface (XIF) file on the OpenLNS Server or a XIF file in the device. If you load an application with a new or different device template, the new application must use a different program ID than the original application.

If you load a compatible application image into a device that you have configured and connected to other devices, OpenLNS CT will preserve all compatible configuration and connection information. You can load an application image file with an .apb or .nxe extension.

To load a new application into a device, the devices must have writeable application memory, and OpenLNS CT must be attached to the network and OnNet.

To load a new application into a device, follow the five steps described in the subsequent sections:

1. Select the device or devices to upgrade.
2. Select or create a device template.
3. Select an application image to load into the device(s).
4. Select the initial state of the device(s), source of configuration property values, and management of device-specific configuration properties.
5. Install the device(s), if necessary.

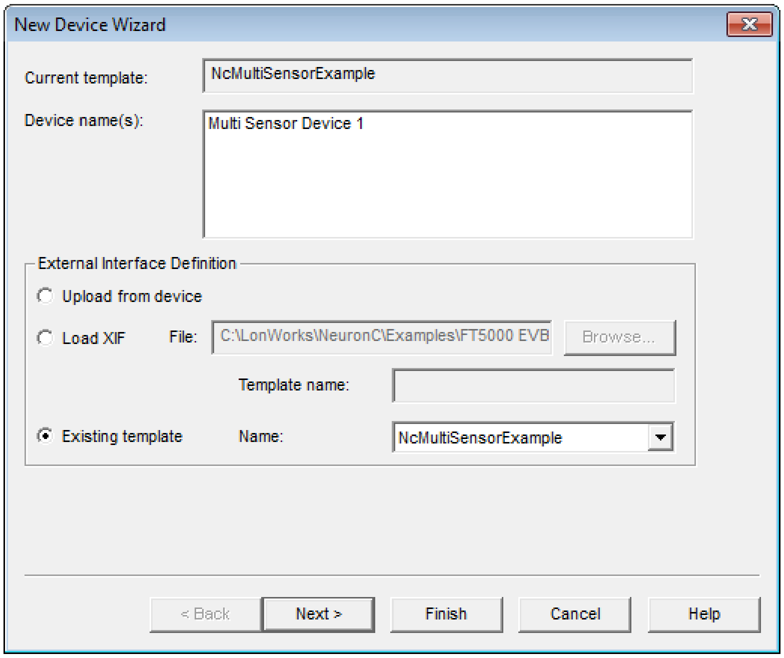
Selecting the Devices to Load

To select devices to load, follow these steps:

1. Select one or more applications devices to load.
2. Right-click the device to load (or one of the devices if you are loading multiple devices), point to **Commissioning**, and then click **Load** on the shortcut menu.
3. The Application Image Wizard opens. The device template page lists all the selected devices associated with the device template displayed in the **Current Template** box.

Selecting or Creating a Device Template

To select or create a device template, follow these steps:



Current template: NcMultiSensorExample

Device name(s): Multi Sensor Device 1

External Interface Definition

○ Upload from device

○ Load XIF File: C:\LonWorks\NeuronC\Examples\FT5000 EVB

Template name:

● Existing template Name: NcMultiSensorExample

1. Choose a method for selecting or creating a device template. You have three choices:
   - **Upload From Device**. Creates a device template by uploading the device interface (XIF) from the device. This option is only available if OpenLNS CT is attached to the network. Enter a name for the new device template in the **Template Name** box. The name must be unique to this network and may be up to 85 characters long.
   - **Load XIF**. Creates a device template by reading an XIF file on the OpenLNS Server. You can either enter the path or click **Browse** to locate the desired XIF file. You can enter the file

name without a path only if the XIF file is in the folder specified in the **XIF Search Path** box **Network Properties: Device** options category (to access this property, click **Add-ins**, click **OpenLNS CT**, click **OpenLNS CT Options**, and then select **Device** from the **Options Category** list). The device template name in the **Template Name** box is automatically set to the name of the XIF file (without the extension), but you can change it to any name that is unique to the network.

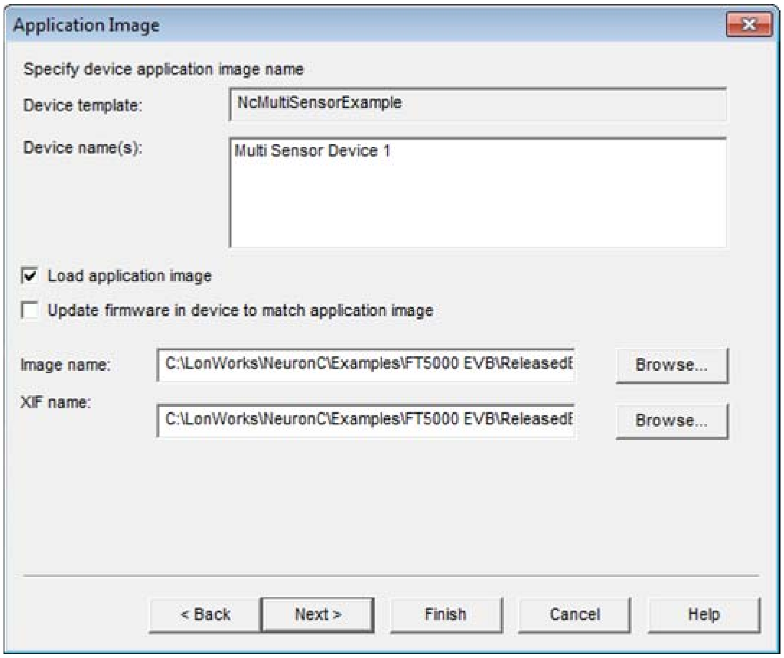
- **Existing Template**. Uses an existing device template. This is the default. By default, the current device template is displayed in the **Existing Template Name** list. You can select any device template currently defined in the network from this list.

  If you select **Upload From Device** or **Load XIF**, the change is applied to all the devices listed in the **Device Names** box. OpenLNS CT then maps all the functional blocks, network variables, message tags, and connections from the old device template to the new one, attempting to preserve as much of the original device configuration as possible. If objects cannot be mapped into the new device template, the OpenLNS CT informs you which objects will be lost and lets you cancel the operation.

2. Click **Next**. The Device Application Image page opens, unless you are loading new application files into multiple devices with different device templates. In this case, this page is repeated for each unique device template used by the selected devices. Once you have defined the device template for all the selected devices, the wizard will then proceed to the Device Application Image page.

Selecting an Application Image and a Neuron Firmware Image

In the Device Application Image page, you select an application image to load into the device.



Device template: NcMultiSensorExample

Device name(s): Multi Sensor Device 1

☑ Load application image

☐ Update firmware in device to match application image

Image name: C:\LonWorks\NeuronC\Examples\FT5000 EVB\ReleasedE

XIF name: C:\LonWorks\NeuronC\Examples\FT5000 EVB\ReleasedE

1. Select the device application you want to load into the device by setting the following options:

Load Application Image

Select this check box to enable loading a new application image into the device.

Update Firmware in Device to Match Application Image

Select this check box to load a new Neuron firmware image to make the Neuron firmware on the device compatible with the new application image file.

For a device to support Neuron firmware updates, it must be based on a Nueron 5000 or Neuron 3150 core with a nonvolatile memory layout supporting at least 32K starting from address 0.

If this check box is cleared, and there is a version of the selected application image file that matches the current device firmware (<*appImage*>fw<*n*>.apb, where *n* is the firmware version) in the same folder of the selected application image file, OpenLNS CT automatically loads that application image file into the device. Otherwise, OpenLNS CT will load the selected application image file into the device.

Note: A device may become inoperable if there is a power failure during a firmware update.

Image Name

Specify the pathname of the application image file (**.apb** or **.nxe** extension).

If you are working on a remote client, OpenLNS CT looks for the application image file on the OpenLNS Server. Specify the full pathname (including the drive) of the application image file on the OpenLNS Server. You cannot load an application image file directly from a remote client computer onto a device.

XIF Name

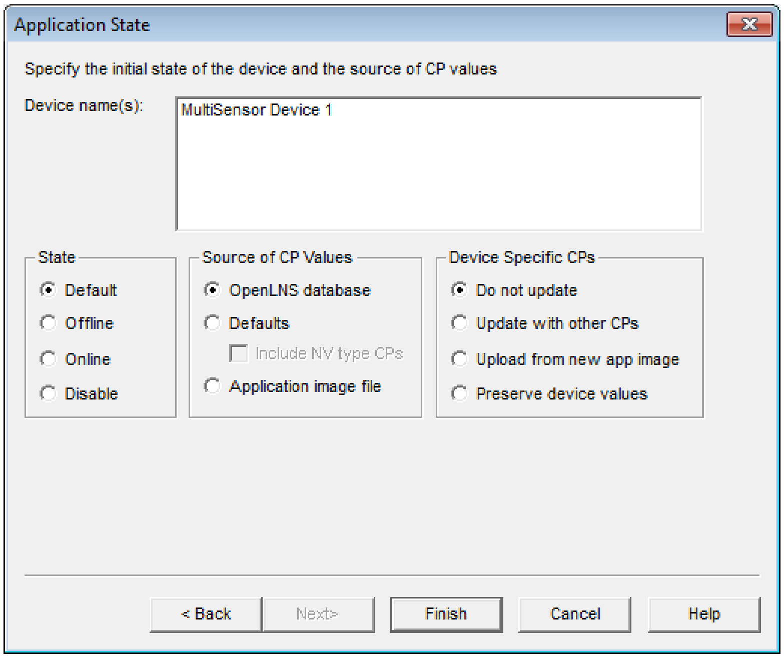
If you are loading an application image file with an .nxe extension, select the pathname of the XIF file. OpenLNS CT only requires the binary application and device interface files (**.apb** and **.xfb** extensions); however, it automatically creates binary files from existing text application and device interface files (**.nxe** and **.xif** extensions) and saves them in the application file folder that contains the corresponding .nxe file.

If you specified a XIF file for the device in the previous page in the wizard (the Device Template page), you cannot select a different XIF file. If an application image file is found in the same directory as the XIF file of the device template, that directory is the default.

2. Click **Next** to continue. The Device Initial State page opens, unless you are loading new application files into multiple devices with different device templates. In this case, this page is repeated for each unique device template used by the selected devices. Once you have defined the application image for all the selected devices, the wizard will then proceed to the Device Initial State page.

Selecting Initial Device State and Source of Configuration Property Values

To select the initial device state and the source of configuration property values, follow these steps:



Application State

Specify the initial state of the device and the source of CP values

Device name(s): MultiSensor Device 1

State
- ● Default
- ○ Offline
- ○ Online
- ○ Disable

Source of CP Values
- ● OpenLNS database
- ○ Defaults
- □ Include NV type CPs
- ○ Application image file

Device Specific CPs
- ● Do not update
- ○ Update with other CPs
- ○ Upload from new app image
- ○ Preserve device values

1. Specify the initial state of the device applications, and select the source of the configuration property values and management of device-specific configuration properties as follows:

State Specify the state of the device application after it has been commissioned. You have four choices:

- **Default**. The state of the device application is determined by the current setting under the **New Device Options** box in the **Network Properties: Device** options category. If the device was previously commissioned and is able to communicate, the default state is the state of the device before the current operation.

- **Online**. Places the device in the online state. The behavior in the online state depends on the device. A Neuron-hosted device, for example, will run its application after it has been commissioned.

- **Offline**. Places the device in the offline state. The behavior in the offline state depends on the device. A Neuron-hosted device, for example, will not run its application after it has been commissioned. This is the default and can be used to bring up a system incrementally.

  When a device is offline, portions of its application may still function and it will still respond to Online, Wink, and Test commands. An offline device still receives network variable updates, but it typically does not process updated network variable values or send network variable updates.

  Resetting an offline device makes it go online, unless the device is in the hard offline state, in which case it will remain offline after a reset.

- **Disable**. Keeps the device application disabled even if the device is reset. For a device to be disabled, it must contain a Node Object functional block. If **Disable** is selected and the device does not contain a Node Object, the device is placed offline. As with offline operation, portions of a disabled device's application may continue to function.

Source of CP Values

Specify the source of the device's configuration property values. You have three choices:

- **OpenLNS Database**. The current configuration property values stored in the OpenLNS network database will be written to the device.
- **Default**. The default configuration property values stored in the device template will be written to the device. This option requires a device template based on a version 4 or higher XIF file with specified default values. See the *LONMARK Device Interface File Developer's Guide* for more details. This is the default.

  If you select **Default**, select whether network variable type configuration properties (**SCPTnvType**) are restored to their default values when the device is commissioned. If you want them to be restored, select the **Include NV Type CPs** check box. This check box is cleared by default. This operation will not work if any of the affected network variables are connected.

- **Application Image File**. The configuration property values specified in the application image file will be written to the OpenLNS network database.

Device Specific CPs

Select how device-specific configuration properties are managed when a device is commissioned. Device-specific configuration properties are configuration properties that may be changed independently from the OpenLNS network database (by the device itself or by a tool not based on OpenLNS, for example) .

- **Do Not Update**. Device-specific configuration properties will not be modified in the device or in the OpenLNS network database. This is the default.

- **Update with Other CPs**. The device-specific configuration property values will be set based on the option selected in the **Source of CP Values** box. For example, if you select **OpenLNS Database** in the **Source of CP Values** box, the device-specific configuration property values stored in the OpenLNS network database will be written to the device.

- **Upload from New App Image**. New device-specific configuration property values are read from the updated device application and written to the OpenLNS network database.

- **Preserve Device Values**. The current device-specific configuration property values set in the device will be written to the OpenLNS network database before the new application image is written to the device. Once the new application image has been installed, the device-specific configuration property values in the OpenLNS network database are written into the new device.

2. Click **Next**. The Device Identification Method window opens. Follow the steps in *Installing Devices and Routers* in Chapter 5, *Installing Networks*, to continue loading the devices.

Replacing Devices

You can replace a device with a new device if the device fails or a newer version of the device becomes available.

The replacement device can use the same device template used by the old device or it can use a different device template. You can also create a new device template for the replacement device using a device interface (XIF) file on the OpenLNS Server or an XIF file stored in the replacement device. If the replacement device uses a new or different device template, the new device application must use a different program ID than the original application.

If you replace a device that is configured and connected to other devices, OpenLNS CT will preserve all compatible configuration and connection information. You can use the same procedure to replace multiple devices at the same time.

To replace a device, follow the three steps described in the subsequent sections:

1. Attach the new device to the network.
2. Replace the device in the OpenLNS CT network.
3. Remove the old device.

Attaching a New Device to the Network

To attach a new device to the network, follow these steps:

1. Set any hardware settings on the new device to match those on the device you are replacing.

2. Attach the replacement device to the network by applying power to the device and attaching its network connection as documented by the device manufacturer.

Replacing a Device in the OpenLNS CT Network

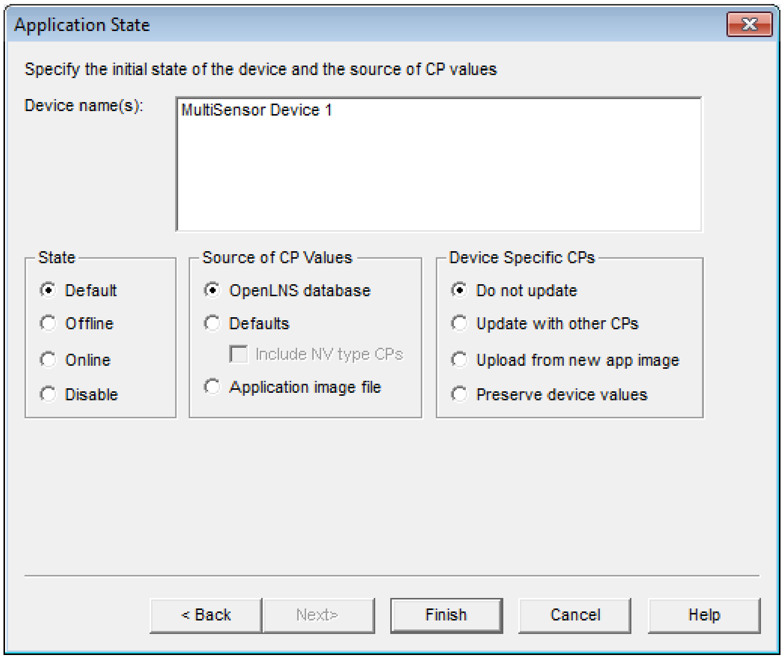
To replace a device in an OpenLNS CT network, follow these steps:

1. If you are replacing an uncommissioned NodeBuilder device, delete the device from the NodeBuilder **Devices** folder.

2. Right-click the device, or select multiple devices and right-click one of them, point to **Commissioning**, and then click **Replace** on the shortcut menu. The Replace Device Wizard opens on the device template page.

3. Select the device template that is associated with the replacement device following the steps in *Selecting or Creating a Device Template* in this chapter. The Specify Device Application Image page opens.

4. Select an application image, and optionally a Neuron firmware image, to load into the replacement device following the steps in *Selecting an Application Image and a Neuron Firmware Image* in the previous *Loading Devices* section in this chapter. The Device Initial State page opens.



Application State

Specify the initial state of the device and the source of CP values

Device name(s): MultiSensor Device 1

State
- ● Default
- ○ Offline
- ○ Online
- ○ Disable

Source of CP Values
- ● OpenLNS database
- ○ Defaults
  - □ Include NV type CPs
- ○ Application image file

Device Specific CPs
- ● Do not update
- ○ Update with other CPs
- ○ Upload from new app image
- ○ Preserve device values

5. Specify the initial state of the replacement device's application, and select the source of the configuration property values and management of device-specific configuration properties as follows:

State — Set the state of the device application after the replacement device has been commissioned. You have four choices:

- **Default**. The state of the replacement device's application is determined by the current setting under the **New Device Options** box in the **Network Properties: Device** options category. If the device was previously commissioned and is able to communicate, the default state is the state of the device before the current operation.
- **Online**. Places the replacement device in the online state. The behavior in the online state depends on the device. A Neuron-hosted device, for example, will run its application after it has been commissioned.
- **Offline**. Places the replacement device in the offline state. The behavior in the offline state depends on the device. A Neuron-hosted device, for example, will not run its application after it has been commissioned. This is the default and can be used to bring up a system incrementally.

  When a device is offline, portions of its application may still function and it will still respond to Online, Wink, and Test commands. An offline device still receives network variable updates, but it typically does not process updated network variable values or send network variable updates.

  Resetting an offline device makes it go online, unless the device is in the hard offline state, in which case it will remain offline after a reset.
- **Disable**. Keeps the replacement device's application disabled even if the device is reset. For a device to be disabled, it must contain a Node Object functional block. If **Disable** is selected

and the device does not contain a Node Object functional block, the device is placed offline. As with offline operation, portions of a disabled device’s application may continue to function. This option is not available for routers.

Source of CP Values

Specify the source of the replacement device’s configuration property values. You have three choices:

- **OpenLNS Database**. The configuration property values stored in the OpenLNS network database for the old device will be written to the replacement device.
- **Defaults**. The default configuration property values stored in the device template will be written to the replacement device. This option requires a device template based on a version 4 or higher XIF file with specified default values. See the *LONMARK Device Interface File Developer’s Guide* for more details. This is the default.

  If you select **Defaults**, select whether network variable type configuration properties (**SCPTnvType**) are restored to their default values when the device is commissioned. If you want them to be restored, select the **Include NV Type CPs** check box. This check box is cleared by default. This operation will not work if any of the affected network variables are connected.
- **New Device Values**. The configuration property values set in the replacement device will be written to the OpenLNS network database.

Device Specific CPs

Select how device-specific configuration properties are managed when an application is loaded into the replacement device. Device-specific configuration properties are configuration properties that may be changed independently from the OpenLNS network database (by the device itself or by a tool not based on OpenLNS, for example).

- **Do Not Update**. Device-specific configuration properties will not be modified in the device or in the OpenLNS network database. This is the default.
- **Update with Other CPs**. The device-specific configuration property values in the replacement device will be set based on the option selected in the **Source of CP Values** box. For example, if you select **OpenLNS Database** in the **Source of CP Values** box, the device-specific configuration property values stored in the OpenLNS network database will be written to the device.
- **Upload from New Device**. The current device-specific configuration property values set in the replacement device will be written to the OpenLNS network database.
- **Transfer from Old Device**. If the old device is available, its device-specific configuration property values will be written to both the OpenLNS network database and the replacement device. If the old device is not available, the device-specific configuration property values in the replacement device will not be changed.

6. Click **Next**. If you are replacing application devices and routers at the same time, the Initial Router State window opens. Otherwise, the Device Identification Method window opens. In this

case, follow the steps in *Installing Devices and Routers* in Chapter 5, *Installing Networks,* to continue installing the devices).

Removing the Old Device

If the old device still functions, and if physically possible, leave the old device connected to the network until the device replacement has been completed. This allows OpenLNS CT to deconfigure the old device so that you can easily reuse it in a new network. This step is not required if the device has failed.

Decommissioning Devices

You can decommission a device if you are no longer using it or you are moving it to a new network. In addition, you can temporarily decommission a device to optimize, troubleshoot, or repair your network. Decommissioning logically removes the device from the network.

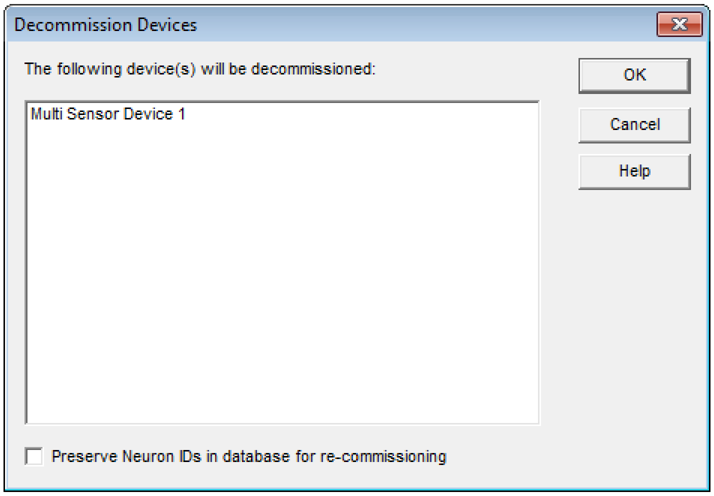
You can preserve the Neuron ID of the decommissioned device in the OpenLNS network database so that you can subsequently recommission the device. All configuration information, including configuration properties and connections, are preserved in the OpenLNS network database. You can then later recommission the same or different devices without re-entering the configuration data.

To decommission a device, follow these steps:

1. Attach OpenLNS CT to the network and set the management mode to OnNet. To do this, click **Add-ins**, point to **LonWorks Network**, and then click **Network Properties**. Click the **Network Interface** tab, select **OnNet** under the **Management Mode** box, and then click **OK**.

2. Right-click the device you want to decommission, or select multiple devices and right-click one of them, point to **Commission**, and then click **Decommission** on the shortcut menu.

3. The **Decommission Devices** dialog opens.



The following device(s) will be decommissioned:

Multi Sensor Device 1

□ Preserve Neuron IDs in database for re-commissioning

4. Select the **Preserve Neuron ID** check box to recommission the same device or router later without re-entering the configuration data.

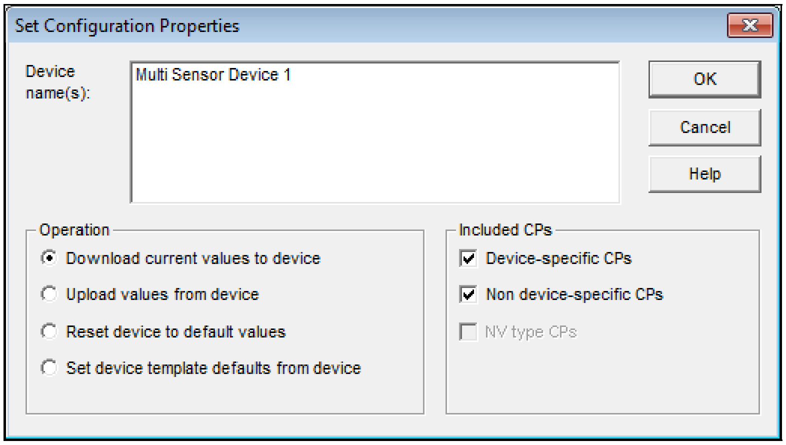
5. Click **OK**. The color of the device SmartShape changes to crosshatched yellow, indicating that the device is unconfigured.

Resynchronizing Configuration Properties

You can resynchronize the configuration property values in the OpenLNS network database with those in the physical devices. You may need to do this if the values get out of synchronization when a network tool not based on OpenLNS is used to update the physical devices, or if the devices themselves update their own configuration properties.

To resynchronize the configuration properties for a device, follow these steps:

1. Right-click the device, or select multiple devices and right-click one of them, point to **Commissioning**, and then click **Resync CPs** on the shortcut menu. The **Set Configuration Properties** dialog opens.



Set Configuration Properties

Device name(s): Multi Sensor Device 1

Operation

● Download current values to device

○ Upload values from device

○ Reset device to default values

○ Set device template defaults from device

Included CPs

☑ Device-specific CPs

☑ Non device-specific CPs

□ NV type CPs

2. Select a synchronization operation and which types of configuration properties to synchronize.

Operation Select a synchronization operation. You have four choices:

- **Download Current Values to Device**. Writes the configuration property values stored in the OpenLNS network database to the physical device. This is the default.
- **Upload Values from Device**. Writes the current configuration property values in the physical device to the OpenLNS network database.
- **Reset Device to Default Values**. Writes the default values defined in the device template to the device and the OpenLNS network database.
- **Set Device Template Defaults from Device**. Sets the default configuration property values in the device template to the current values in the physical device, and writes the configuration property values to the OpenLNS network database.

Included CPs Select which configuration properties are resynchronized. You can include the following configuration properties:

- **Device-Specific CPs**. Updates device-specific configuration properties. This check box is selected by default.
- **Non Device-Specific CPs**. Updates non–device-specific configuration properties. This check box is selected by default.
- **NV Type CPs**. Updates configuration properties which specify the network variable type (**SCPTnvType**). This check box is only available if **Reset Device to Default Values** is selected in the **Operation** box. If any of the affected network variables are connected (with a different NV type), the resynchronization operation will fail.

3. Click **OK**. OpenLNS CT updates the device and the OpenLNS network database, as required.

Using OpenLNS CT as a Passive Configuration Tool

You can use OpenLNS CT as a passive configuration tool. A *passive configuration tool* is a tool that that configures the application in a device without affecting its network configuration. A passive configuration tool is required when a network tool not based on OpenLNS is being used to manage the configuration of a network. This mode is useful when you need to use a plug-in in passive mode to configure a device on a network managed with a network tool not based on OpenLNS. To use OpenLNS CT and one or more plug-ins as a passive configuration tool, follow these steps:

1. Attach OpenLNS CT to the network. If you do not already have an OpenLNS CT network for the network, recover the network as described in *Restoring an OpenLNS CT Network*. In the final steps of the procedure, do not synchronize the network to the database, and leave the management mode OffNet.

   If you synchronize or change the management mode to OnNet, OpenLNS CT will become an active management tool and will no longer be passive. Skip to step 3.

2. If you have an OpenLNS CT network design for the network, open it. Set the management mode to OffNet. To do this, click **Add-ins**, point to **LonWorks Network**, and then click **Network Properties**. Click the **Network Interface** tab, select **OffNet** under the **Management Mode** box, and then click **OK**. If you set the management mode to OnNet, OpenLNS CT will become an active management tool and will no longer be passive.
3. To configure a device with a manufacturer’s plug-in that supports passive configuration, right-click the device or a functional block on the device and then click **Configure** on the shortcut menu. The plug-in starts.
4. Configure the device using the manufacturer’s plug-in.
5. Right-click the device, point to **Commissioning**, and then click **Propagate CP Values to Device** on the shortcut menu.

Backing up an OpenLNS CT Network Design

You can back up an OpenLNS CT network design to an OpenLNS CT backup file to protect against accidental file corruption or hardware failure, copy an OpenLNS CT network design from one computer to another, or replicate an OpenLNS CT network design in a new installation. You can also use an OpenLNS CT backup file to archive an OpenLNS CT network design at a site without an OpenLNS Server or to archive multiple network configurations so a previous configuration can quickly be restored, if necessary.

An OpenLNS CT backup file may contain the following components:

- OpenLNS CT drawing
- OpenLNS network database
- NodeBuilder project (only if you are using the NodeBuilder tool)
- Device interface (XIF) files in the Import folder
- Resource files in the Types folder
- All files in the OpenLNS CT drawing directory. Backing up files in the OpenLNS CT drawing directory provides an easy way to back up and copy files associated with an OpenLNS CT network design such as HMI application files or schedule design files.

You can manually back up an OpenLNS CT network design, or you can have OpenLNS CT schedule periodic backups.

Manually Backing Up an OpenLNS CT Network

You can back up an OpenLNS CT network design from an OpenLNS CT drawing or from the OpenLNS CT Design Manager.

When you back up the network database, you can close the network database and create a *standard backup*, or you can create a *hot backup*, which is a shadow copy of the network database created while it is open. You can use a hot backup to create a backup without disrupting any remote clients or other OpenLNS applications accessing the network database.

When you backup an OpenLNS CT drawing, OpenLNS CT temporarily closes the drawing while making the backup.

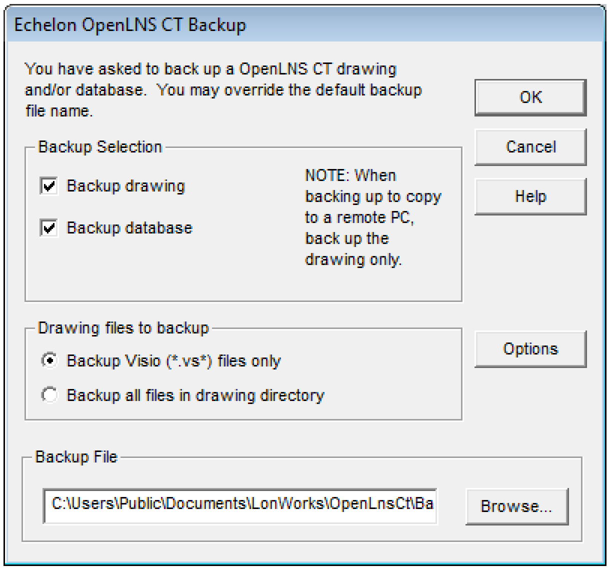
The following sections describe how to create a standard OpenLNS CT backup and how to create a hot backup.

Note: Utilities such as Google Desktop Search may cause backups to fail if they have any of the database or drawing files open at the time of the backup. If you receive a backup failure error, disable the utilities during the backup, or exclude the LM and LONWORKS folders from the search of the utility.

Creating a Backup from an OpenLNS CT Drawing

To create a backup while working in an OpenLNS CT drawing, follow these steps:

1. Click **Add-Ins**, point to **OpenLNS CT Documents**, and then click **Backup**. The **OpenLNS CT Backup** dialog box opens.



You have asked to back up a OpenLNS CT drawing and/or database. You may override the default backup file name.

Backup Selection

☑ Backup drawing

☑ Backup database

NOTE: When backing up to copy to a remote PC, back up the drawing only.

Drawing files to backup

● Backup Visio (*.vs*) files only

○ Backup all files in drawing directory

Backup File

C:\Users\Public\Documents\LonWorks\OpenLnsCt\Ba

2. Select which components of the OpenLNS CT network design to back up.

   - Select both the **Backup Drawing** and **Backup Database** check boxes to back up the entire OpenLNS CT network design. Use this option to move an OpenLNS CT network design from one computer to another, replicate the OpenLNS CT network design for another installation, or to store a backup of the OpenLNS CT network design to protect against accidental file corruption or hardware failure.

   - Select the **Backup Drawing** check box to back up only the OpenLNS CT drawing. Use this option to create a backup of the drawing for use by an OpenLNS CT remote client.

   - Select the **Backup Database** check box to back up the OpenLNS network database only. Use this option to move the network database to an OpenLNS Server computer that does not have OpenLNS CT installed.

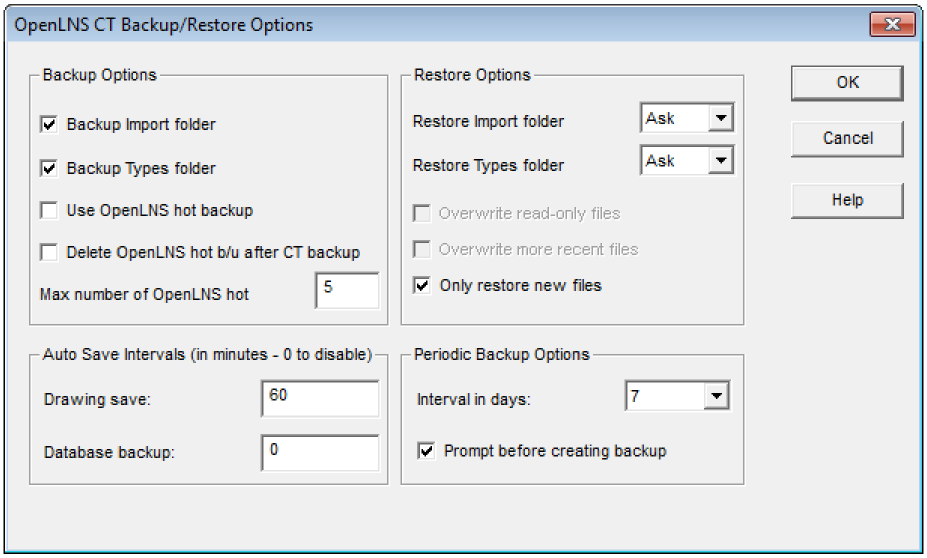
- Select the **Backup NodeBuilder Project** check box to back up the NodeBuilder project associated with the OpenLNS CT network, if there is one. This option is only available if you are using the NodeBuilder Development Tool.

3. If you are backing up the OpenLNS CT drawing, select which drawing files to back up under the **Drawing Files to Backup** box.

   - Select **Backup Visio (*.vs*) Files only** to back up only the Visio files in the drawing directory.

   - Select **Backup all Files in the Drawing Directory** to backup all the files in the drawing directory. This may include HMI projects and documentation. You can also include the plug-ins used by the network by copying them to the drawing directory.

4. To create a hot backup, or to include or exclude the Import and Types folders in the backup file, click **Options**. The **OpenLNS CT Backup/Restore Options** dialog opens.



OpenLNS CT Backup/Restore Options

Backup Options

☑ Backup Import folder

☑ Backup Types folder

☐ Use OpenLNS hot backup

☐ Delete OpenLNS hot b/u after CT backup

Max number of OpenLNS hot: 5

Restore Options

Restore Import folder: Ask

Restore Types folder: Ask

☐ Overwrite read-only files

☐ Overwrite more recent files

☑ Only restore new files

Auto Save Intervals (in minutes - 0 to disable)

Drawing save: 60

Database backup: 0

Periodic Backup Options

Interval in days: 7

☑ Prompt before creating backup

   a. To enable hot backups, select the **Use OpenLNS Hot Backup** check box under **Backup Options**. If you select this check box, OpenLNS CT creates a hot backup of the network database if an application is accessing the OpenLNS network database when you attempt to back up the network. Creating a hot backup prevents remote clients or other OpenLNS applications accessing the network database from being disrupted. If you clear this check box, you can still create a hot backup (see step 6 for more information)

   b. Select or clear the **Backup Import Folder** check box to include or exclude the Import folder (includes XIF and .APB files) in the backup file, and select or clear the **Backup Types Folder** check box to include or exclude the Types folder (includes LONMARK resource files).

   Tip: You can set whether the options for using OpenLNS hot backups and backing up the Import and Types folders are enabled by default in the **Network Properties: Backup/Restore** options category (see Appendix A for more information on setting these options).

5. To change the default backup file name, click **Browse** and navigate to a different directory and enter the file name, or enter the file name in the **Backup File** field. The default backup directory is **C:\Users\Public\Documents\LonWorks\OpenLnsCt\Backup\**<*OpenLNS CT network*>. The default backup file depends on the which components you included in the backup file:

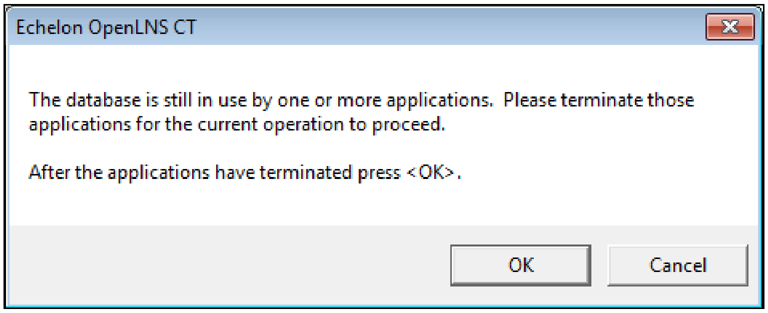
   - If the backup file includes the drawing and database, the default backup file is <*OpenLNS CT network*>[_index]**.zip**.

   - If the backup file includes only the drawing, the default backup file is <*OpenLNS CT network*>Dwg[_index]**.zip**.

- If the backup file includes only the database, the default backup file is <*OpenLNS CT network*>Db[_index]**.zip**.

6. Click **OK**.

   - If the OpenLNS network database is not being accessed, OpenLNS CT automatically creates a standard backup of the database.
   - If the OpenLNS network database is being accessed and hot backups are enabled, OpenLNS CT automatically creates a hot backup of the database.
   - If the database is being accessed and hot backups are disabled, a prompt will open informing you that you need to close all applications accessing the database, including the OpenLNS Server, before creating the backup.


Echelon OpenLNS CT

The database is still in use by one or more applications. Please terminate those applications for the current operation to proceed.

After the applications have terminated press <OK>.

     - To create a backup, close all other OpenLNS applications accessing the network database, including the OpenLNS Server, and then click **OK**. OpenLNS CT will then temporarily close the OpenLNS CT drawing and/or database and then re-open them after creating the backup.
     - Click **Cancel** to stop the backup and return to the OpenLNS CT drawing.

7. The selected OpenLNS CT drawing, database, and associated files will be archived to an OpenLNS CT backup file (**.zip** extension) in the specified backup directory.

Creating a Backup from the OpenLNS CT Design Manager

To create an OpenLNS CT backup from the OpenLNS CT Design, follow these steps:

1. Close all drawing files of the network you want to back up.
2. Open the OpenLNS CT Design Manager. Click **Start** on the taskbar, point to **Programs**, point to the **Echelon OpenLNS CT** folder, and then click **OpenLNS CT**. The **General** tab of the OpenLNS CT Design Manager opens.
3. Select the network to back up by selecting the corresponding network database from the **Database Name** list. If your network contains multiple drawing files, all drawing files in the network will be backed up regardless of which one is selected in the **Drawing Name** list.
4. Click **Backup**. The OpenLNS CT **Backup** dialog box opens.
5. Follow steps 2–7 in the previous section, *Creating a Backup from an OpenLNS CT Drawing*.

Scheduling Drawing Saves and Database Backups

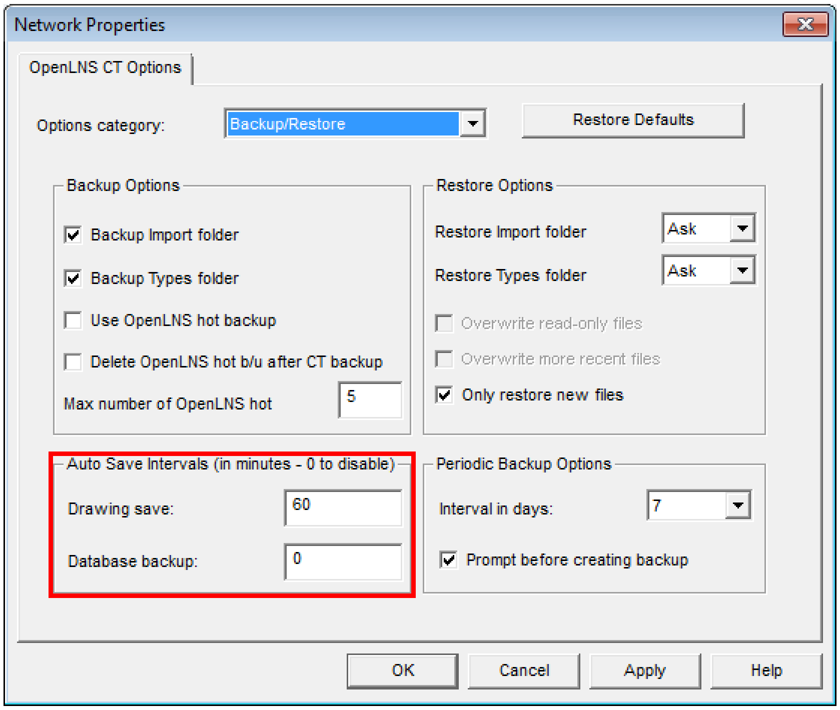
You can schedule regular backups of an OpenLNS CT network using OpenLNS CT or the Windows Task Scheduler.

Scheduling Backups with OpenLNS CT

You can schedule OpenLNS CT to automatically save an OpenLNS CT drawing and back up the network database. To use this feature, you set an interval in which the OpenLNS CT drawing is to be saved and the network database backed up.

To set the automatic OpenLNS CT drawing save interval, follow these steps:

1. In the OpenLNS CT drawing, click **Add-ins**, point to **OpenLNS CT**, and then select **OpenLNS CT Options**. The **Network Properties** dialog box opens.
2. Select **Backup/Restore** from the **Options Category** list.
3. In the **Drawing Save** field under the **Auto Save Interval** box, enter the desired number of minutes between the automatic saves in the network properties. The default value is **60**, which means that the OpenLNS CT drawing is saved every hour by default.
4. In the **Database Backup** field under the **Auto Save Interval** box, enter the desired number of minutes between the automatic database back ups. The default value is **0**, which means that automatic backups are disabled by default.



Options category: Backup/Restore

Backup Options
☑ Backup Import folder
☑ Backup Types folder
□ Use OpenLNS hot backup
□ Delete OpenLNS hot b/u after CT backup
Max number of OpenLNS hot: 5

Restore Options
Restore Import folder: Ask
Restore Types folder: Ask
□ Overwrite read-only files
□ Overwrite more recent files
☑ Only restore new files

Auto Save Intervals (in minutes - 0 to disable)
Drawing save: 60
Database backup: 0

Periodic Backup Options
Interval in days: 7
☑ Prompt before creating backup

Once you set a database backup interval, OpenLNS CT will make the automatic hot backups on the specified interval as long as changes have been made to the network database since the previous backup, the changes were made using OpenLNS CT or reported by an OpenLNS event, and you are using OpenLNS CT on a local client. The hot backup will be stored in the DbBackup<*dd*>\<*network name*> folder, where <*dd*> is the number in the sequence of hot backups made for that network (for example, the third hot backup made for a network named "Building" would be stored in the **DbBackup03 \Building** folder, provided that the maximum number of hot backups is set to 3 or more).

5. You can set the maximum number of hot backups that are saved for all OpenLNS CT networks (includes hot backups created automatically and manually). By default, OpenLNS CT will save five hot backups of a given network database before deleting older hot backups to create new ones. To change the maximum number of hot backups, enter the desired maximum in the **Max Number of OpenLNS Hot Backups** box.
6. You can also set a periodic backup interval, in which OpenLNS CT creates a standard OpenLNS CT backup of the network drawing and database. When you close an OpenLNS CT drawing and the periodic backup interval has elapsed, OpenLNS CT automatically creates a network backup or prompt you as to whether you want to create a backup (you can select whether you want to be prompted to create the backup in the **Periodic Backup Options** box. If you create the network

backup, the backup will be stored in the **LM\Backup\**<*network name*> folder and the backup file will include the date on which it was created and an index (if necessary to make the file unique).

7. To set the periodic backup interval, select the desired number of days between backups in the **Interval in Days** list under the **Periodic Backup Options** box. The default value is **7**, which means that a standard OpenLNS CT backup of the network drawing(s) and database is created every week by default. For networks created with a previous release of OpenLNS CT, you will be prompted as to whether to create a backup the first time you close the OpenLNS CT drawing. The **Periodic Backup Options** box also includes a **Prompt before Creating Backup** check box that you can clear so that OpenLNS CT automatically creates the backup without prompting you. This option is selected by default.

8. Click **OK**.

Scheduling Backups with the Windows Task Scheduler

To schedule a regular backup of a network with the Windows Task Scheduler, you create a scheduled task that runs **OpenLNSCTDesignMgr.exe**. The steps for this depend on your Windows OS.

Windows 7 / Windows Vista

To create a scheduled task in Windows 7 or Vista, follow these steps:

1. Open the Task Scheduler. Click the **Start** button, click **Control Panel**, click **System and Security**, click **Administrative Tools**, and then double-click **Task Scheduler**. If prompted, enter the administrator password or provide confirmation.

2. Click **Action**, and then click **Create Basic Task**.

3. Enter a name for the task (for example, **OpenLNS CT Backup**) and an optional description, and then click **Next**.

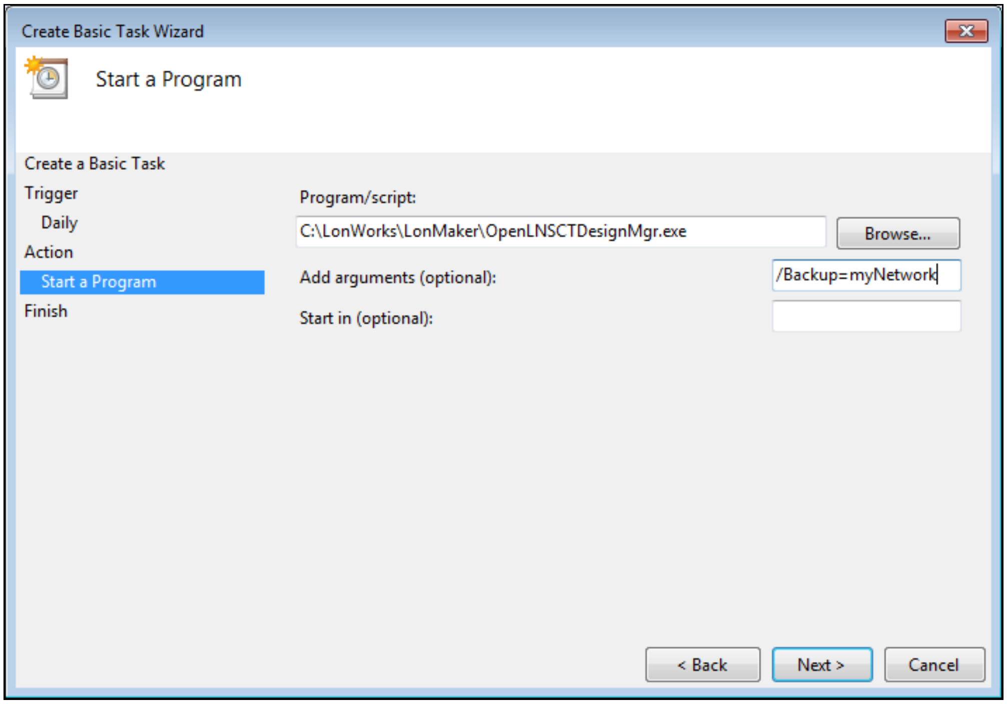
4. Click **Daily, Weekly, Monthly, or One Time** to select a schedule based on the calendar, click **Next**, specify the schedule you want to use, and then click **Next**

5. Click **Start a Program**, and then click **Next**.

   In the **Program Script** box, click **Browse** and select the **OpenLNSCTDesignMgr.exe** file. This file is stored in the **C:\Program Files\LonWorks** or **C:\Program Files (x86)\LonWorks** folders or in the **C:\LonWorks\LonMaker** folder for computers on which OpenLNS CT was upgraded from LonMaker Turbo Editions.

6. In the **Add Arguments** box, enter the following script: **/Backup=<***network name***>**[/**Quiet**] [/**DB**||/**Drawing**]. The following are explanations for the command line options.

   - */Quiet*. Specifies that the network is to be backed up without any prompts.
   - */DB*. Specifies that only the OpenLNS network database is to be backed up (the default is to backup both the OpenLNS network database and the drawing).
   - */Drawing*. Specifies that only the OpenLNS CT drawing is to be backed up.



Program/script: C:\LonWorks\LonMaker\OpenLNSCTDesignMgr.exe

Add arguments (optional): /Backup=myNetwork

Start in (optional):

7. Click **Next**.
8. Click **Finish**.

Windows XP SP3

To create a scheduled task in Windows XP3, follow these steps:

1. Open the Windows Task manager. Click **Start** on the taskbar, click **Run**, and the enter **C:\Windows\tasks**. The **Windows Task** dialog opens.
2. Double-click **Add Scheduled Task**. The Scheduled Task Wizard opens.
3. Click **Next**.
4. Under the **Application** column, locate and then click **LonMaker**. Click **Next**.
5. Modify the task name, if desired, and select how often you want the backup performed. Click **Next**.
6. Select the time, frequency, and date you want to begin running the backup task.
7. Optionally, you can enter a user name and password.
8. Select the **Open Advanced Properties** check box and then click **Finish**. The **Task Scheduler** dialog opens.
9. In the **Run** field, append the following to the command line: **/Backup=**<*network name*>[/**Quiet**] [/**DB**||/**Drawing**]. The following are explanations for the command line options.
   - */Quiet*. Specifies that the network is to be backed up without any prompts.
   - */DB*. Specifies that only the LNS network database is to be backed up (the default is to backup both the LNS network database and the drawing).
   - */Drawing*. Specifies that only the LonMaker drawing is to be backed up.

Restoring an OpenLNS CT Network

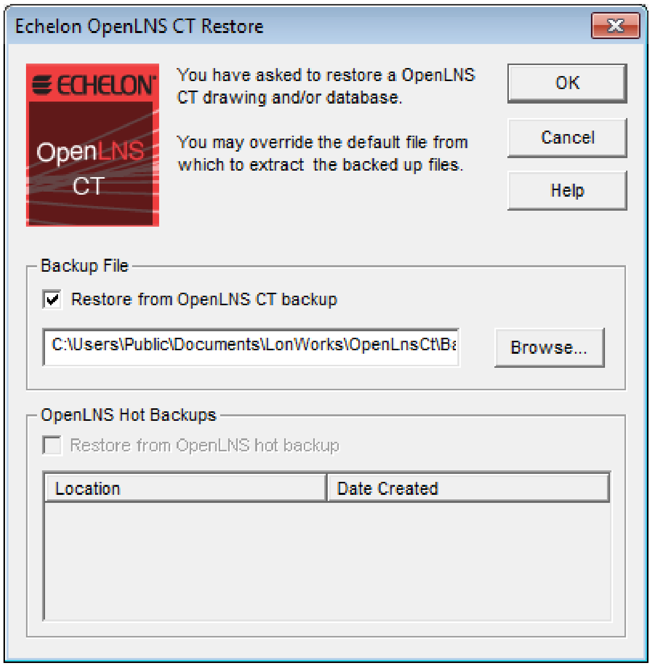
You can restore an OpenLNS CT network from a standard OpenLNS CT backup or a hot backup. If you restore an OpenLNS CT network containing only a network database or an OpenLNS CT drawing, OpenLNS CT replaces the current drawing file and/or database with the backup copy (unless the network does not already exist on a local client). If you restore a backup containing both components, OpenLNS CT replaces the existing files with the backup copies and then lets you open the OpenLNS CT drawing to recommission any devices that have changed since the OpenLNS CT network was backed up.

The following sections describe how to restore a network database backup (standard OpenLNS CT backup or hot backup), an OpenLNS CT drawing backup, and a full network backup containing both network database and OpenLNS CT drawing files.

Restoring a Network Database Backup

To restore a network database backup, follow these steps:

1. Open the OpenLNS CT Design Manager. Click **Start** on the taskbar, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **OpenLNS CT**.
2. Select the network database to restore from the **Database Name** list.
3. Click **Restore**. The OpenLNS CT **Restore** dialog box opens, unless the currently selected network does not have any backups saved in the default location. In this case, proceed to step 4 in which the **Select Backup File** dialog opens.



You have asked to restore a OpenLNS CT drawing and/or database.

You may override the default file from which to extract the backed up files.

Backup File

☑ Restore from OpenLNS CT backup

C:\Users\Public\Documents\LonWorks\OpenLnsCt\Ba

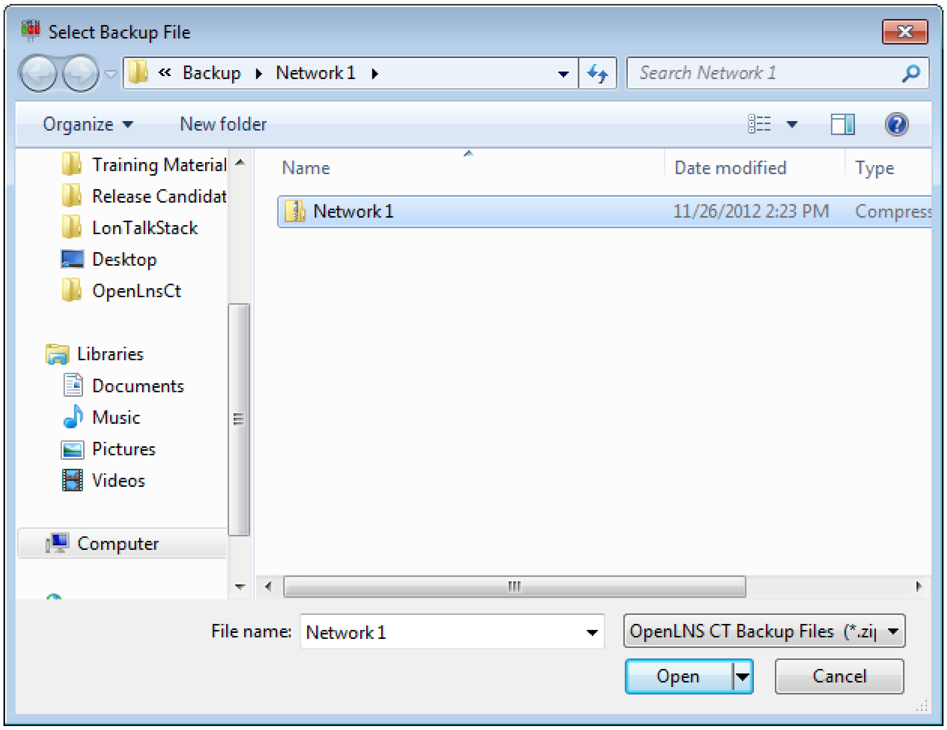
OpenLNS Hot Backups

□ Restore from OpenLNS hot backup

Location	Date Created

4. Select whether to restore a standard OpenLNS CT backup or an OpenLNS hot backup (backup made while the OpenLNS Server is running).

   - If you select the **Restore from OpenLNS CT Backup** check box, specify the backup you want to restore. By default, the last file containing both a network database and OpenLNS CT drawing backup is selected (unless one does not exist). To select a network database backup, click **Browse**. The **Select Backup File** dialog opens. By default, OpenLNS CT backup files are located in the **C:\Users\Public\Documents\LonWorks\OpenLnsCt\Backup**\<*network*>folder. Select the OpenLNS CT backup file (**.zip** extension) of the desired database backup (the database backup file names are in the following format by default: <*network name*>Db[_index]**.zip**), and click **Open** or double-click the OpenLNS CT backup file.



Network 1 11/26/2012 2:23 PM Compress

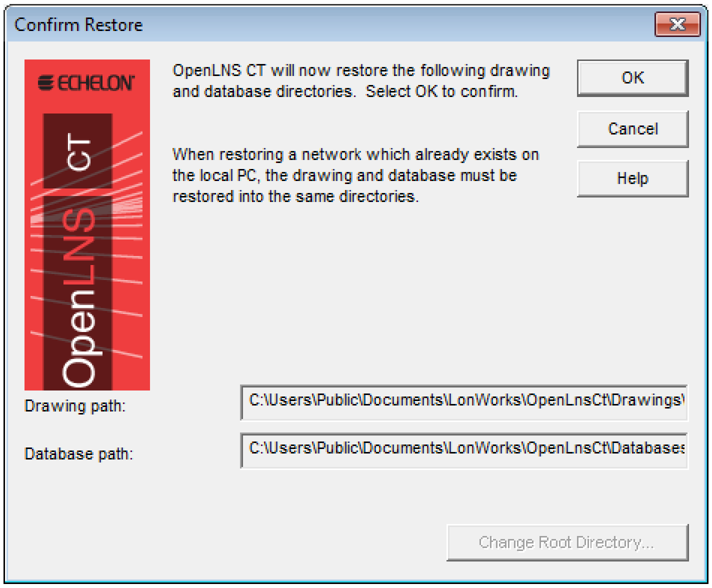
File name: Network 1

OpenLNS CT Backup Files (*.zi|

- If you select the **Restore from OpenLNS Hot Backup** check box, select the OpenLNS hot backup to restore. The OpenLNS hot backups appear in the order they were created. By default, the first OpenLNS hot backup created is selected. To select a different OpenLNS hot backup, click the file name of the OpenLNS hot backup you want to restore.

5. Click **OK** in the OpenLNS CT **Restore** dialog box.

6. OpenLNS CT checks for user privileges. If this OpenLNS CT network is password protected (see *User Profiles* in Chapter 3, *Getting Started*), enter your user name and password. Your user profile must have the **Modify** privilege selected for the network object to restore a network. The **Confirm Restore** dialog box opens.



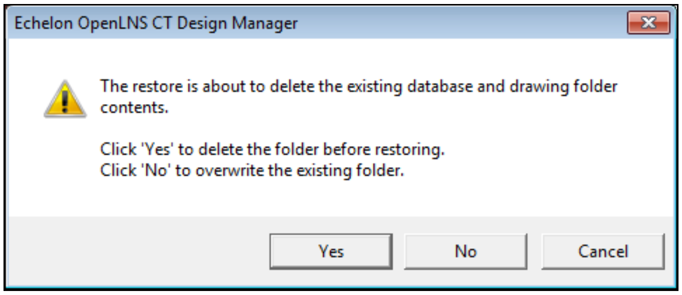
OpenLNS CT will now restore the following drawing and database directories. Select OK to confirm.

When restoring a network which already exists on the local PC, the drawing and database must be restored into the same directories.

Drawing path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Drawings\

Database path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Databases

7. This dialog box displays the paths where the restored network database will be saved. You cannot change the directory if the database already exists on your computer. Click **OK**. A message will appear prompting you to select whether to delete or overwrite the existing directory.



The restore is about to delete the existing database and drawing folder contents.

Click 'Yes' to delete the folder before restoring.
Click 'No' to overwrite the existing folder.

8. Click **Yes** to delete the existing drawing folder before restoring the network, or click **No** to overwrite it. The database restore operation begins.

9. When the network database has been restored, a confirmation dialog opens. Click **OK**.

Note: When you open a network restored from a hot backup of the network database, a message may appear prompting you to resynchronize the OpenLNS CT drawing to match the database. This occurs if changes were made to the drawing after the hot backup of the network database was created.

Restoring an OpenLNS CT Drawing Backup

To restore an OpenLNS CT drawing backup, follow these steps:

1. Open the OpenLNS CT Design Manager. Click **Start** on the taskbar, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **OpenLNS CT**.

2. Select the OpenLNS CT drawing to restore from the **Drawing Directory** list.

3. Click **Restore**. The OpenLNS CT **Restore** dialog box opens, unless the currently selected network does not have any backups saved in the default location. In this case, proceed to step 6 to select the drawing backup to restore in the **Select Backup File** dialog.

4. Select the **Restore from OpenLNS CT Backup** check box.

5. If there is no file containing both network database and OpenLNS CT drawing backups, the **Select Backup File** dialog opens. Otherwise, click **Browse** to open this dialog.

6. Select the OpenLNS CT backup file (.zip extension) of the desired OpenLNS CT drawing to restore (the drawing backup files are in the following format by default: <*network name*>Dwg[_index]**.zip**), and click **Open** or double-click the OpenLNS CT backup file.

7. Click **OK** in the **OpenLNS CT Restore** dialog box.

8. OpenLNS CT checks for user privileges. If this OpenLNS CT network is password protected (see *User Profiles* in Chapter 3, *Getting Started*), enter your user name and password and then click **OK**. Your user profile must have the **Modify** privilege selected for the network object to restore an OpenLNS CT drawing. The **Confirm Restore** dialog box opens.

9. Click **OK**. A message opens prompting you to select whether to delete or overwrite the existing directory.

10. Click **Yes** to delete the existing drawing folder before restoring the network, or click **No** to overwrite it. The drawing restore operation begins.

11. When the drawing has been restored, a message opens informing you that the drawing was restored successfully.

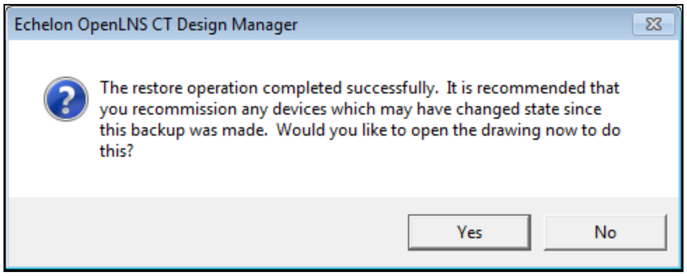
Restoring a Full Network Backup

To restore a full network backup containing both the network database and OpenLNS CT drawing files, follow these steps:

1. Open the OpenLNS CT Design Manager. Click **Start** on the taskbar, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **OpenLNS CT**.
2. Select the network to restore by selecting either the database from the **Database Name** list or the OpenLNS CT drawing from the **Drawing Directory** list.
3. Click **Restore**. The OpenLNS CT **Restore** dialog box opens.
4. Select the Restore from OpenLNS CT Backup check box.
5. By default, the last file containing both a network database and OpenLNS CT drawing backup for the selected network opens in the **Restore from OpenLNS CT Backup** box. To select a different OpenLNS CT backup file (.zip extension), click **Browse**. The **Select Backup File** dialog opens. Select the OpenLNS CT backup file to restore (the full network backup files are in the following format by default: <*network name*> [_index]**.zip**), and click **Open** or double-click the OpenLNS CT backup file.
6. Click **OK** in the OpenLNS CT **Restore** dialog box.
7. OpenLNS CT checks for user privileges. If this OpenLNS CT network is password protected (see *User Profiles*), enter your user name and password and then click **OK**. Your user profile must have the Modify privilege selected for the network object to restore a network. The **Confirm Restore** dialog box opens.
8. This dialog box displays the paths where the restored network drawing and database will be saved. You cannot change the directory if the database or the drawing already exists on your computer. Click **OK**. A message opens prompting you to select whether to delete or overwrite the existing directory.
9. Click **Yes** to delete the existing drawing folder before restoring the network, or click **No** to overwrite it. The full network restore begins.
10. By default, OpenLNS CT will prompt you to select whether to install any new files in the Import folder (includes LONMARK resource files) and then any new files in the **Types** folder (includes **.XIF** and **.APB** files).

    Click **Yes** to restore the files; click **No** to skip restoring the files; or clear the **Only Restore New Files** check box, select one or both of the **Overwrite Read-only Files** and **Overwrite more Recent Files** check boxes, and click **Yes**. The overwrite options will replace existing files with backup versions.

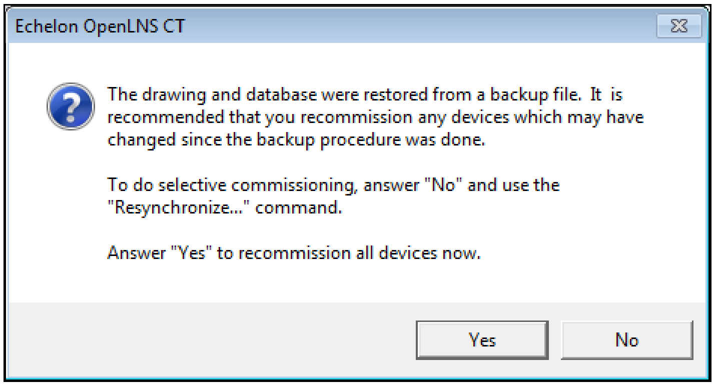
    If you do not want to be prompted, in the **Network Properties: Backup/Restore** options category, you can set whether OpenLNS CT automatically installs the backup files in the Import and Types folders.
11. A message opens informing you that the network restore operation has been completed, and prompting you to select whether to open the OpenLNS CT network in order to recommission devices that have changed since the network was backed up.


Echelon OpenLNS CT Design Manager

The restore operation completed successfully. It is recommended that you recommission any devices which may have changed state since this backup was made. Would you like to open the drawing now to do this?

- Click **Yes** if you made *any* changes to the network since it was backed up. This prevents the network from behaving unpredictably if the OpenLNS CT network design is not in sync with the physical devices.
- Click **No** only if changes have not been made to the configuration of the existing physical devices on the network since it was backed up. This happens if OpenLNS CT was OffNet the entire time, or if you added new devices and functional blocks but did not modify any existing devices or functional blocks. The OpenLNS CT drawing will not be opened.

12. Complete the Network Wizard by following the steps in *Opening an Existing OpenLNS CT Network Design* in Chapter 3, *Getting Started*.
13. When you open an OpenLNS CT drawing after restoring the database from a backup file, a message opens recommending that you recommission devices that have changed since the network was backed up.



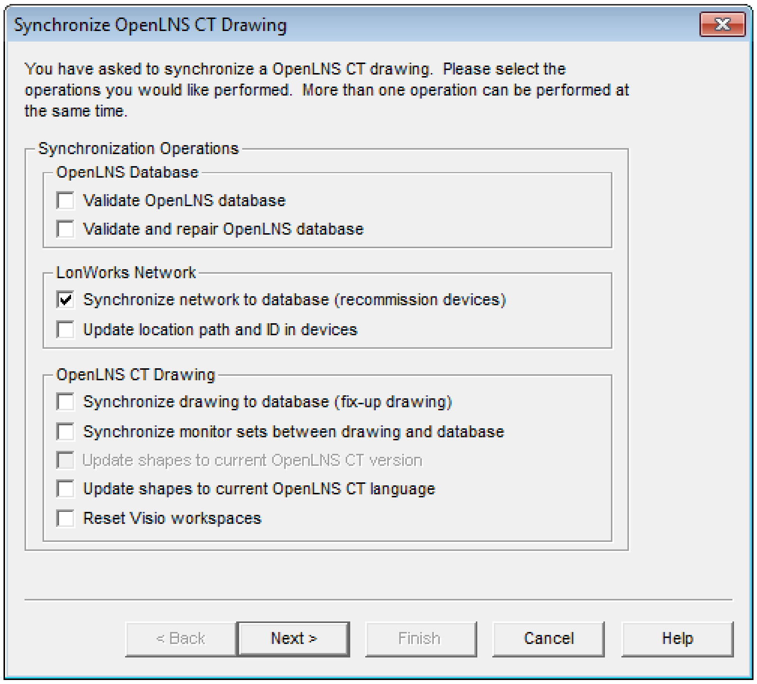
The drawing and database were restored from a backup file. It is recommended that you recommission any devices which may have changed since the backup procedure was done.

To do selective commissioning, answer "No" and use the "Resynchronize..." command.

Answer "Yes" to recommission all devices now.

- Click **Yes** if you want OpenLNS CT to recommission all the devices in the network. OpenLNS CT opens the restored network and starts the Network Wizard.
- Click **No** if you want to use the **Resynchronize** command after the drawing has been opened and select the desired scope of the recommissioning process. To recommission devices after the drawing has been opened, click **Add-ins**, point to **Synchronization**, click **Resynchronize**, and follow steps 14–18.

14. The **Synchronize OpenLNS CT Drawing** dialog box opens.



Synchronize OpenLNS CT Drawing

You have asked to synchronize a OpenLNS CT drawing. Please select the operations you would like performed. More than one operation can be performed at the same time.

Synchronization Operations

OpenLNS Database

- □ Validate OpenLNS database
- □ Validate and repair OpenLNS database

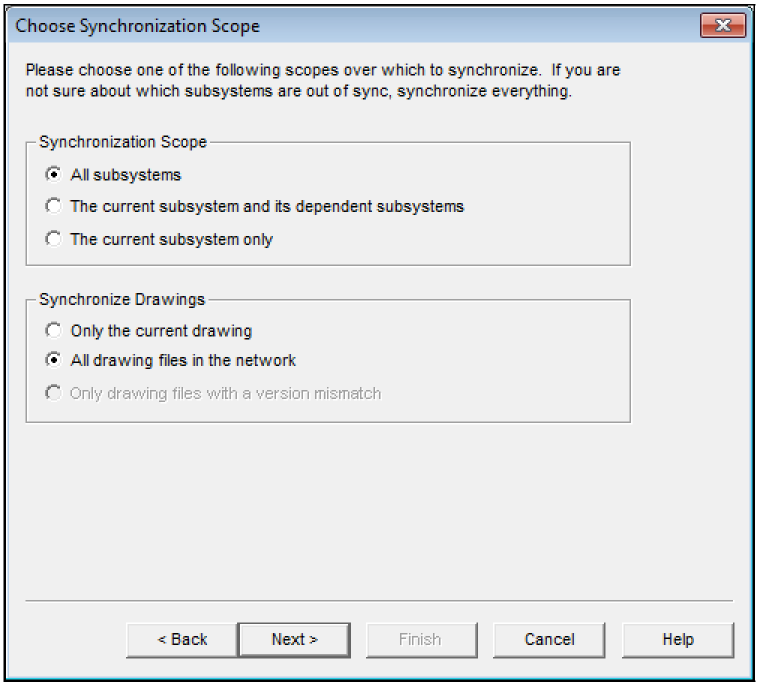
LonWorks Network

- ☑ Synchronize network to database (recommission devices)
- □ Update location path and ID in devices

OpenLNS CT Drawing

- □ Synchronize drawing to database (fix-up drawing)
- □ Synchronize monitor sets between drawing and database
- □ Update shapes to current OpenLNS CT version
- □ Update shapes to current OpenLNS CT language
- □ Reset Visio workspaces

15. Click **Next** to next to accept the default synchronization operation, which is **Synchronize Network to Database** (the option for recommissioning devices). The **Choose Synchronization Scope** dialog opens.



Choose Synchronization Scope

Please choose one of the following scopes over which to synchronize. If you are not sure about which subsystems are out of sync, synchronize everything.

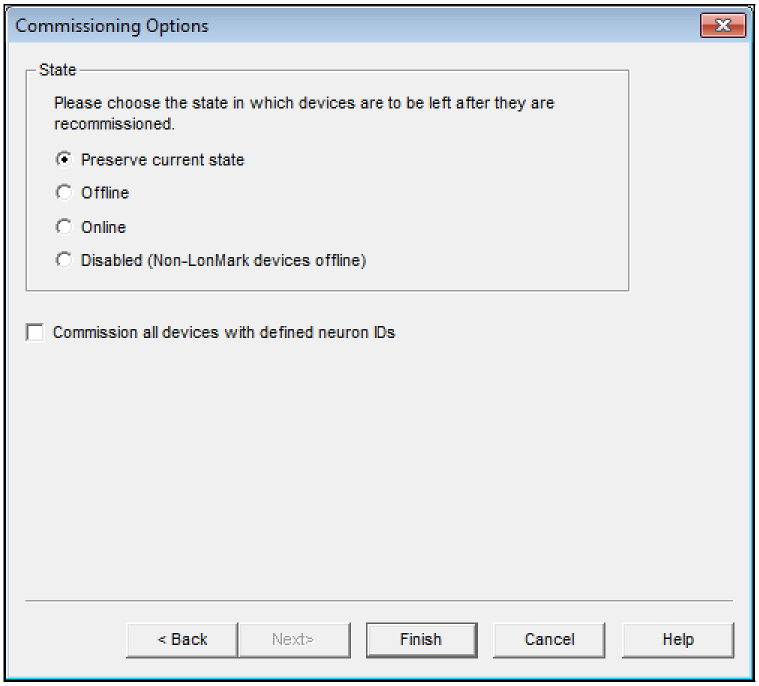
Synchronization Scope

- ● All subsystems
- ○ The current subsystem and its dependent subsystems
- ○ The current subsystem only

Synchronize Drawings

- ○ Only the current drawing
- ● All drawing files in the network
- ○ Only drawing files with a version mismatch

16. Click **Next** to accept the default synchronization scope (all the subsystems and drawing files in the network). The **Commissioning Options** dialog box opens.



Commissioning Options

State

Please choose the state in which devices are to be left after they are recommissioned.

● Preserve current state

○ Offline

○ Online

○ Disabled (Non-LonMark devices offline)

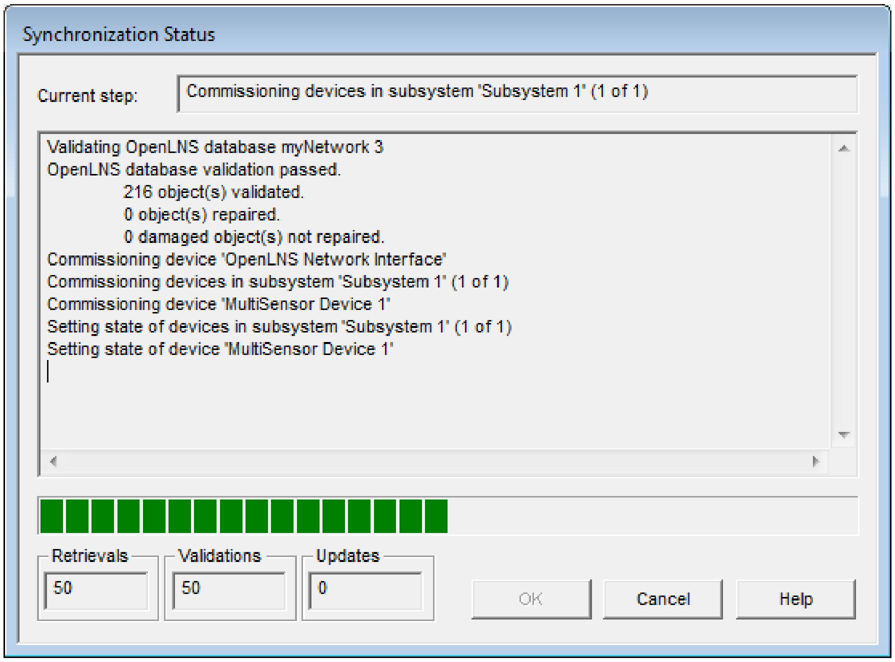
□ Commission all devices with defined neuron IDs

17. Select the state of the devices after they have been recommissioned. You have four choices:

    - **Preserve Current State**. The devices will be restored to the state they were in prior to being recommissioned. This is the default.
    - **Offline**. All devices will be left in the offline state. The behavior in the offline state depends on the device. A Neuron-hosted device, for example, will not run its application. You can set all device offline so that you can recommission the system incrementally. Resetting a device in the offline state puts the device in the online state.
    - **Online**. All devices will be put in the online state. The behavior in the online state depends on the device. A Neuron-hosted device, for example, will run its application after it has been recommissioned, sending and responding to network and external inputs.
    - **Disabled**. All devices with a Node Object functional block will be disabled; all other devices will be placed offline. Selecting **Disabled** keeps the device's application disabled even if the device is reset.

    Select the **Commission all Devices with Defined Neuron IDs** check box to commission all the previously commissioned devices as well as any unconfigured devices within the selected synchronization scope.

18. Click **Finish**. The **Synchronization Status** dialog opens.



Current step: Commissioning devices in subsystem 'Subsystem 1' (1 of 1)

```
Validating OpenLNS database myNetwork 3
OpenLNS database validation passed.
            216 object(s) validated.
            0 object(s) repaired.
            0 damaged object(s) not repaired.
Commissioning device 'OpenLNS Network Interface'
Commissioning devices in subsystem 'Subsystem 1' (1 of 1)
Commissioning device 'MultiSensor Device 1'
Setting state of devices in subsystem 'Subsystem 1' (1 of 1)
Setting state of device 'MultiSensor Device 1'
```

Retrievals: 50
Validations: 50
Updates: 0

19. This window displays the status of the resynchronization. Resynchronization events will be listed in the order they occur. Click **OK** when the resynchronization is complete. The resynchronization events, including any errors, are shown in the status pane and recorded in a log file (**Resync.log**). The resync log is stored in the drawing directory of the current OpenLNS CT network.

    Notes:

    - Recommissioning a device may temporarily interrupt its operation. Recommission devices when the network can tolerate brief service interruptions.
    - When recommissioning the network, OpenLNS CT is placed OnNet and it remains OnNet after the recommissioning process has been completed.
    - If you defined and commissioned new devices in between backing up and restoring the network, recommissioning the network does not modify the new devices. You can redefine and commission those devices so that they communicate properly with existing devices after you have restored and recommissioned your network.

Recovering an OpenLNS CT Network

You can recover an OpenLNS CT network design from a physical network. The network recovery creates a new OpenLNS CT drawing and OpenLNS network database based on the devices recovered. This is useful for networks that were installed with legacy network tools not based on OpenLNS; self-installed networks, including ISI networks; and OpenLNS CT networks or networks installed with OpenLNS in which the original network database has been lost.

To recover a network, follow these steps:

1. Ensure that all components of the network are present and functioning properly before starting recovery. Any missing devices or routers will cause portions of the network design to be missing. For example, if a router is not recovered, all devices on the router's far side will not be included in the recovered network design.
2. Run the OpenLNS Network Database Recovery Wizard. The recovery wizard includes an option to automatically generate the OpenLNS CT drawing.

3. Verify that the recovered network opens to be correct. The recovery wizard logs the recovery tasks as well as any errors that occurred in the detailed recovery log. A trace log lists the steps, with the times and dates, that the wizard took to recover the database. You can view the logs after the database has been recovered. These logs are useful for tracking any errors that the recovery process encounters, such as connection errors or lost data.

4. To actively manage the recovered network, change the management mode to OnNet. To do this, click **Add-ins**, point to **LonWorks Network**, and then click **Network Properties**. Click the **Network Interface** tab, select **OnNet** under the **Management Mode** box, and then click **OK**.

The following sections detail each of these steps. The subsystem recovery options that you can set in an OpenLNS CT network design are also described. These subsystem recovery options help you recover your network in the event you lose the network design and all backups.

Subsystem Recovery Options

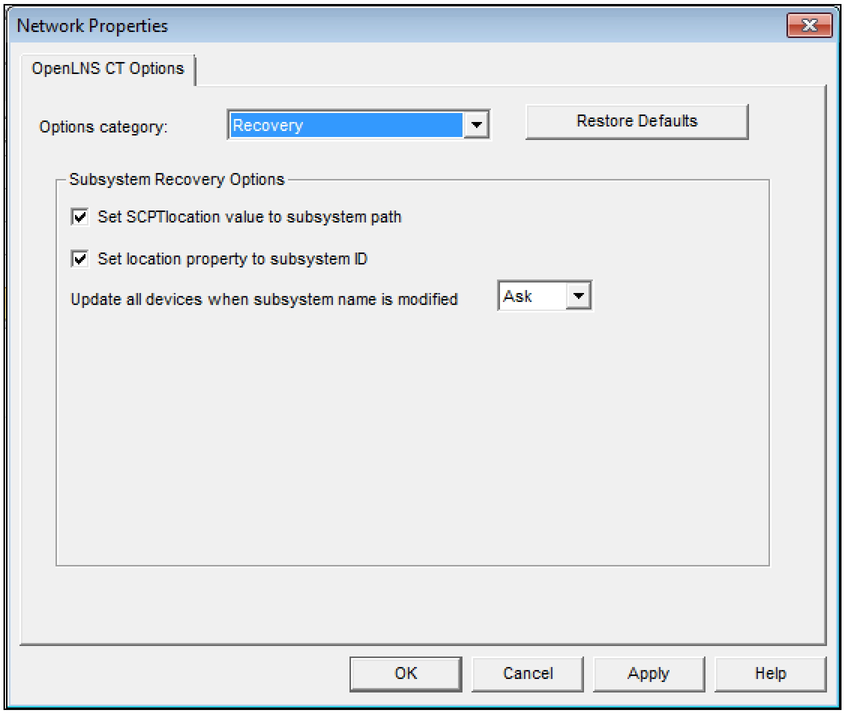
You can make your OpenLNS CT network designs easier to recover and enhance a recovered network design by storing recovery data in the physical devices. You can store the subsystem ID of a device in its location string and its subsystem path in its **SCPTlocation** configuration property (if it has one). This recovery data is not part of the network configuration of a device.

When you recover a network, the following data is used in the listed order:

1. Subsystem path stored in a CSV file.
2. Subsystem path stored in a **SCPTlocation** configuration property that is associated with the Node Object, if present, or associated with the application device.
3. Subsystem ID stored in the location property (the most-significant bit of the first byte of the location string must be set).
4. The **Recovered** subsystem. A device is placed in a subsystem named “**Recovered**” if a CSV file is not imported or subsystem data is not included in the file, and the device does not have a valid subsystem ID or a valid **SCPTlocation** configuration property

By default, both subsystem recovery options are enabled in new networks. For OpenLNS CT network designs created with previous versions of OpenLNS CT, they are disabled. To enable the subsystem recovery options and update existing devices in a network, follow these steps:

1. Click **Add-ins**, point to **OpenLNS CT**, and then click **OpenLNS CT Options**. The **Network Properties** dialog opens.
2. Select **Recovery** in the **Options Category** list.
3. Select the check boxes under the **Subsystem Recovery Options** box. This will enable subsystem location data to be stored and updated in all future application devices and routers you create.



Options category: Recovery

Subsystem Recovery Options

☑ Set SCPTlocation value to subsystem path

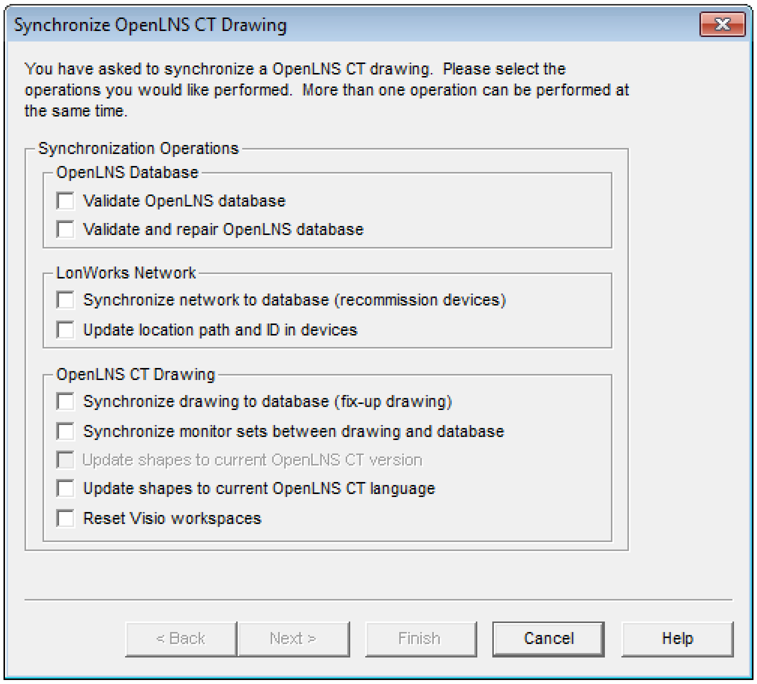
☑ Set location property to subsystem ID

Update all devices when subsystem name is modified: Ask

4. Click **OK**.

You can update the recovery data in the physical devices in a network after setting the subsystem recovery options and after recovering a network. To update the recovery data, follow these steps:

1. Click **Add-ins**, point to **Synchronization**, and then click **Resynchronize**. The **Synchronize OpenLNS CT Drawing** dialog opens.



You have asked to synchronize a OpenLNS CT drawing. Please select the operations you would like performed. More than one operation can be performed at the same time.

Synchronization Operations

OpenLNS Database

☐ Validate OpenLNS database

☐ Validate and repair OpenLNS database

LonWorks Network

☐ Synchronize network to database (recommission devices)

☐ Update location path and ID in devices

OpenLNS CT Drawing

☐ Synchronize drawing to database (fix-up drawing)

☐ Synchronize monitor sets between drawing and database

☐ Update shapes to current OpenLNS CT version

☐ Update shapes to current OpenLNS CT language

☐ Reset Visio workspaces

2. Under **LONWORKS Network**, select the **Update Location Path and ID in Devices** check box (this option is only available if at least one of the subsystem recovery options is selected). Click **Next**. The **Choose Synchronization Scope** dialog opens.

3. Select the subsystems and drawing files to update.
4. Click **Finish**. The **Synchronization Status** dialog opens and OpenLNS CT begins updating the subsystem data in the devices within the specified network scope.
5. When the synchronization operation has been completed, click **OK**.

Using Subsystem Paths

If you select the **Set SCPT Location Value to Subsystem Path** option, OpenLNS CT stores the subsystem path in the **SCPTlocation** configuration property of application devices when they are commissioned. OpenLNS CT automatically updates the subsystem path stored in **SCPTlocation** when the application device is moved. If you set the **Update all Devices when Subsystem Name is Modified** option in the **Network Properties: Recovery** options category to **Yes**, OpenLNS CT also automatically updates the subsystem path of the device when you move or rename its subsystem.

If you do not import a CSV file that includes the subsystem location data for an application device, the recovery wizard places application devices in the subsystem path specified in the **SCPTlocation** configuration property. If the device also has a subsystem ID, the recovery wizard will link the subsystem path to it so that all devices with same subsystem ID but no subsystem path can be placed in that subsystem path.

Tip:

To optimize the accuracy of the network recovery, keep subsystem paths under 32 characters. If you are storing subsystem IDs in the **SCPTlocation** configuration property of devices and you commission a device that has a subsystem path longer than 31 characters, OpenLNS CT truncates the subsystem path to fit within 31 characters. This may result in the recovery wizard losing part of the subsystem path, which may cause the device to be moved to a parent subsystem.

Using Subsystem IDs

If you select the **Set Location Property to Subsystem ID** option, OpenLNS CT stores the subsystem ID in application devices or routers when they are commissioned. The subsystem ID is equal to the lower 15 bits of a device’s subsystem handle, and it opens as a 6-character ASCII or 12-digit hex string in the **Location** field of the **Basic Properties** dialog box. OpenLNS CT automatically updates the subsystem ID when you move the device. If you set the **Update all Devices when Subsystem Name is Modified** option in the **Network Properties: Recovery** options category to **Yes**, OpenLNS CT also automatically updates the subsystem ID of the device when you move or rename its subsystem.

The recovery wizard will place devices in the subsystem path specified by the subsystem ID if the following is true: you do not import a CSV file that includes the subsystem data; the device does not have a valid **SCPTlocation** configuration property; and no other devices that have the same subsystem ID have a valid **SCPTlocation** configuration property. In this case, the device will be placed in a subsystem named “**Subsystem** <*subsystemID*>”.

If there are one or more discovered devices that have a subsystem ID or a valid **SCPTlocation** configuration property, the following occurs, listed in order of precedence:

- A discovered device with the same subsystem ID but no valid **SCPTlocation** configuration property is placed in the subsystem path specified by the first **SCPTlocation** configuration property value discovered.
- A discovered device with the same **SCPTlocation** configuration property but no valid subsystem ID is placed in the subsystem specified by the first subsystem ID value discovered. If there is no discovered device that has a valid subsystem ID, all discovered devices with the same **SCPTlocation** configuration property are assigned the same new subsystem ID.

  Based on this, if OpenLNS CT discovers both devices with the same subsystem ID but different **SCPTlocation** configuration properties and devices with the same **SCPTlocation** configuration property but different subsystem IDs, OpenLNS CT places the discovered devices in the subsystem path specified by the first valid **SCPTlocation** configuration property value discovered.

OpenLNS CT then assigns the discovered devices the first subsystem ID value discovered. OpenLNS CT resolves any conflict that occurs in this scenario.

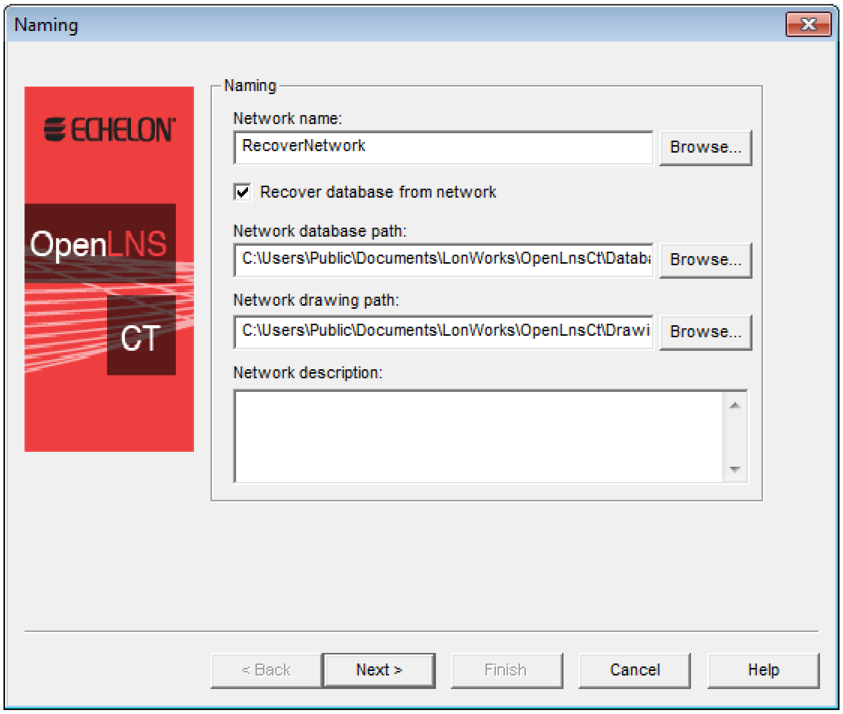
Using the OpenLNS Database Recovery Wizard

To run the OpenLNS Database Recovery Wizard in order to recover a network, follow these steps:

1. Click **Start** on the taskbar, point to **Programs**, point to **Echelon OpenLNS CT**, and then select **OpenLNS CT**. The **General** tab of the OpenLNS CT Design opens.

2. Under **New Network**, select the **Show all Options** check box and then click **Create Network**. OpenLNS CT opens the new network, and the **Naming** dialog in the Network Wizard opens.

3. Enter a unique network name, and select the **Recover Database from Network** check box. Click **Next**.



Naming

Network name: RecoverNetwork

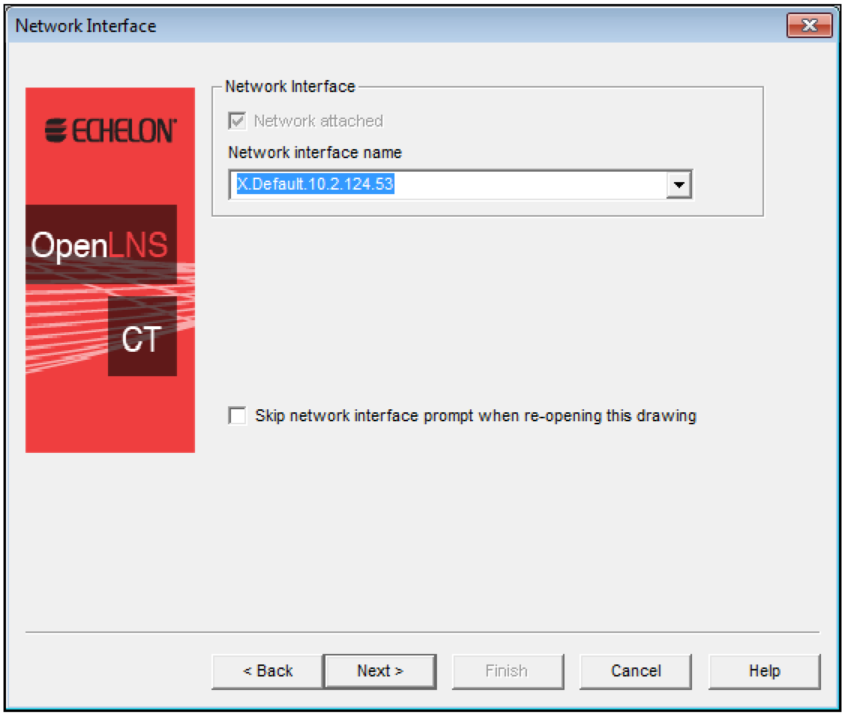
☑ Recover database from network

Network database path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Datab

Network drawing path: C:\Users\Public\Documents\LonWorks\OpenLnsCt\Drawi

Network description:

4. The **Network Interface** dialog opens. Select the network interface and then click **Next**



Network Interface

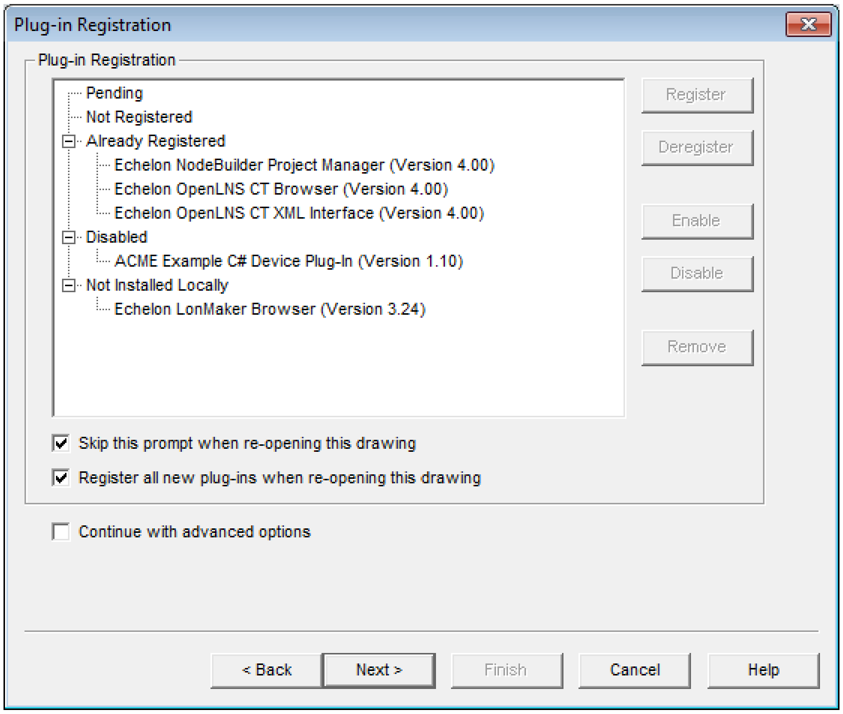
Network Interface

☑ Network attached

Network interface name: X.Default.10.2.124.53

□ Skip network interface prompt when re-opening this drawing

5. The **Plug-In Registration** dialog opens. Register or disable plug-ins as desired and click **Next**.



Plug-in Registration

Plug-in Registration

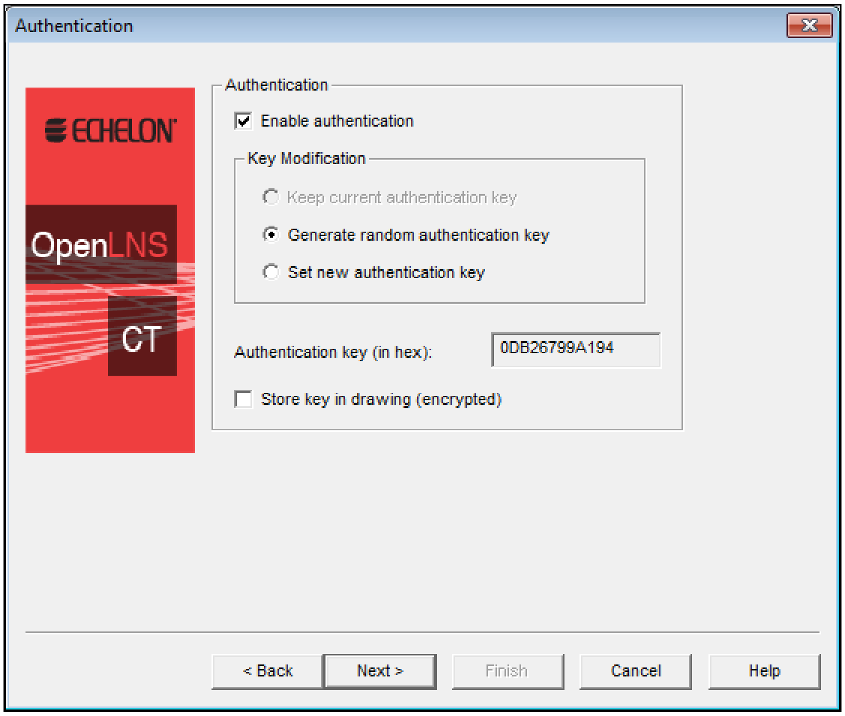
- Pending
- Not Registered
- Already Registered
  - Echelon NodeBuilder Project Manager (Version 4.00)
  - Echelon OpenLNS CT Browser (Version 4.00)
  - Echelon OpenLNS CT XML Interface (Version 4.00)
- Disabled
  - ACME Example C# Device Plug-In (Version 1.10)
- Not Installed Locally
  - Echelon LonMaker Browser (Version 3.24)

☑ Skip this prompt when re-opening this drawing

☑ Register all new plug-ins when re-opening this drawing

□ Continue with advanced options

6. The **Authentication** dialog opens. If you use authentication on your network, select the **Enable Authentication** check box and click **Set New Authentication Key**. Enter the correct authentication key for your network (OpenLNS CT cannot recover your database without the authentication key). Click **Next**.



Authentication

☑ Enable authentication

Key Modification

○ Keep current authentication key

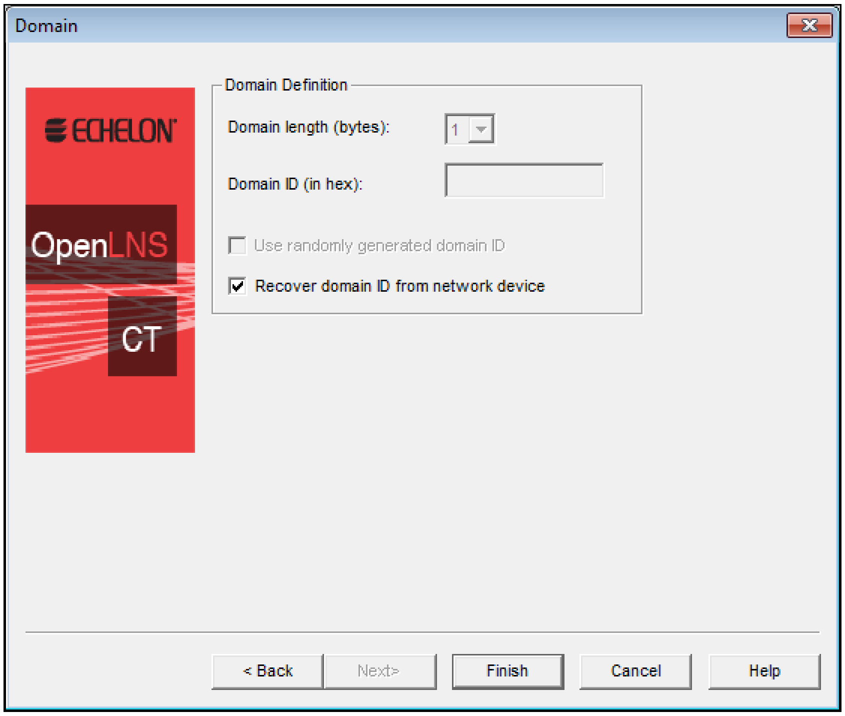
● Generate random authentication key

○ Set new authentication key

Authentication key (in hex): 0DB26799A194

□ Store key in drawing (encrypted)

7. The **Domain Definition** dialog opens. If you know the domain ID of the network that you want to recover, enter it in hexadecimal format into the **Domain ID** box. If you do not know the domain ID, select the **Recover Domain ID from Network Device** check box. To recover a domain ID, you must have access to the service pin of an unauthenticated device that is on the same channel as the network interface.



Domain Definition

Domain length (bytes): 1

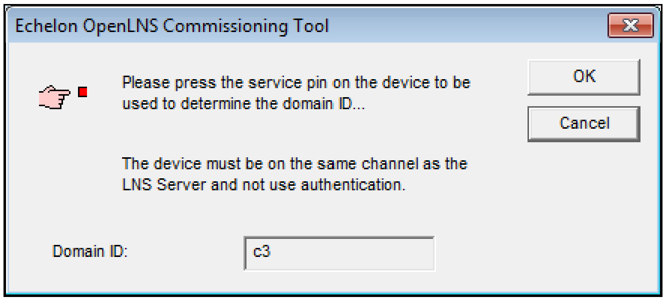
Domain ID (in hex):

□ Use randomly generated domain ID

☑ Recover domain ID from network device

8. Click **Finish**.

9. If you selected the **Recover Domain ID from Network Device** check box, the **Service Pin** dialog box opens. Press the service pin on one of the unauthenticated devices that are on the same channel as the network interface, and then click **OK**.



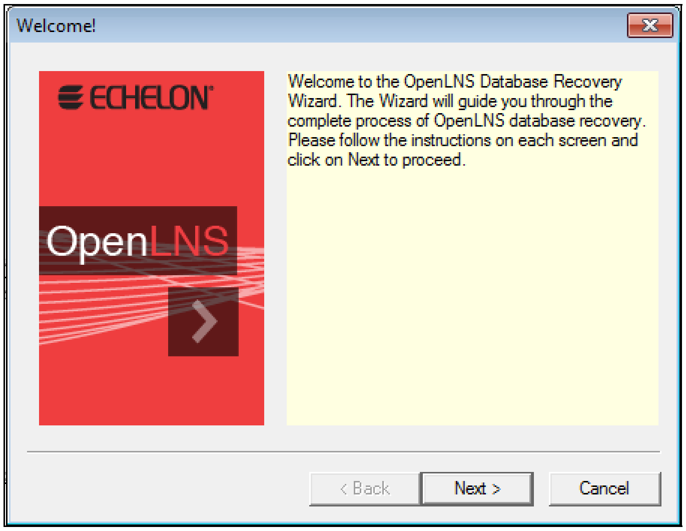
Echelon OpenLNS Commissioning Tool

Please press the service pin on the device to be used to determine the domain ID...

The device must be on the same channel as the LNS Server and not use authentication.

Domain ID: c3

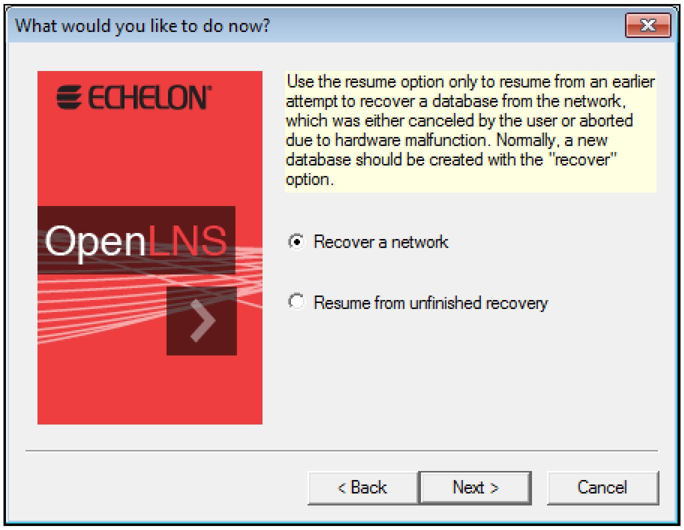
10. The OpenLNS Database Recovery Wizard opens. Click **Next**.



Welcome!

Welcome to the OpenLNS Database Recovery Wizard. The Wizard will guide you through the complete process of OpenLNS database recovery. Please follow the instructions on each screen and click on Next to proceed.

11. **The What Would You Like To Do Now?** dialog box opens. Click **Recover a Network** and click **Next**.



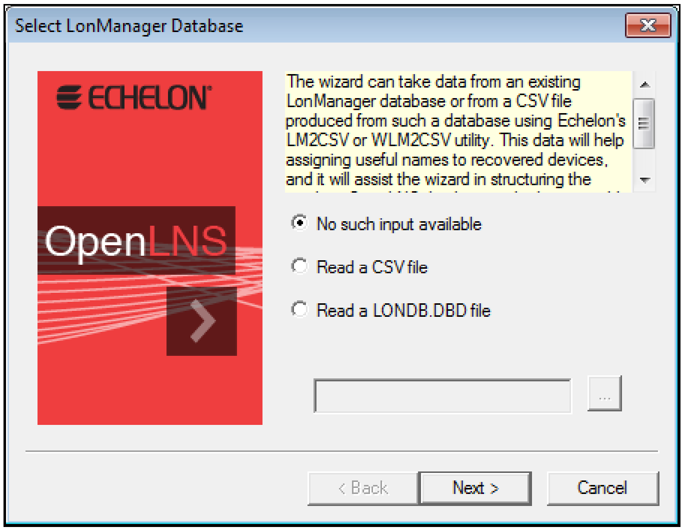
What would you like to do now?

Use the resume option only to resume from an earlier attempt to recover a database from the network, which was either canceled by the user or aborted due to hardware malfunction. Normally, a new database should be created with the "recover" option.

● Recover a network

○ Resume from unfinished recovery

12. The **Select LonManager Database** dialog box opens. If you have a LonManager database or CSV file, select the corresponding option; otherwise, click **No Such Input Available**. Click **Next**



Select LonManager Database

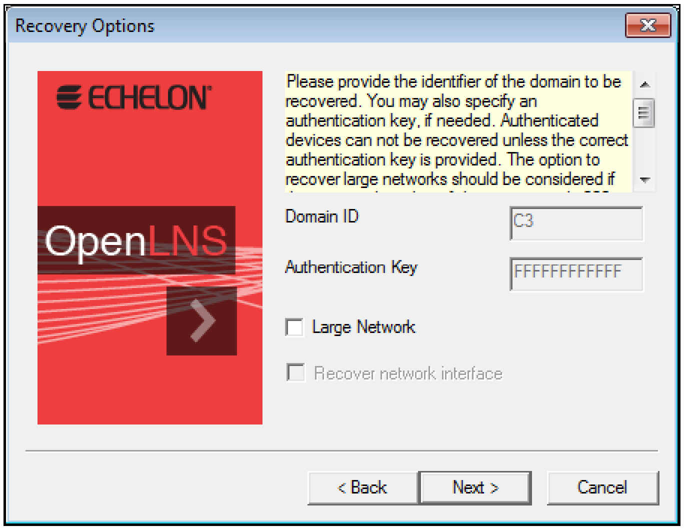
The wizard can take data from an existing LonManager database or from a CSV file produced from such a database using Echelon's LM2CSV or WLM2CSV utility. This data will help assigning useful names to recovered devices, and it will assist the wizard in structuring the

● No such input available

○ Read a CSV file

○ Read a LONDB.DBD file

13. The **Recovery Options** dialog box opens. If you expect your network to have more than 200 devices, select the **Large Network** check box. The **Domain ID** and **Authentication Key** boxes are read-only, as these settings are determined by the input from step 7. Click **Next**.



Recovery Options

Please provide the identifier of the domain to be recovered. You may also specify an authentication key, if needed. Authenticated devices can not be recovered unless the correct authentication key is provided. The option to recover large networks should be considered if

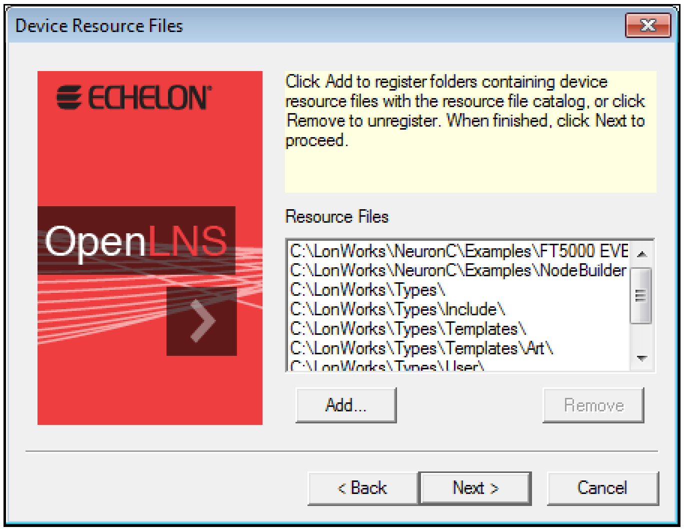
Domain ID: C3

Authentication Key: FFFFFFFFFFFF

☐ Large Network

☐ Recover network interface

14. The **Device Resource Files** dialog box opens. Before proceeding, check if any LONMARK resource files are contained in directories not displayed in this dialog box. If there are any such LONMARK resource files, create a new folder in your LONWORKS **Types\User** folder and then copy the LONMARK resource files to it. To add the device resource files, click **Add**, and then select the folders containing the device resource files and click **Open**, or double-click the folders. Click **Next** when all desired folders containing the device resource files are displayed.

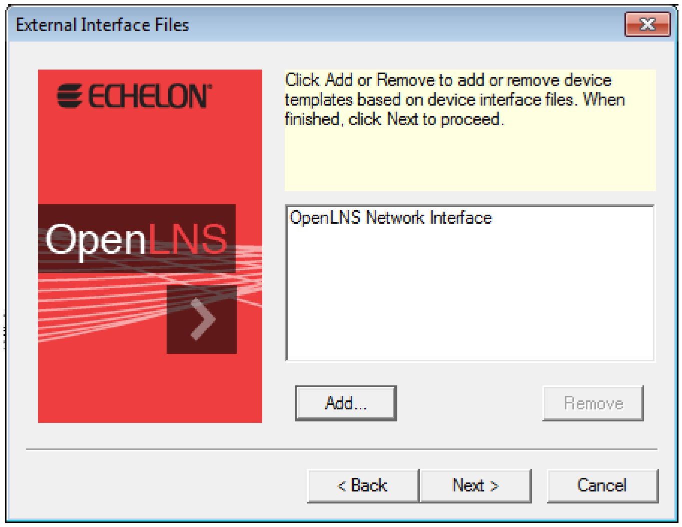

Device Resource Files

Click Add to register folders containing device resource files with the resource file catalog, or click Remove to unregister. When finished, click Next to proceed.

Resource Files

```
C:\LonWorks\NeuronC\Examples\FT5000 EVE
C:\LonWorks\NeuronC\Examples\NodeBuilder
C:\LonWorks\Types\
C:\LonWorks\Types\Include\
C:\LonWorks\Types\Templates\
C:\LonWorks\Types\Templates\Art\
C:\LonWorks\Types\User\
```

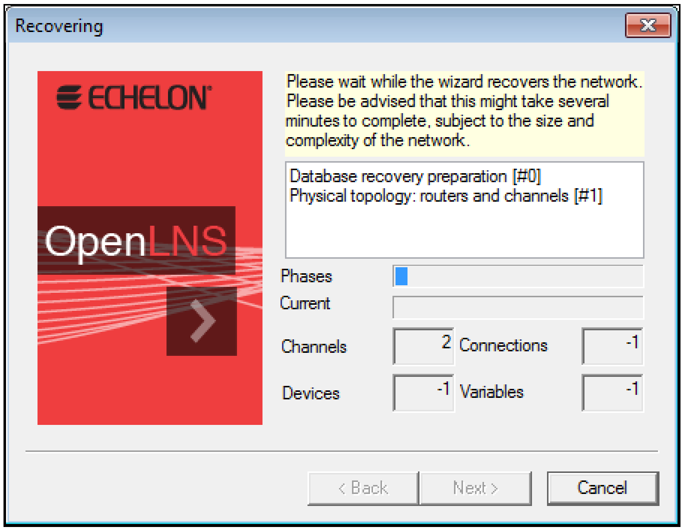
15. The **External Interface Files** dialog box opens. Add or remove device templates based on device interface (XIF) files. When finished, click **Next**.


External Interface Files

Click Add or Remove to add or remove device templates based on device interface files. When finished, click Next to proceed.

OpenLNS Network Interface

16. The **Recovering** dialog box opens. Click **Next** to begin the network recovery. The time of the recovery process depends on the size and complexity of the network. Throughout this process, messages will appear.


Recovering

Please wait while the wizard recovers the network. Please be advised that this might take several minutes to complete, subject to the size and complexity of the network.

```
Database recovery preparation [#0]
Physical topology: routers and channels [#1]
```

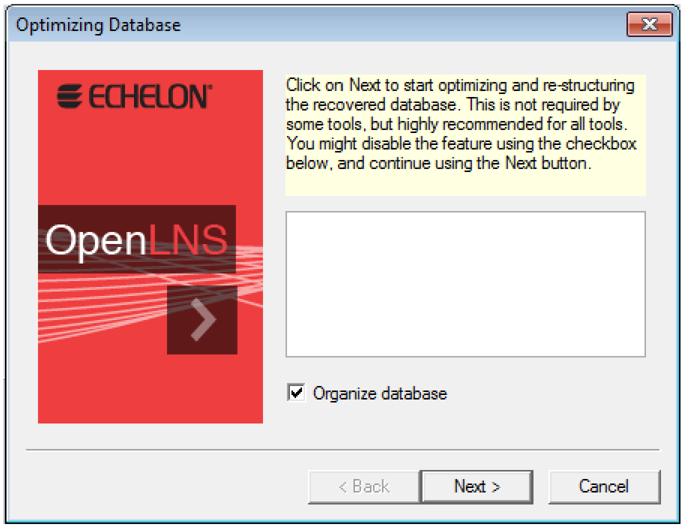
Phases

Current

Channels 2 Connections -1

Devices -1 Variables -1

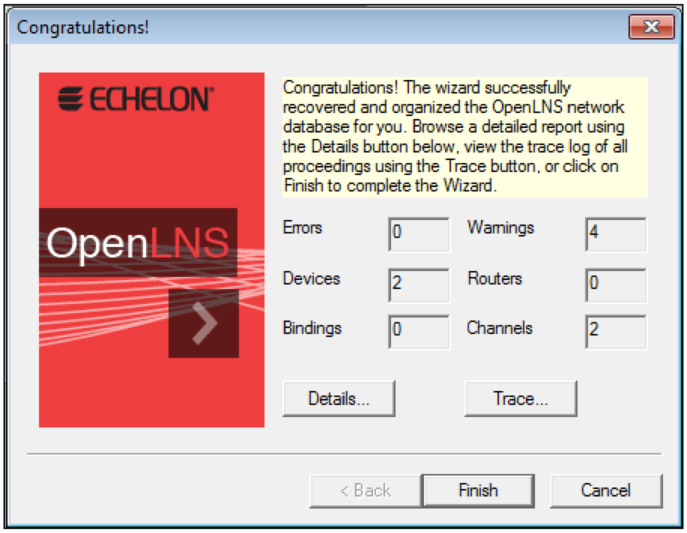
17. When the data recovery is complete, the **Optimizing Database** dialog box opens. Select the **Organize Database** check box and click **Next**.


Optimizing Database

Click on Next to start optimizing and re-structuring the recovered database. This is not required by some tools, but highly recommended for all tools. You might disable the feature using the checkbox below, and continue using the Next button.

☑ Organize database

18. Throughout this process numerous messages will appear. When the database organization is complete, a **Congratulations** dialog box opens. This dialog box displays the number of errors and warnings, and the number of devices, bindings, warnings, routers, and channels found. This window provides valuable information to help you validate your OpenLNS CT network design after it has been synchronized. Verify that the number of devices, routers, bindings, and channels found match your physical network



Congratulations!

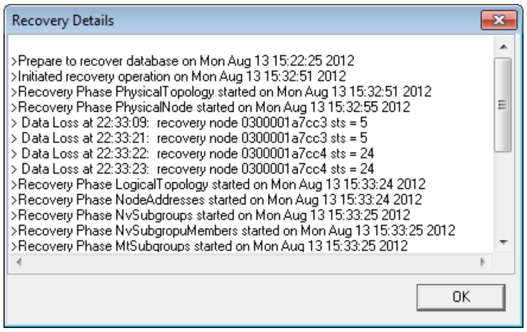
Congratulations! The wizard successfully recovered and organized the OpenLNS network database for you. Browse a detailed report using the Details button below, view the trace log of all proceedings using the Trace button, or click on Finish to complete the Wizard.

Errors: 0 Warnings: 4

Devices: 2 Routers: 0

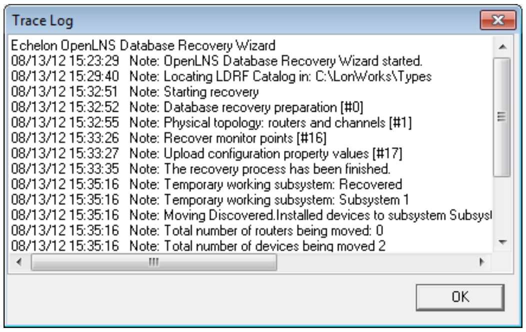
Bindings: 0 Channels: 2

19. If there are any errors or warnings, click **Details** to view the recovery log listing each recovery phase initiated and any problems encountered. Investigate any errors before you set the final network design OnNet. You can also research warnings, but they are not considered as serious as errors. Click **Trace** to view the trace log listing the steps taken by the OpenLNS Database Recovery Wizard to recover your network. The trace log file is placed in **C:\Users\Public\Documents\LonWorks\OpenLnsCt\Databases\**< *network*> or in the network database path you specified in the naming window of the Network Wizard.



Recovery Details

```
>Prepare to recover database on Mon Aug 13 15:22:25 2012
>Initiated recovery operation on Mon Aug 13 15:32:51 2012
>Recovery Phase PhysicalTopology started on Mon Aug 13 15:32:51 2012
>Recovery Phase PhysicalNode started on Mon Aug 13 15:32:55 2012
> Data Loss at 22:33:09:  recovery node 0300001a7cc3 sts = 5
> Data Loss at 22:33:21:  recovery node 0300001a7cc3 sts = 5
> Data Loss at 22:33:22:  recovery node 0300001a7cc4 sts = 24
> Data Loss at 22:33:23:  recovery node 0300001a7cc4 sts = 24
>Recovery Phase LogicalTopology started on Mon Aug 13 15:33:24 2012
>Recovery Phase NodeAddresses started on Mon Aug 13 15:33:24 2012
>Recovery Phase NvSubgroups started on Mon Aug 13 15:33:25 2012
>Recovery Phase NvSubgropuMembers started on Mon Aug 13 15:33:25 2012
>Recovery Phase MtSubgroups started on Mon Aug 13 15:33:25 2012
```

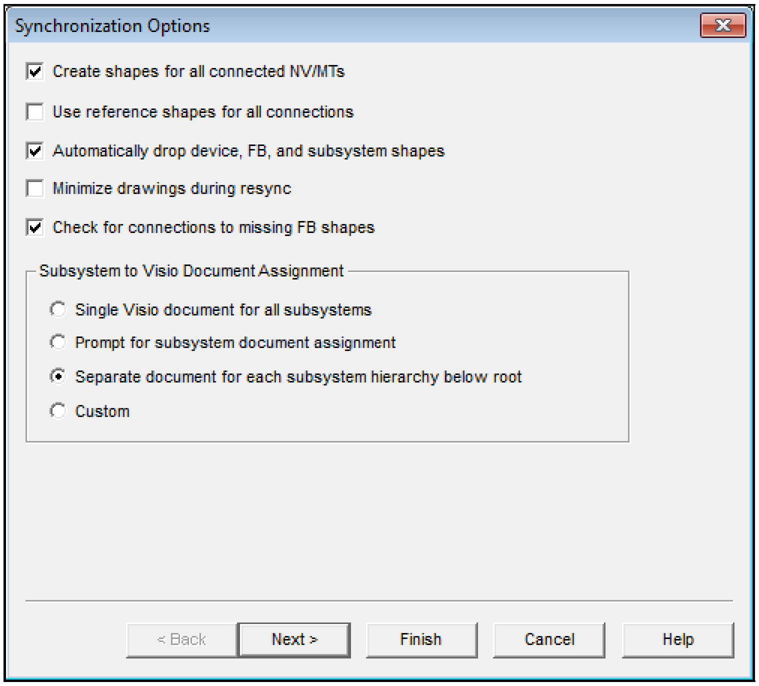


Trace Log

```
Echelon OpenLNS Database Recovery Wizard
08/13/12 15:23:29   Note: OpenLNS Database Recovery Wizard started.
08/13/12 15:29:40   Note: Locating LDRF Catalog in: C:\LonWorks\Types
08/13/12 15:32:51   Note: Starting recovery
08/13/12 15:32:52   Note: Database recovery preparation [#0]
08/13/12 15:32:55   Note: Physical topology: routers and channels [#1]
08/13/12 15:33:26   Note: Recover monitor points [#16]
08/13/12 15:33:27   Note: Upload configuration property values [#17]
08/13/12 15:33:35   Note: The recovery process has been finished.
08/13/12 15:35:16   Note: Temporary working subsystem: Recovered
08/13/12 15:35:16   Note: Temporary working subsystem: Subsystem 1
08/13/12 15:35:16   Note: Moving Discovered.Installed devices to subsystem Subsyst
08/13/12 15:35:16   Note: Total number of routers being moved: 0
08/13/12 15:35:16   Note: Total number of devices being moved 2
```

20. Click **Finish** to continue.

21. The **Synchronization Options** dialog box opens. Select the **Automatically Drop Device, FB, and Subsystem Shapes** check box to automatically add the appropriate SmartShapes from the stencils to the drawing. You can clear the **Create Shapes for all Connected NVs/MTs** check box to simplify the drawing (alternatively, you can add the connection SmartShapes and then hide them temporarily as described in *Hiding and Showing Connector SmartShapes* in Chapter 4, *Designing Networks*. Select the **Use Reference Shapes for all Connections** check box to prevent a drawing with many connections from becoming too complex. Click **Next**.



Synchronization Options

☑ Create shapes for all connected NV/MTs

☐ Use reference shapes for all connections

☑ Automatically drop device, FB, and subsystem shapes

☐ Minimize drawings during resync

☑ Check for connections to missing FB shapes

Subsystem to Visio Document Assignment

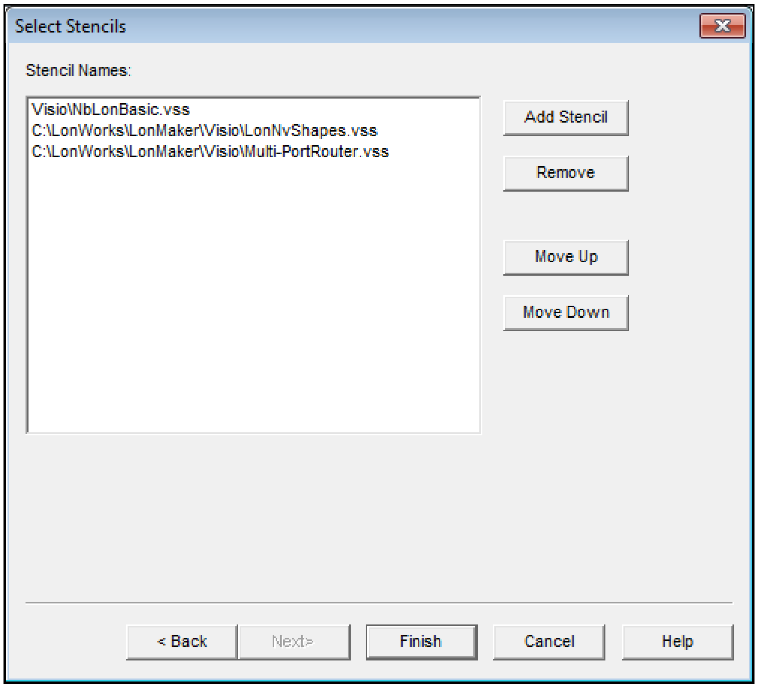
○ Single Visio document for all subsystems

○ Prompt for subsystem document assignment

● Separate document for each subsystem hierarchy below root

○ Custom

22. The **Select Stencils** dialog box opens. Add, remove, or re-order stencils. Click **Finish**



Select Stencils

Stencil Names:

```
Visio\NbLonBasic.vss
C:\LonWorks\LonMaker\Visio\LonNvShapes.vss
C:\LonWorks\LonMaker\Visio\Multi-PortRouter.vss
```

23. The **Synchronization Status** dialog box opens as the synchronization process begins. It then displays the progress of the synchronization. Once "Synchronization Complete" opens, click **OK** to open your OpenLNS CT network.

24. Verify that your OpenLNS CT drawing accurately represents the network. Ensure that all devices on the physical network and all the connections between the devices appear on the drawing. You can refer to the trace log created during the synchronization process. The trace log file is located in **C:\Users\Public\Documents\LonWorks\OpenLnsCt\Databases\**<*network*> or in the network database path you specified in the naming window of the Network Wizard. Make changes to your OpenLNS CT network, if necessary, until it matches the physical network.

25. Compare the recovered OpenLNS network database to your expected results. Because network recovery occurs while OpenLNS CT is OffNet, required changes to device configuration properties are stored in the OpenLNS network database instead of being propagated to the network. Once you have checked the recovered OpenLNS network database and validated your OpenLNS CT network design, change the management mode to OnNet. To do this, click **Add-ins**, click **LonWorks Network**, click **Network Properties**, click the **Network Interface** tab, and under **Management Mode**, select **OnNet.** If changes to device configurations are required, OpenLNS CT will prompt you to resynchronize the network with the OpenLNS network database (recommission devices).

Network Recovery vs. Database Backup

Even though you can use network recovery to rebuild an entire network database, you should not use recovery as a replacement for backing up your OpenLNS CT network design. A backup is faster, more accurate, and more reliable than using network recovery. You should therefore regularly backup your OpenLNS CT network design as the primary means of protecting against drawing and database loss. If your application warrants greater protection, you can use a redundant array of independent disks (RAID) for the OpenLNS Server. For maximum performance and reliability, use a RAID 5 disk array.

Some of the advantages of regularly performing database backups over using network recovery are as follows:

- **Speed.** Creating and restoring an OpenLNS CT backup file is much faster than recovering a network. To recover the network, the OpenLNS Server must scan the network to discover all the devices, upload a small portion of the overall network configuration stored in each device, and then create a drawing and network database based on the data collected from the physical devices. This data collection and reconstruction process generally grows with the complexity of the network (the more devices, network variables, channels, and connections in a network, the more complex the network is considered to be).
- **Accuracy.** Recovery only recovers attributes and properties stored in the devices themselves; information that is not stored in devices, such as device names and the specified hubs and targets in connections, may not be recovered.

  Recovery may fail to create an exact duplicate of the original database. Specifically, if you compare a recovered database to the original database, there will be differences because there are many objects, properties, and attributes that cannot be identified uniquely and unambiguously. For example, the handles assigned to devices, routers, and so on will probably differ, and connection hubs and targets may change.

  Recovery may also be incomplete if the network itself is inconsistent. A network may become inconsistent if, for example, some of the devices have failed or are powered off, or if the network tool failed while updating the configuration of a series of devices. In such a case, some devices would contain new configuration data while others would have the old data. When recovering this network, OpenLNS CT will not be able to distinguish the new configuration data from the old.
- **Reliability**. Although recovery is designed to be as reliable as possible, the successful recovery of a given network depends on that network being fully functional, properly configured, and not saturated by regular network traffic. Additionally, authentication can prevent parts of a network from being recovered.

In the event you cannot restore a network from a database backup or your OpenLNS database and OpenLNS CT drawing files are lost, you can use recovery. However, the following inconsistencies may result in changes to network variable and address table entry configurations:

- If a bound source network variable does not have any associated source or target, it is marked as unbound and removed from the associated connection.
- If a bound dedicated message tag does not have any targets, its address table entry is marked as empty and removed from the associated connection.

- Any address table entries that are not associated with a network variable or message tag (source or target) are marked as empty. A lost source or a lost target may create this scenario. Group use counts are updated as necessary, and group IDs are freed as necessary.
- When recovering a network, OpenLNS CT cannot determine which network variables are hubs and which ones are targets. These relationships are arbitrarily assigned.
- Because the hub and target relationships may change, the recovery may not accurately restore all connection descriptions. If the recovery is unable to determine the correct connection description for a connection, it uses the default connection description.

Resynchronizing an OpenLNS CT Network

You can resynchronize an OpenLNS CT drawing and the physical devices in a network with the OpenLNS network database. OpenLNS CT keeps an OpenLNS CT drawing and its network database synchronized by tracking and processing OpenLNS events. In the case where multiple OpenLNS CT users are simultaneously accessing the same network database, it uses an event log that keeps all copies of the OpenLNS CT drawing synchronized with the OpenLNS network database and with each other.

An OpenLNS CT drawing and its network database, though, can lose synchronization due to a hardware or software failure. When OpenLNS CT detects that an OpenLNS CT drawing and its network database have lost synchronization or that an inconsistency exists in the drawing, it will automatically try to resynchronize the OpenLNS CT drawing to the network database. This resynchronization will be transparent, except that you may notice a slightly lower response time during the resynchronization.

However, the automatic resynchronization may fail or other scenarios may occur in which you need to manually resynchronize the network. For example, some types of changes to the OpenLNS network database by an OpenLNS application other than OpenLNS CT, an improperly saved drawing, or deleted entries in the OpenLNS CT event log can cause an OpenLNS CT drawing to lose synchronization with its OpenLNS network database. You may also need to manually resynchronize physical devices with the OpenLNS network database (to recommission all the devices on the network, for example).

The following sections describe how OpenLNS CT automatically keeps an OpenLNS CT drawing synchronized with its network database when one or more users are making changes to the drawing and how to manually resynchronize the physical network, network database, and the OpenLNS CT drawing when automatic resynchronization fails.

Automatic OpenLNS CT Drawing Synchronization

You can make changes to an OpenLNS CT drawing at the same time that another user is modifying the network, even if the other user is modifying the same drawing page as you. OpenLNS CT uses OpenLNS events and the OpenLNS CT event log described in the following sections to keep the OpenLNS CT drawings synchronized with the OpenLNS network database and with each other. The OpenLNS CT drawings will remain synchronized even as multiple users and applications access and modify the drawings and the network database.

The OpenLNS events and OpenLNS CT event log events enable multiple OpenLNS CT users to stay synchronized, even if they are not operating at the same time. For example, if two users start working on a network and one of the users shuts down their computer to move to another location while the other user continues to work, the first user’s OpenLNS CT will automatically pick up all changes created by the other user once the first OpenLNS CT is restarted—even if the other user is no longer using OpenLNS CT.

While it is possible for multiple users to be modifying the same page of an OpenLNS CT drawing at the same time, if the users are working in the same subsystem, it is possible for one user to delete an

OpenLNS CT object that the other user is trying to modify at the same time. As a result, multiple users will typically be more productive if they are working in different subsystems.

OpenLNS Event Tracking

OpenLNS CT uses OpenLNS events to detect changes to an OpenLNS CT drawing made by OpenLNS applications other than OpenLNS CT, as well as changes in the status of devices as reported by the OpenLNS server. Actions tracked exclusively by OpenLNS events consist of detecting and commissioning application devices and routers. OpenLNS events track other actions that are also tracked in the OpenLNS CT event log described in the next section. These actions include upgrading application devices; creating application devices, routers, subsystems, and connections; deleting application devices, routers, and connections; and renaming application devices, routers, functional blocks, and subsystems. OpenLNS CT can only process OpenLNS events if it is active when the events are generated.

OpenLNS CT Event Log

OpenLNS CT stores every change made to an OpenLNS CT drawing by every user in a log called the OpenLNS CT *event log*.   The OpenLNS CT event log is kept in the OpenLNS database so that all OpenLNS CTs on a network can view the events logged by all other OpenLNS CTs on the same network—including tools that are not active at the same time. All OpenLNS CTs check the log when they open an OpenLNS CT drawing and periodically thereafter (every 10 seconds by default), and update their copies of the OpenLNS CT drawing based on the log entries. The OpenLNS CT event log stores up to a fixed number of entries per drawing file (100 by default). Once the limit is reached, older entries will be deleted to enable new entries to be added.

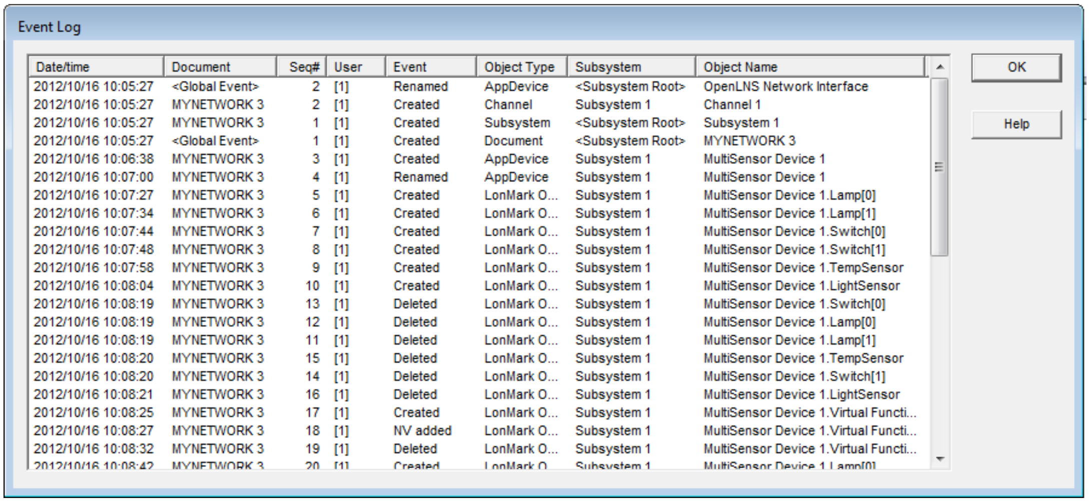
You can change the defaults for how frequently your OpenLNS CT checks the OpenLNS CT event log, and the maximum number of entries it stores in the log. To change the default synchronization settings, select the **Network Properties: Synchronization** options category.

Additional benefits of the OpenLNS CT event log are that it can be used by OpenLNS CT to restore your work if you computer fails while you are updating a drawing, and it can reduce the number of times that you need to resynchronize an OpenLNS CT drawing. You may still need to resynchronize if applications other than OpenLNS CT make changes to the OpenLNS database that affect an OpenLNS CT drawing but are not reported by OpenLNS events, or if log entries are deleted.

You can view the OpenLNS CT event log, and you can export it to an XML file.

Viewing the OpenLNS CT Event Log

You can view the OpenLNS CT event log to see a summary of the changes made by OpenLNS CT users to the objects and subsystems in an OpenLNS CT network. To view the OpenLNS CT event log, click **Add-ins**, point to **Synchronization**, and then click **Display Event Log**.



Date/time	Document	Seq#	User	Event	Object Type	Subsystem	Object Name
2012/10/16 10:05:27	<Global Event>	2	[1]	Renamed	AppDevice	<Subsystem Root>	OpenLNS Network Interface
2012/10/16 10:05:27	MYNETWORK 3	2	[1]	Created	Channel	Subsystem 1	Channel 1
2012/10/16 10:05:27	MYNETWORK 3	1	[1]	Created	Subsystem	<Subsystem Root>	Subsystem 1
2012/10/16 10:05:27	<Global Event>	1	[1]	Created	Document	<Subsystem Root>	MYNETWORK 3
2012/10/16 10:06:38	MYNETWORK 3	3	[1]	Created	AppDevice	Subsystem 1	MultiSensor Device 1
2012/10/16 10:07:00	MYNETWORK 3	4	[1]	Renamed	AppDevice	Subsystem 1	MultiSensor Device 1
2012/10/16 10:07:27	MYNETWORK 3	5	[1]	Created	LonMark O...	Subsystem 1	MultiSensor Device 1.Lamp[0]
2012/10/16 10:07:34	MYNETWORK 3	6	[1]	Created	LonMark O...	Subsystem 1	MultiSensor Device 1.Lamp[1]
2012/10/16 10:07:44	MYNETWORK 3	7	[1]	Created	LonMark O...	Subsystem 1	MultiSensor Device 1.Switch[0]
2012/10/16 10:07:48	MYNETWORK 3	8	[1]	Created	LonMark O...	Subsystem 1	MultiSensor Device 1.Switch[1]
2012/10/16 10:07:58	MYNETWORK 3	9	[1]	Created	LonMark O...	Subsystem 1	MultiSensor Device 1.TempSensor
2012/10/16 10:08:04	MYNETWORK 3	10	[1]	Created	LonMark O...	Subsystem 1	MultiSensor Device 1.LightSensor
2012/10/16 10:08:19	MYNETWORK 3	13	[1]	Deleted	LonMark O...	Subsystem 1	MultiSensor Device 1.Switch[0]
2012/10/16 10:08:19	MYNETWORK 3	12	[1]	Deleted	LonMark O...	Subsystem 1	MultiSensor Device 1.Lamp[0]
2012/10/16 10:08:19	MYNETWORK 3	11	[1]	Deleted	LonMark O...	Subsystem 1	MultiSensor Device 1.Lamp[1]
2012/10/16 10:08:20	MYNETWORK 3	15	[1]	Deleted	LonMark O...	Subsystem 1	MultiSensor Device 1.TempSensor
2012/10/16 10:08:20	MYNETWORK 3	14	[1]	Deleted	LonMark O...	Subsystem 1	MultiSensor Device 1.Switch[1]
2012/10/16 10:08:21	MYNETWORK 3	16	[1]	Deleted	LonMark O...	Subsystem 1	MultiSensor Device 1.LightSensor
2012/10/16 10:08:25	MYNETWORK 3	17	[1]	Created	LonMark O...	Subsystem 1	MultiSensor Device 1.Virtual Functi...
2012/10/16 10:08:27	MYNETWORK 3	18	[1]	NV added	LonMark O...	Subsystem 1	MultiSensor Device 1.Virtual Functi...
2012/10/16 10:08:32	MYNETWORK 3	19	[1]	Deleted	LonMark O...	Subsystem 1	MultiSensor Device 1.Virtual Functi...
2012/10/16 10:08:42	MYNETWORK 3	20	[1]	Created	LonMark O...	Subsystem 1	MultiSensor Device 1.Lamp[0]

You can sort the OpenLNS CT event log by any of the column headers. To sort by a column header, click the header. Click the header again to toggle the order in which the data in that column is listed (ascending or descending). The columns contain the following data:

Date/Time The date and time the event occurred. By default, the OpenLNS CT Event Log lists the changes by date/time in ascending order.

Document The name of the document in which the change occurred.

Sequence # The sequence number of the event. Each OpenLNS CT drawing has its own sequence numbers.

Event The type of event that occurred.

User Name The user name, or if not specified, the application instance (in square brackets).

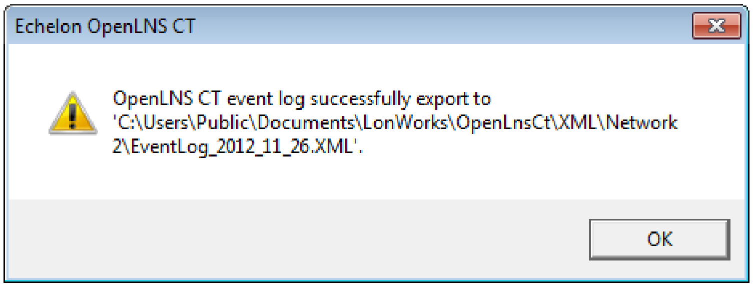
Object Type The type of the object affected by the event.

Subsystem The subsystem in which the event occurred.

Object Name The name of the object affected by the change. If the object is no longer defined in the OpenLNS database, its handle will be listed.

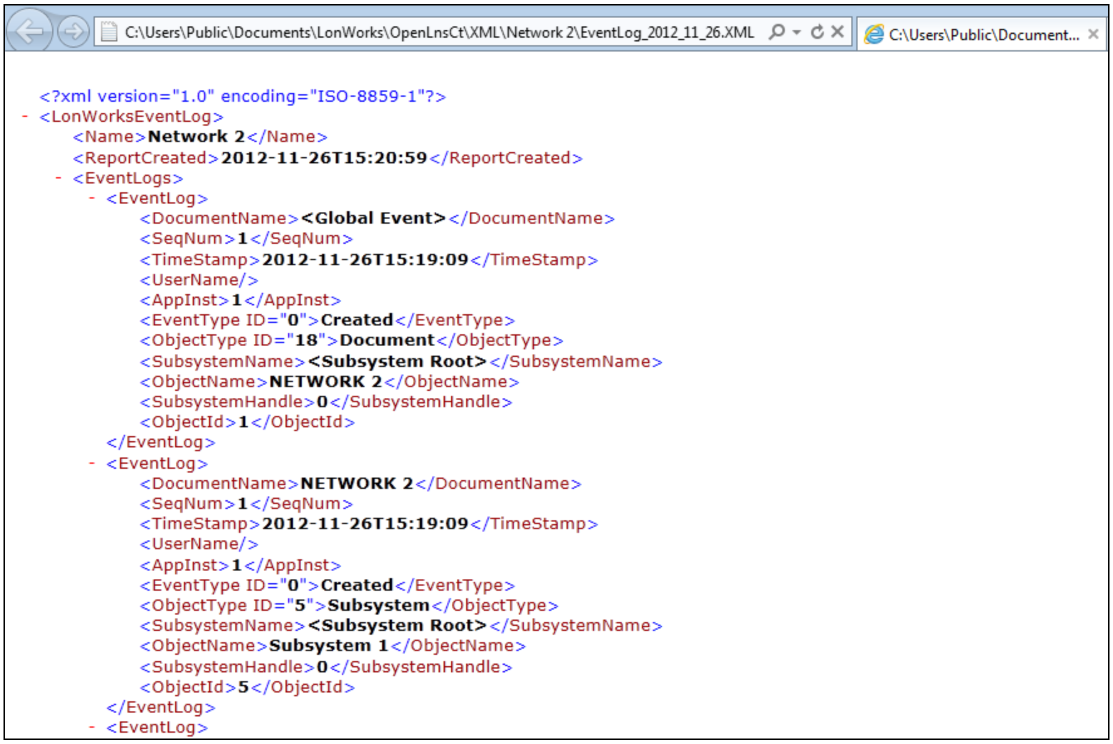
Exporting the OpenLNS CT Event Log

You can export the OpenLNS CT event log to an XML file to work with the log information in another application. To export the OpenLNS CT event log to an XML file, click **Add-ins**, point to **Synchronization**, and then click **Export Event Log**. A message opens stating that the OpenLNS CT event log has been successfully exported, and displaying the full path of the XML file. The XML file will be exported to **C:\Users\Public\Documents\LonWorks\OpenLnsCt\XML\**<*network name*>\EventLog_*year_month_date*_[*version*].



OpenLNS CT event log successfully export to 'C:\Users\Public\Documents\LonWorks\OpenLnsCt\XML\Network 2\EventLog_2012_11_26.XML'.

To view the exported OpenLNS CT event log, find the XML file in **C:\Users\Public\Documents\LonWorks\OpenLnsCt\XML***network name* and double-click it. The file opens in your default XML viewing application, which is Internet Explorer unless you change it. You can also open the XML file from a text or XML editor or report generator.


C:\Users\Public\Documents\LonWorks\OpenLnsCt\XML\Network 2\EventLog_2012_11_26.XML
C:\Users\Public\Document...

```
  <?xml version="1.0" encoding="ISO-8859-1"?>
- <LonWorksEventLog>
      <Name>Network 2</Name>
      <ReportCreated>2012-11-26T15:20:59</ReportCreated>
    - <EventLogs>
      - <EventLog>
            <DocumentName><Global Event></DocumentName>
            <SeqNum>1</SeqNum>
            <TimeStamp>2012-11-26T15:19:09</TimeStamp>
            <UserName/>
            <AppInst>1</AppInst>
            <EventType ID="0">Created</EventType>
            <ObjectType ID="18">Document</ObjectType>
            <SubsystemName><Subsystem Root></SubsystemName>
            <ObjectName>NETWORK 2</ObjectName>
            <SubsystemHandle>0</SubsystemHandle>
            <ObjectId>1</ObjectId>
        </EventLog>
      - <EventLog>
            <DocumentName>NETWORK 2</DocumentName>
            <SeqNum>1</SeqNum>
            <TimeStamp>2012-11-26T15:19:09</TimeStamp>
            <UserName/>
            <AppInst>1</AppInst>
            <EventType ID="0">Created</EventType>
            <ObjectType ID="5">Subsystem</ObjectType>
            <SubsystemName><Subsystem Root></SubsystemName>
            <ObjectName>Subsystem 1</ObjectName>
            <SubsystemHandle>0</SubsystemHandle>
            <ObjectId>5</ObjectId>
        </EventLog>
      - <EventLog>
```

The exported XML version of the OpenLNS CT event log contains the following information for each log entry.

DocumentName Name of the OpenLNS CT drawing in which the event occurred.

SeqNum The unique ID (within the OpenLNS CT drawing) of the event.

TimeStamp Date and time at which event occurred in the format *year-month-date-time*.

UserName The user who generated the event.

AppInst The instance of the OpenLNS CT application that generated the event.

EventType ID The type of event that occurred (for example, created, deleted, renamed, SmartShape moved, connection added).

Object Type ID The type of object modified by the event (for example, AppDevice, Channel, Router, Subnet, NSI, Subsystem, Device template, LonMark Object).

SubsystemName The name of the subsystem in which the object is defined. This field can be empty if the object spans multiple subsystems (for example, channels), or if the subsystem no longer exists.

ObjectName The name of the object that was modified. If the object no longer exists, the name will be "**Object ID:**<*n*>" where <*n*> is the handle of the object. For events related to network variables and message tags, the name will be that of their functional block.

SubsystemHandle The handle of the subsystem in which the object is defined. Can be 0 for objects that span multiple subsystems.

ObjectId The handle of the object that was modified. For events related to network variables and message tags, the object ID will be the index of their functional block.

DeviceHandle The handle of the device in which the object is defined. This field only applies to events related to network variables, message tags, and functional blocks.

NvHandle The index of the network variable in the device. This field only applies to events related to network variables.

Mthandle The index of the message tag in the device. This field only applies to events related to message tags.

Manual Network Resynchronization

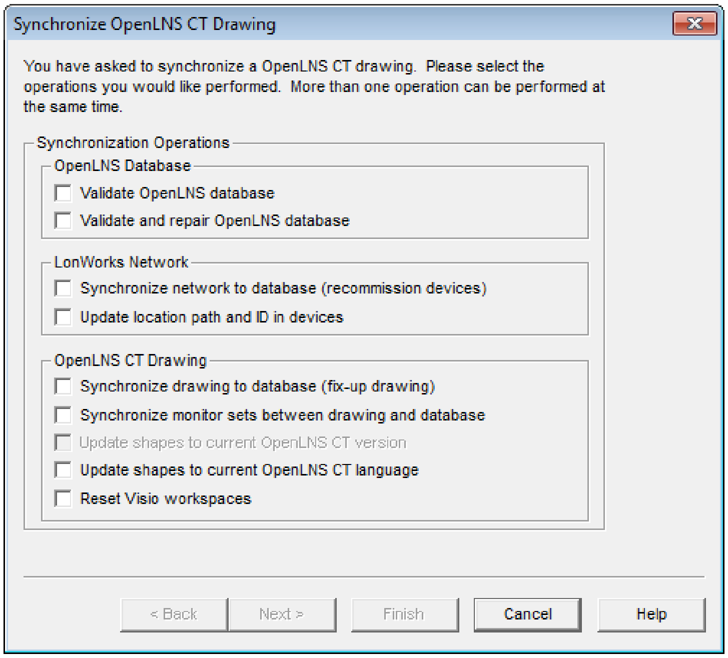
You may need to manually resynchronize the OpenLNS network database, the network, or the OpenLNS CT drawing if automatic resynchronization fails or if one the following scenarios occur:

- An OpenLNS application other than OpenLNS CT makes changes to the OpenLNS network database that are not tracked by OpenLNS events.
- An OpenLNS CT drawing is saved improperly.
- Log entries are deleted from the OpenLNS CT event log before they are applied to an OpenLNS CT drawing
- Changes are made to an OpenLNS CT drawing in between creating and restoring a hot backup of the network database.
- An OpenLNS CT backup file containing an OpenLNS network database and OpenLNS CT drawing files is restored and devices need to be recommissioned.
- An OpenLNS CT network design is created using network recovery.

To manually resynchronize an OpenLNS CT network, follow these steps:

1. Back up the OpenLNS CT network following the steps in *Backing up an OpenLNS CT Network Design* in this chapter. This enables you to restore the OpenLNS CT network design in the event a serious resynchronization error occurs.
2. In the OpenLNS CT drawing, click **Add-ins**, point to **Synchronization**, and then click **Resynchronize**. The **Synchronize OpenLNS CT Drawing** dialog opens.

   If you did not back up the network and changes have been made since the last backup, a message opens recommending that you back up the network before proceeding. If this occurs, you should click **No**, close the drawing, and back up the network.



Synchronize OpenLNS CT Drawing

You have asked to synchronize a OpenLNS CT drawing. Please select the operations you would like performed. More than one operation can be performed at the same time.

Synchronization Operations

OpenLNS Database

□ Validate OpenLNS database

□ Validate and repair OpenLNS database

LonWorks Network

□ Synchronize network to database (recommission devices)

□ Update location path and ID in devices

OpenLNS CT Drawing

□ Synchronize drawing to database (fix-up drawing)

□ Synchronize monitor sets between drawing and database

□ Update shapes to current OpenLNS CT version

□ Update shapes to current OpenLNS CT language

□ Reset Visio workspaces

3. Choose the synchronization operation(s) you want to perform by selecting the following options.

OpenLNS Database — The network database resynchronization options are only available if OpenLNS CT is running on the same computer as the OpenLNS Server.

Validate OpenLNS Database — Validates the structure of the OpenLNS network database.

Validate and Repair OpenLNS Database — Validates the structure of the OpenLNS network database and repairs the database if any errors are detected.

LONWORKS Network

Synchronize Network to Database (recommission devices) — Commissions the physical devices in the LONWORKS network to make them consistent with the OpenLNS network database. To use this option, OpenLNS CT must be OnNet.

Update Location Path and ID in Devices — Stores the current subsystem path in the **SCPTlocation** configuration property of application devices that have a **SCPTlocation** configuration property, and stores the current subsystem ID in the **Location** field of application devices and routers. This option is only available if at least one of the check boxes in the **Network Properties: Recovery** options category is selected.

If an application device and its functional blocks span multiple subsystems, OpenLNS CT updates the **SCPTlocation** configuration property with the subsystem path in which the device SmartShape is located.

The subsystem ID is the first 15 bits of the OpenLNS subsystem handle (in a very large network with many subsystems where more than 32,767 subsystems have been created and deleted, this could result in multiple unique subsystems having the same ID; however, this is very unlikely).

OpenLNS CT Drawing

Synchronize Drawing to Database (fix-up Drawing)
Updates the OpenLNS CT drawing to be consistent with the OpenLNS database.

Synchronize Monitor Sets between Drawing and Database
Updates the monitor sets in the OpenLNS CT drawing to be consistent with the OpenLNS database. A monitor set is a group of point definitions that OpenLNS CT uses to monitor and control network variables. If the type or format of a monitored point has changed, the value displayed will be updated when the point is monitored with OpenLNS CT.

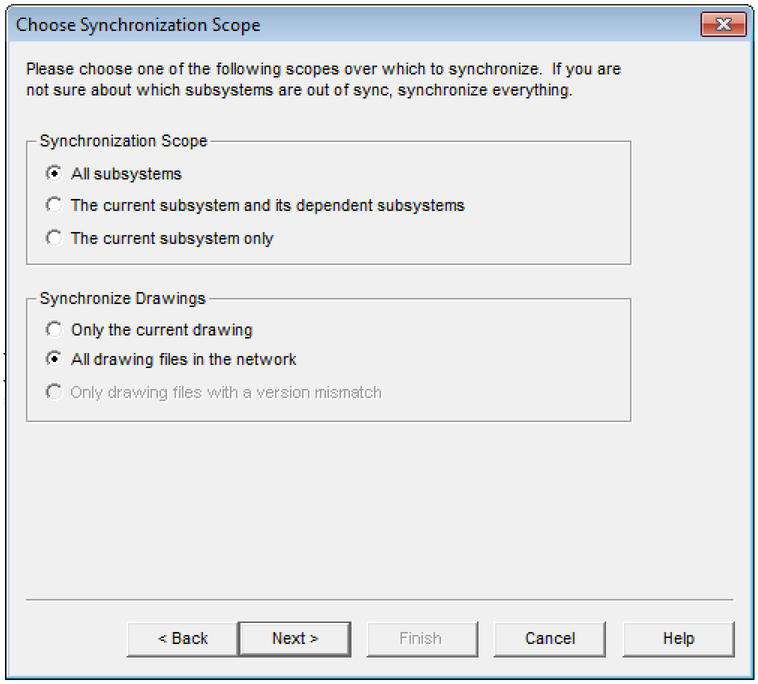
Update Shapes to Current OpenLNS CT Version
Updates SmartShapes to the current version of OpenLNS CT if the OpenLNS CT drawing was originally created with an earlier version of OpenLNS CT.

Update Shapes to Current OpenLNS CT Language
Updates SmartShape menus to current OpenLNS CT language. For example, if you open a network with the German version of OpenLNS CT that was created with the English version, this option will convert the SmartShape menus to German. All the dialogs are in the language of the respective version of OpenLNS CT. The language conversion may take some time; therefore, this option should not be used frequently.

Reset Visio Workspaces
Corrects instances in which Visio has added multiple documents and stencils to the drawing in an OpenLNS CT network. For each drawing in the network, OpenLNS CT opens the drawing file, closes all other open drawings, opens the standard set of stencils, and saves the drawing as a workspace.

This option also automatically upgrades the drawings in the OpenLNS CT network to the current version of Visio.

4. If you are only validating or repairing the OpenLNS network database, click **Finish** to begin the resynchronization process and skip to step 16 in which the **Synchronization Status** dialog opens. Otherwise, click **Next**. The **Synchronization Scope** dialog box opens.



Choose Synchronization Scope

Please choose one of the following scopes over which to synchronize. If you are not sure about which subsystems are out of sync, synchronize everything.

Synchronization Scope

● All subsystems

○ The current subsystem and its dependent subsystems

○ The current subsystem only

Synchronize Drawings

○ Only the current drawing

● All drawing files in the network

○ Only drawing files with a version mismatch

5. Select the scope of the selected synchronization operation(s) by choosing which subsystems and drawing files to resynchronize. If you know that the current subsystem is the only one needing resynchronization, resynchronize just that subsystem. If you aren't sure which subsystem needs to resynchronized, resynchronize all subsystems. The same logic applies if your OpenLNS CT network contains multiple drawing files. If you know which drawing file requires resynchronization, resynchronize just that drawing to save time. However, if you aren't sure which drawing file needs to be resynchronized, resynchronize all the drawings.

Synchronization Scope

Select which subsystems to resynchronize. You have three choices:

- **All Subsystems**. All the subsystems in the OpenLNS CT network design are resynchronized. This is the default.
- **The Current Subsystem and Dependent Subsystem**. Only the current subsystems and any subsystems nested within in it are resynchronized.
- **The Current Subsystem Only**. Only the current subsystem is resynchronized.

Synchronize Drawings

Select which drawings files to resynchronize. You have three choices:

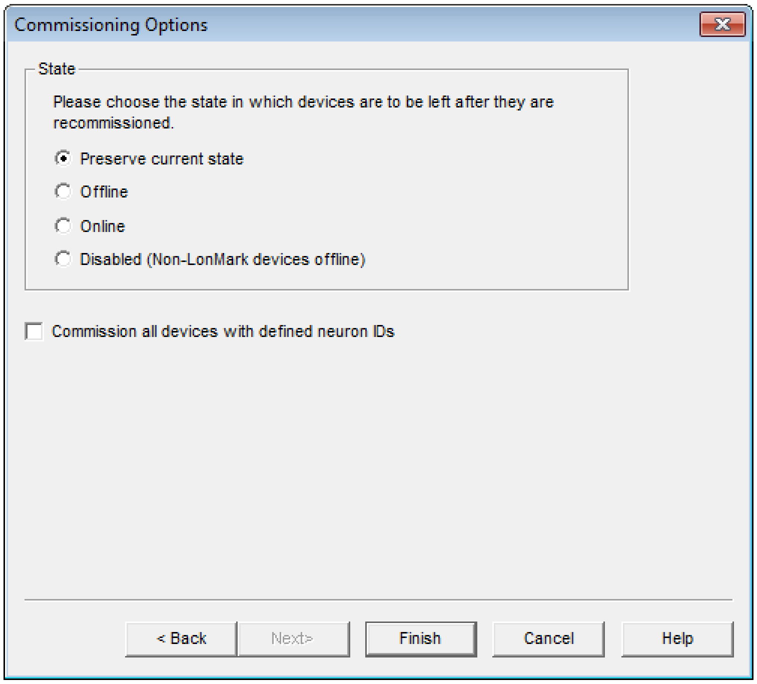
- **Only the Current Drawing**. Only the current drawing file is resynchronized.
- **All Drawing Files in the Network**. All the drawing files in the OpenLNS CT network are resynchronized.
- **Only Drawing Files with a Version Mismatch**. Only drawing files with IDs not matching the version number of the network database are synchronized. This option is only available if you are resynchronizing all subsystems.

6. If you are re-commissioning devices (you selected the **Synchronize Network to Database** check box in the **Synchronize OpenLNS CT Drawing** dialog), click **Next**. The **Commissioning Options** dialog box opens.

Otherwise, if you are synchronizing the network or the OpenLNS CT drawing to the network database, click **Next** and skip to step 10 in which the **Synchronization Options** dialog opens.

If you are performing any other synchronization operation, click **Finish** to begin resynchronization. Skip to step 16 in which the **Synchronization Status** dialog opens.

7. If you are recommissioning devices, the **Commissioning Options** dialog box opens.



Commissioning Options

State

Please choose the state in which devices are to be left after they are recommissioned.

● Preserve current state
○ Offline
○ Online
○ Disabled (Non-LonMark devices offline)

□ Commission all devices with defined neuron IDs

Select the state of the devices after they have been recommissioned. You have four choices:

- **Preserve Current State**. The devices will be restored to the state they were in prior to being recommissioned. This is the default.
- **Offline**. All devices will be left in the offline state. The behavior in the offline state depends on the device. A Neuron-hosted device, for example, will not run its application. You can set all device offline so that you can recommission the system incrementally. Resetting a device in the offline state puts the device in the online state.
- **Online**. All devices will be put in the online state. The behavior in the online state depends on the device. A Neuron-hosted device, for example, will run its application after it has been recommissioned, sending and responding to network and external inputs.
- **Disabled**. All devices with a Node Object functional block will be disabled; all other devices will be placed offline. Selecting **Disabled** keeps the device's application disabled even if the device is reset.

Select the **Commission all Devices with Defined Neuron IDs** check box to recommission all the previously commissioned devices as well as any unconfigured devices within the selected synchronization scope.

8. If you are resynchronizing the OpenLNS CT drawing, click **Next**. The **Choose Shapes and Items to Update** dialog box opens if you selected the **Update Shapes to Current OpenLNS CT Version** check box in the **Synchronize OpenLNS CT Drawing** dialog.

Otherwise, the **Synchronization Options** dialog opens. Skip to step 10.

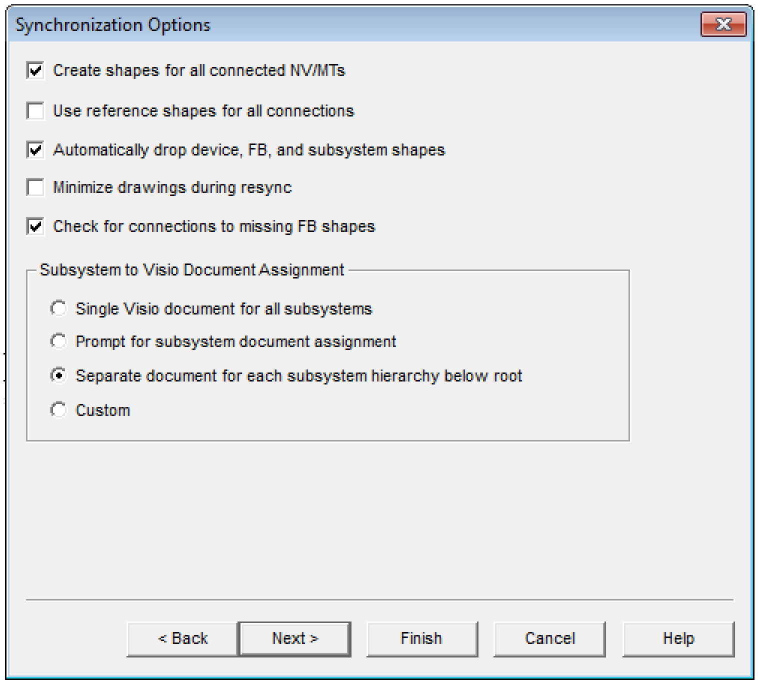
If you are not resynchronizing the OpenLNS CT drawing, click **Finish** to begin resynchronization. Skip to step 16 in which the **Synchronization Status** dialog opens.

9. Select the following options and then click **Next**.

Shapes
All SmartShapes that have newer versions available than the ones currently in the drawing are listed. Select the SmartShapes to update.

Items
Select **Menu** to update the shortcut menus for the selected SmartShapes. Set **Appearance** to have the appearance of the SmartShapes updated.

10. The **Synchronization Options** dialog opens.



Synchronization Options

☑ Create shapes for all connected NV/MTs

□ Use reference shapes for all connections

☑ Automatically drop device, FB, and subsystem shapes

□ Minimize drawings during resync

☑ Check for connections to missing FB shapes

Subsystem to Visio Document Assignment

○ Single Visio document for all subsystems

○ Prompt for subsystem document assignment

● Separate document for each subsystem hierarchy below root

○ Custom

11. Select the options to be applied during the synchronization of the OpenLNS CT drawing to the network database.

Create Shapes for all Connected NVs/MTs
Adds SmartShapes to functional blocks for all connected network variables or message tags created during resynchronization. If this check box is cleared, OpenLNS CT uses generic connections when possible. It only creates network variable or message tag SmartShapes for connections to supernodes and for those functional blocks that already contain network variable or message tag SmartShapes.

Use Reference Shapes for all Connections
Represents all network variable connections created during resynchronization with reference connection SmartShapes. If this check box is cleared, OpenLNS CT uses standard connection SmartShapes for network variable connections in the same subsystem. OpenLNS CT always uses reference connection SmartShapes for connections that span subsystems.

Automatically Drop Device, FB, and Subsystem Shapes
Automatically adds SmartShapes to the drawing for all devices, functional blocks, and subsystems created during resynchronization. If this check box is cleared, OpenLNS CT prompts the user to select whether a SmartShape should be added for a device, functional block, or subsystem created during resynchronization.

Minimize Drawings
Minimizes all the drawings in the OpenLNS CT network during

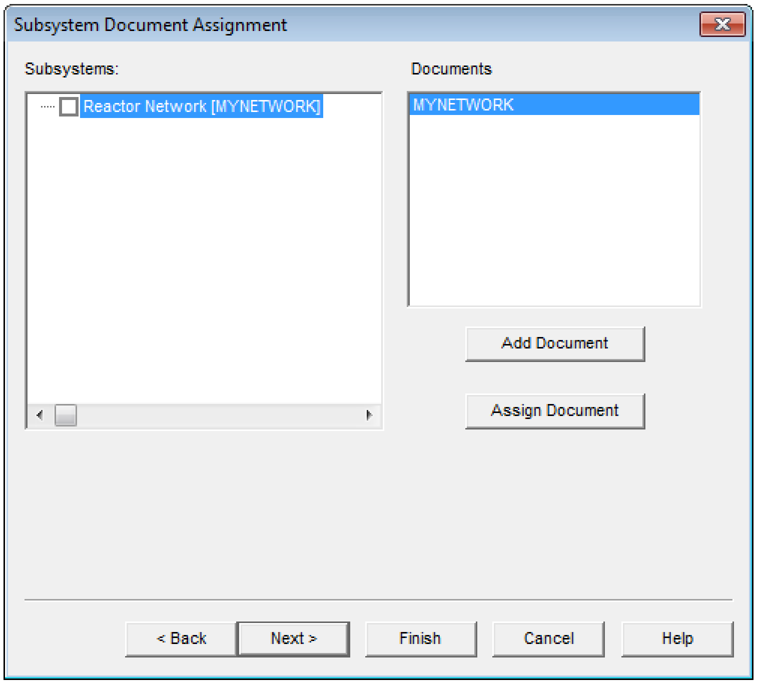
during Resync — resynchronization. This speeds up resynchronization, as it reduces the time Visio spends updating the drawings.

Check for Connections for Missing FB Shapes — Enables OpenLNS CT to check for connections to functional blocks that are in the network database but are not represented by functional block SmartShapes in the OpenLNS CT drawing.

Subsystem to Visio Document Assignment — Select in which Visio drawing file to place the subsystems in your network. You have four choices:

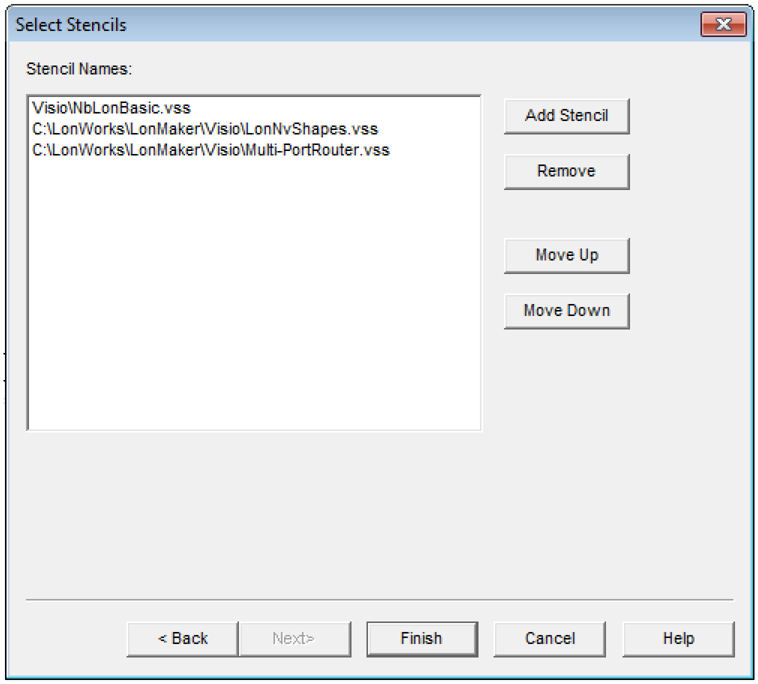
- **Single Visio Document for all Subsystems**. Places all subsystems in a single Visio document.
- **Prompt for Subsystem Document Assignment**. Prompts you to specify the Visio document for each subsystem.
- **Separate Document for each Subsystem Hierarchy below Root**. Places each subsystem hierarchy below the top-level subsystem in a separate, new Visio document.
- **Custom**. Opens the **Subsystem Document Assignment** dialog. In this dialog, you specify the Visio document for each subsystem.

12. Click **Next**. If you selected the **Custom** subsystem document assignment option, the **Subsystem Document Assignment** dialog opens. Otherwise, skip to step 15 in which the **Select Stencils** dialog opens.



13. Customize the drawing document assignment of subsystems found during resynchronization. Subsystems are listed in the left hand pane listed under **Subsystems**, while the Visio drawing documents are listed in the right hand pane listed under **Documents**. To assign a subsystem, select the subsystem name, select the desired document name, and then click **Assign Document**. To add additional drawing documents, click **Add Document**.

14. Click **Next**. The **Select Stencils** dialog opens.

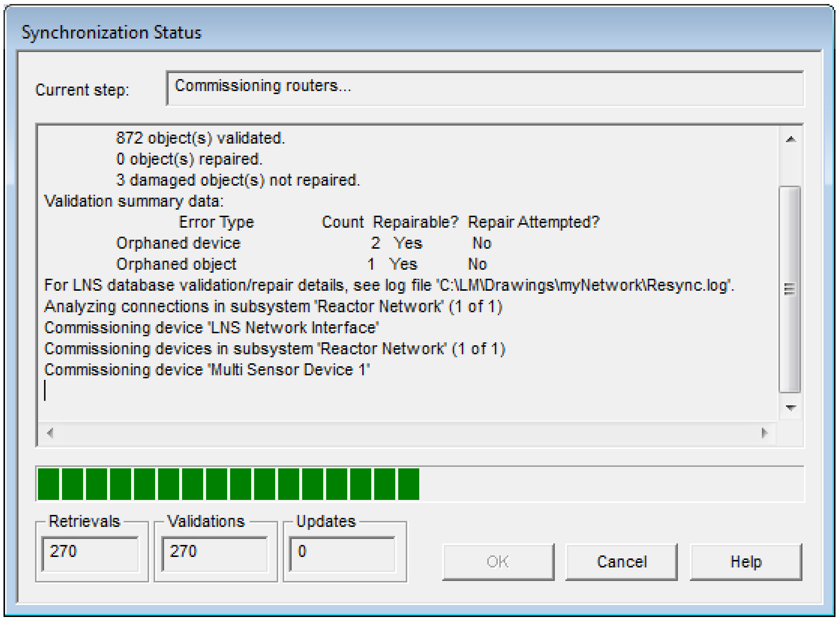


```
Visio\NbLonBasic.vss
C:\LonWorks\LonMaker\Visio\LonNvShapes.vss
C:\LonWorks\LonMaker\Visio\Multi-PortRouter.vss
```

15. Specify the stencils that will be scanned for master SmartShapes during resynchronization. When objects are found within the OpenLNS network database that are not already represented in the drawing, OpenLNS CT scans the listed stencils and automatically adds the appropriate SmartShape to the drawing. If the object's particular master SmartShape is not found, an appropriate generic master SmartShape (e.g., a LONWORKS Device SmartShape) will be used.

    To add additional stencils, click **Add Stencil**. To remove a stencil, select the desired stencil and then click **Remove**.

16. Click **Finish**. Resynchronization begins and the **Synchronization Status** dialog opens.



Current step: Commissioning routers...

```
          872 object(s) validated.
          0 object(s) repaired.
          3 damaged object(s) not repaired.
Validation summary data:
                    Error Type           Count  Repairable?  Repair Attempted?
          Orphaned device                    2   Yes          No
          Orphaned object                    1   Yes          No
For LNS database validation/repair details, see log file 'C:\LM\Drawings\myNetwork\Resync.log'.
Analyzing connections in subsystem 'Reactor Network' (1 of 1)
Commissioning device 'LNS Network Interface'
Commissioning devices in subsystem 'Reactor Network' (1 of 1)
Commissioning device 'Multi Sensor Device 1'
```

Retrievals: 270
Validations: 270
Updates: 0

This dialog displays the status of the resynchronization. Resynchronization events are listed in order as they occur. Click **OK** when the resynchronization is complete. The resynchronization events, including any errors, are shown in the status pane and recorded in a log file (**Resync.log**). The resync log is stored in the drawing directory of the current OpenLNS CT network design.

17. If the OpenLNS CT drawing is being synchronized to match the network database and the **Automatically Drop Device, FB, and Subsystem Shapes** check box was cleared, the **Missing Shape** dialog opens for each object in the network database that is not represented in the OpenLNS CT drawing.

    a. Click **OK**. The **Missing Shape** dialog closes and you are returned to the OpenLNS CT drawing.

    b. Drag a single SmartShape from an OpenLNS CT stencil to the OpenLNS CT drawing to represent the missing SmartShape. OpenLNS CT assigns the SmartShape to the specified object in the OpenLNS network database.

    c. Click **Add-ins**, point to **Synchronization**, and then click **Resynchronize** to re-open the **Missing Shape** dialog and assign the next missing SmartShape.

    d. Repeat steps a – c for each missing SmartShape. Once all missing SmartShapes have been added and the connections have been updated, resynchronization will be complete.

Tips:

- The first time you resynchronize an OpenLNS CT drawing created with a previous version of OpenLNS CT, most connections that span multiple documents will automatically be deleted and then re-created. This speeds up any subsequent resynchronizations.
- The first time you resynchronize an OpenLNS CT drawing created with an older version of Visio, select the **Reset Visio Workspaces** option in the **Synchronize OpenLNS CT Drawing** dialog. This automatically saves the documents in your drawing to the Visio 2010 or Visio 2003 format. Otherwise, OpenLNS CT prompts you whether to save the documents in your drawing in Visio 2002 format or the current Visio format. If you answer Visio 2002, OpenLNS CT will continue to prompt you every time you resynchronize the drawing.
- If you have stencils from an older version of Visio that you want to use, open and save the stencils individually in Visio 2010 or Visio 2003. Otherwise, Visio will prompt you each time the drawing is closed as to whether to save the changes to the stencil, but it won’t save any changes because it opens these stencils as read-only.

Refreshing the OpenLNS CT Network

OpenLNS CT caches data from the OpenLNS network database to improve performance. If another OpenLNS application or plug-in updates data that has been cached by OpenLNS CT and does not generate an appropriate OpenLNS event, the OpenLNS CT cache may lose synchronization with the OpenLNS network database. You can force OpenLNS CT to process and release the cache by clicking **Add-ins**, pointing to **Synchronization**, and then clicking **Refresh**.

Merging OpenLNS CT Networks

You can initially install a large LONWORKS network as a number of smaller independent networks—each with their own network database—and then later merge the networks into one network. Examples of installations that may use this technique include the following scenarios:

- **Multiple systems**. When multiple systems are being installed, the installers for each system want to install and debug their installation without interference from other systems. When each system has been correctly installed, they can be merged into a single network.
- **Incomplete infrastructure**. Portions of a building, such as the floors of a high-rise, may need to be installed before connectivity exists for the entire site (before there is a backbone channel connecting the floors together, for instance). You can install each floor of the network as part of a single network using a portable network server under a sequential network installation scenario, in

which no permanent server is needed. In most cases, however, it may be easier (if not necessary) to install each floor as a separate network. Once the entire site has connectivity, the individual floor networks can be merged into a single network representing the high rise.

To support these scenarios, OpenLNS CT provides the ability to merge OpenLNS CT networks.

Network Merge Considerations

When you create a new LONWORKS network that will entail merging individual networks, minimize the number of objects in the top-level subsystem. When you merge two OpenLNS CT network designs, the merge process will be disruptive to the operation of the networks. For example, devices in the source network will be assigned new subnet/node addresses, connections will be recreated, and authenticated devices will have authentication disabled for a period of time. Devices in the destination network will generally be impacted less, but they can be affected if their connections are modified to accommodate the new devices from the source networks.

Limitations

The network merge process has several limitations:

- **Server support only**. You can only merge networks from a local client. You cannot merge networks from a remote client.
- **Single direction only**. You cannot stop the merge process once it has begun. You can, however, restore a network from an OpenLNS CT backup file, if needed.
- **Single root subsystem**. You can only merge networks that have a single top-level subsystem (this is the default for OpenLNS CT drawings). If the network has more than one top-level subsystem, as may be the case when using an OpenLNS network database created by another network tool, the pre-merge utility will fail. You must remove or relocate all but one of the top-level subsystems, before retrying the pre-merge utility.

Information Loss

When you merge networks, the following information in the source network will be lost:

- **Registered plug-ins**. Plug-ins registered in the source database but not in the destination database must be re-registered following the merge.
- **Non-channel object descriptions**. Channel descriptions are stored in the channel SmartShapes as well as in the network database; therefore, they are transferred to the destination database. Other object descriptions that are not stored in the SmartShapes are lost.
- **Source network addresses, group IDs, NV selectors, and other related information**. All devices in the source network are assigned new addresses in the destination network domain and all connections are reconstructed. Therefore, all subnet/node IDs, group IDs, and NV selectors, will likely change. In addition, all previously commissioned devices in the source network must be recommissioned.
- **Commission Status**. Devices in the source network are unconfigured in the destination network upon the completion of the network merge. However, the Neuron IDs of the devices are preserved in the destination network.
- **Network service devices**. All network service devices (and their functional blocks and connections) in the source network are removed during the merge.
- **Unreferenced device templates**. Device templates that have been imported but do not have any corresponding devices are not created in the destination network.
- **Unreferenced connection descriptions**. Connection descriptions that have been created but do not have any corresponding connections are not created in the destination network.
- **User profiles**. User profiles in the source network are not created in the destination network.

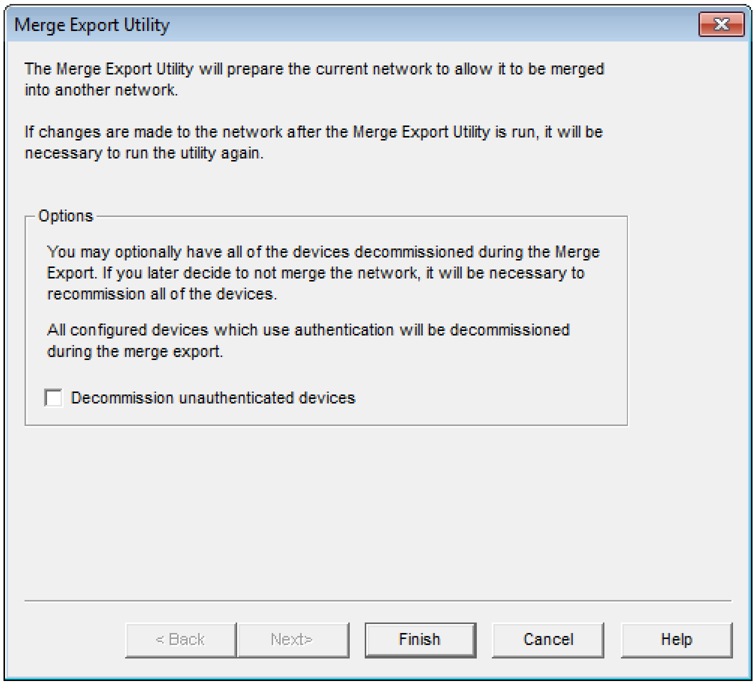
- **Non OpenLNS CT extension records**. Non OpenLNS CT extension records will only be copied if you selected the **Enable Configuration Properties** check box the **Network Properties: Configuration Properties** options category, and the extension record is marked for copying

Merging OpenLNS CT Networks

To merge two OpenLNS CT networks, follow these steps:

1. Identify the source and destination networks. The supernodes, subsystems, devices, functional blocks, and connections in the source network are added to the destination network. The devices in the destination network are not modified.
2. Back up the source OpenLNS CT network design following the steps in *Manually Backing Up an OpenLNS CT Network Design* in this chapter. You can use the backup to restore the network if the merge process fails for any reason. You can also use it to restore the network if you later choose not to merge the networks after running the pre-merge utility.

   In the event that you do not merge the networks after running the pre-merge utility in the source network, restore the OpenLNS CT backup file. This is because the pre-merge utility adds significant data to the OpenLNS CT drawing that will degrade the performance of the network database.
3. Open the source OpenLNS CT network design following the steps described in *Opening an OpenLNS CT Network Design* in Chapter 3, *Getting Started*.
4. Click **Add-ins**, point to **Synchronization**, and then click **Database Merge Export Utility**. The **Merge Export Utility** dialog box opens.



Merge Export Utility

The Merge Export Utility will prepare the current network to allow it to be merged into another network.

If changes are made to the network after the Merge Export Utility is run, it will be necessary to run the utility again.

Options

You may optionally have all of the devices decommissioned during the Merge Export. If you later decide to not merge the network, it will be necessary to recommission all of the devices.

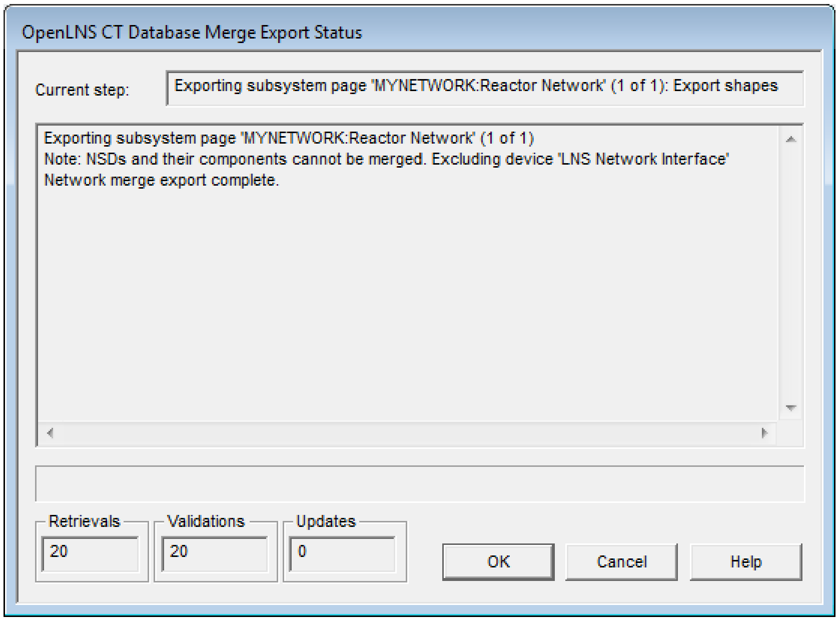
All configured devices which use authentication will be decommissioned during the merge export.

☐ Decommission unauthenticated devices

The merge export utility prepares the network to be merged into another network. The utility adds data from the source OpenLNS network database to the source OpenLNS CT drawing.

The merge export utility also automatically decommissions all configured devices in the network that use authenticated message service so that they can be recommissioned in the new network. Select the **Decommission Unauthenticated Devices** check box to decommission all the configured devices in the network.

5. Click **Finish** to prepare the OpenLNS CT network design for merging. The OpenLNS CT **Database Merge Export Status** dialog opens. This dialog displays the export status and lists merge export events in the order they occur.



OpenLNS CT Database Merge Export Status

Current step: Exporting subsystem page 'MYNETWORK:Reactor Network' (1 of 1): Export shapes

```
Exporting subsystem page 'MYNETWORK:Reactor Network' (1 of 1)
Note: NSDs and their components cannot be merged. Excluding device 'LNS Network Interface'
Network merge export complete.
```

Retrievals: 20

Validations: 20

Updates: 0

6. Click **OK** when the merge export is complete. Click **Cancel** at any time to terminate the merge export.

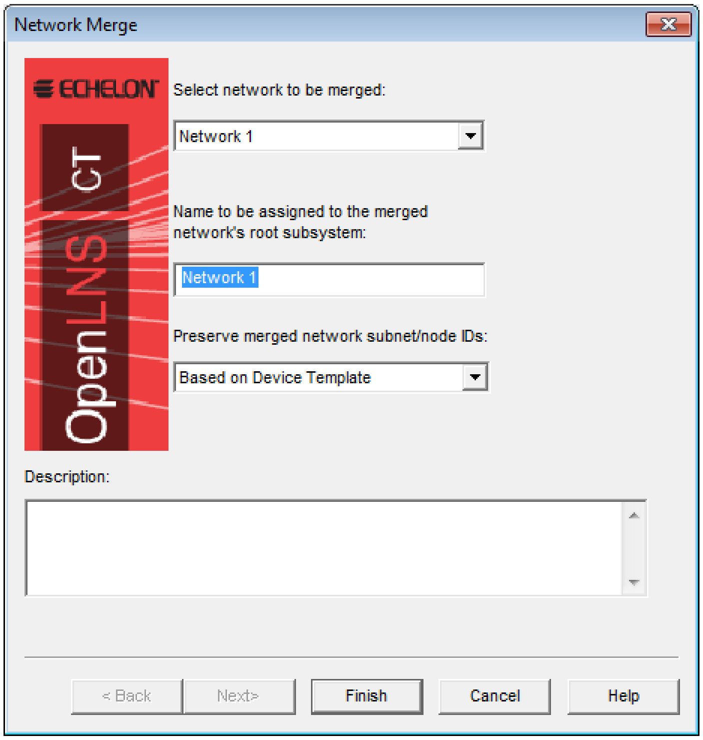
7. Exit the OpenLNS CT drawing of the source network.

8. Back up the destination OpenLNS CT network design following the steps in *Manually Backing Up an OpenLNS CT Network Design* in this chapter. You can use the backup to restore the network if the merge process fails for any reason. If a failure occurs during the merge process, the destination network will be left in a partially merged state and OpenLNS CT will not let you access the network. To return the network to its original state, restore the backup of the destination network.

9. Ensure that the device interface (XIF) files for all the devices in the source network are available in the directory specified in the **XIF Search Path** field in the **Network Properties: Device** options category.

10. Open the destination OpenLNS CT network design following the steps described in *Opening an Existing OpenLNS CT Network Design* in Chapter 3, *Getting Started*.

11. Drag the Network Merge SmartShape () from the OpenLNS CT Basic Stencil to a subsystem page within the destination network. The Network Merge Wizard opens.



Select network to be merged: Network 1

Name to be assigned to the merged network's root subsystem: Network 1

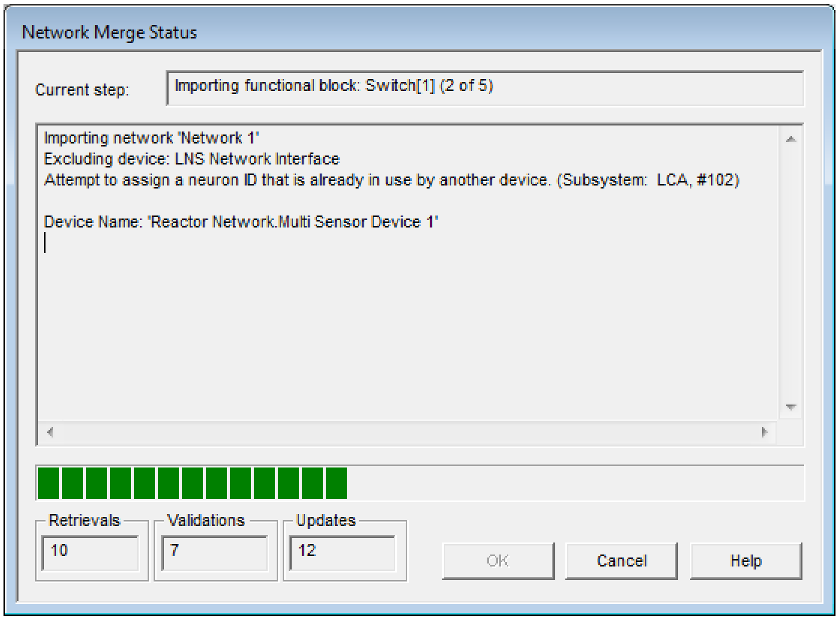
Preserve merged network subnet/node IDs: Based on Device Template

Description:

12. Select the source network you exported in steps 4 – 6 from the **Select Network to be Merged** list. This list contains only those networks that have been exported using the Merge Export Utility.

13. Enter a name for the root subsystem of the source network in the **Name to Be Assigned** box.

14. Click **Finish** to begin the network merge. The source network and its underlying subsystem hierarchy will be added as a new subsystem. The merge process copies all documents from the source drawing directory into the destination drawing directory and renames them, as required. The SmartShapes are then scanned and added to the destination network database. A network merge status window opens and indicates the progress of the merge.



Current step: Importing functional block: Switch[1] (2 of 5)

```
Importing network 'Network 1'
Excluding device: LNS Network Interface
Attempt to assign a neuron ID that is already in use by another device. (Subsystem:  LCA, #102)

Device Name: 'Reactor Network.Multi Sensor Device 1'
```

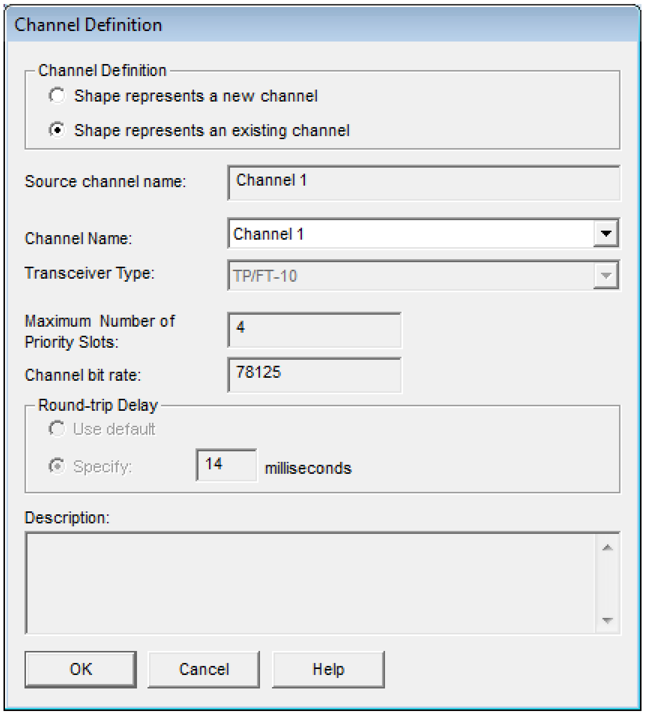
Retrievals: 10

Validations: 7

Updates: 12

This dialog displays the status of the network merge and lists the events in the order they occur. Any errors encountered are shown in the status pane, and they are also available in a log file for the network merge. The log file is named Merge_<*merged network*>.log. The merge log is stored in the drawing directory of the destination network.

15. During the network merge, OpenLNS CT prompts you to specify whether the channels in the merged network already exist in or are new to the current network. The **Channel Definition** dialog opens.



Channel Definition

Channel Definition
○ Shape represents a new channel
● Shape represents an existing channel

Source channel name: Channel 1

Channel Name: Channel 1

Transceiver Type: TP/FT-10

Maximum Number of Priority Slots: 4

Channel bit rate: 78125

Round-trip Delay
○ Use default
● Specify: 14 milliseconds

Description:

16. If the channel already exists in the current network, select its name from the **Channel Name** list. If the channel is new to the current network, enter a unique name for the channel in the **Channel Name** box and select the channel type from the **Channel Type** list. Click **OK** to advance to the next channel.

17. When the network merge is complete, click **OK**. Click **Cancel** at any time to terminate the network merge.

8

Managing Networks

This chapter explains how to test and verify application devices, functional blocks, and routers. It also describes OpenLNS CT styles, and it details how to generate device status summary reports, network resource reports, and OpenLNS network reports.

Using the OpenLNS CT Device Manager

You can manage application devices, functional blocks, and routers using the OpenLNS CT Device Manager. You can switch between performing management tasks and making changes in your OpenLNS CT drawing without having to close the OpenLNS CT Device Manager.

You can perform the following tasks using the OpenLNS CT Device Manager:

- Test application devices, functional blocks, and routers to ensure they are operating and are configured correctly. Testing generates an error log that you can use to evaluate the performance of the object, and it updates the SmartShape status of the tested objects in the OpenLNS CT drawing.
- Wink application devices to identify them on the network and verify that they are communicating properly.
- Place application devices and routers online and offline.
- Disable application devices and functional blocks to isolate network problems.
- Force functional blocks into an override state to keep segments of the network running while disabling others portions of the network.

To use the OpenLNS CT Device Manager, OpenLNS CT must be attached to the network. OpenLNS CT can either be OnNet or OffNet for testing, winking, and setting the application state of devices and routers, but it must be OnNet to disable application devices, and to disable and override functional blocks.

Opening the OpenLNS CT Device Manager

To open the OpenLNS CT Device Manager, right-click the application device, functional block, or router you want to manage and then click **Manage** on the shortcut menu.

You can perform management tasks on multiple objects at one time. You can use any of the following methods to select the objects to be managed:

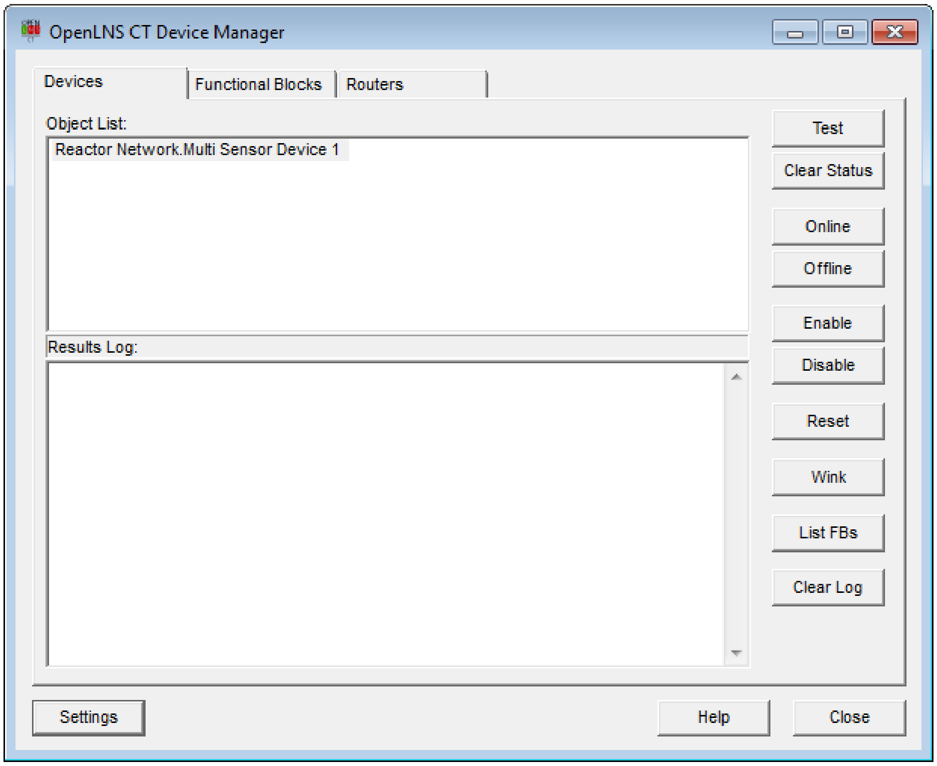
- To select multiple devices within a subsystem, hold down CTRL and click the devices, or click an empty space in the drawing page and drag a selection net around them. Then right-click one of the selected devices and click **Manage** on the shortcut menu.
- To select all the devices within a subsystem, select the **Exclude Nested Subsystems from Subsystem Operations** check box for the subsystem (to access this option, right-click the subsystem SmartShape, or an empty space in the network if you are in the top-level subsystem, and select **Subsystem Properties** from the shortcut menu). Then right-click the subsystem SmartShape, or an empty space in the drawing page if you are in the top-level subsystem, and click **Manage** on the shortcut menu.
- To select all the devices within a subsystem and all of its nested subsystems, clear the **Exclude Nested Subsystems from Subsystem Operations** check box for the subsystem, right-click the Subsystem SmartShape, or an empty space in the drawing page if you are in the top-level subsystem, and click **Manage** on the shortcut menu.
- To select all the devices within a network, right-click an empty space in the drawing page of the top-level subsystem and then click **Manage** on the shortcut menu.

  The OpenLNS CT Device Manager contains individual tabs with distinct functions for managing application devices, functional blocks, and routers. The tab that opens when you initially open the OpenLNS CT Device Manager depends on the objects selected.

Managing Devices

To manage a device, right-click a device (or select multiple objects including at least one application device or functional block and right-click one of them), and then select **Manage** on the shortcut menu. You can also select the **Devices** tab after you have opened one of the other tabs with one or more application devices or functional blocks selected. The **Devices** tab of the OpenLNS CT Device Manager opens.

You can use this tab to manage the selected application devices. The tab contains an **Object List** that displays the selected devices, and a **Results Log** that lists the results of any commands or tests.



Reactor Network.Multi Sensor Device 1

Click one or more devices listed in the **Object List** and then click any of the following buttons:

Test — Tests the device to ensure that it is operating and configured correctly. The tests run are determined by the options set in the **Device Manager Settings** dialog, which you can open by clicking **Settings**. The test results are displayed in the **Results Log**.

Clear Status — Clears the network error statistics for the device and the error log, which are displayed after you click **Test**.

Online — Places the device in the online state. The behavior in the online state depends on the device. A Neuron-hosted device, for example, will run its application. If the device was previously online, the device state does not change.

Offline — Places the device in the offline state. The behavior in the offline state depends on the device. A Neuron-hosted device, for example, will stop running its application.

When a device is in the offline state, it still responds to **Online**, **Wink**, and **Test** commands and receives network variable updates. The device application, though, will not process the updated network variable values or send network variable updates.

If the device was previously offline, the device state does not change. If the device is reset while offline, it will be placed in the online state.

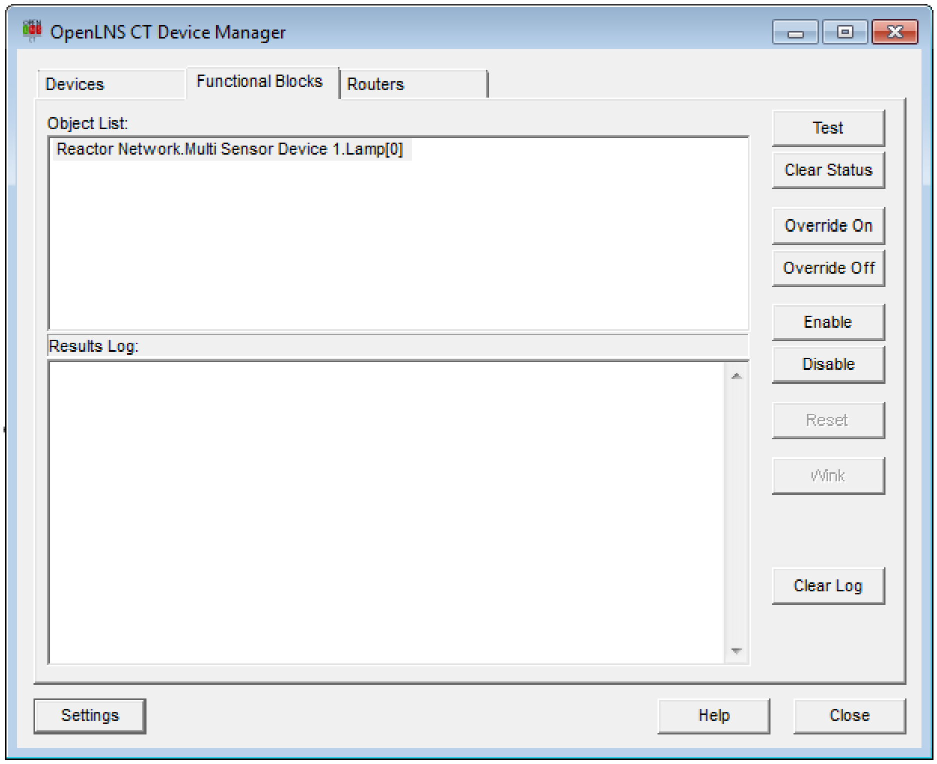
Enable	Activates all functional blocks on a disabled device. The device must be online to be enabled. This command only applies to devices containing a Node Object functional block.
Disable	Keeps the device application disabled even if the device is reset. The device must be online to be disabled. This command only applies to devices containing a Node Object functional block. If **Disable** is selected and the device does not contain a Node Object functional block, the device is placed offline.
Reset	Stops the device application, terminates all incoming and outgoing messages, sets all temporary settings to their initial values, and then restarts the device application. If the device was in the soft offline state, it will be put online; if the device was in the hard offline state, it will remain offline.
Wink	Requests that the device generate an application-dependent audio or visual feedback such as a beep or a flashing service LED. The device must support the Wink function to use this command. This command is useful for identifying devices.
List FBs	Switches to the **Functional Blocks** tab and adds the functional blocks assigned to the selected devices to the object list.
Clear Log	Clears the **Results Log**.

Click **Settings** to open the **Device Manager Settings** dialog box. This dialog lets you set options for testing and winking devices. See *Device Manager Settings* in this section for more information on using this dialog.

Managing Functional Blocks

To manage a functional block, right-click a functional block (or select multiple objects including at least one functional block or application device and right-click one of them), and then select **Manage** on the shortcut menu. You can also select the **Functional Block** tab after you have opened one of the other tabs with one or more application devices or functional blocks selected. The **Functional Blocks** tab of the OpenLNS CT Device Manager opens.

You can use this tab to manage the selected functional blocks. The tab contains an **Object List** that displays the selected devices, and a **Results Log** that lists the results of any commands or tests.



The commands in this dialog require that the device application associated with the functional block includes a Node Object functional block and that the device is online.

Click one or more functional blocks in the **Object List** and then click any of the following buttons:

Test — Tests the functional block to ensure that it is operating correctly. The tests run are determined by the options set in the **Device Manager Settings** dialog, which you can open by clicking **Settings**. The test results are displayed in the **Results Log**.

Clear Status — Clears the network error statistics for the functional block and the error log, which are displayed after you click **Test**.

Override On — Puts the functional block into override mode. The output network variables output their configured override value, even if part of the network is not yet operating and there is no input to the functional block. The override values are set using the configuration properties in the functional block.

See the functional block's documentation for more information on how the override command affects the functional block.

Override Off — Takes the functional block out of override mode so that it functions normally.

Enable — Activates a disabled functional block.

Disable — Disables the functional block. The functional block will send its configured output network variable defaults.

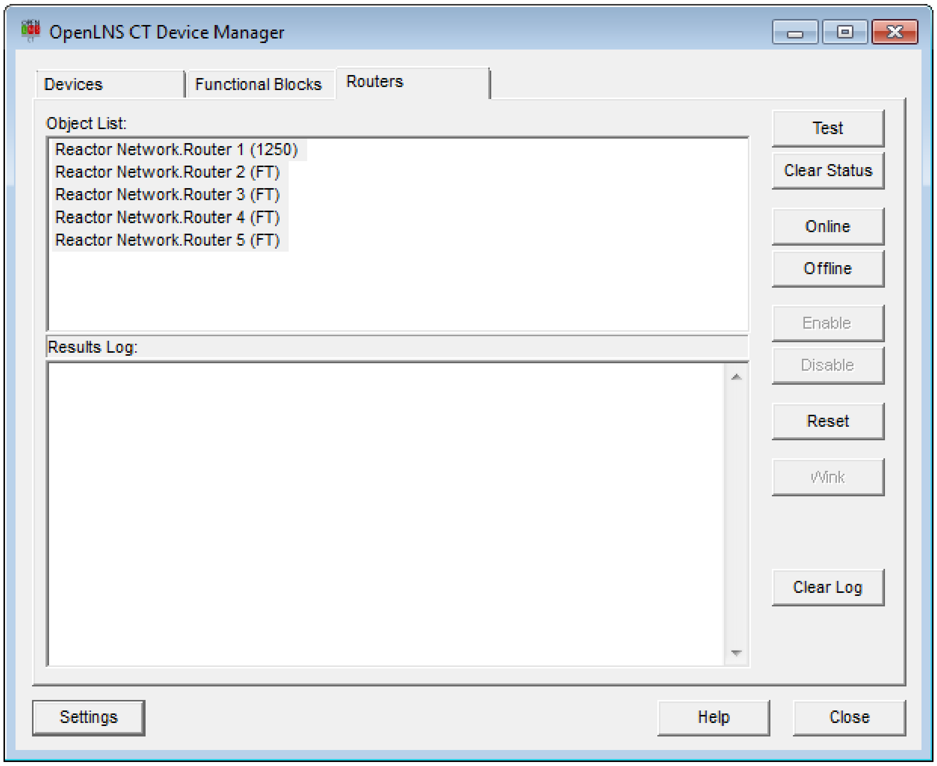
Clear Log — Clears the **Results Log**.

Click **Settings** to open the **Device Manager Settings** dialog box. In this dialog, you can set options for testing functional blocks.

Managing Routers

To manage a router, right-click a router (or select multiple objects including at least one router and right-click one of them), and then and select **Manage** on the shortcut menu. You can also select the **Routers** tab after you have opened one of the other tabs with one or more routers selected. The **Routers** tab of the OpenLNS CT Device Manager opens.

You can use this tab to manage the selected routers. The tab contains an **Object List** that displays the selected routers, and a **Results Log** that lists the results of any commands or tests.



Reactor Network.Router 1 (1250)
Reactor Network.Router 2 (FT)
Reactor Network.Router 3 (FT)
Reactor Network.Router 4 (FT)
Reactor Network.Router 5 (FT)

Click one or more routers in the **Object List** and then click any of the following buttons:

Test — Tests the routers to ensure that they are operating and configured correctly. The tests run are determined by the options set in the **Device Manager Settings** dialog, which you can open by clicking **Settings**, and by the device application. The test results are displayed in the **Results Log**.

Clear Status — Clears the network error statistics for both sides of the router.

Online — Enables the routers to forward packets between the channels connected to them. If any of the routers were previously online, the router state does not change.

Offline — Stops the routers from forwarding packets. If any of the routers were previously offline, their states do not change. If a router is reset while offline, the router will go online, unless it is in the hard offline state.

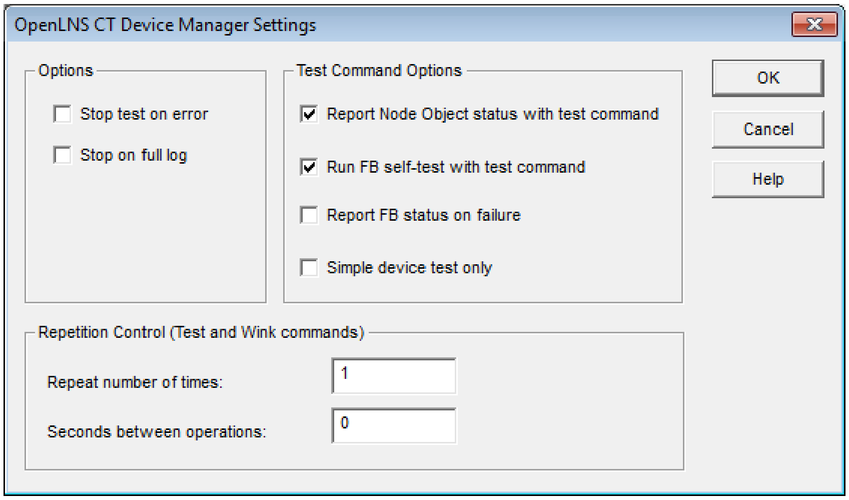
Reset — Stops the routers from forwarding packets, sets all temporary settings to their initial values, and then restarts the router application. If any of the routers were offline, they will be go online after being reset, unless they were in the hard offline state.

Clear Log — Clears the **Results Log**.

Click **Settings** to open the **Device Manager Settings** dialog box. There, you can set options for testing routers and options for the results log.

Device Manager Settings

You can customize the error reporting, testing, and winking operations performed by the OpenLNS CT Device Manager. To customize these setting, click the **Settings** button on any of the tabs in the Device Manager. The **Device Manager Settings** dialog opens.



OpenLNS CT Device Manager Settings

Options

☐ Stop test on error

☐ Stop on full log

Test Command Options

☑ Report Node Object status with test command

☑ Run FB self-test with test command

☐ Report FB status on failure

☐ Simple device test only

Repetition Control (Test and Wink commands)

Repeat number of times: 1

Seconds between operations: 0

Set the following options and then click **OK** to apply any changes.

Options

Stop Test on Error

Stops tests immediately when an error is reported. This option is cleared by default.

Stop on Full Log

Stops tests immediately when the results log becomes full. You cannot click the **Test** command again until you click **Clear Log**. This option is cleared by default.

Test Command Options

Report Node Object Status with Test Command

Displays the status of the Node object for the device being tested in the results log of the **Device** tab. This command only applies to devices containing a Node Object functional block. This option is selected by default.

Run Object Self-Test with Test Command

Sends a self-test command to the object being tested and displays the results of the self-test in the **Results Log**. If cleared, no self-test will be sent as part of the testing procedure. This option is selected by default.

Report Object Status on Failure

Reports an object's status in the **Results Log** when it fails a management command. This option is cleared by default.

Simple Device Test

Limits the device test to a simple **Query Status** command. This option reduces the amount of network traffic generated by the **Test** command, and it may be useful when communication is performed over low-bandwidth channels such as a power line. This option is cleared by default.

Repetition Control

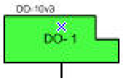
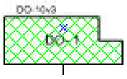
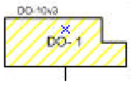
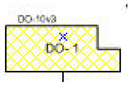
Repeat Number of Times	Specifies how many times a **Wink** or **Test** command will be repeated. The default is **1**, which means that these commands are not repeated. If you enter a value greater that **1** in this box, you must enter a non-zero value in the **Wait Between Operations** box.
Wait Between Operations	Specifies how many seconds the OpenLNS CT Device Manager waits between repeating a **Wink** or **Test** command when the number in the **Repeat Number of Times** box is greater than 1. Shorter wait times may cause a **Wink** or **Test** command to generate an error because it does not get completed before the next one is issued.

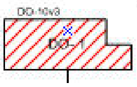
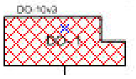
Using OpenLNS CT Styles Overview

You can use Visio styles to view the state of each device, functional block, connection, and network variable/message tag. OpenLNS CT automatically updates SmartShape styles as events are received that indicate changes in SmartShape status. You can change how states are depicted in an OpenLNS CT drawing. If you modify the current drawing file styles, only the styles in the drawing file that you modified will be affected. If you modify the styles in the template (**LonWorks.vst** by default), all networks created after the template modification will use the new styles.

OpenLNS CT Device Styles

Device styles indicate the state of the application devices and routers in an OpenLNS CT drawing. The OpenLNS Server monitors the state of devices that have been enabled, and notifies OpenLNS CT of any changes. The following list describes the device styles:

Device State	Default Style	Description
Normal	Solid green 	All changes made to the network database have been successfully written to the physical device. The device and the database are synchronized.
Soft Offline	Green crosshatching 	The device has an application loaded and is configured, but it is offline. The device will go online when either the online command is invoked on the device or the device is reset.
Uninstalled	Yellow diagonal 	Changes have been made in the network database, but the physical device has not been updated. Devices may not be up to date because OpenLNS CT is unattached or is OffNet. Once the device is commissioned and updated, the device style changes to Normal. For example, if you delete a connection between two application devices while OpenLNS CT is OffNet, the devices are marked with the yellow diagonal lines.
State Failure	Yellow crosshatching 	The OpenLNS Server has detected that the device is not in the expected state.
Error	Red diagonal	Changes have been made in the network database, but an error has occurred while updating the device. The

		device may not be responding properly, a router or channel between the OpenLNS Server and the device may have failed, or a physical connection may be faulty. If OpenLNS CT is OnNet, the OpenLNS Server periodically tries to update the device. Once the device has been updated to match the network database, the device style changes to Normal.
Detached	Red crosshatching 	The OpenLNS Server has detected that the device is not attached or is otherwise unable to communicate with the network. If a router becomes detached, the devices on the other side of the router will not show up as detached on the drawing.

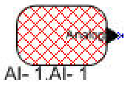
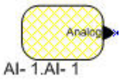
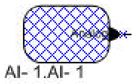
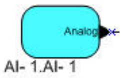
Note: Devices that have a Ping Interval of **Never** are not monitored; therefore, the device SmartShapes may not indicate the correct detached or failure states.

OpenLNS CT Functional Block Styles

OpenLNS CT reflects the last known state of functional blocks. You may update that information in two ways:

- From the OpenLNS CT menu, select **Status Summary** (see *Generating a Device Status Summary Report* in this chapter for more information about this feature).
- Select one or more functional blocks, right-click one of the selected functional blocks, ands then click **Manage** on the shortcut menu. The **Functional Block** tab of the OpenLNS CT Device Manager opens. Click **Test** (see *Managing Functional Blocks* in this chapter for more information.)

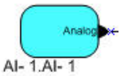
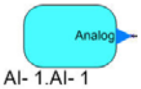
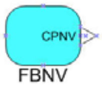
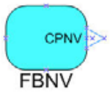
The OpenLNS CT drawing indicates the current functional block states using the following styles:

Functional Block State	Default Style	Description
Self-Test Fail	Red crosshatching 	The functional block has reported a self-test failure.
Disabled	Yellow crosshatching. 	The functional block has been disabled.
Override	Dark blue crosshatching. 	The functional block has been placed in override mode.
Normal	Solid cyan. 	The functional block is in a normal state and none of the states listed above are active.

When multiple functional blocks states are concurrently active, the style displayed is based on the order of the states listed above. For example, a functional block in the normal state that is disabled and placed in override mode will be crosshatched yellow, indicating the disabled state (disabled is the highest-order state of the three). If multiple states are active and the highest-order state is then cleared, the functional block style will be updated to display the next applicable state. Using the previous example, if you enable the functional block, it will be crosshatched dark blue, indicating that it is in override mode.

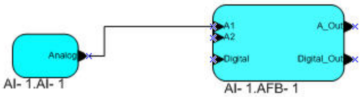
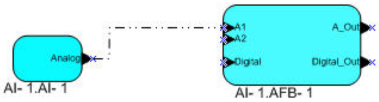
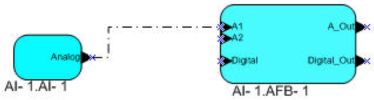
Network Variable and Message Tag SmartShape Styles

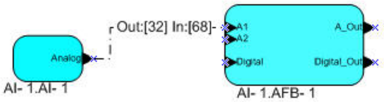
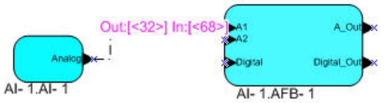
OpenLNS CT indicates the state of network variable and message tag SmartShapes using the following styles:

Connection Point State	Default Style	Description
Connection Point Normal	Solid black fill  	The network variable or message tag SmartShape is normal (not currently exported to a supernode).
Connection Point Exported	Solid blue fill  	The network variable or message tag SmartShape has been exported to one or more supernodes.
Configuration Property Normal	Solid black line with a transparent fill  	The configuration network variable is normal (not currently exported to a supernode).
Configuration Property Exported	Solid blue line with a transparent fill.  	The configuration network variable is currently exported to one or more supernodes.

OpenLNS CT Connector SmartShape Styles

OpenLNS CT indicates the messaging service used for a connection and whether monitoring is enabled with the following styles:

Messaging Service/Monitoring State	**Default Style**
Acknowledged Messaging Service	Solid line.  Analog AI- 1.AI- 1 A1 A2 Digital A_Out Digital_Out AI- 1.AFB- 1
Repeated Messaging Service	Dash separated by two dots.  Analog AI- 1.AI- 1 A1 A2 Digital A_Out Digital_Out AI- 1.AFB- 1
Unacknowledged Messaging Service	Dash separated by one dot.  Analog AI- 1.AI- 1 A1 A2 Digital A_Out Digital_Out AI- 1.AFB- 1

Monitoring Enabled	Text is black.  Out:[32] In:[68] Analog AI- 1.AI- 1 A1 A2 Digital A_Out Digital_Out AI- 1.AFB- 1
Monitoring Disabled	Text is magenta.  Out:[<32>] In:[<68>] Analog AI- 1.AI- 1 A1 A2 Digital A_Out Digital_Out AI- 1.AFB- 1

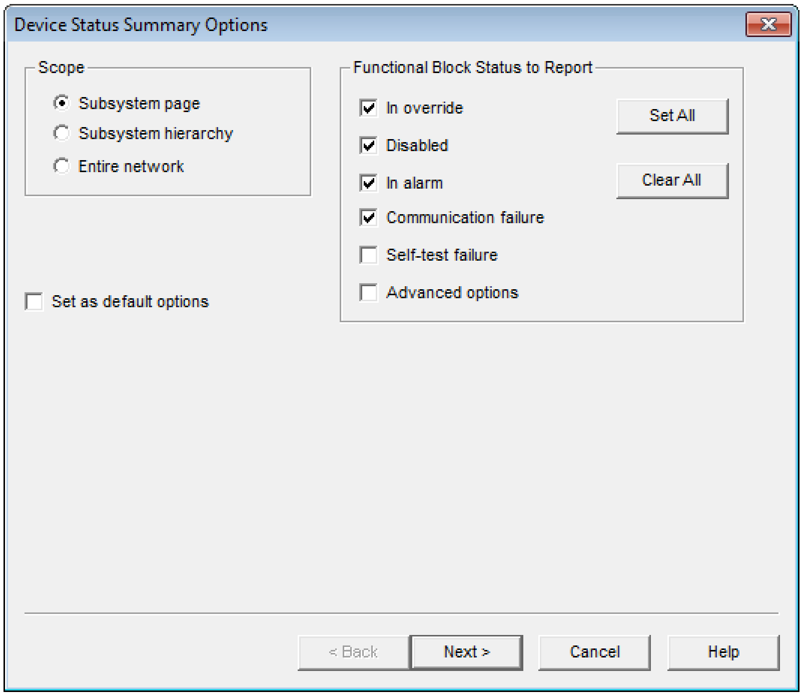
OpenLNS CT updates the style of the connection SmartShapes, as required, if changes are made to the connections that affect the service type used.

Generating a Device Status Summary Report

You can generate a status summary report listing the devices and routers that are unconfigured and offline, and selected functional blocks that are in the override, alarm, or disabled states or have reported communication or self-test failures. For a functional block to be listed in the report, it must be associated with a device that is configured and online.

To generate a status summary report, follow these steps:

1. Click **Add-ins**, point to **Network Properties**, and then click **Device Status Summary**. The **Device Status Summary Options** dialog box opens.



Device Status Summary Options

Scope

● Subsystem page
○ Subsystem hierarchy
○ Entire network

□ Set as default options

Functional Block Status to Report

☑ In override
☑ Disabled
☑ In alarm
☑ Communication failure
□ Self-test failure
□ Advanced options

2. Select the scope of the **Device Status Summary** and the types of functional blocks to include in the report.

Scope Select the scope of the **Device Status Summary Report**. This determines which applicable application devices, routers, and functional blocks will be included in the report. You have three choices:

- Select **Subsystem Page** to include the application devices, routers, and functional blocks in the current subsystem. This is the default.
- Select **Subsystem Hierarchy** to include the application devices, routers, and functional blocks in the current subsystem and subsystems nested in it.
- Select **Entire Network** to include all the application devices, routers, and functional blocks in the current OpenLNS CT network.

Functional Block Status to Report
Select which functional blocks with a specific status to include in the **Device Status Summary**.

For a functional block to be included in the report, its device must be configured and online.

Status reporting is not supported by all functional blocks. If an expected functional block does not appear in the summary, it means that the functional block does report the selected status. In this case, OpenLNS CT treats the functional block as if the status was reported as not set by the device.

Override
Includes functional blocks in the override state in the **Device Status Summary Report**. This check box is selected by default.

Disabled
Includes disabled functional blocks in the **Device Status Summary Report**. This check box is selected by default.

Alarm
Includes functional blocks in the alarm state in the **Device Status Summary Report**. This check box is selected by default.

Communication Failure
Includes functional blocks that have had a communication failure in the **Device Status Summary Report**. This check box is cleared by default.

Self-Test Failure
Includes functional blocks that have had a self-test failure in the **Device Status Summary Report**. This check box is cleared by default.

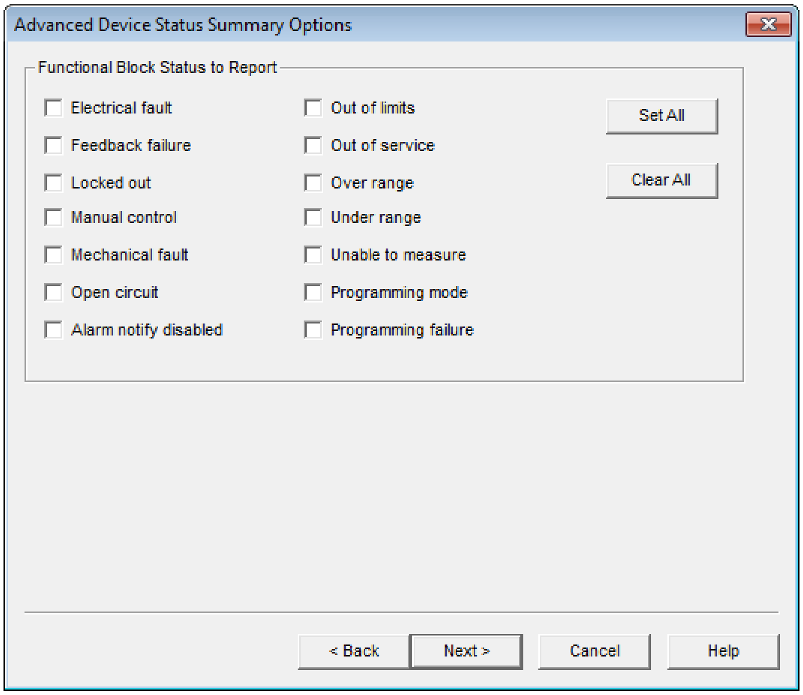
Advance Options
Opens a dialog containing additional status summary options when you click **Next**.

Set All
Selects all the functional block status options.

Clear All
Clears all the functional block status options.

Set as Default Options
Causes the options set in this window to become the defaults for this OpenLNS CT network. The default options are set when you click **Next**.

3. Click **Next**. If you selected the **Advanced Options** check box, the **Advanced Status Summary Options** dialog box opens. Otherwise, you can skip to step 5, in which the **Device Status Summary Results** dialog opens.



Advanced Device Status Summary Options

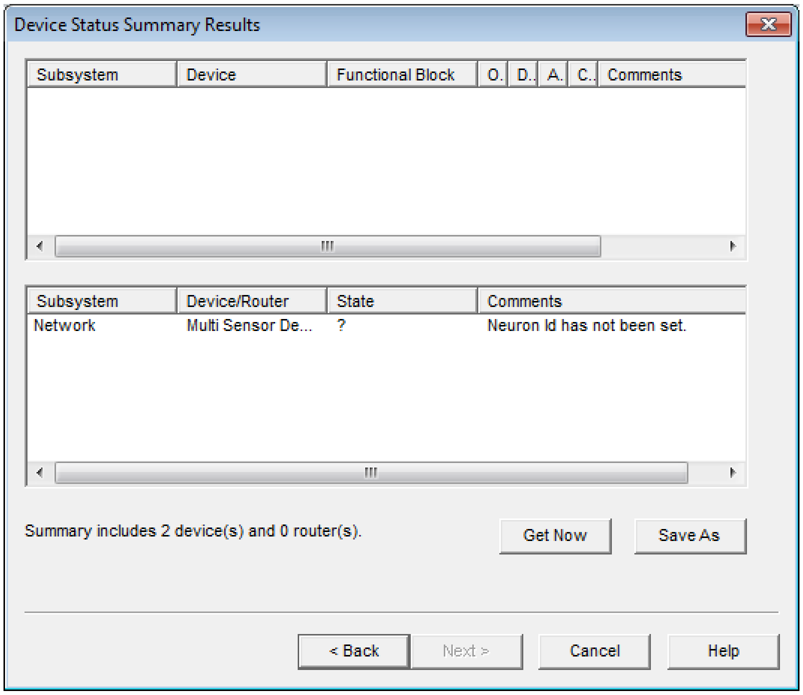
Functional Block Status to Report

□ Electrical fault	□ Out of limits
□ Feedback failure	□ Out of service
□ Locked out	□ Over range
□ Manual control	□ Under range
□ Mechanical fault	□ Unable to measure
□ Open circuit	□ Programming mode
□ Alarm notify disabled	□ Programming failure

4. Select additional functional blocks to include in the status summary based on the type of errors or exceptions they have reported. For more information on the status items themselves, refer to hardware manuals or LONMARK documentation.

5. Click **Next**. The **Device Status Summary Results** dialog opens. This dialog displays the status of devices, routers, and specified functional block types within the scope specified in the **Device Status Summary Options** dialog. The functional blocks are listed in the top pane; the devices and routers are listed in the bottom pane.

6. Click **Get Now** to generate the status summary. A one-line summary states the number of devices and routers included in the summary results.



Device Status Summary Results

Subsystem	Device	Functional Block	O.	D.	A.	C.	Comments

Subsystem	Device/Router	State	Comments
Network	Multi Sensor De...	?	Neuron Id has not been set.

Summary includes 2 device(s) and 0 router(s).

For functional blocks, this dialog displays the following data in the top pane: subsystem, device, functional block name, applicable status (indicated by "**YES**"), and any comments about the functional block.

For application devices and routers, this dialog displays the following data in the bottom pane: subsystem, device name, state, and any comments about the device or router.

7. To save the status summary to a text file (**.txt** extension), click **Save As**.

Using Resource Usage Reports

You can generate a *network resource report* to get a summary of group ID, subnet ID, and selector usage in your network. You can display a summary of the usage and availability of alias table entries for a device.

Network Resource Report

You can display a summary of the usage of group IDs, subnet IDs, and selectors in your network. The usage of these resources is displayed in a network resource report. This report displays the number of group IDs, subnet IDs, and selectors that have been allocated out of those available in your OpenLNS CT drawing. Individual color status bars indicate the percentage consumed and available for each resource. As more network resources are consumed, the status bar fills up and changes color. The status indicated by each color is as follows:

- **Green**. Resource is readily available (up to 75 percent used).
- **Yellow**. Resource is available, but beginning to run low (between 75 and 90 percent used).
- **Red**. Resource is critically low (over 90 percent used).

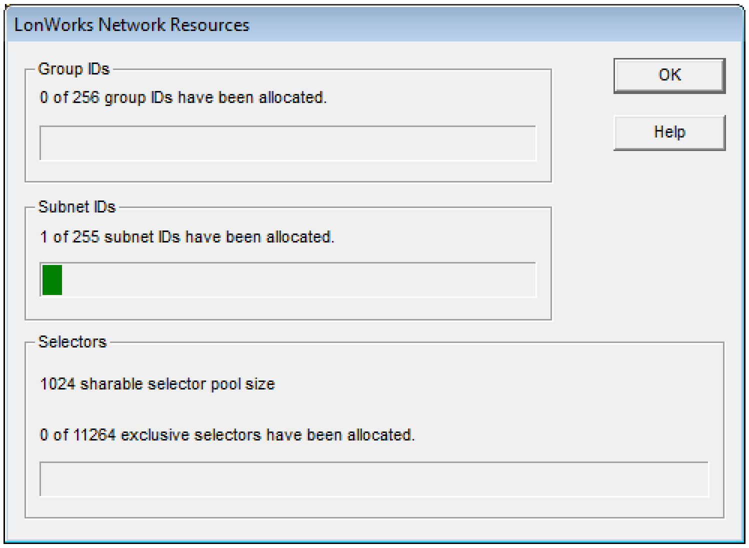
Each of these resources is described as follows:

- *Group ID*. A group ID is allocated when you create a multicast connection with a group broadcast. Once a group ID is allocated for a connection, it cannot be reused. There are 256 group IDs available per network. You will not be able to create connections with group multicast once all group IDs are allocated. You can prevent group ID usage by using other forms of multicast such as subnet or domain broadcast, or alias multi-unicast.

- *Subnet ID*. A subnet ID is allocated when you create a new channel, when you add more than 127 devices to an existing subnet, or when you explicitly create a subnet. Once a subnet ID is allocated, it cannot be reused for a different channel. There are 255 subnet IDs per network. You will not be able to add new channels or subnets once all subnet IDs have been allocated.

- *NV Selector*. An NV selector is allocated when you create a new network variable connection. A pool of NV selectors may be shared among multiple connections, but the remaining NV selectors are exclusive and can only each be used for one connection. You will not be able to create more connections once all exclusive NV selectors are allocated unless OpenLNS CT can successfully share one of the shared selectors. The network resource report shows the size of the shareable NV selector pool, and it shows the size and consumption of the exclusive selector pool.

To view the network resources report, click **Add-ins**, point to **Network Properties**, and then click **Network Resources**. The **LonWorks Network Resources** dialog box opens.



Group IDs

0 of 256 group IDs have been allocated.

Subnet IDs

1 of 255 subnet IDs have been allocated.

Selectors

1024 sharable selector pool size

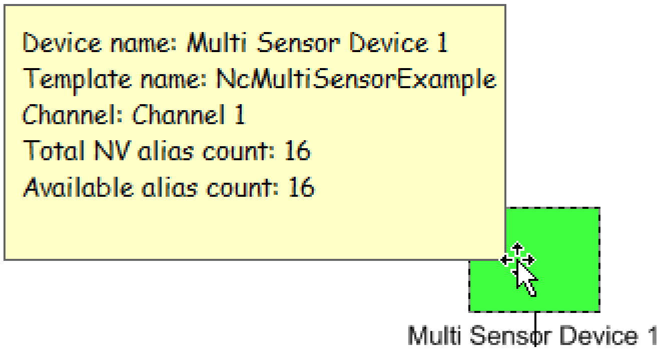
0 of 11264 exclusive selectors have been allocated.

Alias Table Summary

You can display a summary of the usage and availability of alias table entries for a device. The usage and availability is displayed in a ToolTip for a device. The device SmartShape ToolTip lists the total number of network variable aliases for the device, and the number of aliases currently available.

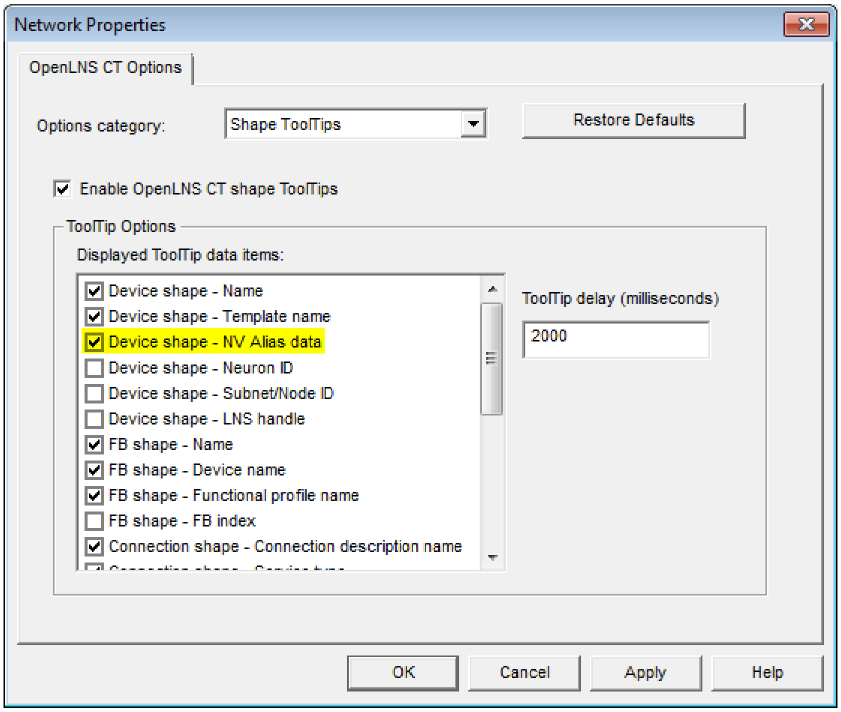
An alias table entry is allocated on a sending device when you create an alias multi-unicast connection and is also consumed on either the sending or destination device when required to resolve network variable selector conflicts. The number of alias table entries available on a device is determined by the device manufacturer. OpenLNS CT will use alias table entries on a device when required if they are available.

To view the alias table summary, move the pointer over the desired device SmartShape and keep it there for the ToolTip delay. The default ToolTip delay (number of milliseconds the pointer needs to be positioned over an OpenLNS CT SmartShape for the ToolTip to appear) is two seconds. The device SmartShape ToolTip will then appear. The total and available network available alias counts are by default the last two items listed in the ToolTip.


Device name: Multi Sensor Device 1
Template name: NcMultiSensorExample
Channel: Channel 1
Total NV alias count: 16
Available alias count: 16

To enable the device shape ToolTip to display the alias table summary, follow these steps:

1. In the OpenLNS CT drawing, click **Add-ins**, point to **OpenLNS CT**, and then select **OpenLNS CT Options**. The **Network Properties** dialog box opens.
2. Select **Shape ToolTips** from the **Options Category** list.
3. Select the **Device Shape – NV Alias Data** check box.



Options category: Shape ToolTips

☑ Enable OpenLNS CT shape ToolTips

ToolTip Options

Displayed ToolTip data items:

- ☑ Device shape - Name
- ☑ Device shape - Template name
- ☑ Device shape - NV Alias data
- ☐ Device shape - Neuron ID
- ☐ Device shape - Subnet/Node ID
- ☐ Device shape - LNS handle
- ☑ FB shape - Name
- ☑ FB shape - Device name
- ☑ FB shape - Functional profile name
- ☐ FB shape - FB index
- ☑ Connection shape - Connection description name
- ☑ Connection shape - Service type

ToolTip delay (milliseconds): 2000

9

Exporting and Importing Networks with XML

This chapter describes how to export a LONWORKS network to a LONWORKS network XML file, view a LONWORKS network XML file in a web browser, edit a LONWORKS network XML file, and import a LONWORKS network XML file to update a network.

Using XML Export/Import Overview

You can speed up your network design and ensure your devices are configured correctly by creating an OpenLNS CT network report or by creating a custom OpenLNS CT user interface. OpenLNS CT includes an OpenLNS CT XML plug-in that you can use to export all the devices, connections, network variables, and configuration properties within all or a part of an OpenLNS CT network design to a LONWORKS network XML file. You can also include device templates, connection description templates, and channels in the XML file. You can create an OpenLNS CT network report from this file, or you can process this file with a custom application. You can also use the OpenLNS CT XML plug-in to modify an OpenLNS CT network design based on changes that you specify in a LONWORKS network XML file.

To create a custom OpenLNS CT interface, you create an application that reads or writes LONWORKS network XML files and then invokes the OpenLNS CT XML Plug in to automatically export the OpenLNS CT network created and import it into your custom application. To develop a custom OpenLNS CT user interface, you need to create an application that does the following:

1. Optionally invokes the OpenLNS CT XML Plug-in and exports all or part of an OpenLNS CT drawing as an XML file.
2. Creates or modifies the XML file to create and configure the desired devices, functional blocks, network variables, and network variable/message tag connections in an OpenLNS CT network.
3. Invokes the OpenLNS CT XML Plug-in and imports the XML file created into OpenLNS CT.

In an exported LONWORKS network XML file, you can create, update, and delete most OpenLNS CT objects, including the following: subsystems, application devices and routers, functional blocks, network variables, connections, and channels. In addition, you can commission application devices and routers. If you commission a device, you can specify the application image to load into it, set the initial state of its application, and specify the source of configuration property value and how device-specific configuration properties are handled. See the *LONWORKS Network XML Programmer's Guide* for more information on editing a LONWORKS network XML file.

After editing the LONWORKS network XML file, you can import it back into your network. OpenLNS CT updates the network based on the attributes and actions specified for the OpenLNS CT objects in the file.

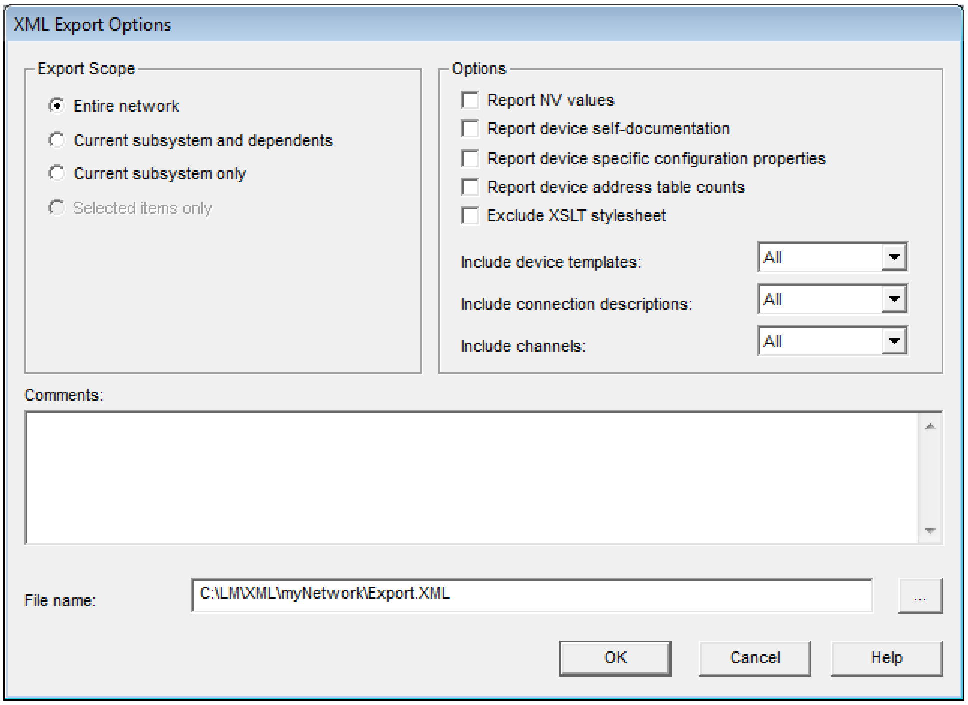
Exporting a LONWORKS Network XML File

You can export your LONWORKS network to an XML file using the OpenLNS CT XML Plug-in included with OpenLNS CT, or you can manually export your OpenLNS CT network using OpenLNS CT. You can export selected items, subsystems, or your entire network to the XML file. The XML file consists of a report of all the devices, connections, network variables, configuration properties, and other data that you select within a specified network scope. In addition, the exported XML file by default includes a XSLT style sheet that you can use to transform the file into an OpenLNS CT network report, which you can view in your Web browser (see *Viewing an OpenLNS CT Network Report* for more information).

If you are using an OpenLNS application other than OpenLNS CT, you can use the OpenLNS CT XML Plug-in to automate exporting your LONWORKS network to the other application. The OpenLNS CT XML Plug-in provides a programmatic interface to the XML import and export features in OpenLNS CT. The OpenLNS CT XML Plug-in is registered at the network level; therefore, it only needs to be registered once on a computer for it to be used to import or export any LONWORKS network. See the *LONWORKS Network XML Programmer's Guide* for details on how applications can use the plug-in and the schema used for the XML file.

To manually export your OpenLNS CT drawing to a LONWORKS network XML file, follow these steps.

1. Click **Add-ins**, point to **OpenLNS CT Documents**, and then click **Export to XML**. The **XML Export Options** dialog box opens.



XML Export Options

Export Scope

● Entire network

○ Current subsystem and dependents

○ Current subsystem only

○ Selected items only

Options

□ Report NV values

□ Report device self-documentation

□ Report device specific configuration properties

□ Report device address table counts

□ Exclude XSLT stylesheet

Include device templates: All

Include connection descriptions: All

Include channels: All

Comments:

File name: C:\LM\XML\myNetwork\Export.XML

2. Set the following options to select the data to be included in the XML file:

Export Scope Select the portion of the network to be included in the XML file. There are four choices:

- **Entire Network**. All application devices and routers in all subsystems in the network will be exported. This is the default.
- **Current Subsystem and Dependents**. Application devices and routers in the currently displayed subsystem and those in any nested subsystems will be exported.
- **Current Subsystem Only**. Only application devices and routers in the currently displayed subsystem will be exported.
- **Selected Items Only**. Only selected application devices, routers, and subsystems in the currently displayed subsystem will be exported. This option is only available if you select one or more items before opening the **XML Export Options** dialog box. To select an item click it. To select multiple items, click one, hold down SHIFT, and click all other desired items.

Options

Report NV Values Includes the current values of all exported NVs. This option is unavailable if the OpenLNS CT computer is not attached to the physical network. This option can add significant processing time to the export if there are many network variables within the export scope.

Report Self-Documentation Includes the self-documentation data for all exported devices and their network variables. This option is unavailable if the OpenLNS CT computer is not attached to the physical network. This option can add significant processing time to the export if there are many network variables within the export scope.

Report Device Includes the values of all device-specific configuration properties.

Specific CP Values Device-specific CPs are CPs that may be changed by the device itself or by another device that is not based on OpenLNS. OpenLNS CT reads device-specific values from the devices in the network—this option is unavailable if the OpenLNS CT computer is not connected to the physical network. This option can add significant processing time to the export if there are many device-specific CPs within the export scope.

Report Device Address Table Counts Includes the address table counts of all exported devices. This option is unavailable if you are using OpenLNS CT on a remote client. This option can add significant processing time to the export if there are many devices within the export scope

Exclude XSLT Stylesheet Includes the XSLT style sheet. See *OpenLNS CT Network Report* later in this chapter.

Include Channels Selects which channels are reported in the XML file. Choose **All**, **None**, or **Referenced**. **Referenced** means that only channels referenced by devices within the selected network scope will be reported.

Include Device Templates Selects which device templates are reported in the XML file. Choose **All**, **None**, or **Referenced**. **Referenced** means that only device templates referenced by devices within the selected network scope will be reported.

Include Connection Descriptions Selects which connection descriptions are reported in the XML file. Choose **All**, **None**, or **Referenced**. Referenced means that only connection descriptions referenced by connections within the selected network scope will be reported.

Comments Specifies optional comments that you can use to document the export file. The comments are inserted as a set of comment tags containing the comments in the XML file.

File Name Specifies the name of the XML file. The default name is **C:\Users\Public\Documents\LonWorks\OpenLnsCt\XML\***network*\**Export**[*_version*]**.xml.**

3. Click **OK**. The **OpenLNS CT XML Export Status** window displaying the report status opens, and OpenLNS CT generates the XML file. You can click **Cancel** to stop the XML export. When “Export complete” opens, click **OK** to close the window.
4. You can open the exported LONWORKS network XML file in an XML editor and begin viewing, creating, deleting, or modifying items; view the exported XML file as an OpenLNS CT network report; or view or edit the OpenLNS CT XML file with another application. If you make changes to the XML file, save the file. You can then import the updated XML file back into the network as described in *Importing an OpenLNS CT XML File*

Viewing an OpenLNS CT Network Report

You can view a LONWORKS network XML file as an OpenLNS CT network report in Internet Explorer (other Web browsers may not display the data or may display it in a text-only format). This report is available as long as the **Exclude XSLT Stylesheet** option in the **XML Export Options** dialog box is cleared.

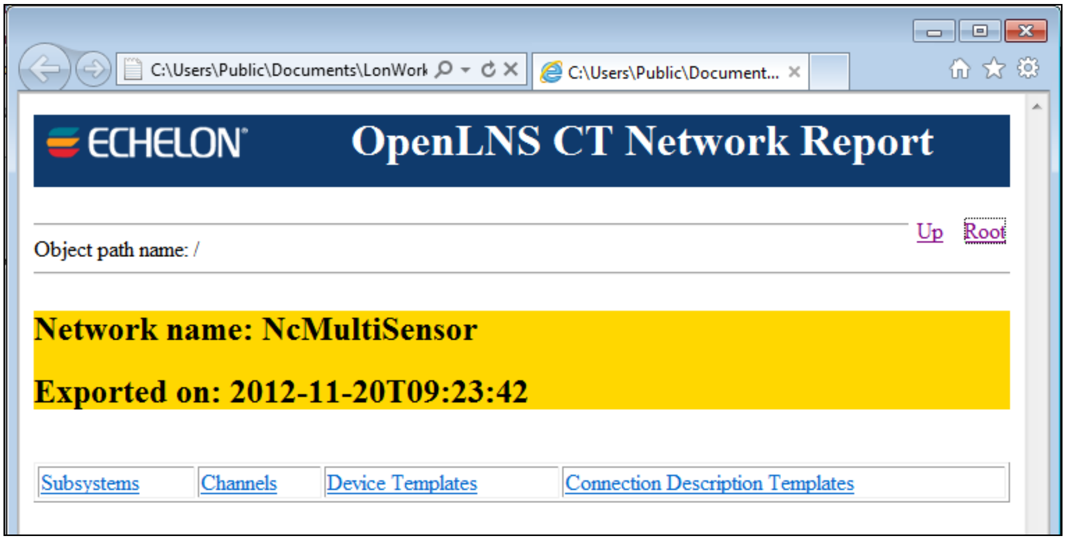
To view an OpenLNS CT network report, follow these steps:

1. Generate a LONWORKS network XML file.
2. Clear the **Exclude XSLT Stylesheet** check box in the **XML Export Options** dialog box.
3. Optionally modify the LONWORKS network XML file directly or with another application.

4. Either open the OpenLNS CT XML file from Internet Explorer, or open the folder containing the OpenLNS CT XML file and double-click the file (Internet Explorer must be your default Web browser for this method to work). The OpenLNS CT network report opens.

5. If you are using Internet Explorer 9, press F12 to open the Developer Tools, click **Browser Mode**, and select **Internet Explorer 9 Compatibility Mode**. This enables Internet Explorer 9 to display the objects in the report.

This report consists of a main network page with the network name, export date and time, and tables with hyperlinks to view the exported data for subsystems, channels, device templates, and connection descriptions.


C:\Users\Public\Documents\LonWork
C:\Users\Public\Document...
ECHELON

OpenLNS CT Network Report

Up Root

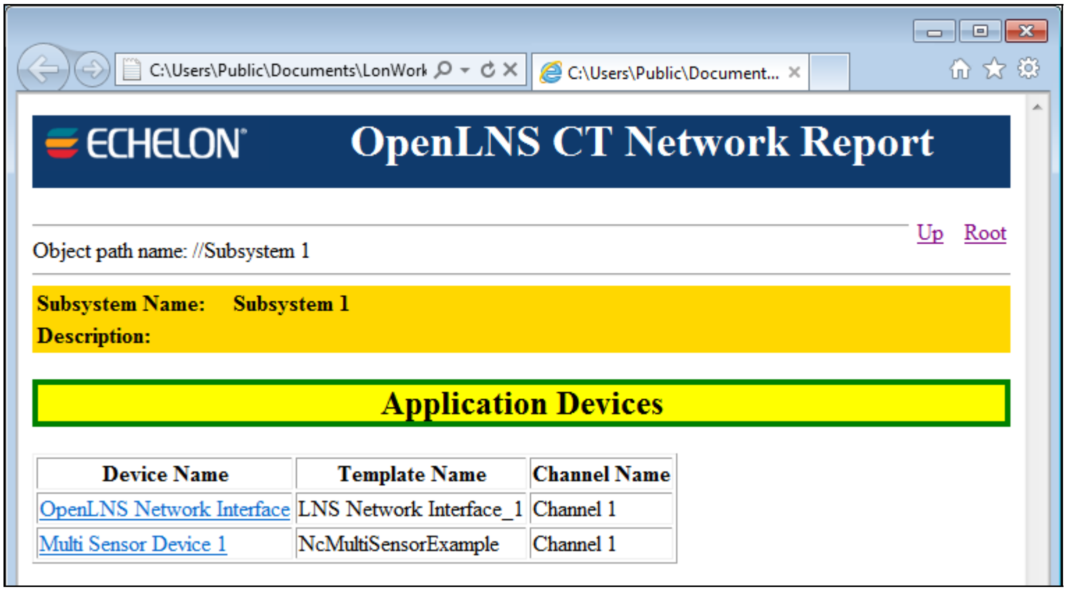
Object path name: /

Network name: NcMultiSensor

Exported on: 2012-11-20T09:23:42

Subsystems	Channels	Device Templates	Connection Description Templates

You can view the exported data following the network hierarchy—starting from the root subsystem through the contained subsystems, devices, functional blocks, and network variables down to the configuration properties associated with a network variable. For example, click the **Subsystems** link to open a page with tables displaying basic data for all the routers, devices, and nested subsystems in the selected subsystem.


C:\Users\Public\Documents\LonWork
C:\Users\Public\Document...
ECHELON

OpenLNS CT Network Report

Up Root

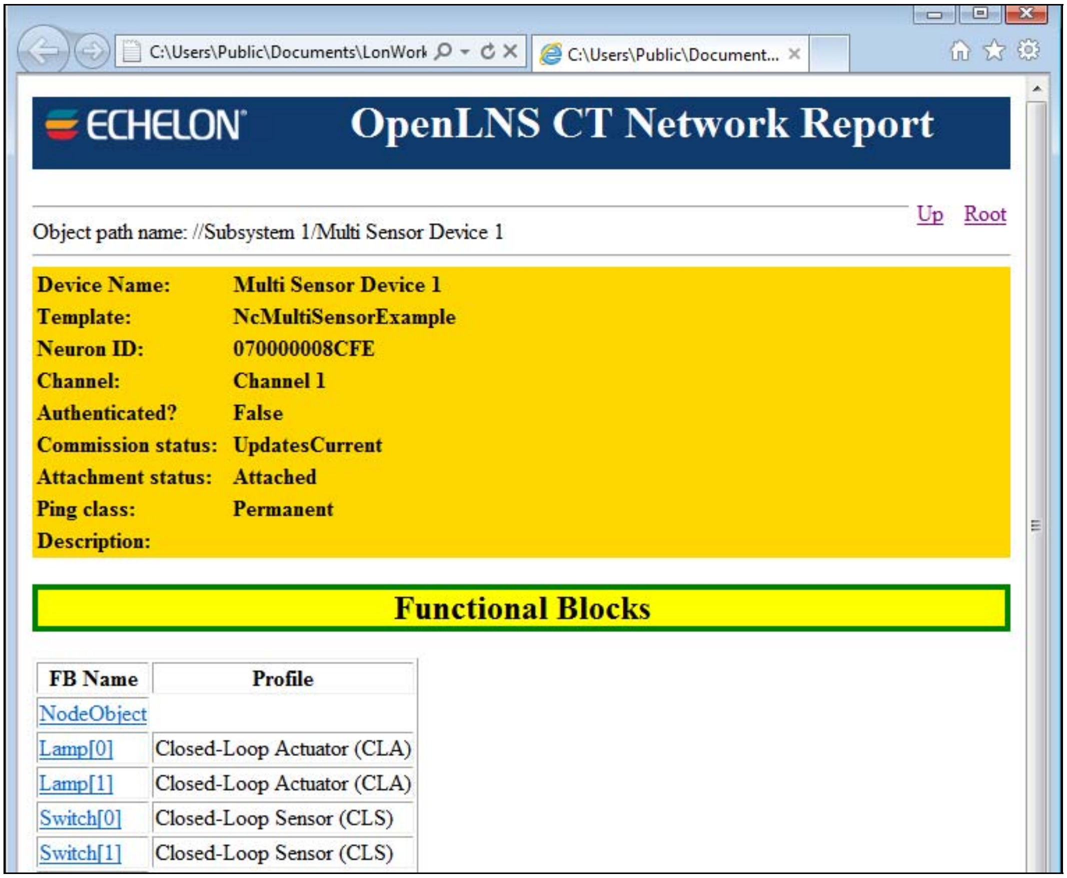
Object path name: //Subsystem 1

Subsystem Name: **Subsystem 1**
Description:

Application Devices

Device Name	Template Name	Channel Name
OpenLNS Network Interface	LNS Network Interface_1	Channel 1
Multi Sensor Device 1	NcMultiSensorExample	Channel 1

Click a device link to open a page with detailed data on that device and a table listing its functional blocks, with links to view detailed data on each one.


C:\Users\Public\Documents\LonWork
C:\Users\Public\Document...
ECHELON

OpenLNS CT Network Report

Up Root

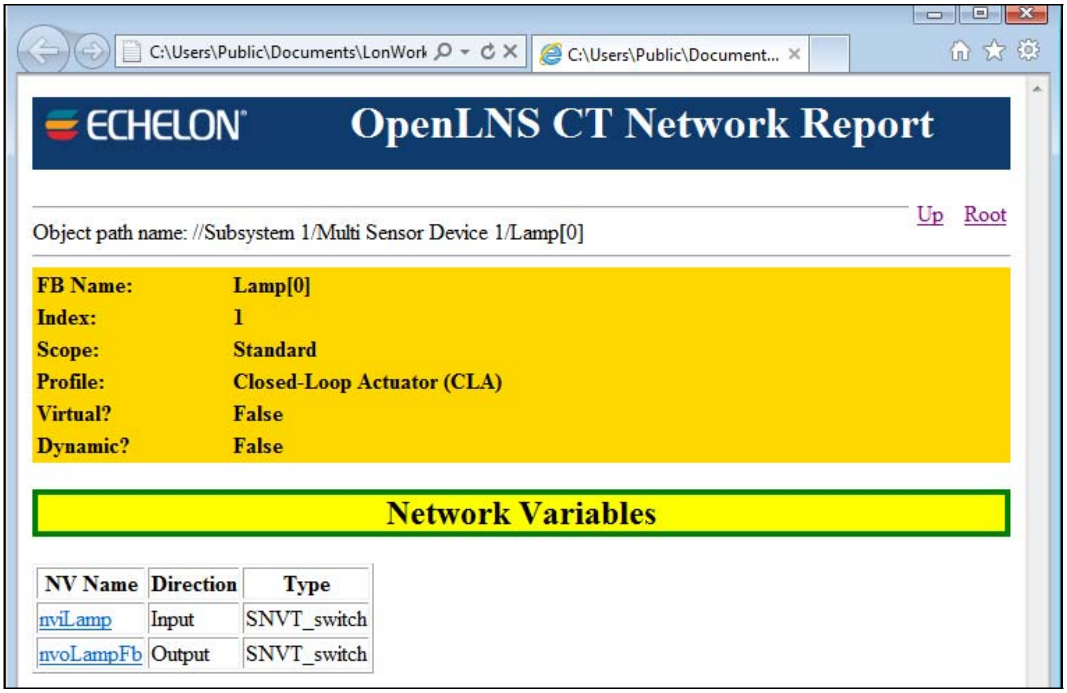
Object path name: //Subsystem 1/Multi Sensor Device 1

Device Name:	**Multi Sensor Device 1**
Template:	**NcMultiSensorExample**
Neuron ID:	**070000008CFE**
Channel:	**Channel 1**
Authenticated?	**False**
Commission status:	**UpdatesCurrent**
Attachment status:	**Attached**
Ping class:	**Permanent**
Description:	

Functional Blocks

FB Name	Profile
NodeObject	
Lamp[0]	Closed-Loop Actuator (CLA)
Lamp[1]	Closed-Loop Actuator (CLA)
Switch[0]	Closed-Loop Sensor (CLS)
Switch[1]	Closed-Loop Sensor (CLS)

Click a functional block link to open a page with detailed data on that functional block and a table listing its network variables and configuration properties, with links to view detailed data on each one.


C:\Users\Public\Documents\LonWork
C:\Users\Public\Document...
ECHELON

OpenLNS CT Network Report

Up Root

Object path name: //Subsystem 1/Multi Sensor Device 1/Lamp[0]

FB Name:	**Lamp[0]**
Index:	**1**
Scope:	**Standard**
Profile:	**Closed-Loop Actuator (CLA)**
Virtual?	**False**
Dynamic?	**False**

Network Variables

NV Name	Direction	Type
nviLamp	Input	SNVT_switch
nvoLampFb	Output	SNVT_switch

You can continue moving down the network hierarchy until reaching the end, which is at the network variable\configuration property or configuration property paths. Click **Up** to go back one level in the network hierarchy, or click **Root** to return to the main network page.

Importing a LONWORKS Network XML File

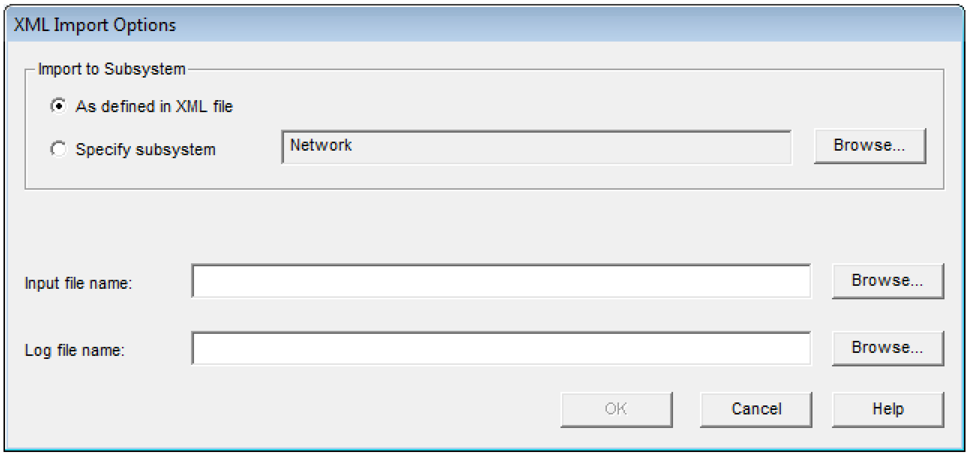
You can import a LONWORKS network XML file from another application using the OpenLNS CT XML Plug-in, or you can manually import a LONWORKS network XML file into an existing network using OpenLNS CT. Importing a LONWORKS network XML file will update the network based on the actions and attributes specified for the objects in the XML file. Objects in the OpenLNS network database are created or updated and the OpenLNS CT drawing is updated accordingly.

As described in *Exporting a LONWORKS Network XML File*, if you are using an OpenLNS application other than OpenLNS CT, you can use the OpenLNS CT XML Plug-in to automate importing your LONWORKS network from the other application. The OpenLNS CT XML Plug-in provides a programmatic interface to the XML import and export features in OpenLNS CT. See the *LONWORKS Network XML Programmer's Guide* for details on how applications can use the plug-in and the schema used for the XML file.

When OpenLNS CT imports a LONWORKS network XML file, it attempts to match each object in the XML file with an object in the OpenLNS database and OpenLNS CT drawing. OpenLNS CT matches objects based on Neuron ID, name, or handle (in that order). For each matching object, OpenLNS CT updates the attributes of the object based on any changes in the XML file. For each object that cannot be matched, OpenLNS CT creates a new matching object in the OpenLNS network database and the OpenLNS CT drawing. As a result, it is possible to create a new network from a LONWORKS network XML file by starting with a new OpenLNS CT drawing and importing the XML file. It is also possible to export an OpenLNS CT XML file from an existing network, make changes in the XML file, and then re-import the XML file to make the changes.

To manually import an XML file, follow these steps:

1. Click **Add-ins**, point to **OpenLNS CT Documents**, and then click **Import from XML**. The **XML Import Options** dialog box opens.



XML Import Options

Import to Subsystem

● As defined in XML file

○ Specify subsystem: Network

Input file name:

Log file name:

2. Set the following options:

Import to Subsystem

Select the root subsystem for the XML file. There are two choices:

- **As Defined in XML File**. Imports the XML file into the subsystem specified in the XML file.
- **Select Subsystem**. Imports the XML file into the subsystem you specify. By default, the top-level root subsystem will be selected. To change the subsystem, click **Browse**.

Input File Name Specifies the name of the XML file to import.

Log File Name Specifies the name of the output XML log file that is generated during the XML import. This log lists the objects created, modified, or deleted, and any errors encountered. The default name is **C:\Users\Public\Documents\LonWorks\OpenLnsCt\XML\** *network***Export**[*_version*]**Log.xml.**

If this field is clear, an XML log file is not created.

3. Click **OK**. The OpenLNS CT **XML Import Status** window displaying the report status opens, and OpenLNS CT imports the XML file. You can click **Cancel** to stop the XML import. When “Import complete” opens, click **OK** to close the window.

Tips:

- If you are making changes that affect the OpenLNS CT drawing, set the OpenLNS CT event log limit high enough to hold all of the events that may be generated by the XML file import. You can set the maximum number of log entries in the **Network Properties: Synchronization** options category (see Appendix A for more information on this option). Alternatively, in the <**LONWORKSNetwork**> section of the XML file, you can insert the limit between a pair of EventLimit tags, using the following format: **<EventLimit>***n***</EventLimit>**, where *n* is the maximum number of log entries saved per drawing file.
- To reduce the time required to import a LONWORKS network XML file, do the following: remove entries from the LONWORKS network XML file that do not need to be updated before importing the network (especially collections of objects); import the XML file with OpenLNS CT OffNet; close any remote applications if you are importing a large amount of data.

10

Managing OpenLNS CT Licenses

This chapter provides an overview of OpenLNS CT licensing. It describes how to upgrade OpenLNS CT, upgrade an OpenLNS CT Standard Edition to the Professional Edition, renew your OpenLNS CT annual maintenance, and transfer an OpenLNS CT license.

Commissioning Devices with OpenLNS CT

OpenLNS CT lets you install devices without commissioning fees (known as “credits” in LonMaker Turbo Editions and previous versions of the LonMaker software). OpenLNS CT features no-cost installation for all LONWORKS devices that comply with the ISO/IEC 14908-1 Control Network Protocol. This includes devices based on the FT 5000 Smart Transceiver, Neuron® 5000 Processor, FT 3150®/3120® Smart Transceiver, or PL 3170/3150/3120 Smart Transceiver, and also includes devices based on third-party ISO/IEC 14908-1 protocol processors.

Overview of OpenLNS CT Licensing

The OpenLNS CT software product includes up to two licenses for both OpenLNS Server and OpenLNS CT. You can install the OpenLNS CT licenses on up to two computers—a primary and a secondary computer—provided that the following requirements are met: only the OpenLNS CT user on the primary computer can use the copy of OpenLNS CT on the secondary computer and the OpenLNS CT software cannot be used on the primary and secondary computers at the same time.

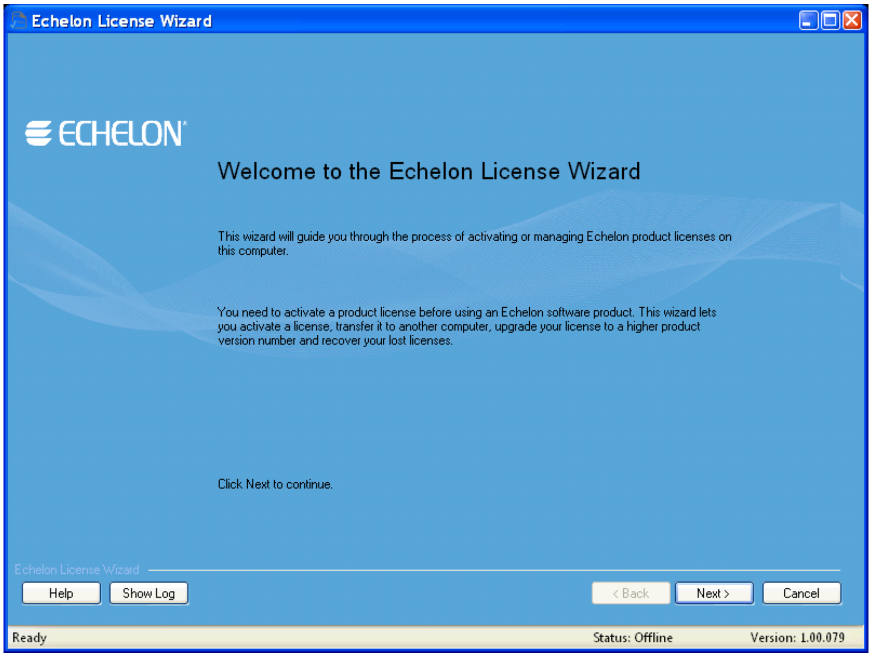
Maintaining OpenLNS CT

The OpenLNS CT Standard and Professional Editions include one-year maintenance for both OpenLNS and OpenLNS CT during which you can download and install software updates and upgrades for both OpenLNS CT and OpenLNS Server for free (Visio upgrades are not included). Your OpenLNS CT software will continue to operate normally if you let your maintenance coverage lapse, but you will not be able to install any updates or upgrades. When you update or upgrade the OpenLNS CT software (and your OpenLNS CT computer has an Internet connection), the Echelon License Manager will automatically verify whether your maintenance coverage is active for the release you are installing.

- If your maintenance contract is valid for the release you are installing, it will allow you to continue installing your OpenLNS CT update or upgrade.
- If your maintenance contract has expired, the installer will prompt you as to whether you want to continue installing the OpenLNS CT update or upgrade. If you continue installing the update or upgrade, the OpenLNS CT software will not run until you renew your maintenance. See *Renewing OpenLNS CT Annual Maintenance* later in this chapter for more information on how to do this.

If your OpenLNS CT computer did not have an Internet connection when you installed the software updates or upgrades and you have valid maintenance coverage for OpenLNS Server and OpenLNS CT, you can activate your licenses for the updated or upgraded OpenLNS and OpenLNS CT software with the **Echelon License Wizard** following these steps:

1. Open the Echelon License Wizard. Click **Start**, click **Programs**, point to **Echelon License Wizard**, and then click **License Wizard**. The **Echelon License Wizard** opens.



Welcome to the Echelon License Wizard

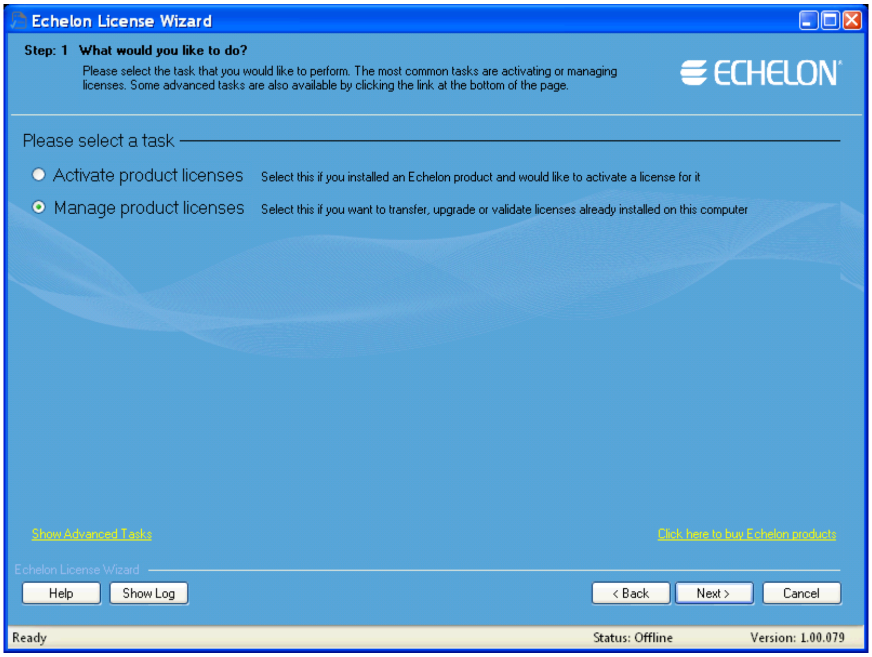
This wizard will guide you through the process of activating or managing Echelon product licenses on this computer.

You need to activate a product license before using an Echelon software product. This wizard lets you activate a license, transfer it to another computer, upgrade your license to a higher product version number and recover your lost licenses.

Click Next to continue.

2. The License Wizard guides you through the process of manually updating or upgrading your OpenLNS CT license. Click **Next.**

3. The **Step 1: What would you like to do?** dialog opens.



Step: 1 What would you like to do?

Please select the task that you would like to perform. The most common tasks are activating or managing licenses. Some advanced tasks are also available by clicking the link at the bottom of the page.

Please select a task

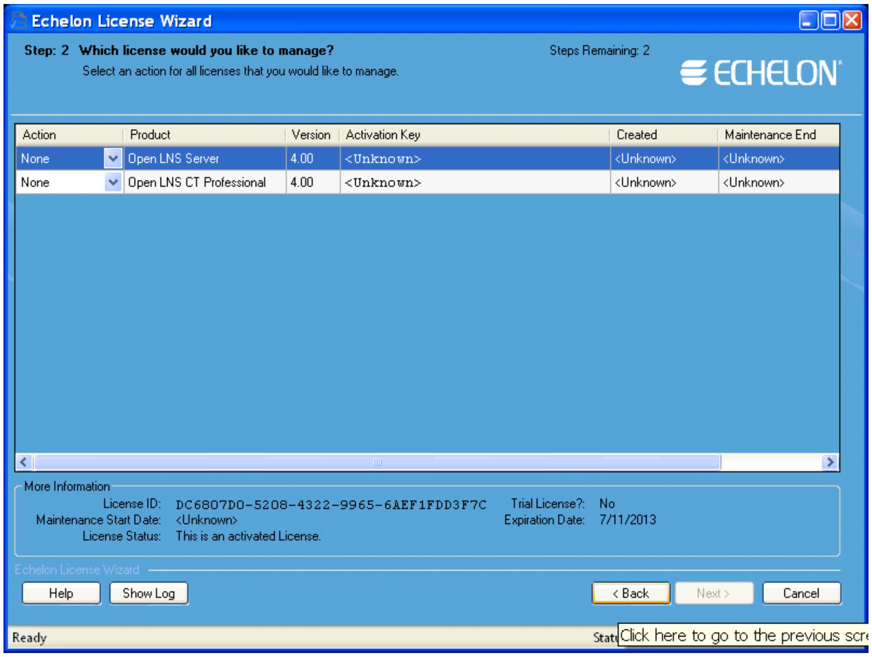
○ Activate product licenses Select this if you installed an Echelon product and would like to activate a license for it

● Manage product licenses Select this if you want to transfer, upgrade or validate licenses already installed on this computer

4. To update or upgrade your OpenLNS CT software using the over the Internet or via e-mail, click **Activate Product Licenses** and then click **Next**.

   To upgrade over the phone, click **Show Advanced Tasks**, **Upgrade a Product License over the Phone**, and then follow the instructions in the License Wizard. If your computer is offline, update or upgrade via e-mail if you have e-mail access because updating and upgrading over the phone requires manually entering many lengthy alpha-numeric strings.

5. The **Step 2: Which License Would You Like to Manage?** opens.



Echelon License Wizard

Step: 2 Which license would you like to manage?

Select an action for all licenses that you would like to manage.

Steps Remaining: 2

Action	Product	Version	Activation Key	Created	Maintenance End
None	Open LNS Server	4.00	<Unknown>	<Unknown>	<Unknown>
None	Open LNS CT Professional	4.00	<Unknown>	<Unknown>	<Unknown>

More Information

License ID: DC6807D0-5208-4322-9965-6AEF1FDD3F7C

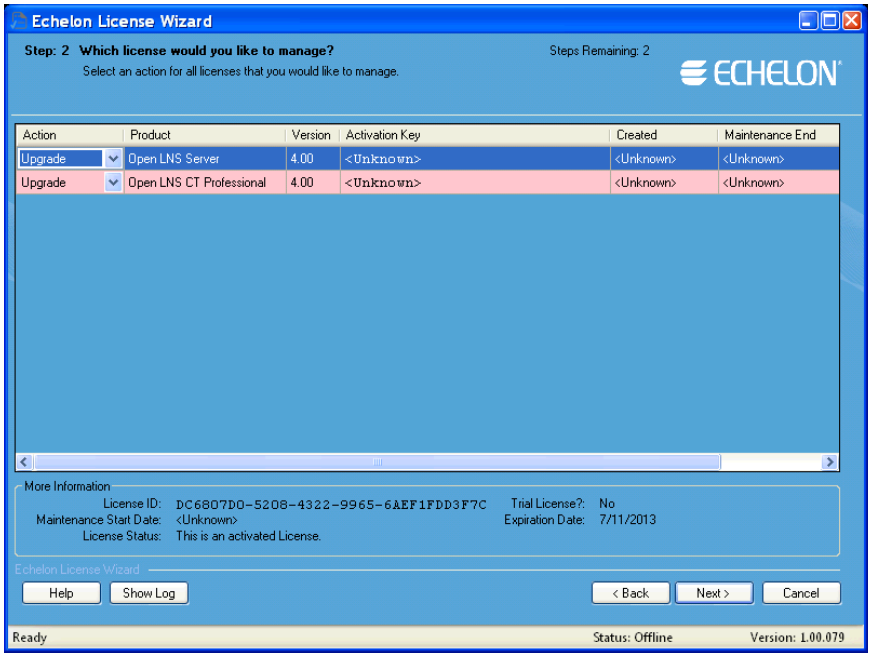
Maintenance Start Date: <Unknown>

License Status: This is an activated License.

Trial License?: No

Expiration Date: 7/11/2013

6. Select **Upgrade** from the **Action** list for both OpenLNS Server and OpenLNS CT, and then click **Next**.



Echelon License Wizard

Step: 2 Which license would you like to manage?

Select an action for all licenses that you would like to manage.

Steps Remaining: 2

Action	Product	Version	Activation Key	Created	Maintenance End
Upgrade	Open LNS Server	4.00	<Unknown>	<Unknown>	<Unknown>
Upgrade	Open LNS CT Professional	4.00	<Unknown>	<Unknown>	<Unknown>

More Information

License ID: DC6807D0-5208-4322-9965-6AEF1FDD3F7C

Maintenance Start Date: <Unknown>

License Status: This is an activated License.

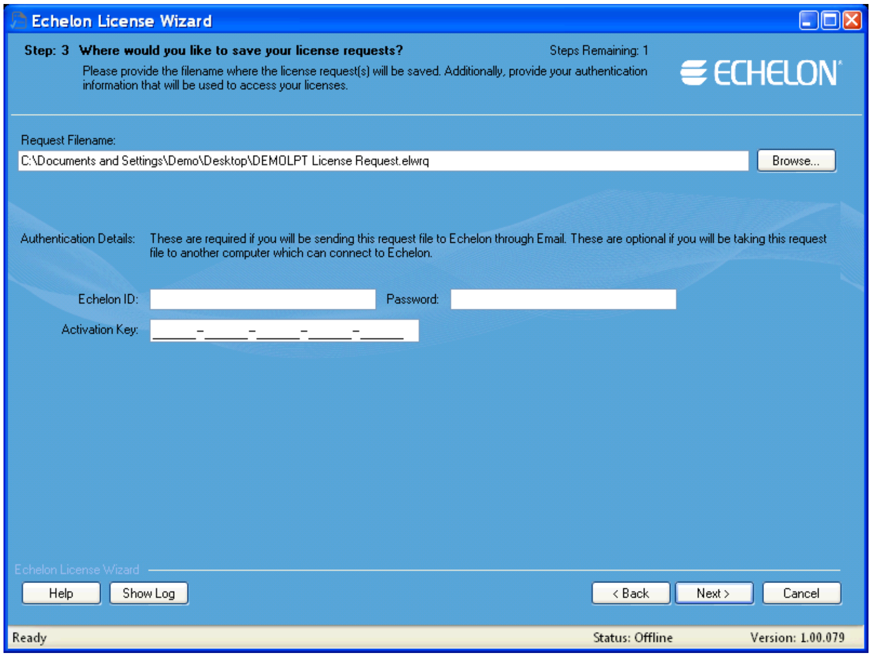
Trial License?: No

Expiration Date: 7/11/2013

Status: Offline

Version: 1.00.079

7. If your computer is online, the **Step 3: Status** dialog opens, and the License Server upgrades OpenLNS Server and OpenLNS CT. If your computer is offline, the **Step 3: Where Would You Like to Save your License Requests?** dialog opens.



Step: 3 Where would you like to save your license requests? Steps Remaining: 1

Please provide the filename where the license request(s) will be saved. Additionally, provide your authentication information that will be used to access your licenses.

Request Filename:
C:\Documents and Settings\Demo\Desktop\DEMOLPT License Request.elwrq

Authentication Details: These are required if you will be sending this request file to Echelon through Email. These are optional if you will be taking this request file to another computer which can connect to Echelon.

Echelon ID:
Password:
Activation Key: _____-_____-_____-_____-_____

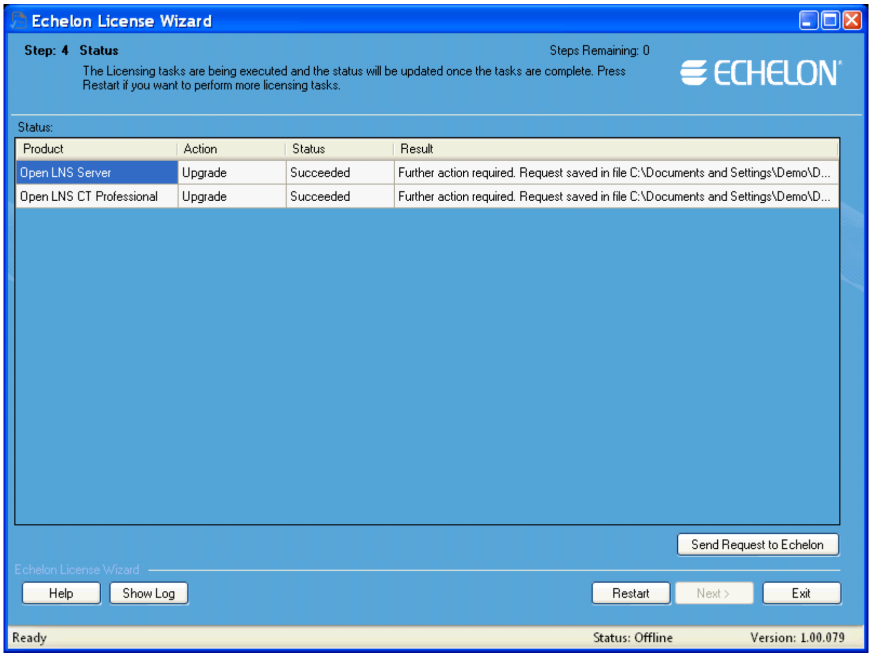
Ready Status: Offline Version: 1.00.079

8. In the **Request File Name** property, enter a full path where a license request file (**.elwrq** extension) is to be saved, or click **Browse**, specify the directory where the license request file is to be saved, and then enter a name for the file.

   Under **Authentication Details**, enter the Echelon ID (e-mail address) and Password you created for your Echelon download account. The **Activation Key** box displays the 25-character alphanumeric string for the Echelon product to be upgraded. Do not modify this property.

   Click **Next**.

9. The **Step 4**: **Status** dialog opens. The License Wizard confirms the creation of the license request file.



Step: 4 Status Steps Remaining: 0

The Licensing tasks are being executed and the status will be updated once the tasks are complete. Press Restart if you want to perform more licensing tasks.

Status:

Product	Action	Status	Result
Open LNS Server	Upgrade	Succeeded	Further action required. Request saved in file C:\Documents and Settings\Demo\D...
Open LNS CT Professional	Upgrade	Succeeded	Further action required. Request saved in file C:\Documents and Settings\Demo\D...

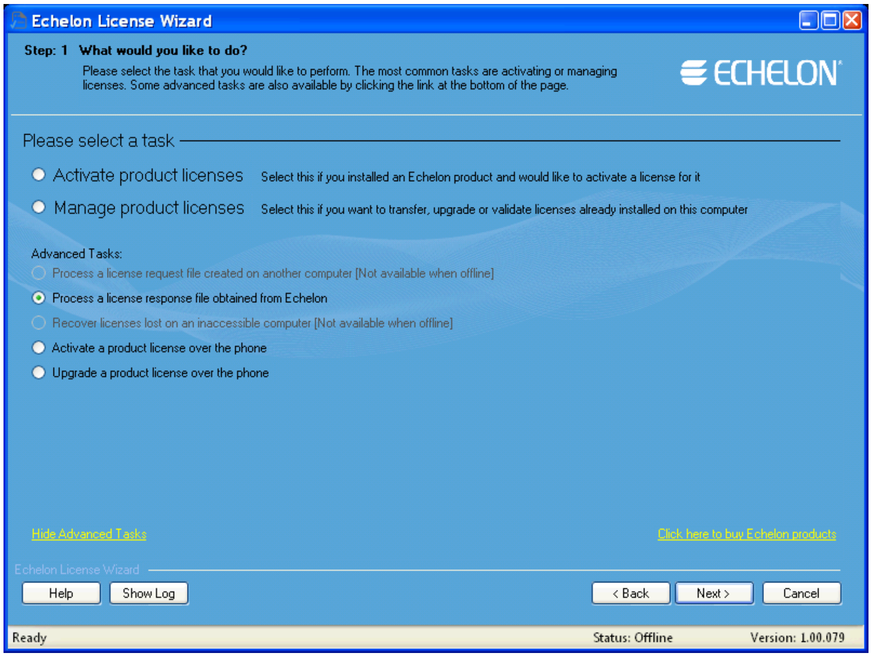
Ready Status: Offline Version: 1.00.079

10. Click **Send Request to Echelon** to e-mail the license request file to Echelon at *activation@echelon.com*. Echelon will send you an e-mail message with a license response (**.elwrs** extension) file. You can upgrade your license faster by copying the license request file to an Internet-connected computer with the License Wizard installed on it and importing it into the License Wizard. The License Wizard creates the license response file, which you can then copy to the original computer.

11. Import the license response (**.elwrs** extension) file into the License Wizard and update or upgrade your OpenLNS Server and OpenLNS CT licenses following these steps:

    a. Start the Echelon License Wizard.

    b. In the **Task Selection** dialog, click the **Show Advanced Topics** link, and then click the **Process a License Response File Obtained from Echelon** option.



Echelon License Wizard

Step: 1 What would you like to do?

Please select the task that you would like to perform. The most common tasks are activating or managing licenses. Some advanced tasks are also available by clicking the link at the bottom of the page.

Please select a task

○ Activate product licenses — Select this if you installed an Echelon product and would like to activate a license for it

○ Manage product licenses — Select this if you want to transfer, upgrade or validate licenses already installed on this computer

Advanced Tasks:

○ Process a license request file created on another computer [Not available when offline]

● Process a license response file obtained from Echelon

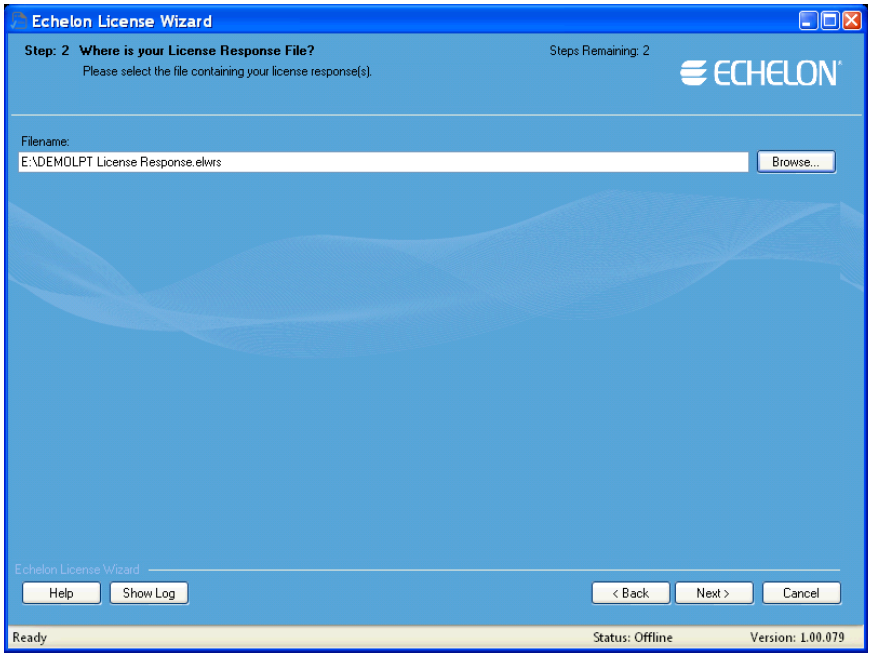
○ Recover licenses lost on an inaccessible computer [Not available when offline]

○ Activate a product license over the phone

○ Upgrade a product license over the phone

Ready — Status: Offline — Version: 1.00.079

    c. The **Import License Request or Response File** dialog opens. In the **File Name** property, enter the full path of the license response file (**.elwrs** extension), or click **Browse** and then select the file. Click **Next**.



Echelon License Wizard

Step: 2 **Where is your License Response File?**

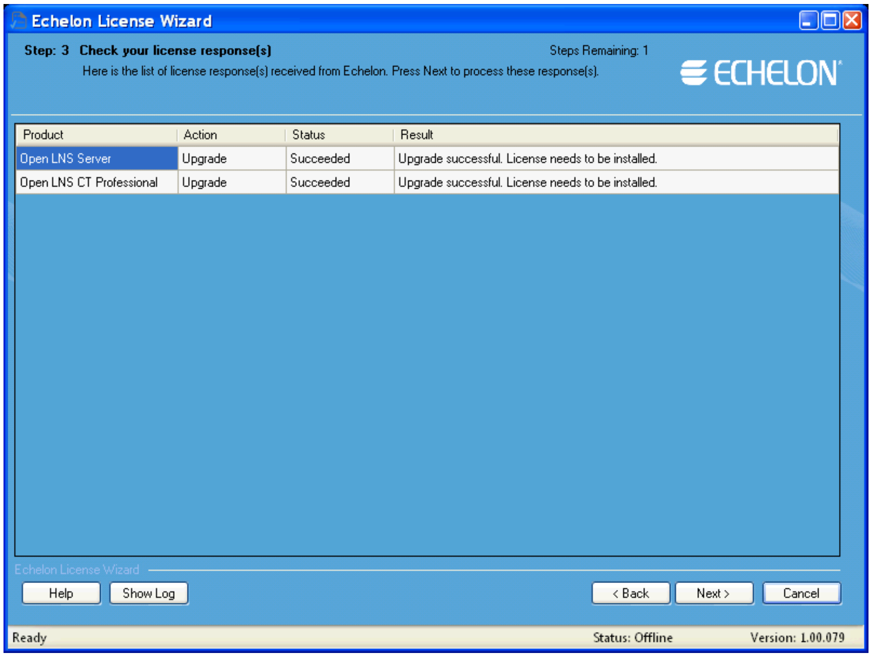
Please select the file containing your license response(s).

Steps Remaining: 2

Filename: E:\DEMOLPT License Response.elwrs

Ready Status: Offline Version: 1.00.079

d. The **Check License Response File** dialog opens. Confirm that the Echelon product licenses in the license response file are the ones you want to activate. Click **Next**. The License Server installs and upgrades the licenses in the license response file.



Echelon License Wizard

Step: 3 **Check your license response(s)**

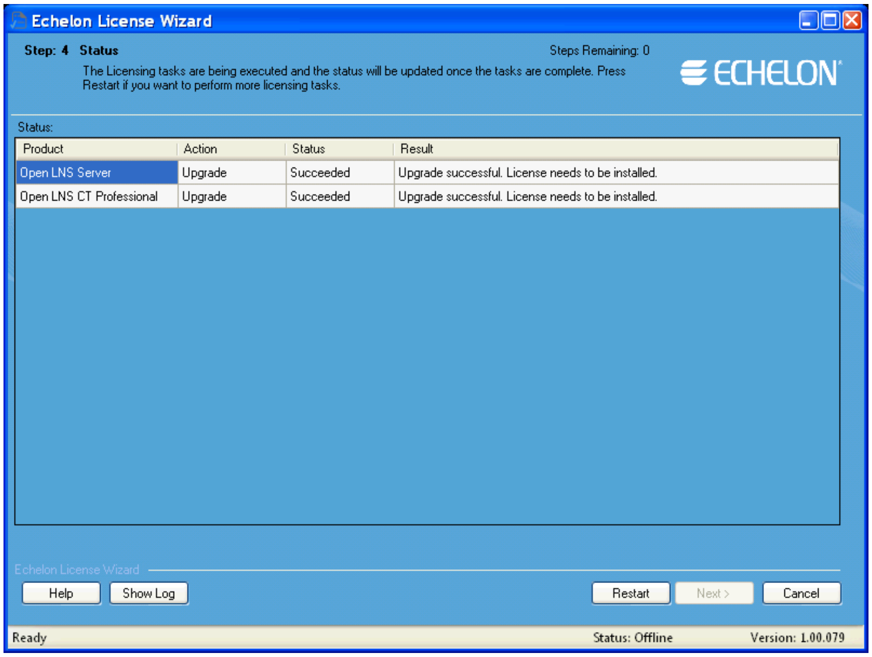
Here is the list of license response(s) received from Echelon. Press Next to process these response(s).

Steps Remaining: 1

Product	Action	Status	Result
Open LNS Server	Upgrade	Succeeded	Upgrade successful. License needs to be installed.
Open LNS CT Professional	Upgrade	Succeeded	Upgrade successful. License needs to be installed.

Ready Status: Offline Version: 1.00.079

e. The **Status** dialog opens and confirms that the OpenLNS Server and OpenLNS CT licenses have been successfully upgraded.



Step: 4 Status

Steps Remaining: 0

The Licensing tasks are being executed and the status will be updated once the tasks are complete. Press Restart if you want to perform more licensing tasks.

Status:

Product	Action	Status	Result
Open LNS Server	Upgrade	Succeeded	Upgrade successful. License needs to be installed.
Open LNS CT Professional	Upgrade	Succeeded	Upgrade successful. License needs to be installed.

Ready   Status: Offline   Version: 1.00.079

12. Click **Exit** to close the License Wizard.

Upgrading OpenLNS CT Standard Edition to OpenLNS CT Professional Edition

You can upgrade a copy of OpenLNS CT Standard Edition to OpenLNS CT Professional Edition if you need support for more networks. The Standard Edition supports up to five networks and the Professional Edition supports an unlimited number of networks. Upgrading from OpenLNS CT Standard to OpenLNS CT Professional does not upgrade the Visio edition. To upgrade from OpenLNS CT Standard Edition to OpenLNS CT Professional Edition, follow these steps:

1. Order the OpenLNS CT Standard to Professional Upgrade Key (Echelon model number 38070-400) from your Echelon Sales representative, Echelon distributor, or the Echelon online store at *store.echelon.com*. You will receive an e-mail message that contains instructions for upgrading to OpenLNS CT Professional Edition.

2. Use the License Wizard to return your OpenLNS Standard Edition following these steps:

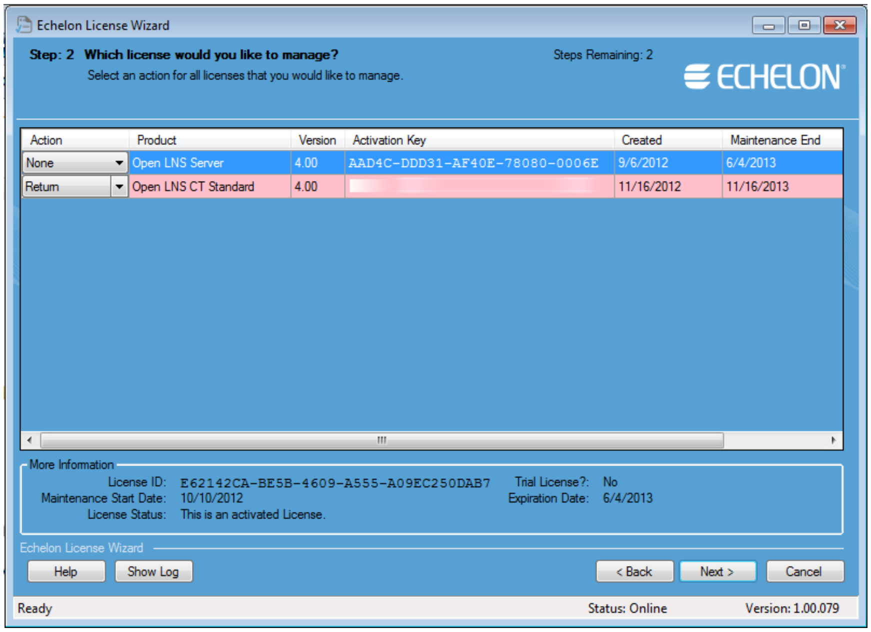
   a. Open the Echelon License Wizard. Click **Start**, click **Programs**, point to **Echelon License Wizard**, and then click **License Wizard**. The **Echelon License Wizard** opens. Click **Next**.

   b. The **Step 1: What would you like to do?** dialog opens.

      To return your OpenLNS CT software using the over the Internet or via e-mail, click **Manage Product Licenses** and then click **Next**.

      To upgrade over the phone, click **Show Advanced Tasks**, **Upgrade a Product License over the Phone**, and then follow the instructions in the License Wizard. If your computer is offline, upgrade via e-mail if you have e-mail access because upgrading over the phone requires manually entering many lengthy alpha-numeric strings.

   c. The **Step 2: Which License Would You Like to Manage?** dialog opens. Select **Return** from the **Action** column of your OpenLNS CT license, and then click **Next**.



Step: 2 Which license would you like to manage? Steps Remaining: 2

Select an action for all licenses that you would like to manage.

Action	Product	Version	Activation Key	Created	Maintenance End
None	Open LNS Server	4.00	AAD4C-DDD31-AF40E-78080-0006E	9/6/2012	6/4/2013
Return	Open LNS CT Standard	4.00	[redacted]	11/16/2012	11/16/2013

More Information

License ID: E62142CA-BE5B-4609-A555-A09EC250DAB7
Maintenance Start Date: 10/10/2012
License Status: This is an activated License.
Trial License?: No
Expiration Date: 6/4/2013

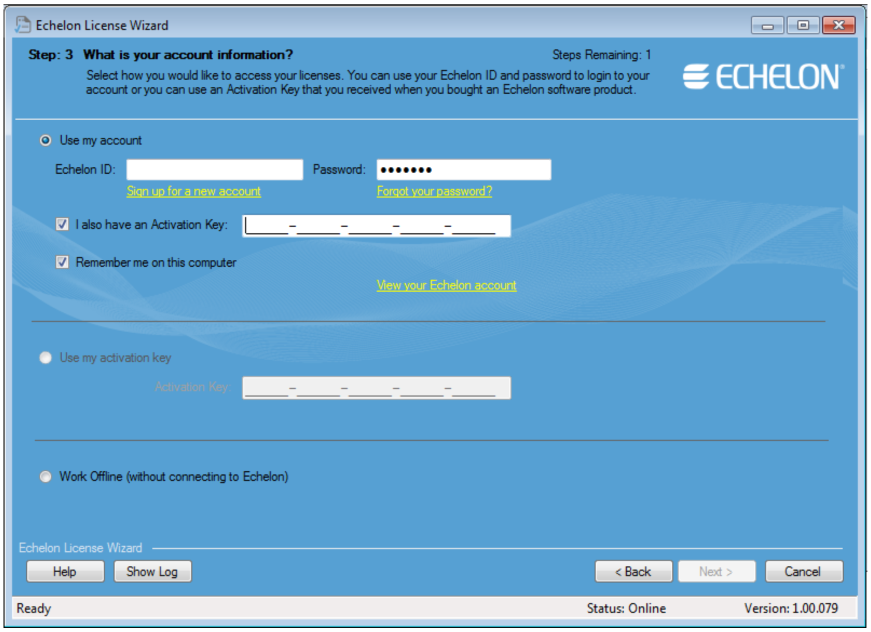
Ready Status: Online Version: 1.00.079

d. If your computer is online, the **Step 3: Enter Customer Information** dialog opens. Complete the following steps to return your software (if your computer is offline, proceed to step e):

- Enter the User ID (e-mail address) and Password you created for your Echelon download account. If you do not have an Echelon download account, click the **Sign Up for a New Account** link, and create one on the *Echelon Web site*. If you forgot your password, you can retrieve it by clicking the **Forgot your Password** link. This opens the Echelon Web site, where you can request the password to be e-mailed to you.

  Optionally, you can select the **Activation Key** check box and enter your product's 25-character alphanumeric string (five groups of five characters) that you received when you purchased your Echelon software product or products. This associates the product's activation key with your account, and it identifies all the Echelon products that you purchased in a single transaction.

  Optionally, you can select the **Remember Me on this Computer** check box to preserve your login credentials. This check box is selected by default, which means that your credentials will automatically be entered the next time you start the License Wizard.



Echelon License Wizard

Step: 3 What is your account information? Steps Remaining: 1

Select how you would like to access your licenses. You can use your Echelon ID and password to login to your account or you can use an Activation Key that you received when you bought an Echelon software product.

● Use my account

Echelon ID: Password: •••••••

Sign up for a new account Forgot your password?

☑ I also have an Activation Key: _____-_____-_____-_____-_____

☑ Remember me on this computer

View your Echelon account

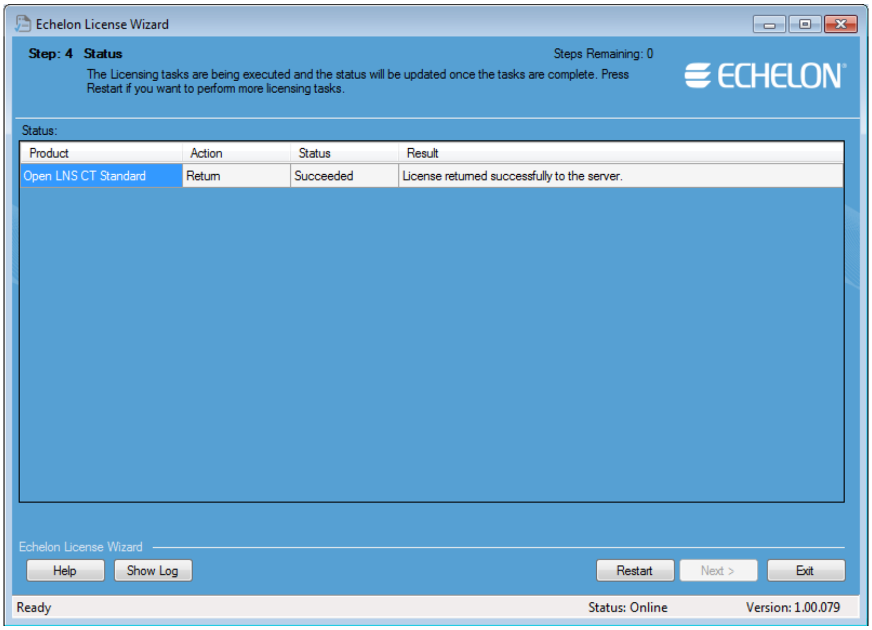
○ Use my activation key

Activation Key: _____-_____-_____-_____-_____

○ Work Offline (without connecting to Echelon)

Ready Status: Online Version: 1.00.079

- The **Step 4: Status** dialog opens, and the License Server returns your OpenLNS CT Standard license.



Echelon License Wizard

Step: 4 Status Steps Remaining: 0

The Licensing tasks are being executed and the status will be updated once the tasks are complete. Press Restart if you want to perform more licensing tasks.

Status:

Product	Action	Status	Result
Open LNS CT Standard	Return	Succeeded	License returned successfully to the server.

Ready Status: Online Version: 1.00.079

- Click **Exit** to close the License Wizard.

e. If your computer is offline, the **Step 3: Where Would You Like to Save your License Requests?** dialog opens.

f. In the **Request File Name** property, enter a full path where a license request file (**.elwrq** extension) is to be saved, or click **Browse**, specify the directory where the license request file is to be saved, and then enter a name for the file.

Under **Authentication Details**, enter the Echelon ID (e-mail address) and Password you created for your Echelon download account. The **Activation Key** box displays the 25-character alphanumeric string for the Echelon product to be upgraded. Do not modify this property.

Click **Next**.

g. The **Step 4**: **Status** dialog opens. The License Wizard confirms the creation of the license request file.

h. Click **Send Request to Echelon** to e-mail the license request file to Echelon at *activation@echelon.com*. Echelon will send you an e-mail message with a license response (**.elwrs** extension) file. You can upgrade your license faster by copying the license request file to an Internet-connected computer with the License Wizard installed on it and importing it into the License Wizard. The License Wizard creates the license response file, which you can then copy to the original computer.

i. Import the license response (**.elwrs** extension) file into the License Wizard and return your OpenLNS CT Standard license following these steps:

- Start the Echelon License Wizard.
- In the **Task Selection** dialog, click the **Show Advanced Topics** link, and then click the **Process a License Response File Obtained from Echelon** option.
- The **Import License Request or Response File** dialog opens. In the **File Name** property, enter the full path of the license response file (**.elwrs** extension), or click **Browse** and then select the file. Click **Next**.
- The **Check License Response File** dialog opens. Confirm that the Echelon product licenses in the license response file are the ones you want to activate. Click **Next**. The License Server installs and upgrades the licenses in the license response file.
- The **Status** dialog opens and confirms that the OpenLNS CT licenses have been successfully upgraded.

3. Your OpenLNS CT Standard Edition license is disabled.
4. Echelon will send you an e-mail message with an activation key for the OpenLNS Commissioning Tool Professional (Without Visio) Activation Key product.
5. Use the License Wizard to install your license for the OpenLNS Commissioning Tool Professional (Without Visio) Activation Key product. See *Manually Activating OpenLNS CT* in Chapter 2 for how to activate your OpenLNS CT Professional license.

Renewing OpenLNS CT Annual Maintenance

You can renew your annual maintenance for OpenLNS Server and OpenLNS CT before they expire. Renewing your OpenLNS and OpenLNS CT annual maintenance enables you to continue installing software updates and upgrades. If you do not renew the products' maintenance, you can still use the products; however, you will not be able to upgrade it with updates or upgrades released after your annual maintenance expired. You cannot renew your annual maintenance after it has expired. Once it has expired, you must purchase a new activation key to be able to install any updates or upgrades released after your maintenance expired.

To renew your annual maintenance, purchase an OpenLNS Commissioning Tool Professional One-Year Maintenance Renewal (Echelon model number 93800) or an OpenLNS Commissioning Tool Standard One-Year Maintenance Renewal (Echelon model number 93810). Echelon will process your order, and your maintenance coverage will be extended so that you can continue updating and upgrading OpenLNS and OpenLNS CT for an additional year.

Transferring OpenLNS CT Licenses

You can transfer your OpenLNS CT license from one computer to another using the **Echelon License Wizard**. Transferring your license lets you upgrade or switch your OpenLNS CT computer without losing your license. You cannot use the **License Wizard** to install one OpenLNS CT license on multiple computers.

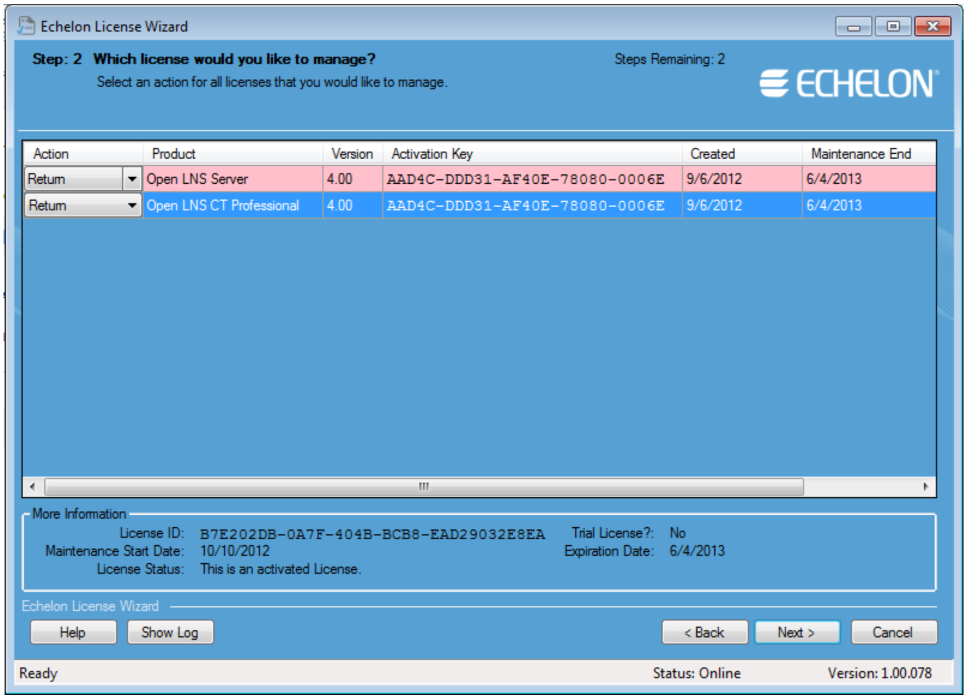
To transfer your OpenLNS CT license, follow these steps:

1. On the source computer, return the OpenLNS CT license following these steps:

   a. Open the Echelon License Wizard. Click **Start**, click **Programs**, point to **Echelon License Wizard**, and then click **License Wizard**. The **Echelon License Wizard** opens. Click **Next**.

   b. The **Step 1: What Would You Like to Do?** dialog opens. Click **Manage Product Licenses** and then click **Next**.

   c. The **Step 2: Which License Would You Like to Manage?** dialog opens. Select **Return** from the **Action** column of your OpenLNS Server and OpenLNS CT licenses, and then click **Next**



Step: 2 Which license would you like to manage? Steps Remaining: 2

Select an action for all licenses that you would like to manage.

Action	Product	Version	Activation Key	Created	Maintenance End
Return	Open LNS Server	4.00	AAD4C-DDD31-AF40E-78080-0006E	9/6/2012	6/4/2013
Return	Open LNS CT Professional	4.00	AAD4C-DDD31-AF40E-78080-0006E	9/6/2012	6/4/2013

More Information

License ID: B7E202DB-0A7F-404B-BCB8-EAD29032E8EA

Maintenance Start Date: 10/10/2012

License Status: This is an activated License.

Trial License?: No

Expiration Date: 6/4/2013

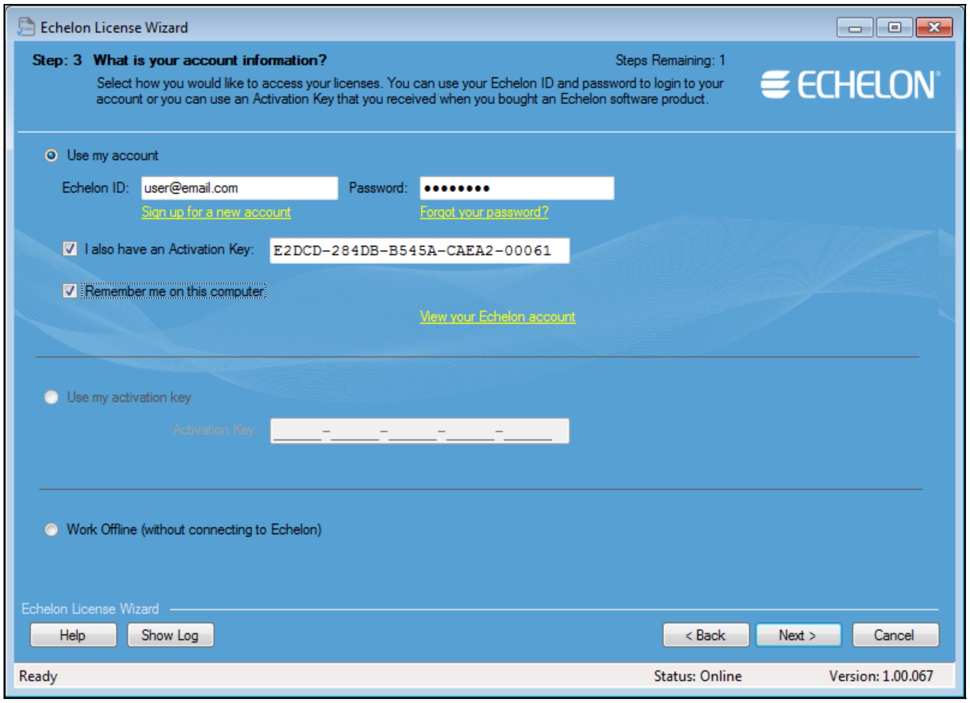
Ready Status: Online Version: 1.00.078

   d. The **Step 3: Enter Customer Information** dialog opens. Enter the User ID (e-mail address) and Password you created for your Echelon download account. If you do not have an Echelon download account, click the **Sign Up for a New Account** link, and create one on the *Echelon Web site*. If you forgot your password, you can retrieve it by clicking the **Forgot Your Password** link. This opens the Echelon Web site, where you can request the password to be e-mailed to you.

   Optionally, you can select the **Activation Key** check box and enter your product's 25-character alphanumeric string (five groups of five characters) that you received when you purchased your Echelon software product or products. This associates the product's activation key with your account, and it identifies all the Echelon products that you purchased in a single transaction.

   Optionally, you can select the **Remember Me on this Computer** check box to preserve your login credentials. This check box is selected by default, which means that your credentials will automatically be entered the next time you start the License Wizard.

   If your computer does not have an Internet connection, click **Work Offline**, and then create a license request file (**.elwrq** extension) in the **Step 4: Where Would You Like to Save Your License Requests** dialog. After you save your license request, you can activate the license on the target computer by e-mailing the license request file to Echelon at *activation@echelon.com*, or by copying it to the target computer (if it has an Internet connection) and importing it into the License Wizard.



Echelon License Wizard

Step: 3 What is your account information? Steps Remaining: 1

Select how you would like to access your licenses. You can use your Echelon ID and password to login to your account or you can use an Activation Key that you received when you bought an Echelon software product.

● Use my account

Echelon ID: user@email.com Password: ••••••••

☑ I also have an Activation Key: E2DCD-284DB-B545A-CAEA2-00061

☑ Remember me on this computer

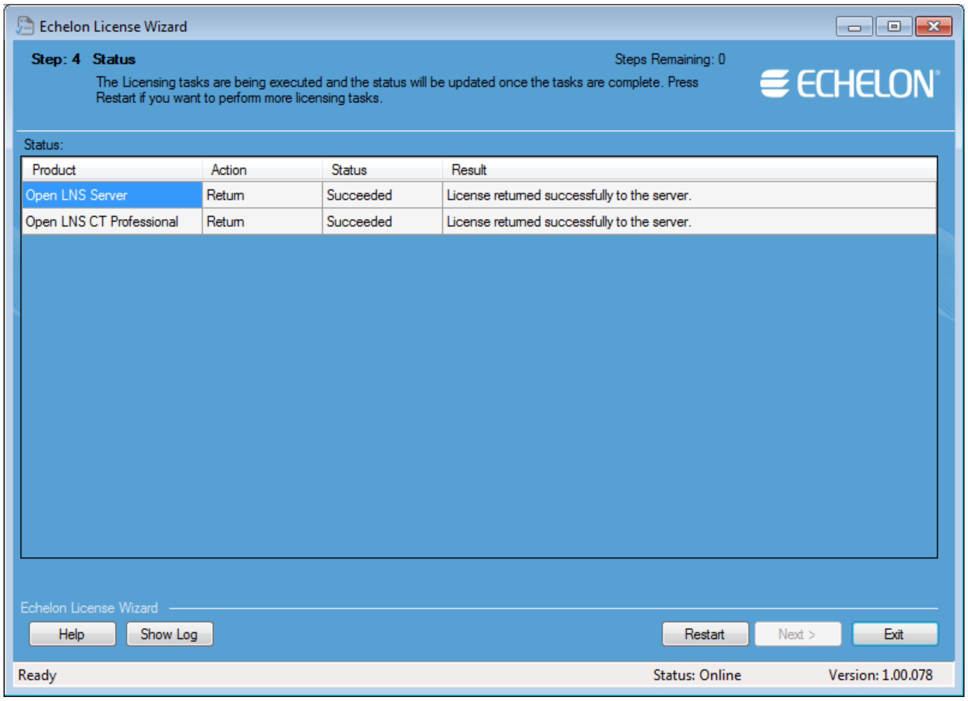
○ Use my activation key

Activation Key: _____-_____-_____-_____-_____

○ Work Offline (without connecting to Echelon)

Ready Status: Online Version: 1.00.067

e. Your OpenLNS Server and OpenLNS CT licenses are returned to the Echelon License Server. The **Step 4: Status** dialog opens and report the results of the license return.



Echelon License Wizard

Step: 4 Status Steps Remaining: 0

The Licensing tasks are being executed and the status will be updated once the tasks are complete. Press Restart if you want to perform more licensing tasks.

Status:

Product	Action	Status	Result
Open LNS Server	Return	Succeeded	License returned successfully to the server.
Open LNS CT Professional	Return	Succeeded	License returned successfully to the server.

Ready Status: Online Version: 1.00.078

2. Install OpenLNS CT on the target computer following the steps described in Chapter 2, *Installing and Activating OpenLNS CT*. In the Customer Information window, enter the activation key you returned in step 1.

11

Using Plug-ins

This chapter provides an overview of plug-ins and then describes how to start a plug-in, get plug-in information, and re-register, enable, and disable plug-ins.

Using Plug-ins Overview

Plug-ins are applications that you can start from within OpenLNS CT to perform a specialized task, often relating to a specific type of device or functional block. Device manufacturers may provide plug-ins that you can use with OpenLNS CT to configure their devices. Any plug-in conforming to the OpenLNS plug-in guidelines may be used with OpenLNS CT. You can view and download free Echelon and third-party plug-ins at *www.echelon.com/plugins*.

Plug-ins do not have to be specific to a device or a functional block type. System-wide plug-ins can provide generic services that may be used with multiple device types. For example, the OpenLNS CT *Browser* is a generic plug-in that can be used on any functional block to view and modify its network variables and configuration properties.

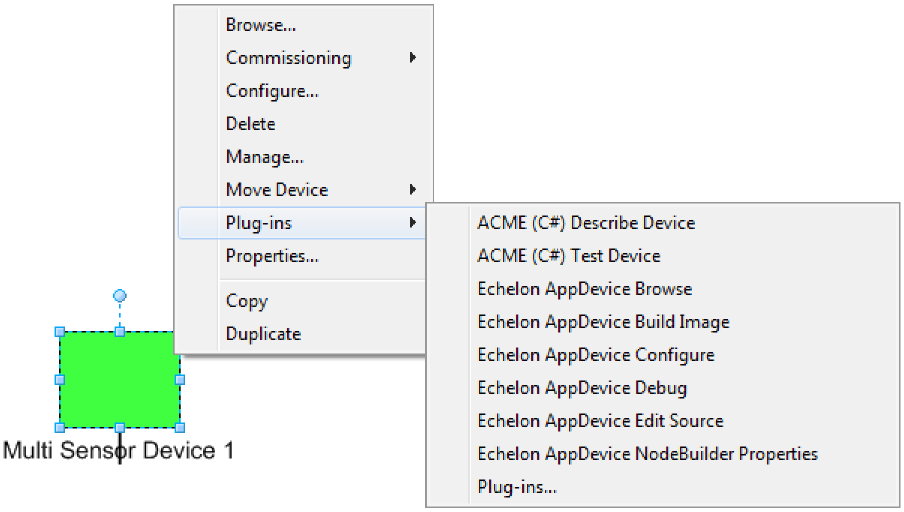
Each plug-in can implement multiple commands. For example, a typical device plug-in implements a **Configure** command for each functional block type used in the device. Each plug-in command may be associated with a device type, functional block type, a subsystem, or an entire network.

Starting a Plug-in

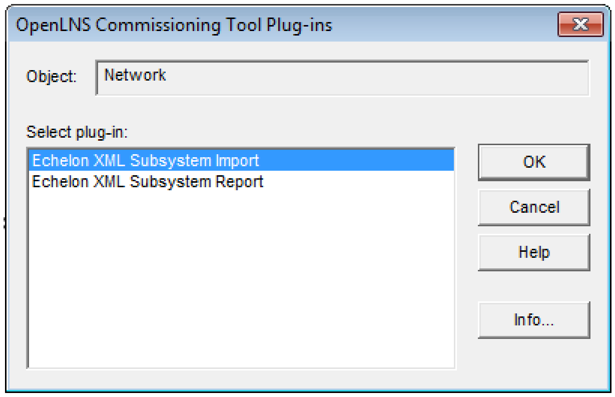
You can start a plug-in from a plug-in command list or by using a default command.

A plug-in command lists provides all the available plug-ins for an object within your OpenLNS CT drawing. Plug-in command lists are available for any device SmartShape, functional block SmartShape, subsystem, OpenLNS CT system, or OpenLNS CT network. To start a plug-in for a device or functional block SmartShape, subsystem, or network, do one of the following:

- For a device or functional block SmartShape, right-click the SmartShape, point to **Plug-Ins**, and click the desired plug-in on the shortcut menu.



- For a subsystem, right-click an empty space in the drawing page, point to **Plug-Ins**, and click the desired plug-in on the shortcut menu.
- For the network, click **Add-ins**, click **OpenLNS CT**, click **Plug-Ins**, and then click **Network Plug-Ins**. The **OpenLNS CT Plug-ins** dialog opens. This dialog lists the plug-ins that have been registered for the network. Select the desired plug-in and then click **OK**.

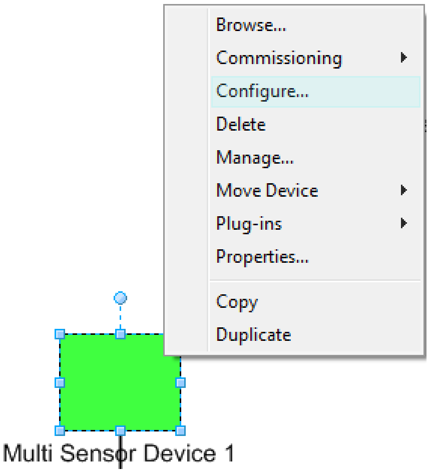

OpenLNS Commissioning Tool Plug-ins

Object: Network

Select plug-in:

Echelon XML Subsystem Import
Echelon XML Subsystem Report

- To start a plug-in using a default command, right click a device or functional block SmartShape, and then select **Configure** or **Browse**. If a plug-in is registered for the selected command and functional block type, the plug-in window opens.



You can display information about the plug-in commands that are installed on your computer and registered for the network, and which commands are default commands, as described in *Getting Plug-in Information*.

Notes:

- If you try to use a functional block plug-in on a virtual functional block, the plug-in may fail. Plug-ins are typically designed to be used with functional blocks. A virtual functional block is how OpenLNS CT represents the network variables and configuration properties on a device that are not contained by a functional block. The virtual functional block is not a real functional block, but a convenient way provided by OpenLNS CT to access the portions of a device interface that are not part of a functional block.
- To use a plug-in, it must be properly installed and registered. See *Re-Registering Plug-ins* and your plug-in documentation for more information.

Viewing Plug-in Information

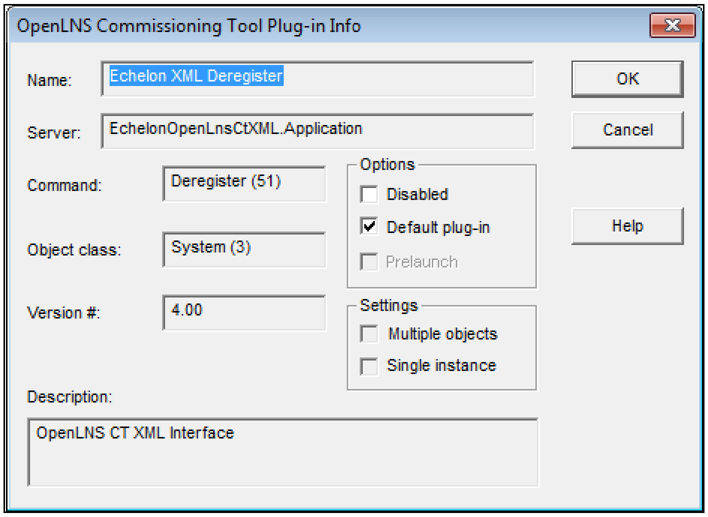
You can perform the following operations for registered plug-ins:

- View general information about the selected plug-in.
- Disable the plug-in.
- Set whether the plug-in is the default for a given command.
- Select whether the plug-in is pre-launched for commands that invoke it.
- View whether the plug-in supports multiple objects and single instance use.

To perform any of these operations, open the **OpenLNS CT Plug-in Info** dialog from the drawing page or the Visio menu.

- To open the **OpenLNS CT Plug-in Info** dialog from the drawing page, right-click a device SmartShape, functional block SmartShape, or empty space in the drawing page, point to **Plug-Ins**, and click **Plug-ins**. Click **Info**.

- To open the **OpenLNS CT Plug-In Info** from the Visio menu, click **Add-ins**, point to **Plug-Ins**, and click **Network Plug-Ins**. Click **Info**.



OpenLNS Commissioning Tool Plug-in Info

Name: Echelon XML Deregister

Server: EchelonOpenLnsCtXML.Application

Command: Deregister (51)

Object class: System (3)

Version #: 4.00

Options
☐ Disabled
☑ Default plug-in
☐ Prelaunch

Settings
☐ Multiple objects
☐ Single instance

Description:
OpenLNS CT XML Interface

This dialog provides the following information for the selected plug-in:

Name Specifies the name of the plug-in.

Server Specifies the registered ActiveX name for the plug-in.

Command Specifies the command name or number that the plug-in implements.

Object Class Specifies the class of the object on which the plug-in operates.

Version # Indicates the version of the plug-in.

Options

Disabled Disables the selected plug-in so that it cannot be opened when the command for the object class of the plug-in is selected. If the default or first available plug-in is disabled, OpenLNS CT will open the next available plug-in for the object class.

Default Plug-In Registers the plug-in as the default application for the specified command and object class. OpenLNS CT removes the default attribute, if set, from all other plug-ins registered for that command and object class. If this check box is cleared, OpenLNS CT does not change the attribute on any other plug-ins.

If multiple plug-ins are registered for the same command and object class, the plug-in with this check box selected will be the one invoked. If multiple plug-ins have this check box selected, OpenLNS CT invokes the first plug-in that it finds with this option set.

Prelaunch

Keeps one instance of the plug-in program active (if registered) so that it is immediately available for commands that invoke it.

When you select or clear this check box, OpenLNS CT will select or clear it for all OpenLNS CT networks on your computer. If you clear this option, the change is not effective until you close and then re-open the network.

If the plug-in does not support pre-launch capability, this check box is disabled.

Settings

Multiple Objects

Indicates whether the plug-in supports multiple objects. For example, the OpenLNS CT Browser supports browsing multiple functional blocks.

Single Instance

Indicates whether the plug-in supports using a single instance of the plug-in when it is requested for the same object multiple times.

Description

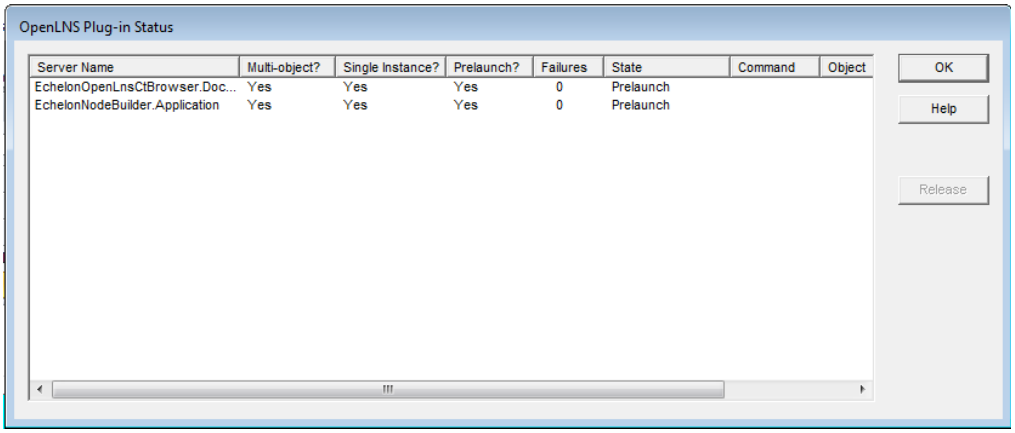
Provides a brief description of the plug-in.

Click **OK** to return to the OpenLNS CT **Plug-ins** dialog.

Viewing Plug-in Status

You can view the status of plug-ins that OpenLNS CT has started and to which it still holds a reference. OpenLNS CT holds a reference to a plug-in after starting one if the plug-in was pre-launched, and it is a single-instance, multi-object plug-in. When OpenLNS CT holds a reference to a plug-in, it will typically prevent the plug-in from being terminated unless you release the plug-in.

To access the **OpenLNS Plug-in Status** dialog, click **Add-ins**, point to **Plug-Ins**, and then click **Plug-In Status**.



Server Name	Multi-object?	Single Instance?	Prelaunch?	Failures	State	Command	Object
EchelonOpenLnsCtBrowser.Doc...	Yes	Yes	Yes	0	Prelaunch		
EchelonNodeBuilder.Application	Yes	Yes	Yes	0	Prelaunch		

The columns contain the following data. You can sort the data by clicking a column header. Click a header again to toggle the order (ascending or descending) in which the data in that column opens.

Server Name The registered ActiveX name for the plug-in.

Multi-Object Whether the plug-in supports multiple objects simultaneously.

Single Instance Whether the plug-in supports using a single instance of the plug-in when it is requested for the same object multiple times.

Pre-Launch Whether OpenLNS CT has pre-launched an instance of the plug-in.

Failures. The number of times in the current session that the plug-in has been pre-launched but has failed to launch or has terminated unexpectedly. If this number exceeds five, OpenLNS CT will stop trying to pre-launch the plug-in.

State The current state of the plug-in (pre-launch or released).

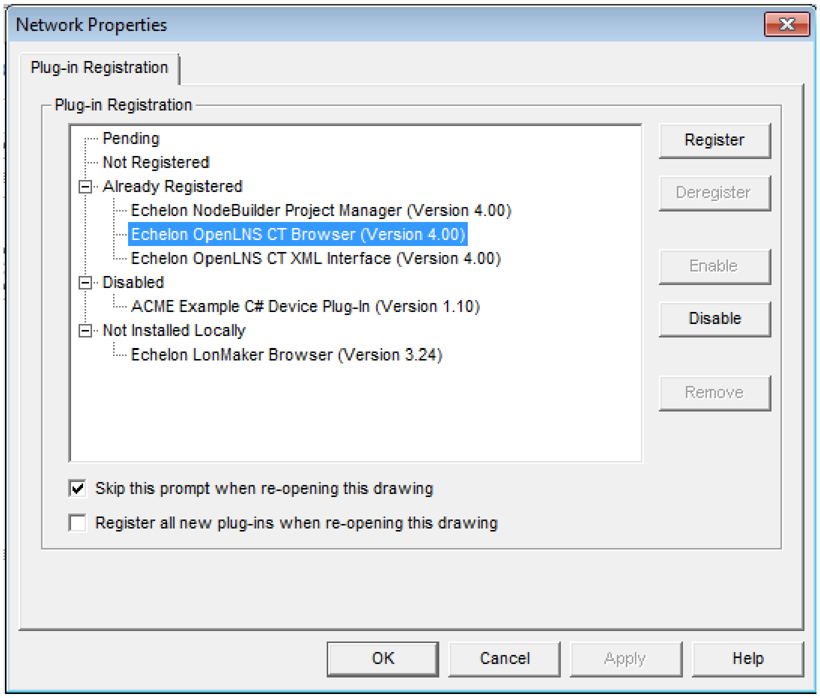
Command The command name or number that the plug-in implements.

Object The class of the object on which the plug-in operates for the command.

To terminate a plug-in to which the OpenLNS CT has a reference, click the Server Name field of the desired plug-in and click **Release**.

Re-Registering Plug-ins

You can register a plug-in that you chose not to register when opening an OpenLNS CT network, or re-register a plug-in due to an update to it. You can register or re-register a plug-in from the **Network Properties: Plug-in Registration** dialog in the OpenLNS CT drawing. To access this dialog, click **Add-ins**, point to **Plug-Ins**, and then click **Register Plug-Ins**.



- Pending
- Not Registered
- Already Registered
  - Echelon NodeBuilder Project Manager (Version 4.00)
  - Echelon OpenLNS CT Browser (Version 4.00)
  - Echelon OpenLNS CT XML Interface (Version 4.00)
- Disabled
  - ACME Example C# Device Plug-In (Version 1.10)
- Not Installed Locally
  - Echelon LonMaker Browser (Version 3.24)

☑ Skip this prompt when re-opening this drawing

☐ Register all new plug-ins when re-opening this drawing

Follow these steps to register or re-register a plug-in:

1. Choose the plug-in to register from the **Already Registered** or the **Not Registered** list. You can register all the plug-ins listed under these lists by clicking the category label.

2. Click **Register**. The plug-in or plug-ins are added to the **Pending** list (they also remain under their existing lists), and their status, "[Register]", is indicated.

3. Once you have chosen all the plug-ins to register or re-register, click **Apply**. The selected plug-ins will be added to the Already Registered category and be removed from the Pending category.

4. After the plug-ins are registered, click **OK** to return to the OpenLNS CT drawing.

Disabling and Enabling Plug-ins

You can disable or enable plug-ins when or after you open or create an OpenLNS CT network. You can disable a plug-in that you are no longer using to prevent if from being automatically registered when your create or open a network with automatic plug-in registration. If you enable or disable a plug-in, it will be enabled or disabled for all networks on your computer (remote clients are not affected).

You can disable and enable plug-ins when you open or create an OpenLNS CT network using the Plug-In Registration page that opens in the Network Wizard (unless you selected the **Skip Plug-in Options Page** in the **New Network Options** tab in the OpenLNS CT Design Manager, or previously selected the check box at the bottom of the page to have it skipped when re-opening the network).

If you skip the Plug-in Registration page in the Network Wizard, you can still disable and enable plug-ins in the **Network Properties: Plug-In Registration** dialog in the OpenLNS CT drawing. To access this dialog, click **Add-ins**, point to **Plug-ins**, and then click **Register Plug-Ins**.

Disabling a Plug-in

To disable a plug-in, follow these steps:

1. Select the plug-in under the **Not Registered** or **Already Registered** lists. You can disable all the plug-ins listed under these lists by clicking the category label.

2. Click **Disable**. The plug-in or plug-ins are added to the **Pending** list (they also remain under their existing lists), and their status, "[Disable]", is indicated.

3. Click **Apply** to disable the selected plug-in or plug-ins.

4. After the plug-ins are disabled, click **OK** to return to the OpenLNS CT drawing.

When a registered plug-in is disabled, it will not open when you select a default command that would otherwise invoke it. For example, if a plug-in is disabled and you right-click a functional block and click **Configure**, the next available plug-in, which is typically the OpenLNS CT Browser, opens instead of the functional block plug-in from the device manufacturer. You can still use a disabled plug-in by manually invoking it. To do this, right-click the object on which you want to use the plug-in or an empty space in the drawing page, point to **Plug-ins**, and then select the desired plug-in from the shortcut menu.

If you disable an unregistered plug-in, it will not be registered in new or existing networks, even if the **Register New Plug-ins** check box is selected in the **New Network Options** tab of the OpenLNS CT Design Manager or the **Register All New Plug-ins** check box is selected in the **Plug-in Registration** dialog. You can still manually register a disabled plug-in in the **Network Properties: Plug-In Registration** dialog in the OpenLNS CT drawing.

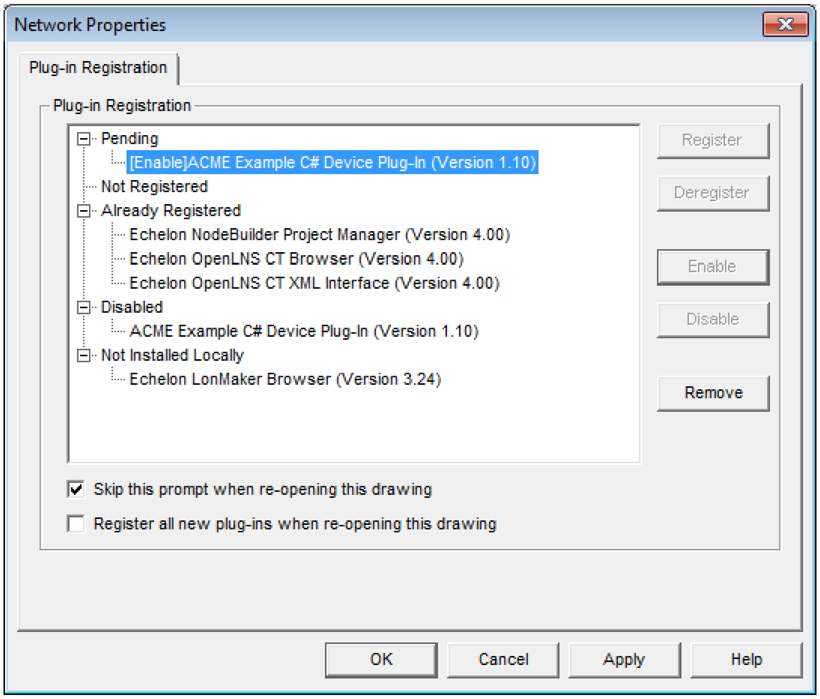
Enabling a Plug-in

To enable a plug-in, follow these steps:

1. Select the plug-in under the **Disabled** list. You can enable all of the disabled plug-ins by clicking the category label.

2. Click **Enable**. The selected plug-in or plug-ins are added to the **Pending** list (they also remain under their existing categories), and their pending status, "[Enable]", is indicated.

3. Click **Apply** to enable the selected plug-in or plug-ins.
4. After the plug-ins are enabled, click **OK** to return to the OpenLNS CT drawing.



Network Properties
Plug-in Registration
Plug-in Registration
Pending
[Enable]ACME Example C# Device Plug-In (Version 1.10)
Not Registered
Already Registered
Echelon NodeBuilder Project Manager (Version 4.00)
Echelon OpenLNS CT Browser (Version 4.00)
Echelon OpenLNS CT XML Interface (Version 4.00)
Disabled
ACME Example C# Device Plug-In (Version 1.10)
Not Installed Locally
Echelon LonMaker Browser (Version 3.24)
Register
Deregister
Enable
Disable
Remove
☑ Skip this prompt when re-opening this drawing
□ Register all new plug-ins when re-opening this drawing
OK
Cancel
Apply
Help

12

Creating and Using Custom OpenLNS CT SmartShapes and Stencils

This chapter describes OpenLNS CT stencils and the process to create them; creating and using custom master SmartShapes for devices, functional blocks, subsystems, and connections; using additional user cells for devices, functional blocks, and routers; and setting user-defined functional block modes and types.

OpenLNS CT Stencils

OpenLNS CT uses Visio *stencils* to speed your LONWORKS network engineering. A stencil is a collection of *master SmartShapes* (also called *masters*) that you can easily reuse in your drawings. To add a master SmartShape to your drawing, you drag the master SmartShape from the stencil to your drawing. OpenLNS CT includes LONWORKS specific master SmartShapes called *OpenLNS CT SmartShapes*. For more information on Visio stencils and master SmartShapes, see your Visio documentation. Stencils allow you to combine multiple SmartShapes into larger master SmartShapes. If you include OpenLNS CT SmartShapes in a master SmartShape, OpenLNS CT saves the configuration of these SmartShapes with the master SmartShape, including information such as device template name and configuration property settings. This allows you to create master SmartShapes that represent, for example, all of the functional blocks in a device, or all of the devices in a subsystem. You can also create a master SmartShape that is based on a single SmartShape.

You can create custom OpenLNS CT stencils to further customize OpenLNS CT. See *Creating a Custom OpenLNS CT Stencil* for more information. You can save custom stencils you create for a specific network in the drawing folder of that network. If you create a custom stencil to use across many networks, you can save it to a global location such as the LonWorks\LonMaker\Visio directory (**C:\Program Files\LonWorks\LonMaker\Visio** [32-bit systems] or **C:\Program Files (x86)\LonWorks\LonMaker\Visio** [64-bit systems] by default).

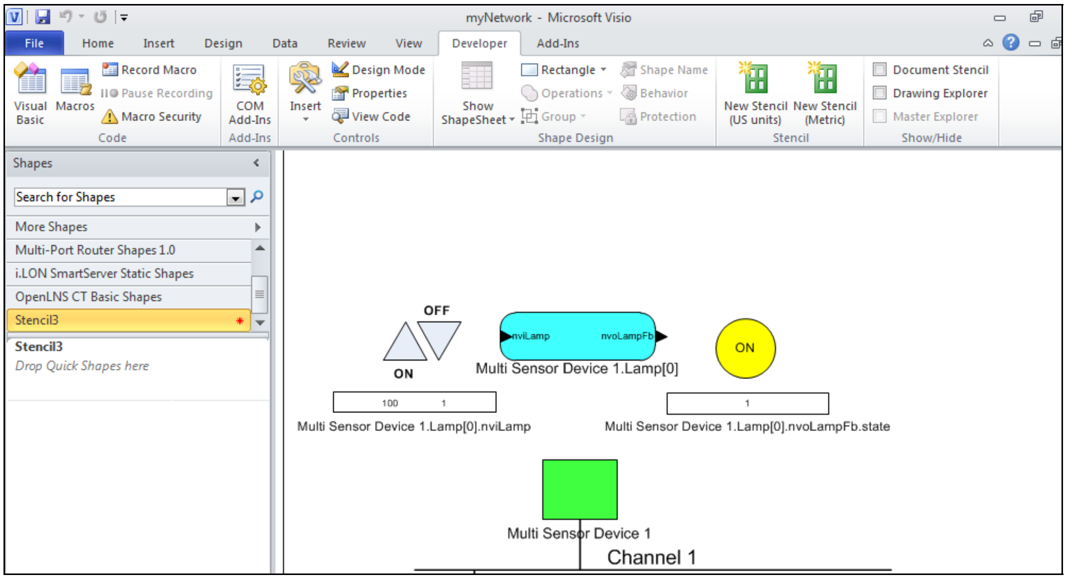
You can create custom OpenLNS CT stencils to further customize OpenLNS CT. See *Creating a Custom OpenLNS CT Stencil* for more information.

Creating a Custom OpenLNS CT Stencil

To create a new custom OpenLNS CT stencil, follow these steps:

1. Enable the Visio **Developer** tab. To do this, click **File** and then click **Options**. In the **Visio Options** dialog, click the **Customize Ribbon** section. On the right-hand side, enable the **Developer** check box, and then click **OK**.

2. Click **New Stencil**. A blank stencil opens with the default name of **Stencil**.



3. Create new OpenLNS CT master SmartShapes as described in *Creating Custom OpenLNS CT Master SmartShapes.*

4. Right-click the title bar of the new stencil and then click **Save** on the shortcut menu. Select a folder, assign a name for the stencil file, and click **Save**. Save the stencil to the

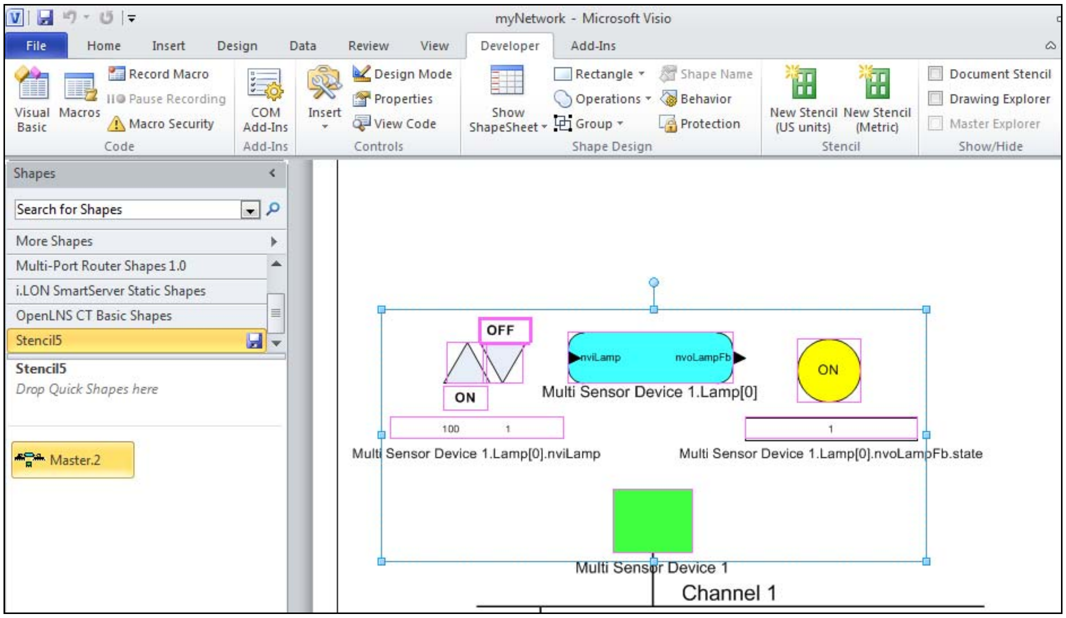
LonWorks\LonMaker\Visio directory (by default, **C:\Program Files\LonWorks\LonMaker\Visio** [32-bit system] or **C:\Program Files (x86)\LonWorks\LonMaker\Visio** [64-bit system]) to easily access it with the open stencil (📂) button, or save the stencil to the drawing folder for simplified backup.

Creating Custom OpenLNS CT Master SmartShapes

You can create a custom OpenLNS CT master SmartShape to reuse any part of a design within the same drawing or in many OpenLNS CT drawings. To create a master SmartShape, you must first open a stencil as an original file, with read/write access. Normally, stencils open with read-only access. To work with a read/write stencil while maintaining the original, open a copy of the original.

To open an original stencil, follow these steps:

1. On the **Shapes** window, point to **More Shapes**, and then click **Open Stencil**. Open the folder that contains the stencil, and then select the stencil.
2. Right-click the stencil title bar, and then click **Edit Stencil** on the shortcut menu. The stencil window opens, with a red asterisk in the title bar indicating that you can modify the stencil. Once you have a stencil open for editing, you can create new master SmartShapes in it.
3. Select the object or objects to be included in the master SmartShape. You can create a master SmartShape that contains routers, channels, devices, functional blocks, connections, subsystems, and Data Point SmartShapes. To select multiple objects you can either drag a selection net around the objects or hold down CTRL and click the objects.
4. Drag the selected object or objects to the stencil you opened. A new master SmartShape representing the selected object or objects opens in the stencil.



5. Save the stencil by clicking the **Save** button. You can then rename, modify, or edit the icon of the master SmartShape.

To copy any number of SmartShapes between OpenLNS CT networks, use a custom OpenLNS CT stencil with custom OpenLNS CT SmartShapes as described in this section. Do not hold down CTRL and drag or use copy/paste between drawings from different networks.

Using Custom OpenLNS CT Master SmartShapes

You can reuse a custom OpenLNS CT SmartShape that you create (see *Creating Custom OpenLNS CT Master SmartShapes*). The custom OpenLNS CT master SmartShape may contain any number of the standard OpenLNS CT SmartShapes described in *Designing OpenLNS CT Networks*. The following sections describe how OpenLNS CT processes each of the standard OpenLNS CT SmartShapes when you include them in a custom OpenLNS CT SmartShape.

If you drag a master SmartShape containing multiple SmartShapes and the creation of any of the SmartShapes fails, that SmartShape will not be added, and all subsequent SmartShapes in the master SmartShape will not be processed.

Device Master SmartShapes

You can include one or more OpenLNS CT Device SmartShapes in a custom master SmartShape. Each of these is called a *device master SmartShape*. Each device master SmartShape includes the device template name, device interface (XIF) file name, and the configuration property settings from the original device. The Neuron ID is not included so that you can reuse the master SmartShape with multiple devices.

When you drag a master SmartShape that includes any device master SmartShapes to a drawing and automatic channel selection is enabled in the **Network Properties: Device** options category, the master SmartShape is added to the drawing without any wizards opening. If automatic channel selection is disabled, the New Device Wizard opens and you select the channel for the new device. You can select the same channel as the original device, another existing channel, or create a new channel. If the custom master SmartShape includes multiple devices on the same channel, the Device Wizard opens for the first device only on each channel. OpenLNS CT places all subsequent devices that are on the same channel in the master SmartShape on the channel that you select for the first device on the channel, and have their properties set as specified in the master SmartShape. To manually set the channels and properties for each device, select the **Force Wizard Prompts** check box in the **Network Properties: General** options category as described in Appendix A.

When you drag a master SmartShape that includes a device SmartShape to a drawing, OpenLNS CT does the following for each device master SmartShape:

1. If the master SmartShape includes a program ID in a ProgramID user cell (see *Additional Device User Cells*), OpenLNS CT looks up the device template using the program ID. If the master SmartShape does not include a program ID but includes a device template name in a **TmplName** user cell (see *Additional Device User Cells*), it looks up the device template based on the specified device template name.
2. If OpenLNS CT does not find a device template in step 1, it looks for the device interface (XIF) file specified in the **XifName** user cell. If the device interface file is found, OpenLNS CT looks up the device template by the program ID specified in the device interface file. If a device template with a matching program ID is found, the user is given the option of changing the name of the device template to the name specified by the device SmartShape, or using the existing device template name. In either case, the existing device template is associated with the device SmartShape.
3. If OpenLNS CT does not find a device template in steps 1 and 2, but it does find the device interface file, it creates a new device template by importing the device interface file. OpenLNS CT gives the new device template the name specified in the device SmartShape, and associates it with the device SmartShape.
4. If OpenLNS CT does not find a device template or device interface file in steps 1 through 3, it prompts you to select an existing device template or to define a new device template by importing a device interface file. The resulting device template is associated with the device SmartShape.

A functional block SmartShape may include a device template name and/or program ID. The device template name assigned by the above procedure must match the name in any functional block

SmartShapes that will be used with the newly created device SmartShape. See *Additional Functional Block User Cells* for more information.

Functional Block Master SmartShapes

You can include one or more OpenLNS CT Functional Block SmartShapes in a custom master SmartShape. Each of these is called a *functional block master SmartShape*. Each functional block master SmartShape includes the device template and/or program ID as well as the network variable and configuration property types, formats, and values associated with the original functional block. To eliminate configuration property values, clear **Copy Values on Functional Block Copy**. See *Configuration Property Properties* for more information.

By default, custom functional block master SmartShapes always apply to the device type for which they were created. For example, if you create a master functional block SmartShape to be used with a specific manufacturer's analog output device, that SmartShape can only be used with those analog output devices. To create a functional block SmartShape that can be used with multiple device types, edit the SmartShape as described in *Editing Master SmartShape User-Defined Cells*.

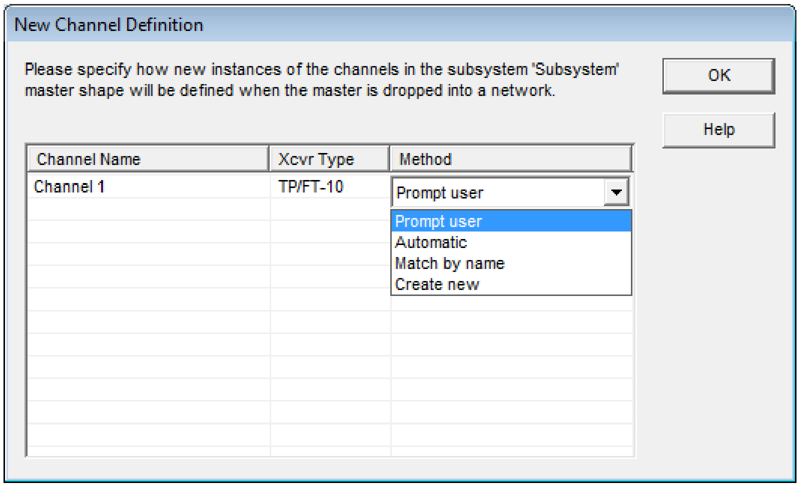
When you drag a custom master SmartShape that includes any functional block SmartShapes to a drawing, the following conditions apply:

- If a custom master SmartShape contains functional blocks, but not their associated devices, you must create the devices to be associated with the functional blocks before dragging the custom master SmartShape to the drawing.
- If you drop a custom master SmartShape containing a functional block and its associated device, the functional block wizard does not appear because it knows its associated device.
- If a custom master SmartShape contains several functional blocks associated with the same device, but not the device itself, the functional block wizard opens for the first functional block only. You must associate the functional block with its device. The remaining functional blocks are assigned to the same device, unless no more functional blocks of the correct type are available on the device. To be prompted for the device to be assigned to each functional block, set **Force Wizard Prompts** in the **Network Properties: General** options category.

Subsystem or Supernode Master SmartShapes

You can include one or more OpenLNS CT subsystem or supernode SmartShapes in a custom master SmartShape. When you create a custom OpenLNS CT SmartShape from a subsystem or supernode SmartShape, OpenLNS CT copies the information for all the devices, functional blocks, connections, routers, channels, and nested subsystems and supernodes to the master.

If you create a custom subsystem or supernode master SmartShape that contains one or more channels the **New Channel Definition** dialog box opens.



Please specify how new instances of the channels in the subsystem 'Subsystem' master shape will be defined when the master is dropped into a network.

Channel Name	Xcvr Type	Method
Channel 1	TP/FT-10	Prompt user

For each channel listed, click anywhere in the **Method** field and then select how the channel is to be defined when the subsystem or supernode master SmartShape is added to your OpenLNS CT drawing. You have four choices:

- **Prompt User**. The **Channel Definition** dialog box will appear, enabling you to select whether the channel represents a new or an existing channel. This is the default.
- **Automatic**. OpenLNS CT will define the channel based on where you drop the subsystem SmartShape in the OpenLNS CT drawing. If you drop the subsystem master SmartShape on top of an existing channel, the channel will be created as an instance of the existing channel. If you do not drop the subsystem master SmartShape on top of an existing channel, the **Channel Definition** dialog box will appear and you select whether the channel represents a new or an existing channel.

  Only one channel can have this option selected.
- **Match by Name**. OpenLNS CT will attempt to match the channel with an existing channel in the network that has the same name. If it finds a match, the channel will be created as an instance of that existing channel. If OpenLNS CT cannot find a match, the channel will be created as a new channel.
- **Create New**. The channel is created as a new channel. The channel name will be modified if there is a naming conflict (if the channel name ends with a number, the channel number will be incremented; if the channel name does not end with a number, an index, starting at 1, will be appended to the channel name).

Notes:

- When creating a subsystem or supernode master SmartShape, OpenLNS CT requires that the stencil has a name. If you just created a new blank stencil, save it before creating the master SmartShape.
- When creating a subsystem or supernode master SmartShape, OpenLNS CT creates one or more hidden master SmartShapes that are placed in the same stencil as the original master SmartShape. Therefore, you cannot copy subsystem or supernode master SmartShapes from one stencil to another.

Connection Master SmartShapes

You can include one or more OpenLNS CT Connector SmartShapes in a custom master SmartShape. Each of these is called a *connection master SmartShape*. Each connection block master SmartShape includes the connection attributes for the original connection. One connection master SmartShape can represent several connections in your OpenLNS CT drawing, allowing you to quickly and easily establish multiple network connections with one operation.

You can use connection master SmartShapes to do the following:

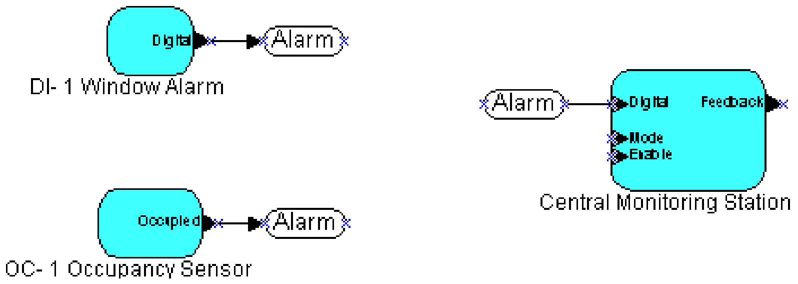
- Specify a connection with a connection description other than the default connection description. A connection description is a set of properties that apply to a connection, such as protocol service, retry count, priority, and authentication.
- Establish multiple connections in a single operation. For example, instead of dropping several individual Connector SmartShapes to establish multiple connections (digital output to digital input, analog input to analog output, and so on), you can create a connection master SmartShape to represent all these connections. The connection is displayed on the OpenLNS CT drawing as a connection between the functional blocks on which the connections exist.
- Append a connection to an existing reference connection. For example, if you have a reference connection between an input and output network variable representing time, you can create a connection master SmartShape to quickly create connections to additional output network variables representing time.

If you include one or more OpenLNS CT Connector SmartShapes in a custom OpenLNS CT master SmartShape, the connections associated with each connector will be saved in the master SmartShape if either of the following conditions is met:

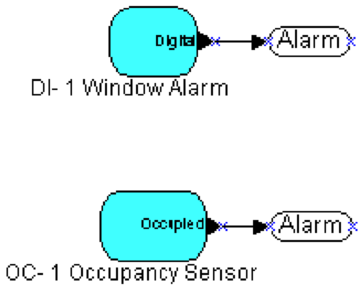
- The functional blocks on both sides of the connection are included.
- The master SmartShape contains the connection, reference SmartShape (if any), and the functional block SmartShape.

You can use reference connections in master SmartShapes to create fan-in or fan-out connections. If all of the members in the reference connection are included in a master SmartShape, OpenLNS CT will create a new connection (with a unique reference name) when you drag the master SmartShape to a drawing.

If you create a master SmartShape with only a subset of a referenced connection, (for example, Functional Block, Connector, and Reference SmartShapes), when you use the master SmartShape the new functional block is added to the same connection as the other reference SmartShapes. For example, assume you have a reference connection as shown in the following figure:



To create a master SmartShape that allows you to fan-in additional inputs to the Central Monitoring Station functional block, copy the following SmartShapes to a new master SmartShape:



If you drag this master SmartShape to any subsystem in the same network, OpenLNS CT creates two new functional blocks, and adds their outputs to the same Alarm connection for the Central Monitoring Station. If you also include the Central Monitoring Station in the master SmartShape, OpenLNS CT creates three new functional blocks and a new reference connection when you drag the master SmartShape to a drawing.

You can also create a master SmartShape containing a connection not attached to any functional blocks. The information in the connection description is saved and becomes part of the master SmartShape. If you create a master SmartShape of a connector and a reference SmartShape (without any associated functional blocks) you can use the master SmartShape to create reference connections without opening the connection dialog.

You can use a master SmartShape that contains reference connections only on the network on which you created it. This is an exception to the general rule that you can use master SmartShapes on more than one network.

Tip: To speed up making connections, put OpenLNS CT OffNet before using a master SmartShape with many connections in it. Updates to devices involved in the connections will be done when you go OnNet.

Creating Additional Channels

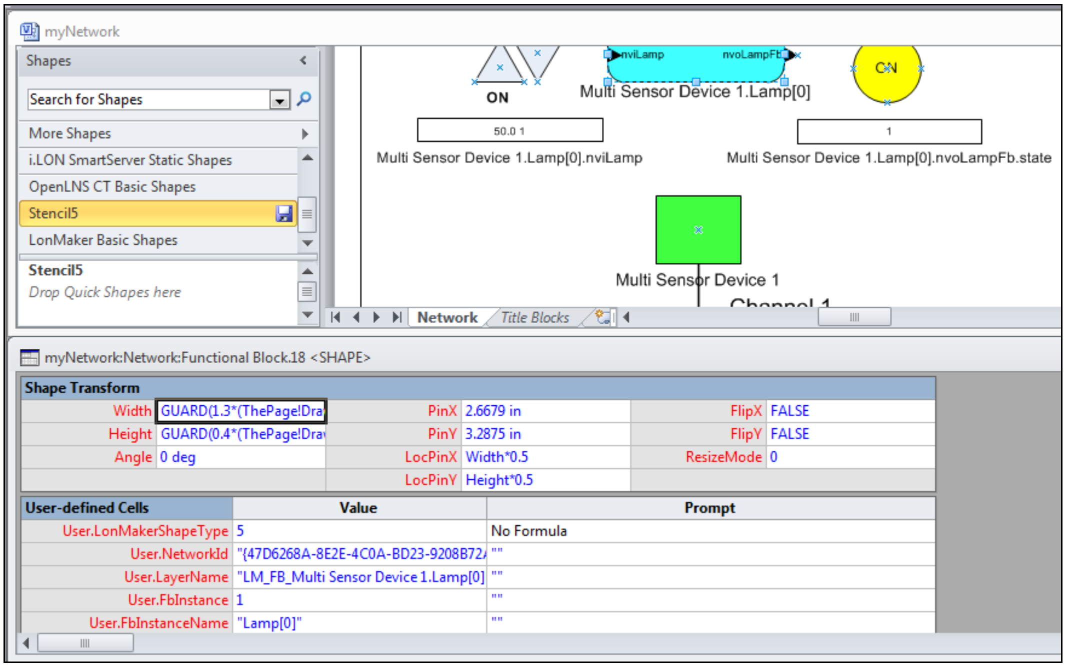
If you drag a master SmartShape that includes a channel SmartShape and device SmartShapes to your drawing, the master SmartShape may limit the channel types that can be used. In that case, only the valid channel names and channel types will be listed.

Editing Master SmartShape User Defined Cells

You can edit user-defined cells in master OpenLNS CT SmartShapes to access advanced OpenLNS CT features such as supporting multiple device types with a single SmartShape. This is an advanced feature for advanced OpenLNS CT and Visio users. You can change the properties of a master SmartShape by changing its user-defined cells. For example, you can extend the list of device types to which a functional block can be applied, as described in *Functional Block Master SmartShapes*. User defined cells are contained within a page called the ShapeSheet.

To view a SmartShape’s ShapeSheet, follow these steps:

1. Enable the Visio **Developer** tab. To do this, click **File** and then click **Options**. In the **Visio Options** dialog, click the **Customize Ribbon** section. On the right-hand side, enable the **Developer** check box, and then click **OK**.
2. Click the target OpenLNS CT SmartShape.
3. Click the **Developer** tab, point to **Show ShapeSheet**, and then click **Shape** (alternatively, you can right-click the shape, and click **Show ShapeSheet** on the shortcut menu). The ShapeSheet for the selected OpenLNS CT SmartShape opens.

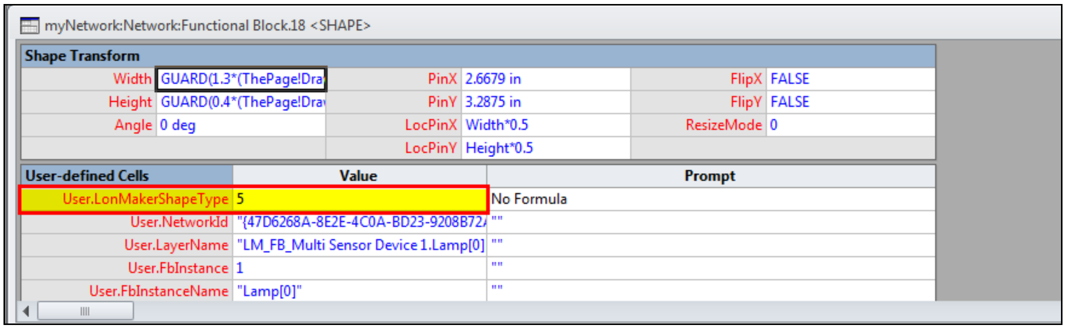


myNetwork:Network:Functional Block.18 <SHAPE>

Shape Transform					
Width	GUARD(1.3*(ThePage!Dra	PinX	2.6679 in	FlipX	FALSE
Height	GUARD(0.4*(ThePage!Dra	PinY	3.2875 in	FlipY	FALSE
Angle	0 deg	LocPinX	Width*0.5	ResizeMode	0
		LocPinY	Height*0.5		

User-defined Cells	Value	Prompt
User.LonMakerShapeType	5	No Formula
User.NetworkId	"{47D6268A-8E2E-4C0A-BD23-9208B72	""
User.LayerName	"LM_FB_Multi Sensor Device 1.Lamp[0]	""
User.FbInstance	1	""
User.FbInstanceName	"Lamp[0]"	""

4. Every OpenLNS CT SmartShape has an OpenLNS CT **ShapeType** user cell. The value of this cell specifies the OpenLNS CT SmartShape type. You are most likely to use the DEVICE (LonMakerShapeType 4), FUNCTIONAL_BLOCK (type 5), and FB_NAME (type 13) SmartShape types.



myNetwork:Network:Functional Block.18 <SHAPE>

Shape Transform					
Width	GUARD(1.3*(ThePage!Dra	PinX	2.6679 in	FlipX	FALSE
Height	GUARD(0.4*(ThePage!Dra	PinY	3.2875 in	FlipY	FALSE
Angle	0 deg	LocPinX	Width*0.5	ResizeMode	0
		LocPinY	Height*0.5		

User-defined Cells	Value	Prompt
User.LonMakerShapeType	5	No Formula
User.NetworkId	"{47D6268A-8E2E-4C0A-BD23-9208B72	""
User.LayerName	"LM_FB_Multi Sensor Device 1.Lamp[0]	""
User.FbInstance	1	""
User.FbInstanceName	"Lamp[0]"	""

5. You can use the **FB_NAME** value to include a functional block within a group that may include other functional block SmartShapes. Set the **ShapeType** value to 13 for SmartShapes in the group with text that reflects the functional block name.

   You can create custom functional block master SmartShapes with network variables by creating a functional block instance in OpenLNS CT, adding the appropriate network variables, and then dragging the SmartShape to a stencil.

Additional Device User Cells

You can include the following additional user cells in a device SmartShape. Not every device has all of these cells, but you can add any of them by right-clicking an existing user-defined cell and selecting **Insert Row**.

Auth	A value of **1** indicates that authentication will be enabled for devices created from this master SmartShape.
AutoLoad	Indicates whether an application will automatically be loaded into the device when the device is commissioned or replaced.

- A value of **0** indicates no.
- A value of **1** indicates yes.
- A value of **2** indicates that the OpenLNS CT user will be prompted.

The default value is **2** to prompt the user.

ChanTypeList

A semicolon-separated list of valid channel type IDs for the device. A value of **-1** in the list indicates that the user has the option of selecting from all channels in addition to the channels of the specific types defined. Channel type IDs correspond to the **ChannelTypeNumber** values contained in the **StdXcvr.xm**l file available from www.lonmark.org/spid. By default, all channel types are allowed.

EditAdvPropCap

Specifies handling of the advanced properties tab in the device properties dialog.

- A value of **0** specifies that the advanced properties dialog is not to be displayed.
- A value of **1** indicates that the advanced properties dialog is to be displayed, but cannot be modified.
- A value of **2** indicates that the advanced properties dialog is displayed and changes can be made.

The default value is **2** (the advanced properties dialog is displayed and changes are allowed).

EditPropCap

Specifies handling of the basic properties tab in the device properties dialog.

- A value of **0** specifies that the basic properties dialog is not to be displayed.
- A value of **1** indicates that the basic properties dialog is to be displayed, but cannot be modified.
- A value of **2** indicates that the basic properties dialog is displayed and changes can be made.

The default value is **2** (the basic properties dialog is displayed and changes are allowed).

FbModes

Specifies a semicolon-separated list of the scopes for each of the functional blocks in this device. This cell's data must be ASCII text in the format <*scopeSelector0*>**;**<*scopeSelector1*>**;**... where *scopeSelector0* applies to the functional block with index 0 (the Node Object), *scopeSelector2* applies to the functional block with index 1, and so on. You can see what functional block in a device has what index by right-clicking the device, selecting **Properties** from the shortcut menu, and clicking the Functional Blocks tab of the device properties dialog.

If this cell is not specified, OpenLNS CT uses scope **0** for functional profile numbers 1–19999, and scope **3** for all other functional profiles. This means that functional block SmartShapes based on user functional profiles with numbers of 20000 and above, as well as all inherited profiles, must either set this value or must include a plug-in that sets the appropriate value.

The scope selector definitions are:

0 Standard functional profile defined in the standard resource file set.

3 User-defined functional profile, defined in a manufacturer-specific resource file set.

4 User-defined functional profile, defined in a manufacturer and device class specific resource file set.

5 User-defined functional profile, defined in a manufacturer and device class/subclass specific resource file set.

6 User-defined functional profile, defined in manufacturer, and device class/subclass/model number specific resource file set.

See *Setting User Functional Block Scopes and Types* for more information.

InitOnline

Specifies the initial state of the device application when it is first commissioned.

- A value of **0** indicates that the device is to be placed offline.
- A value of **1** indicates the device is to be placed online.
- A value of **2** specifies that the device is to be disabled (only applies to devices with a Node Object functional block, otherwise, the devices are placed offline).
- A value of **3** indicates that the initial state of the device will based on the default value set in the **Network Properties: Device** options category.

The default value is **3** (the network properties value is used).

InstallMethod

Specifies the default device installation method (service pin or manual entry).

- A value of **0** specifies service pin installation method.
- A value of **1** specifies manual installation method.

LmwStyle0

Specifies the coloring style applied to a device SmartShape when it is in the Updates Pending state. The default style is **Node Uninstalled**.

LmwStyle1

Specifies the coloring style applied to a device SmartShape when it is up-to-date. The default style is **Node Normal**.

LmwStyle3

Specifies the coloring style applied to a device SmartShape when it is in the Update Failed state. The default style is **Node Error**.

LmwStyle7

Specifies the coloring style applied to a device SmartShape when OpenLNS CT detects that it is detached from the network.

LmwStyle8

Specifies the coloring style applied to a device SmartShape when OpenLNS CT detects it to be in an unexpected state. For further information about Visio styles, see the Visio documentation.

Locn

A 12-digit hex string that specifies the default value of the device's location string. By default, the location string is initialized to all zeroes unless you enable the **Set Location Property to Subsystem ID** check box in the **Network Properties: Recovery** options category.

NewAdvPropCap

Specifies handling of the advanced properties dialog in the new device wizard.

- A value of **0** specifies that the advanced properties dialog is not to be displayed.
- A value of **1** indicates that the advanced properties dialog is to be displayed, but cannot be modified.
- A value of **2** indicates that the advanced properties dialog is displayed and changes can be made.
- The default value is **0** (the advanced properties dialog is not displayed).

NewPropCap

Specifies handling of the basic properties dialog in the new device wizard.

- A value of **0** specifies that the basic properties dialog is not to be displayed.
- A value of **1** indicates that the basic properties dialog is to be displayed, but cannot be modified.
- A value of **2** indicates that the basic properties dialog is displayed and changes can be made.
- The default value is **2** (the basic properties dialog is displayed and changes are allowed).

NonGrpRcvTimer

Specifies the default non-group receive timer; a value of **-1** indicates that the OpenLNS Server default value will be used. Otherwise, the value is encoded as follows:

Encoded Value	**Seconds**
0	0.016
1	0.024
2	0.032
3	0.048
4	0.064
5	0.096
6	0.128
7	0.192
8	0.256
9	0.384
10	0.512
11	0.768
12	1.024
13	1.536
14	2.048
15	3.072

NxeName

Specifies the file name or full path of the application file. If it is not specified here, the OpenLNS CT user will be prompted for an application file when a request is made to load the application image.

ObjStateStyle

Specifies whether the coloring style is applied to the SmartShape's fill, line, text, or a combination of the three.

- A value of **1** indicates that the style applies to the SmartShape’s lines.
- A value of **2** indicates that the value applies to the SmartShape’s fill.
- A value of **4** indicates that the style applies to the SmartShape’s text.

You can add values together to specify combinations. For example, a value of **6** indicates that the style applies to the SmartShape’s fill and text, but not its lines. By default, the style applies to all three SmartShape attributes.

PingClass

Specifies the ping interval.

- A value of **0** specifies that the default ping interval is used for this SmartShape.
- A value of **1** specifies that the 1-minute ping interval be used.
- A value of **2** specifies that the 2-minute ping interval be used.
- A value of **3** specifies that the 15-minute ping interval be used.

A value of **4** specifies that the Never ping interval be used.

Priority

A value of **1** indicates that priority will be enabled for devices created from this master SmartShape.

ProgramId

Device program ID. OpenLNS CT uses this value first to find the device template by name. If not specified, the TmplName value is used.

RequiresVFB

Specifies whether the device requires a virtual functional block.

- A value of **0** specifies that the device does not require a virtual functional block.
- A value of **1** specifies that the device requires virtual functional blocks.
- A value of **-1** specifies that OpenLNS CT should determine this value based on the device’s set of network variables, configuration properties, and message tags.

The default value is **-1**.

TmplName

Device template name. This must be specified if **XifName** is specified. Be sure to specify a unique device template name for each unique device interface (XIF) file.

- If OpenLNS CT does not first find a device template based on the **ProgramId** value, it uses the value in **TmplName** to look up the device template by name. If neither is found, OpenLNS CT searches for a device interface file based on the **XifName** value and the XIF search path. If found, the XIF file is loaded and used. If it is not found, the OpenLNS CT user is prompted to enter a device interface file when the device master SmartShape is dragged to the OpenLNS CT drawing.

XifName

File name or full path of the device interface (XIF) file. If it is not specified here, the OpenLNS CT user may be prompted for a device interface when the device master SmartShape is dragged to the to the OpenLNS CT drawing, depending on the **ProgramId** and **TmplName** values.

Additional Functional Block User Cells

You can include the following additional user cells in a functional block SmartShape. Not every functional block has all of these cells, but you can add any of them by right-clicking an existing user-defined cell and selecting **Insert Row**.

DevProgIDList Specifies a semicolon-separated list of program IDs that are valid for this functional block. If both **DevProgIDList** and **DevTypeList** values are specified, the **DevTypeList** value takes precedence.

DevTypeList Specifies a semicolon-separated list of device types (device template names) that are valid for this functional block. If both **DevProgIDList** and **DevTypeList** values are specified, the **DevTypeList** value takes precedence.

FbInstanceList Specifies the functional blocks on each device type that are valid for this functional block SmartShape. This cell can be used to further restrict which functional blocks are valid beyond what is possible with the **FbTypeList** user cell. The syntax is *<i1>,<i2>.<in>;<j1>,<j2>,<jn>*;..., where the sequence of the semicolon-separated lists corresponds to the device types specified in either the **ProgIdList** or **DevTypeList** fields (the **ProgIdList** field has precedence when both are defined). The comma-separated list indicates the indices of the valid functional blocks for that device type.

FbTypeName Specifies the functional profile name for this functional block, e.g. “Open loop sensor object.”

FbTypeList Specifies a semicolon-separated list of functional profiles that are valid for this functional block SmartShape. This cell’s data must be ASCII text in the following format: *<scopeSelector>***:***<profileNumber>***;***<scopeSelector>***:***<profileNumber>***;**....

The scope selector definitions are as follows:

0 Standard functional profile defined in the standard resource file set.

3 User-defined functional profile, defined in a manufacturer-specific resource file set.

4 User-defined functional profile, defined in a manufacturer and device class specific resource file set.

5 User-defined functional profile, defined in a manufacturer and device class/subclass specific resource file set.

6 User-defined functional profile, defined in manufacturer, and device class/subclass/model number specific resource file set.

See *Setting User Functional Block Scopes and Types* for more information.

Additional Router SmartShape User Cells

You can include the following additional user cells in a router SmartShape. Not every router has all of these cells, but you can add them by right-clicking an existing user-defined cell and selecting **Insert Row**.

Class Specifies the type of router. Valid values are as follows:

0	Configured router
1	Learning router
2	Repeater
3	Bridge
4	Permanent Repeater
5	Permanent Bridge

The default and recommended value is **0** for a configured router.

The router SmartShape supports the following fields defined for device SmartShapes: **Auth**, **Priority**, **PingInterval**, **InstallMethod**, **BitmapFile**, **ChanTypeList**, **Locn**, **NewPropCap**, **NewAdvPropCap**, **EditPropCap**, **EditAdvPropCap**, and **InitOnline**. In addition, the router SmartShape has the following fields you can use to specify defaults for Side B of the router.

PriorityB A value of **1** indicates that priority will be enabled on the B-side for SmartShapes created from this master SmartShape.

ChanTypeListB A semicolon-separated list of valid channel type IDs for the device as described in the **ChanTypeList** device user cell.

Setting User Functional Block Scopes and Types

All functional blocks have scope selectors, which specify the context in which the network variables and configuration properties are interpreted. The scope selector provides a reference to a resource file set that defines data types and display formats. Scope selector **0** is a reference to the LONMARK standard resource file set. If you are developing devices with user-defined functional profiles, network variables, or configuration properties, OpenLNS CT automatically sets the scope of the functional blocks as long as it can find an appropriate scope 3, 4, 5, or 6 resource file set. If this fails for any reason, you can set new scope selectors for the functional blocks that failed.

You can change a scope selector by creating a custom device master SmartShape or by creating a device plug-in that sets your scope selectors. To set the scope selector using a custom device master SmartShape, create a master SmartShape for your device, and then create and set an **FbModes** user cell that sets the scope for each functional block as described in *Additional Device User Cells*.

Adding a Bitmap to a Device Master SmartShape

You can change the appearance of a device master SmartShape by adding any bitmap to the SmartShape and then optionally changing its name. When you add a bitmap, you can leave part of the original form of the SmartShape open so that an OpenLNS CT user can see the fill style that OpenLNS CT uses to represent the device state.

To add a bitmap to a master device SmartShape, follow these steps:

1. Open a new (non-OpenLNS CT) drawing in Visio.
2. Open the **OpenLNS CT Basic Shapes** stencil. To do this, on the **Shapes** window, point to **More Shapes**, and then click **LONBASIC**.
3. Drag the generic Device SmartShape (▦) to the drawing.
4. Open the **Edit** menu and then click **Open Device**. The Group window opens for the device.

5. Open the **Insert** menu, point to **Picture,** and then click **From File**. Select a bitmap file. Position the bitmap on the device as desired. Close the Group window.
6. Click the bitmap, open the **Format** menu, and then click **Behavior**. Clear the three check boxes under the **Selection Highlighting** box.
7. Open the **Windows** menu and then click **Show ShapeSheet**. The ShapeSheet opens. In the **Protection** section, enter **1** in the **LockTextEdit** cell. Close the ShapeSheet.
8. Drag the SmartShape to a stencil to which you have write access. To change the name of the SmartShape, right-click it, point to **Edit Master**, and then click **Master Properties** on the shortcut menu. Modify the name and then click **OK**.
9. Save the stencil.

Viewing and Editing VBA Code Associated with an OpenLNS CT Network Drawing

Each OpenLNS CT drawing has VBA code associated with it that implements its OpenLNS CT functionality. Do not edit this code, as this can cause loss of OpenLNS CT functionality. However, you can add your own VBA code to the drawing (to create HMIs with the Data Point SmartShape, for example). To avoid unintentional changes to the OpenLNS CT VBA code and protect the OpenLNS CT digital signature, the VBA code in the OpenLNS CT templates and drawings is password protected. See *Working with Digital Signatures* in Chapter 3, *Getting Started,* for information on the consequences on modifying VBA code.

To view and edit the VBA code associated with an OpenLNS CT drawing, follow these steps:

1. Enable the Visio **Developer** tab. To do this, click **File** and then click **Options**. In the **Visio Options** dialog, click the **Customize Ribbon** section. On the right-hand side, enable the **Developer** check box, and then click **OK**.
2. Click the **Developer** tab, and then click **Visual Basic**. The Visual Basic development environment opens. You must have Visual Basic installed on your computer to view and edit VBA code.
3. On the left pane of the Visual Basic window, open the network project. You will be prompted for a password.
4. Enter **enumerate** in the **Password** box.
5. View and edit the Visual Basic code associated with the OpenLNS CT drawing.

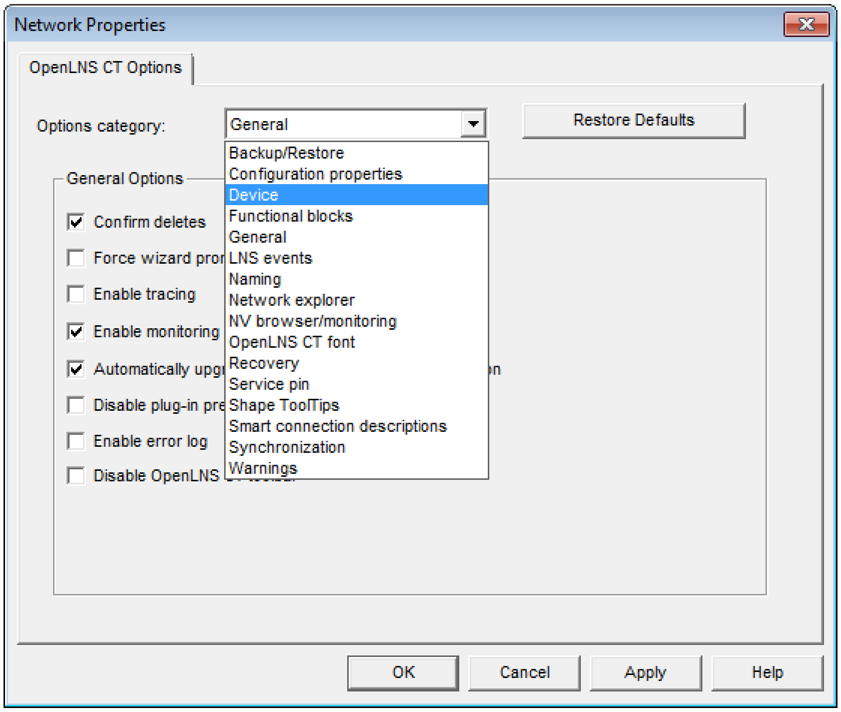
Appendix A
Setting OpenLNS CT Default Options

This appendix documents the OpenLNS CT Default Options you can set either in the OpenLNS CT Design Manager or in your OpenLNS CT drawing.

Setting OpenLNS CT Default Options

You can set OpenLNS CT Default Options either in the OpenLNS CT Design Manager or in your OpenLNS CT drawing. For new networks, you can set the default options in the **Default OpenLNS CT Options** tab in the **OpenLNS CT Design Manger**. For an existing network, you can set them in the OpenLNS CT drawing by clicking **Add-ins**, pointing to **OpenLNS CT**, and then clicking **OpenLNS CT Options**. The dialog that opens lists options that you can set for the following options categories:

- Backup/Restore.
- Configuration Properties.
- Devices.
- Functional Blocks.
- General.
- OpenLNS Events.
- OpenLNS CT Font.
- Naming.
- Network Explorer.
- NV Browser/Monitoring.
- Recovery.
- Service Pin.
- SmartShape ToolTip.
- Smart Connection Descriptions (this options category is only available in the **Network Properties: OpenLNS CT options** dialog in the OpenLNS CT drawing; see *Using Automatic Connection Description Selection* for the options you can set).
- Synchronization.
- Warning.



Options category: General

General Options

☑ Confirm deletes

□ Force wizard pror

□ Enable tracing

☑ Enable monitoring

☑ Automatically upg

□ Disable plug-in pre

□ Enable error log

□ Disable OpenLNS

You can click **Restore Defaults** to reset all the options and values in the currently displayed options category to their defaults.

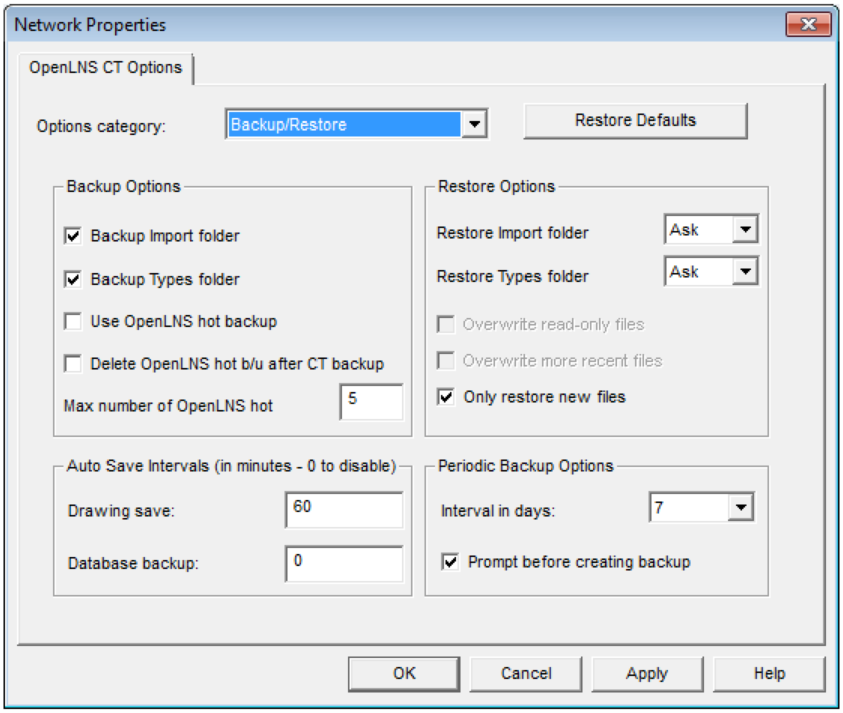
Every default OpenLNS CT option has one of the following three scopes that determine which networks and OpenLNS CT computers are updated when you change the value of an option:

- Current network, current OpenLNS CT computer. Value is updated in the network on the OpenLNS CT computer in which the change was made.
- Current network, all OpenLNS CT computers. Value is updated in the network for all OpenLNS CT computers accessing the network database
- All networks, current OpenLNS CT computer. Value is updated in all networks, and to the OpenLNS CT Design Manager, on the OpenLNS CT computer in which the change was made.

When you change OpenLNS CT options that have the “All Networks” scope from the OpenLNS CT Design Manager, the new values are updated in all existing networks when you re-open them. When you change these types of options in an OpenLNS CT drawing, the new values are updated immediately in that network and all other existing networks on your computer, and they are updated in the OpenLNS CT Design Manager when you re-open it.

The following sections describe the OpenLNS CT default options you can set for each options category. The scope for each OpenLNS CT default option is listed at the end of its description.

Backup/Restore Options



Options category: Backup/Restore

Backup Options

☑ Backup Import folder

☑ Backup Types folder

☐ Use OpenLNS hot backup

☐ Delete OpenLNS hot b/u after CT backup

Max number of OpenLNS hot: 5

Restore Options

Restore Import folder: Ask

Restore Types folder: Ask

☐ Overwrite read-only files

☐ Overwrite more recent files

☑ Only restore new files

Auto Save Intervals (in minutes - 0 to disable)

Drawing save: 60

Database backup: 0

Periodic Backup Options

Interval in days: 7

☑ Prompt before creating backup

The Backup/Restore category contains the following options. The changes you make in the Backup/Restore options category are applied to all OpenLNS CT network designs on your computer.

Backup Options

Backup Import Folder

Includes application files, device interface (**XIF**) files, and all other files in the LonWorks **Import** folder when creating an OpenLNS CT backup file. This option is selected by default.

If selected, a new **Import** folder will be inserted in the network database directory, and all the files in the LonWorks **Import** folder (even those files that are not resource files or are not used in the OpenLNS CT drawing) will be copied into that new **Import** folder prior to creating the OpenLNS CT backup.

Files outside the LONWORKS **Import** folder will not be backed up, even if they have been used in the OpenLNS CT drawing.

Scope: All networks, current OpenLNS CT computer

Backup Types Folder

Include resource files and all other files in the LONWORKS **Types** folder when creating an OpenLNS CT backup file. This option is selected by default.

If selected, a new **Types** folder will be inserted in the network database directory, and all the files in the LONWORKS **Types** folder (even those files that are not resource files or are not used in the OpenLNS CT drawing) will be copied into that new **Types** folder prior to creating the OpenLNS CT backup.

Files outside the LONWORKS **Types** folder will not be backed up, even if they have been used in the OpenLNS CT drawing.

Scope: All networks, current OpenLNS CT computer

Use OpenLNS Hot Backup

Automatically creates a hot backup copy of the OpenLNS network database, if required, when creating an OpenLNS CT backup. This option is selected by default.

If this check box is cleared and the OpenLNS network database is being accessed while attempting to create an OpenLNS CT backup, a prompt opens and offers the following three choices:

- Exit all applications accessing the database and try again.
- Use the OpenLNS hot backup (creates an OpenLNS hot backup).
- Cancel the backup request.

Scope: All networks, current OpenLNS CT computer

Delete OpenLNS Hot Backup after LM Backup

Deletes the hot backup copy of the network database once a standard OpenLNS CT backup has been created. OpenLNS CT may create an OpenLNS hot backup of the OpenLNS network database when creating an OpenLNS CT backup file. When an OpenLNS hot backup copy is created, a copy of it is placed in the OpenLNS CT backup file. Select this check box to delete the extra copy.

If this check box is cleared, the hot backup copy of the network database created by the OpenLNS hot backup will be preserved.

This option is cleared by default.

Scope: All networks, current OpenLNS CT computer

Max Number of OpenLNS Hot Backups

Sets the maximum number of unique hot backup copies of an OpenLNS database that will be preserved. An OpenLNS hot backup copy of the OpenLNS network database may be created when an OpenLNS backup file is created. If the **Delete OpenLNS Hot Backup after LM Backup** check box is cleared, the OpenLNS hot backup copies are preserved. Select a value between 1 and 99. The default value is **5**.

Scope: All networks, current OpenLNS CT computer

Auto Save Intervals

Drawing Save

Sets how frequently (in minutes) the OpenLNS CT drawing (**.VSD**) file is automatically saved. The default value is **60** minutes. To disable this feature, set the interval to **0**.

Scope: All networks, current OpenLNS CT computer

Database Backup

Sets how frequently (in minutes) OpenLNS CT will automatically create a hot backup of the OpenLNS database. The default value is **0** (never).

Scope: All networks, current OpenLNS CT computer

Restore Options

Restore Import folder

Selects how device interface (**XIF**) files, device application files, and other files saved in the **Import** folder are restored when restoring an OpenLNS CT backup. You have three choices:

- **No**. The device files are not installed.
- **Yes**. The device files are automatically installed.
- **Ask**. OpenLNS CT will prompt you to decide whether to install the device files upon restoring the network. This is the default.

Scope: All networks, current OpenLNS CT computer

Restore Types folder

Selects how resource files and other files saved in the **Types** folder are restored when restoring an OpenLNS CT backup. You have three choices:

- **No**. The resource files are not restored.
- **Yes**. The resource files are automatically restored.
- **Ask**. OpenLNS CT will prompt you to decide whether to restore the resource files upon restoring the network. This is the default.

Scope: All networks, current OpenLNS CT computer

Overwrite Read-only Files

Automatically replaces existing read-only device application, device interface, and resource files with backup versions of the files.

This option is cleared by default, and it is unavailable if the **Only Restore New Files** check box is selected.

Scope: All networks, current OpenLNS CT computer

Overwrite More Recent Files

Automatically replaces the existing device application, device interface, and resource files with backup versions of the files that are not as recent as the existing ones.

This option is cleared by default, and it is unavailable if the **Only Restore New Files** check box is selected.

Scope: All networks, current OpenLNS CT computer

Only Restore New Files

Automatically restores unique backup device application, device interface, and resource files. Backup files with the same name as an existing version will not be installed.

If this check box is not selected, OpenLNS CT will install backup device application, device interface, and resource files based on the settings of the **Overwrite Read-only Files** and **Overwrite More Recent Files** check boxes

This option is selected by default, and it is unavailable if either the **Overwrite Read-only Files** or **Overwrite More Recent Files** check boxes are selected.

Scope: All networks, current OpenLNS CT computer

Periodic Backup Options

Interval in Days

Sets the number of days between periodic OpenLNS CT backups. The backup will be automatic or manual depending on the **Prompt Before Creating Backup** setting. You can select 1, 2, 3, 7, 14, or 28 days, or never. The default is **7** days.

Scope: All networks, current OpenLNS CT computer

Prompt Before Creating Backup

Prompts you to create an OpenLNS CT backup when the backup interval has elapsed. This option is selected by default.

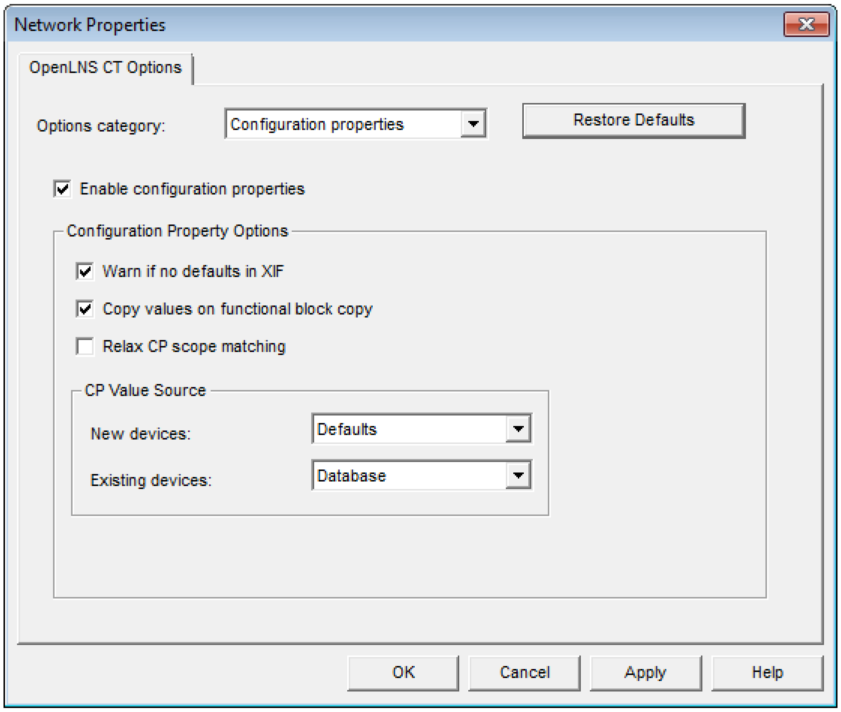
If you choose create an OpenLNS CT backup when prompted, specify the backup settings (such as the backup file name and whether to back up the OpenLNS CT drawing and/or database) and set any desired options (such as backing up the files in the Types and Import folders, and using an OpenLNS hot backup).

If you do not create an OpenLNS CT backup when prompted, you will be prompted again the next time you close your network.

If this check box is clear, OpenLNS CT automatically creates an OpenLNS CT backup when the backup interval has elapsed.

Scope: All networks, current OpenLNS CT computer

Configuration Properties Options



Options category: Configuration properties

☑ Enable configuration properties

Configuration Property Options

☑ Warn if no defaults in XIF

☑ Copy values on functional block copy

☐ Relax CP scope matching

CP Value Source

New devices: Defaults

Existing devices: Database

The Configuration Properties category contains the following options.

Enable Configuration Properties

Enables access to configuration properties. This option is selected by default.

Also enables automatic extension record copying. When this check box is selected, a device's extension records that are marked for copying will automatically be copied when you create a new device by copying an existing one and when you create a new custom device master SmartShape.

If this check box is cleared, you will not be able to view or modify configuration properties from OpenLNS CT. This speeds up the OpenLNS CT Browser, but reduces your ability to configure your devices from OpenLNS CT.

Scope: Current network, all OpenLNS CT computers

Configuration Property Options

Warn if no Default in XIF

Notifies you if you have requested to copy default configuration property values to a device, but the device template was created from a device interface (XIF) file with no default configuration property values. The device manufacturer must include default configuration property values in their XIF file for default configuration property values to be used.

This check box is only available if the **Enable Configuration Properties** check box has been selected, and it is selected by default.

Scope: Current network, all OpenLNS CT computers

Copy Values on Functional Block Copy

Copies CP values when creating a new functional block from a master SmartShape or from another functional block.

This check box is only available if the **Enable Configuration Properties** check box has been selected, and it is selected by default.

Scope: Current network, all OpenLNS CT computers

Relax CP Scope Matching

Enables OpenLNS CT to re-attempt the validation of the creation of a functional block from a custom master SmartShape or the replication of configuration property values from one functional block to another using a lower scope selector if the initial validation operation fails using the defined scope selector.

For example, if you attempt to create a functional block from a custom master SmartShape that has a defined scope selector of 6 but the validation operation fails, OpenLNS CT will then try to validate the functional block again using scope 3. If the validation re-attempt succeeds, the operation will be allowed.

Scope: All networks, current OpenLNS CT computer

CP Value Source

New Devices

Selects the source of configuration property values for new devices. You have two choices:

- **Defaults**. The default configuration property values from the device template will be copied to the device. The default configuration property values in the device template may have been copied from the original manufacturer's device interface (XIF) file, or may have been updated from a device. This is the default
- **Device**. The current configuration properties in the device will be kept and will be copied to the device template.

  A new device may have settings different from the defaults if the device configuration properties were set on a network other than the current network.

Scope: Current network, current OpenLNS CT computer

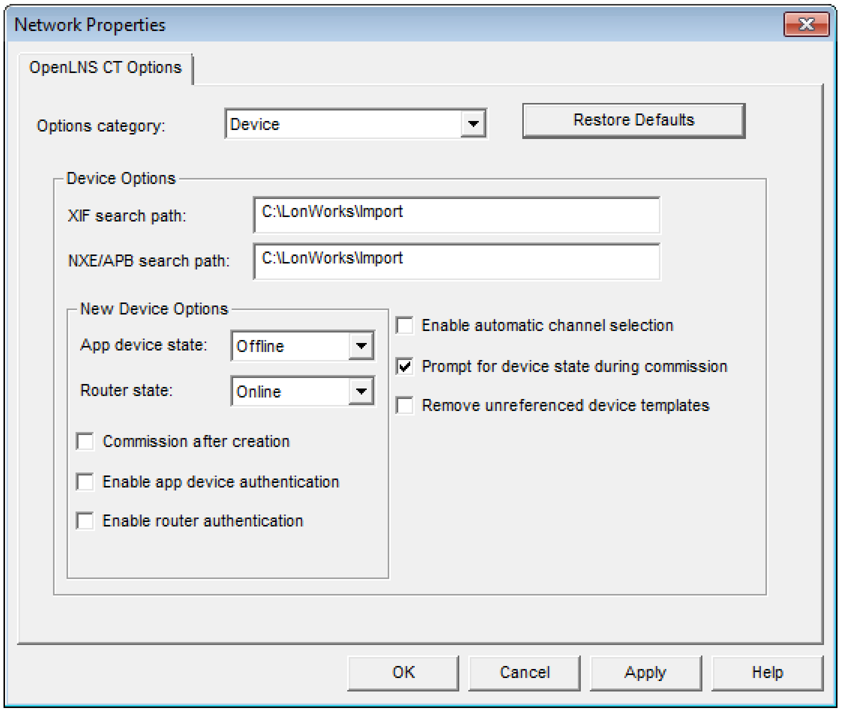
Existing Devices

Selects the source of configuration property values for existing devices. You have three choices:

- **Database**. The configuration property values stored in the OpenLNS database will be copied to the device when a device is commissioned, replaced, or loaded. This is the default
- **Defaults**. The default configuration property values from the device template will be copied to the device when a device is commissioned, replaced, or loaded.
- **Device**. The current configuration properties in the device will be kept and will be copied to the device template when a device is commissioned, replaced, or loaded.

Scope: Current network, current OpenLNS CT computer

Device Options



Options category: Device

Device Options

XIF search path: C:\LonWorks\Import

NXE/APB search path: C:\LonWorks\Import

New Device Options

App device state: Offline

Router state: Online

□ Commission after creation

□ Enable app device authentication

□ Enable router authentication

□ Enable automatic channel selection

☑ Prompt for device state during commission

□ Remove unreferenced device templates

The Device category contains the following options.

Device Options

XIF Search Path

Specifies the path that OpenLNS CT uses to search for device interface (**XIF**) files. You can enter multiple directories by separating them with semicolons.

Scope: All networks, current OpenLNS CT computer

NXE/APB Search Path

Specifies the path that OpenLNS CT uses to search for device application files (**.NXE** and **.APB** extensions). You can enter multiple directories by separating them with semicolons.

Scope: All networks, current OpenLNS CT computer

Enable Automatic Channel Selection

Automatically assigns the appropriate channel to a new device SmartShape, thereby skipping the Device Wizard. This option is cleared by default.

The device SmartShape must contain both the definition of the device template that will be used and the channel type information. This feature will therefore not work for the Device SmartShape in the **OpenLNS CT Basic Shapes** stencil. To accelerate network design, create custom device master SmartShapes for each of the device types that you typically use.

The order of the criteria used by OpenLNS CT to select the appropriate channel is as follows: (1) closest qualified channel to device SmartShape where the channel is within 1 ½ times the width of the device SmartShape; (2) most recently used or created channel; and (3) other available defined channels. If no channel can be selected using the preceding criteria, the Device Wizard will appear.

Scope: All networks, current OpenLNS CT computer

Prompt for Device State During Commissioning

Includes the window in the Commissioning Device Wizard for setting the initial state of application device/router applications and the source of device configuration property values. This option is selected by default.

If this check box is cleared, this window will be skipped. The initial device state will be determined by the current setting in the **New Device Options** box. The source of the device configuration property values will be will be determined by the current setting in the **CP Value Source** box located in the **Configuration Properties** options category.

Scope: Current network, current OpenLNS CT computer

Remove Unreferenced Device Templates

Removes a device template when all devices using that template have been deleted. This option is cleared by default.

Registered plug-ins associated with a device template will become unregistered if that device template is removed. You will need to re-register the plug-ins using the **Plug-In Registration** tab in the **Network Properties** command under the OpenLNS CT menu.

Scope: Current network, all OpenLNS CT computers

New Device Options

App Devices

Specifies the initial state of a new application device after it has been commissioned. You have three choices:

- **Online**. Places the device in the online state. The behavior in the online state depends on the device. A Neuron-hosted device, for example, will run its application after it has been commissioned.
- **Offline**. Places the device in the offline state. The behavior in the offline state depends on the device. A Neuron-hosted device, for example, will not run its application after it has been commissioned. This is the default and can be used to bring up a system incrementally.

  When a device is offline, portions of its application may still function and it will still respond to Online, Wink, and Test commands. An offline device still receives network variable updates, but it typically does not process updated network variable values or send network variable updates.

  Resetting an offline device makes it go online, unless the device is in the hard offline state, in which case it will remain offline after a reset.

- **Disable**. Disables the device’s application. The device application will remain disabled, even if the device is reset. For a device to be disabled, it must contain a Node Object functional block. If **Disable** is selected and the device does not contain a Node Object functional block, the device is placed offline. As with offline operation, portions of a disabled device’s application may continue to function.

Scope: Current network, all OpenLNS CT computers

Routers

Specifies the initial state of a new router after it has been commissioned. You have two choices:

- **Online**. The router will forward packets between the channels connected to it. Use this setting to ensure routers are operational as they are commissioned. This is the default.
- **Offline**. The router will not forward packets between the channels connected to it.

Scope: Current network, all OpenLNS CT computers

Commission After Creation

Selects the check box to commission new devices and routers on the first page of their respective wizards. This option is cleared by default.

Scope: Current network, current OpenLNS CT computer

Enable App Device Authentication

Enables authentication for new devices. Authentication must be enabled for the network for it to be used by a device. This option is cleared by default.

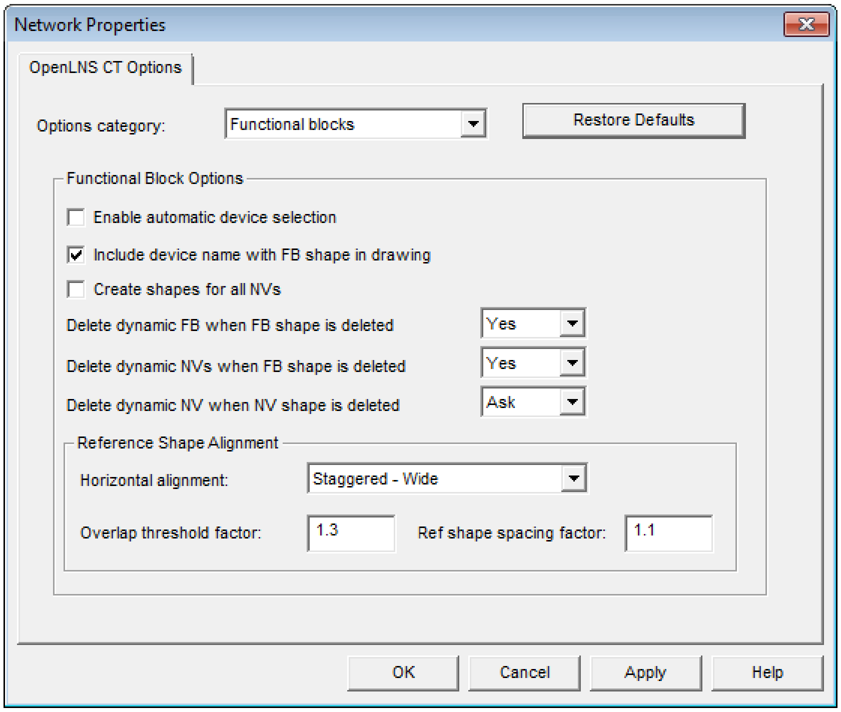
Scope: Current network, all OpenLNS CT computers

Enable Router Authentication

Enables authentication for new routers. Authentication must be enabled for the network for it to be used by a router. This option is cleared by default.

Scope: Current network, all OpenLNS CT computers

Functional Block Options



Network Properties

OpenLNS CT Options

Options category: Functional blocks

Restore Defaults

Functional Block Options

☐ Enable automatic device selection

☑ Include device name with FB shape in drawing

☐ Create shapes for all NVs

Delete dynamic FB when FB shape is deleted: Yes

Delete dynamic NVs when FB shape is deleted: Yes

Delete dynamic NV when NV shape is deleted: Ask

Reference Shape Alignment

Horizontal alignment: Staggered - Wide

Overlap threshold factor: 1.3

Ref shape spacing factor: 1.1

OK | Cancel | Apply | Help

The Functional Block category contains the following options.

Functional Block Options

Enable Automatic

Automatically assigns the appropriate device to a new functional

Device Selection

block SmartShape, thereby skipping the Functional Block wizard. This option is cleared by default.

The functional block SmartShape must contain a functional block type specification for this feature to work. This feature will therefore not work for the Functional Block SmartShape in the **OpenLNS CT Basic Shapes** stencil. To accelerate network design, create custom device master SmartShapes for each of the functional block types that you typically use.

The order of the criteria used by OpenLNS CT to select the appropriate device is as follows: (1) closest qualified device to functional block SmartShape where the device is within 1 ½ times the width of the functional block SmartShape, and then (2) device most recently created or device that has most recently been assigned a functional block. If no device can be selected using the preceding criteria, the Functional Block wizard will appear.

Scope: All networks, current OpenLNS CT computer

Include Device Name with FB SmartShape in Drawing

Includes the device name in the name displayed for a functional block in the OpenLNS CT drawing. The device name followed by a period will precede the functional block name. For example, the default name of a functional block SmartShape associated with a second analog output device created in the OpenLNS CT drawing would be **AO- 2. AO- 1**. This option is selected by default.

Scope: Current network, all OpenLNS CT computers

Create SmartShapes for all NVs

Adds network variable SmartShapes for all network variables on a functional block. This option is cleared by default.

Scope: Current network, all OpenLNS CT computers

Delete Dynamic FB when FB SmartShape is Deleted

Deletes a dynamic functional block from a device when its functional block SmartShape is deleted from an OpenLNS CT drawing. You have three choices:

- **No**. The dynamic functional block is not deleted.
- **Yes**. The dynamic functional block is automatically deleted. This is the default.
- **Ask**. OpenLNS CT will prompt you to decide whether to delete the dynamic functional block.

Scope: All networks, current OpenLNS CT computer

Delete Dynamic NVs when FB SmartShape is Deleted

Deletes a dynamic network variable from a device when its functional block SmartShape is deleted from an OpenLNS CT drawing. You have three choices:

- **No**. The dynamic network variable is not deleted.
- **Yes**. The dynamic network variable is automatically deleted. This is the default.
- **Ask**. OpenLNS CT will prompt you to decide whether to delete the dynamic network variable.

Scope: All networks, current OpenLNS CT computer

Delete Dynamic NV when NV SmartShape

Deletes a dynamic network variable from a device when its network variable SmartShape is deleted from a functional block SmartShape.

is Deleted

You have three choices:

- **No**. The dynamic network variable is not deleted.
- **Yes**. The dynamic network variable is automatically deleted.
- **Ask**. OpenLNS CT will prompt you to decide whether to delete the dynamic network variable. This is the default.

Scope: All networks, current OpenLNS CT computer

Reference Shape Alignment

Select how OpenLNS CT aligns reference connection SmartShapes. The alignment settings are applied if the **Align Reference Shapes** option is set on the functional block. To set this option, right-click a functional block, point to **Display Options**, and click **Align Reference Shapes** if it is not already set.

Horizontal Alignment

Selects how OpenLNS CT aligns overlapping connector reference SmartShapes. You have three choices:

- **Staggered Wide**. Connector reference SmartShapes are staggered horizontally, in two columns, with their centers aligned. This is the default.
- **Staggered Narrow**. Connector reference SmartShapes are staggered horizontally, in two columns, with the outermost reference SmartShapes located close to the functional block SmartShape and somewhat overlapping with reference SmartShapes above and below them.
- **Overlapped**. Connector reference SmartShapes are not staggered and will overlap.

Scope: Current network, current OpenLNS CT computer

Overlap Threshold Factor

Determines the point at which OpenLNS CT will overlap multiple connector reference SmartShapes that are attached to the same network variable SmartShape on one side of a functional block. Overlapping connector reference SmartShapes cannot be aligned with the network variable SmartShapes to which they are connected.

If the overlap threshold factor multiplied by the height of the functional block SmartShape is greater than the total height of the reference connector SmartShapes (including spacing), the reference connector SmartShapes will not be overlapped.

The default overlap threshold factor is **1.3**. For the best appearance, keep this value below 2.

Scope: Current network, current OpenLNS CT computer

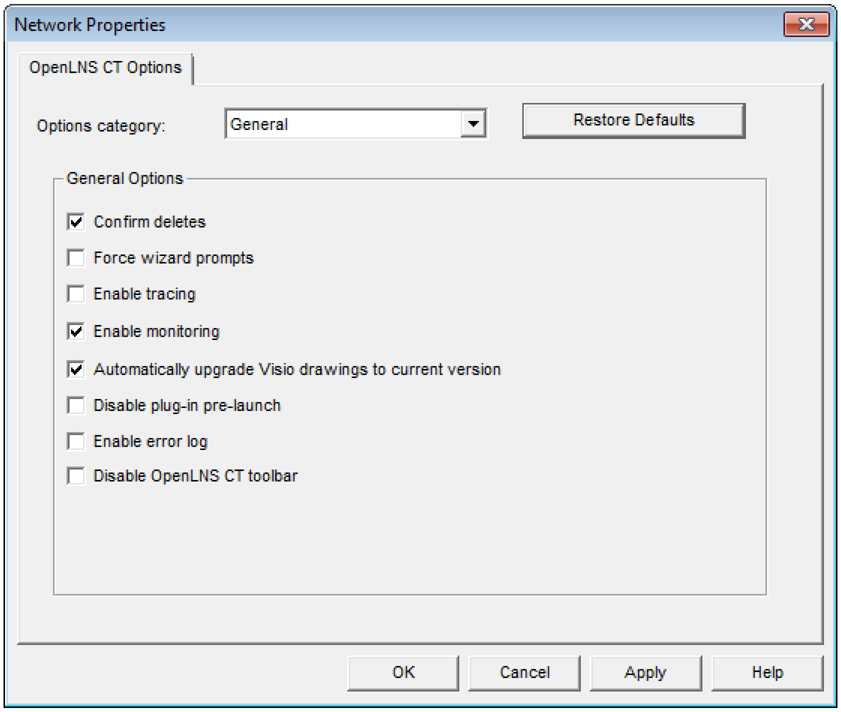
Ref SmartShape Spacing Factor

Determines the vertical spacing between multiple connector reference SmartShapes when they are not aligned with the network variable SmartShapes to which they are connected.

The default reference SmartShape spacing value is **1.1**.

Scope: Current network, current OpenLNS CT computer

General Options



Options category: General

General Options

☑ Confirm deletes

□ Force wizard prompts

□ Enable tracing

☑ Enable monitoring

☑ Automatically upgrade Visio drawings to current version

□ Disable plug-in pre-launch

□ Enable error log

□ Disable OpenLNS CT toolbar

The General category contains the following options.

Confirm Deletes

Prompts you to confirm the deleting of an OpenLNS CT SmartShape. This option is selected by default.

Scope: Current network, current OpenLNS CT computer

Force Wizard Prompts

Displays all wizards when you create a new OpenLNS CT SmartShape, regardless of the settings of any other options. This option is cleared by default.

If this check box is selected, and you create a functional block with automatic device selection enabled, for example, the Functional Block Wizard will still open.

If this check box is cleared and a functional block and its associated device are copied, for example, the new functional block will automatically be associated with the new device and the Functional Block Wizard will not open.

Scope: Current network, current OpenLNS CT computer

Enable Tracing

Enables OpenLNS CT tracing. This option is cleared by default.

The OpenLNS CT tracing output is saved in a file called **trace.log**, which is stored in the **C:\Users\Public\Documents\LonWorks\OpenLnsCt\ \Drawings**\<*network*> folder.

Scope: Current network, current OpenLNS CT computer

Enable Monitoring

Enables connection monitoring and data point monitoring. Clear this check box to disable all NV polling by OpenLNS CT. This check box does not affect the OpenLNS CT Browser. This option is selected by default.

Scope: Current network, current OpenLNS CT computer

Automatically upgrade Visio drawings to current version

Automatically upgrades an OpenLNS CT drawing to the current version of Visio you are using, if possible. This option is selected by default.

Scope: All networks, current OpenLNS CT computer

Disable plug-in pre-launch

Disables the pre-launching of plug-ins when an OpenLNS CT drawing is opened. This option is cleared by default.

Scope: All networks, current OpenLNS CT computer

Enable Error Log

Writes all displayed error messages to the error log, which is stored in the drawing directory of the OpenLNS CT network. This option is cleared by default.

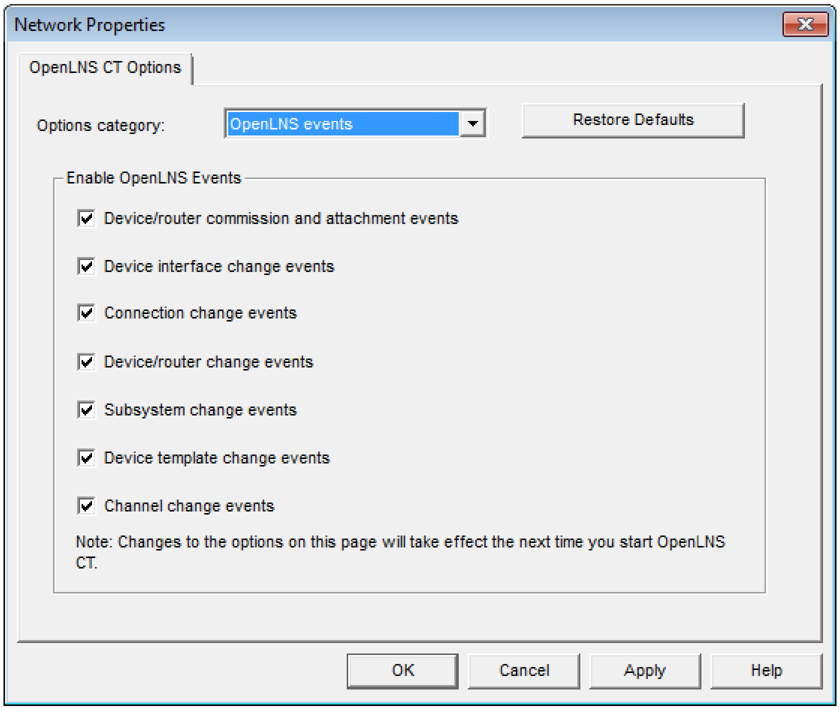
The error log is circular and has a maximum size of 100K; therefore, error messages at the beginning of the log will be overwritten to store new error messages.

Scope: All networks, current OpenLNS CT computer

OpenLNS Event Options

The OpenLNS Event options category contains a set of check boxes for enabling and disabling OpenLNS events that are used to update the OpenLNS CT drawing. To maintain synchronization between your OpenLNS CT drawing, the OpenLNS CT database, and other OpenLNS CT tools, applications, and plug-ins, enable all the OpenLNS events. If you are using an OpenLNS tool, application, or plug-in other than OpenLNS CT to make changes to the OpenLNS network database, OpenLNS CT uses OpenLNS events to keep OpenLNS CT synchronized with that OpenLNS application or plug-in.

If you are only using OpenLNS CT, you can disable all the OpenLNS events except for the device/router commission and attachment events, as OpenLNS CT tracks these events itself. A benefit of disabling the OpenLNS events is that it may improve network performance.



Options category: OpenLNS events

Enable OpenLNS Events

☑ Device/router commission and attachment events

☑ Device interface change events

☑ Connection change events

☑ Device/router change events

☑ Subsystem change events

☑ Device template change events

☑ Channel change events

Note: Changes to the options on this page will take effect the next time you start OpenLNS CT.

The OpenLNS Event category contains the following options. The changes you make in the OpenLNS Events options category are all applied to the current network on your computer.

Device/router Commission and Attachment Events

Updates the OpenLNS CT drawing when devices or routers are attached to the network, detached, commissioned, or decommissioned. This option is selected by default.

Scope: Current network, current OpenLNS CT computer

Device Interface Change Events

Updates the OpenLNS CT drawing when a device's functional blocks, network variables, or configuration properties are added, removed, or modified. This option is selected by default.

Scope: Current network, current OpenLNS CT computer

Connection Change Events

Updates the OpenLNS CT drawing when a connection is created or modified, or when a connection description template being used in your network is modified. This option is selected by default.

Scope: Current network, current OpenLNS CT computer

Device/router Change Events

Updates the OpenLNS CT drawing when new devices or routers are detected. This option is selected by default.

If selected, a device will be automatically upgraded when a new application is loaded into a device or when a device is replaced. OpenLNS CT will automatically make the device compatible with the updated device template, and it will maintain existing compatible connections and configuration property settings.

Scope: Current network, current OpenLNS CT computer

Subsystem Change Events

Updates the OpenLNS CT drawing when subsystems are added, removed, or modified. This option is selected by default.

Scope: Current network, current OpenLNS CT computer

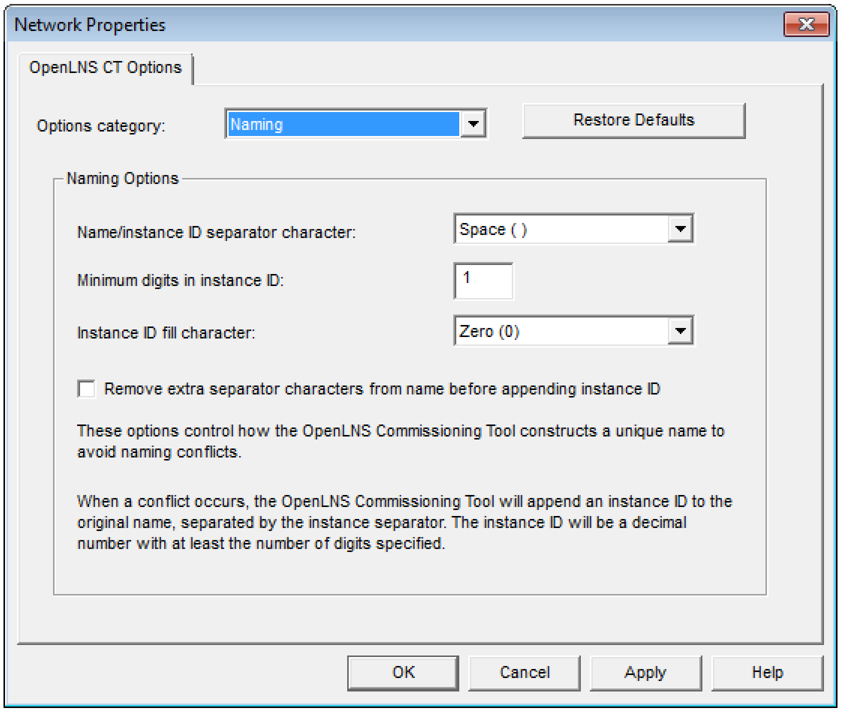
Device Template Change Events

Updates the internal copies of device templates maintained by OpenLNS CT when device templates are added, removed, or modified. This option is selected by default.

Scope: Current network, current OpenLNS CT computer

Channel Change Events	Updates the OpenLNS CT drawing when channels are added, removed, or modified. This option is selected by default. **Scope:** Current network, current OpenLNS CT computer

Naming Options



Options category: Naming

Naming Options

Name/instance ID separator character: Space ( )

Minimum digits in instance ID: 1

Instance ID fill character: Zero (0)

□ Remove extra separator characters from name before appending instance ID

These options control how the OpenLNS Commissioning Tool constructs a unique name to avoid naming conflicts.

When a conflict occurs, the OpenLNS Commissioning Tool will append an instance ID to the original name, separated by the instance separator. The instance ID will be a decimal number with at least the number of digits specified.

The Naming category contains the following options. The changes you make in the Naming options category are applied to the current network on your computer.

Naming Options	Controls how OpenLNS CT constructs a unique name for new OpenLNS CT SmartShapes. If there is a naming conflict, OpenLNS CT will append to a new OpenLNS CT SmartShape name a space, dash, or other selected separator character followed by an instance ID. The instance ID starts at 1 and is incremented by one each time a new SmartShape of a given type (device, router, functional block, and so on) with an identical name of an existing SmartShape is added to your OpenLNS CT drawing. For example, the default name of the first generic device SmartShape added to your OpenLNS CT drawing is **Device**. The second generic device SmartShape is **Device 1**, but you can change this to **Device1**, **Device-1**, or **Device-001** based on the naming options you set. **Scope:** Current network, current OpenLNS CT computer

Name/instance ID Separator Character

Selects the character to be inserted between the OpenLNS CT SmartShape name and the instance ID. The choices are **Space ( )**, **Underscore (_)**, **Dash (-)**, **Exclamation Point (!)**, and **Hash (#)**. The default separator character is **Space ( )**.

For example, the default name of the first analog input (**AI-**) LonPoint device added to your OpenLNS CT drawing with a
space ( ) selected as the separator character is **AI- 1**. If you select a dash as the separator character, the default name is **AI--1**.

Scope: Current network, current OpenLNS CT computer

Minimum Digits in Instance ID

Sets the minimum number of digits to be used in the instance ID. The default value is **1**.

For example, the default name of the first analog input (**AI-**) LonPoint device added to your OpenLNS CT drawing with the minimum digits set to 2 is **AI- 01**, set to 3 it is **AI- 001**, and so on.

To ensure that objects are listed numerically in various OpenLNS CT menus and reports, set the minimum digits in the instance ID to accommodate the maximum number of a specific type of object you plan on creating in a subsystem. For example, if you plan on creating 10 or more of a certain device in a subsystem, you should set the minimum digits to two; if you plan on creating 100 or more of a device, set it to three; and so on.

Scope: Current network, current OpenLNS CT computer

Instance ID Fill Character

Selects the characters to be inserted between the separator character and the instance ID when the number of digits in the instance ID is less than minimum digits set. The choices are **Space ( )** or **Zero (0)**. The default fill character is **Space ( )**.

For example, the default name of the first analog input (**AI-**) LonPoint device added to your OpenLNS CT drawing with the minimum digits set at 2 and a space selected as the fill character is **AI- 1**. If a zero (0) was selected as the fill character, the default name is **AI- 01**.

Scope: Current network, current OpenLNS CT computer

Remove Extra Separator Characters from Name before Appending Instance ID

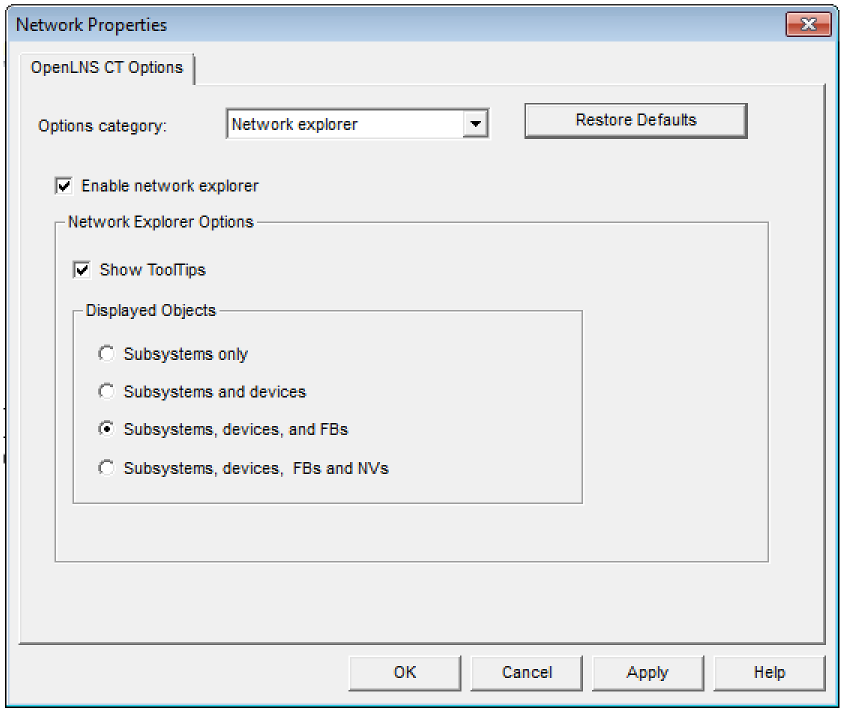
Automatically deletes any separator characters appended to OpenLNS CT SmartShape names. This option is cleared by default.

For example, the default name of the first analog input (**AI-**) LonPoint device added to your OpenLNS CT drawing with this check box cleared is **AI- 1**. If this check box was selected, the default name would be **AI 1**.

Scope: Current network, current OpenLNS CT computer

Network Explorer Options

The Network Explorer is an auto-hide window that displays a hierarchal view of the OpenLNS CT drawing. You can use the Network Explorer to navigate, view, and modify the subsystems, devices, and device templates in your network design.



Network Properties

OpenLNS CT Options

Options category: Network explorer

☑ Enable network explorer

Network Explorer Options

☑ Show ToolTips

Displayed Objects

○ Subsystems only

○ Subsystems and devices

● Subsystems, devices, and FBs

○ Subsystems, devices, FBs and NVs

The Network Explorer category contains the following options. The changes you make in the Network Explorer options category are applied to all networks on your computer.

Enable Network Explorer

Enables the network explorer to appear in your OpenLNS CT drawing. This option is selected by default.

Scope: All networks, current OpenLNS CT computer

Network Explorer Options

Show ToolTips

Enables the name of the OpenLNS CT object and its absolute path to appear in a ToolTip when the mouse pointer is positioned over the object. This option is selected by default.

Scope: All networks, current OpenLNS CT computer

Displayed Objects

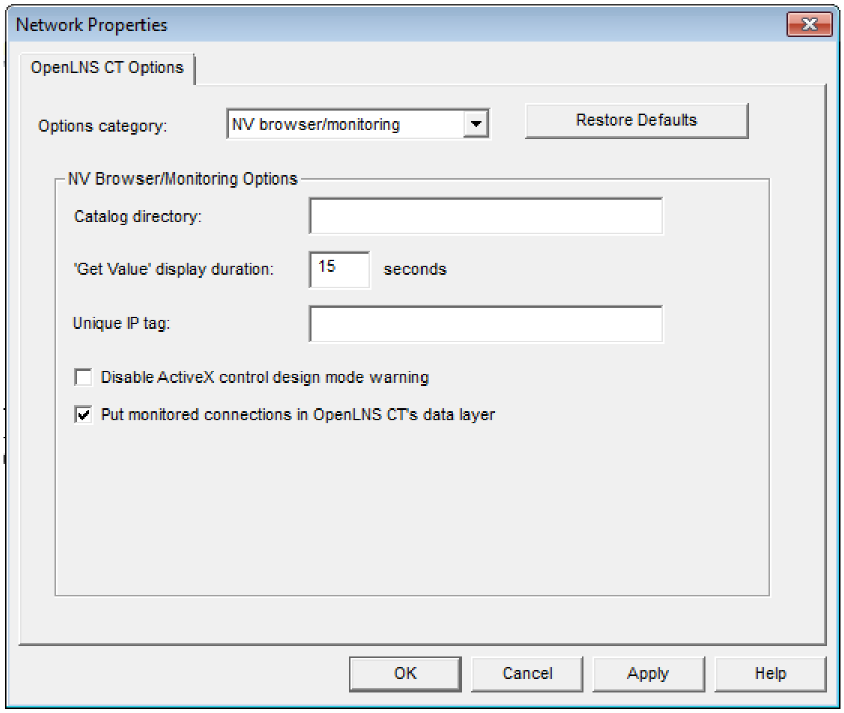
Select the scope of the objects displayed within the network explorer. You have four choices:

- **Subsystems Only**. Lists only the subsystems in the OpenLNS CT network.
- **Subsystems and Devices**. Lists the subsystems and devices in the OpenLNS CT network.
- **Subsystems, Devices, and FBs**. Lists the subsystems, devices, and functional blocks in the OpenLNS CT network.
- **Subsystems, Devices, FBs, and NVs**. Lists the

subsystems, devices, functional blocks, and network variables in the OpenLNS CT network.

Scope: All networks, current OpenLNS CT computer

NV Browser/Monitoring Options



Options category: NV browser/monitoring

NV Browser/Monitoring Options

Catalog directory:

'Get Value' display duration: 15 seconds

Unique IP tag:

□ Disable ActiveX control design mode warning

☑ Put monitored connections in OpenLNS CT's data layer

The NV Browser/Monitoring category contains the following options.

NV Browser/Monitoring Options

Catalog Directory

Specifies the directory containing resource files (**.TYP**, **.FMT**, **.FPT,** and language files). The default directory is LONWORKS\Types.

Scope: Current network, all OpenLNS CT computers

'Get Value' Display Duration

Specifies the number of seconds that a value will be displayed when you right-click a Connector SmartShape and then click **Get Input Value** or **Get Output Value**. If you set this option to 0, the value will be displayed indefinitely. The default time is **15** seconds.

Scope: All networks, current OpenLNS CT computer

Unique IP Tag

Specifies a tag that identifies a remote lightweight client to the OpenLNS server for accessing its monitor sets. A monitor set is a group of point definitions that OpenLNS CT uses to monitor and control network variables. The tag is used to correlate a lightweight client to its monitor sets. By default, this tag is the computer's Windows network name.

If you have existing monitor sets and you change this tag, the existing monitor sets can no longer connect to the lightweight client. To remove the monitor sets, use the **Remove Monitor Sets** command from the **Network Service Devices** command on the OpenLNS CT menu.

Scope: All networks, current OpenLNS CT computer

Disable ActiveX Control Design Mode Warning

Disables the warning that opens when you drag an ActiveX control to your OpenLNS CT drawing. Dragging an ActiveX control to your OpenLNS CT drawing puts OpenLNS CT in design mode, which affects the behavior of all ActiveX controls. This check box is cleared by default.

Scope: Current network, current OpenLNS CT computer

Put Monitored Connections in OpenLNS CT's Data Layer

Automatically displays monitored connections in the OpenLNS CT Data Layer, in addition to displaying them in the OpenLNS CT Logical Layer. If cleared, monitored connections will only be displayed in the OpenLNS CT Logical Layer. This check box is selected by default.

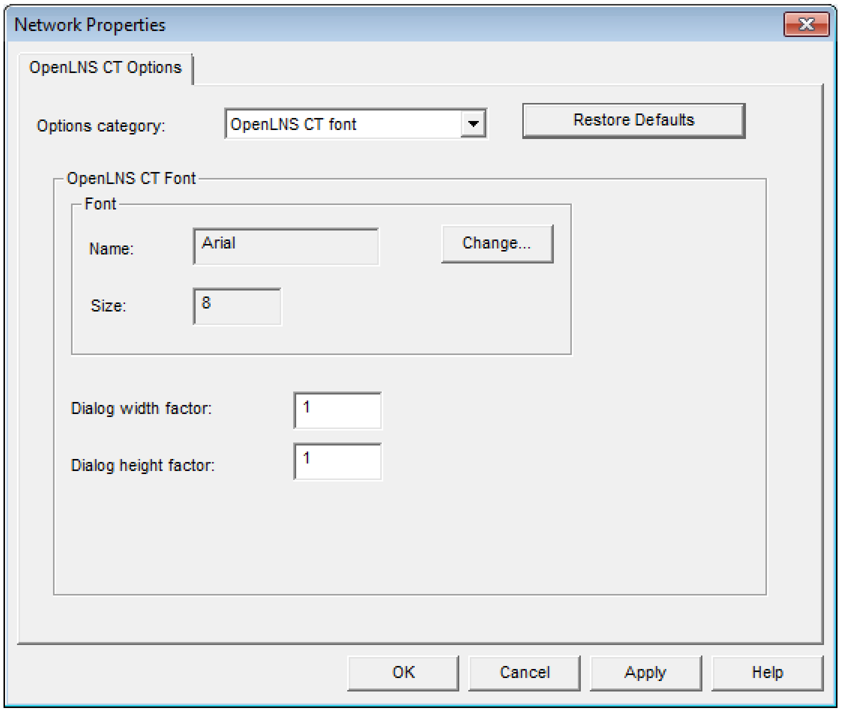
Scope: All networks, current OpenLNS CT computer

OpenLNS CT Font Options

The OpenLNS CT Font options category lets you adjust the font and the size of the OpenLNS CT Design Manager and the dialogs in the OpenLNS CT drawing. This is useful in the event you need to re-size dialogs that have become truncated, which may occur if you are using an Asian version of Windows. Otherwise, you will not likely need to change the settings in this options category.

To apply changes in the OpenLNS CT Font options category to the OpenLNS CT Design Manager, exit and then re-open the OpenLNS CT Design Manager.

The changes you make in the OpenLNS CT Font options category are applied to all networks on your computer.



Network Properties

OpenLNS CT Options

Options category: OpenLNS CT font

OpenLNS CT Font

Font

Name: Arial

Size: 8

Dialog width factor: 1

Dialog height factor: 1

The OpenLNS CT Font category contains the following options.

OpenLNS CT Font

Name — Displays the name of the current OpenLNS CT font. The default value is **Arial**.

Scope: Current network, current OpenLNS CT computer

Size — Displays the size of the current OpenLNS CT font. The default value is **8** points.

Scope: Current network, current OpenLNS CT computer

Change — Opens a dialog in which you can change the font, size, or style of the OpenLNS CT font.

Scope: Current network, current OpenLNS CT computer

Dialog Width Factor — Sets the width of all dialogs in the OpenLNS CT drawing and the width of the OpenLNS CT Design Manager. You can use this setting to adjust the dialog and Design Manager width when changing the OpenLNS CT font. The default value is **1**.

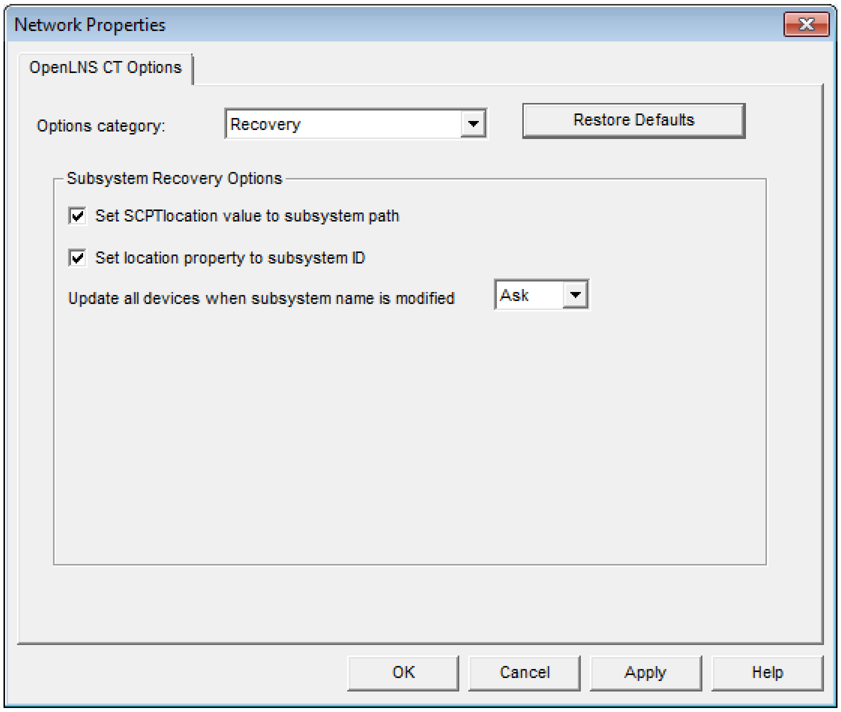
Scope: Current network, current OpenLNS CT computer

Dialog Height Factor — Sets the height of all dialogs in the OpenLNS CT drawing and the height of the OpenLNS CT Design Manager. You can use this setting to adjust the dialog and Design Manager height when changing the OpenLNS CT font. The default value is **1**.

Scope: Current network, current OpenLNS CT computer

Recovery Options

The Recovery options category lets you store and automatically update subsystem data in application devices and routers. This is useful in the event you need to recover your OpenLNS network database from the network. Setting these options enables OpenLNS CT to use the saved subsystem data to place application devices and routers in their correct subsystems. See *Recovering an OpenLNS CT Network* in Chapter 7, *Maintaining Networks*, for more information on the network recovery process.



Options category: Recovery

Subsystem Recovery Options

☑ Set SCPTlocation value to subsystem path

☑ Set location property to subsystem ID

Update all devices when subsystem name is modified: Ask

The Recovery category contains the following options. The changes you make in the Recovery options category are applied to the current network on all OpenLNS CT computers.

Subsystem Recovery Options

If you select either of the following subsystem recovery options in the OpenLNS CT drawing, you need to resynchronize your drawing and select the **Update Location Path and ID in Devices** check box in the **Synchronize OpenLNS CT Drawings** dialog box in order to update the devices with the current subsystem location data.

To resynchronize your drawing, click **Add-ins**, point to **Synchronization**, and click **Resynchronize**.

Set SCPT Location Value to Subsystem Path

Stores the subsystem path in the **SCPTlocation** configuration property of an application device when it is created. Automatically updates **SCPTlocation** when the application device is commissioned. The **SCPTlocation** configuration property must apply to the Node Object functional block if the device has a Node Object functional block, or it must apply to the entire device if the device does not have a Node Object functional block.

Automatically updates the **SCPTlocation** configuration property when the application device is moved or its subsystem is moved or renamed (provided that the **Update all Devices when Subsystem Name is Modified** option is set to **Yes**) , or you select the **Update Location Path and ID in Devices** check box in the **Synchronize OpenLNS CT Drawing** dialog. This check box is cleared by default.

If an application device and its functional blocks span multiple subsystems, OpenLNS CT will use the subsystem path in which the application device SmartShape is located.

If OpenLNS CT cannot locate the applicable **SCPTlocation** configuration property in an application device, the subsystem path will not be stored in the device.

Scope: Current network, all OpenLNS CT computers

Set Location Property to Subsystem ID

Stores the subsystem ID in the location field of an application device and router when the device is commissioned. Automatically updates the subsystem ID in the location field when the application device or router is moved or its subsystem is moved or renamed (provided that the **Update all Devices when Subsystem Name is Modified** option is set to **Yes**), or you select the **Update Location Path and ID in Devices** check box in the **Synchronize OpenLNS CT Drawing** dialog. This check box is selected by default.

The subsystem ID is the first 15 bits of the OpenLNS subsystem handle (in a very large network, with many subsystems where more than 32767 subsystems have been created and deleted this could result in multiple unique subsystems having the same ID).

Scope: Current network, all OpenLNS CT computers

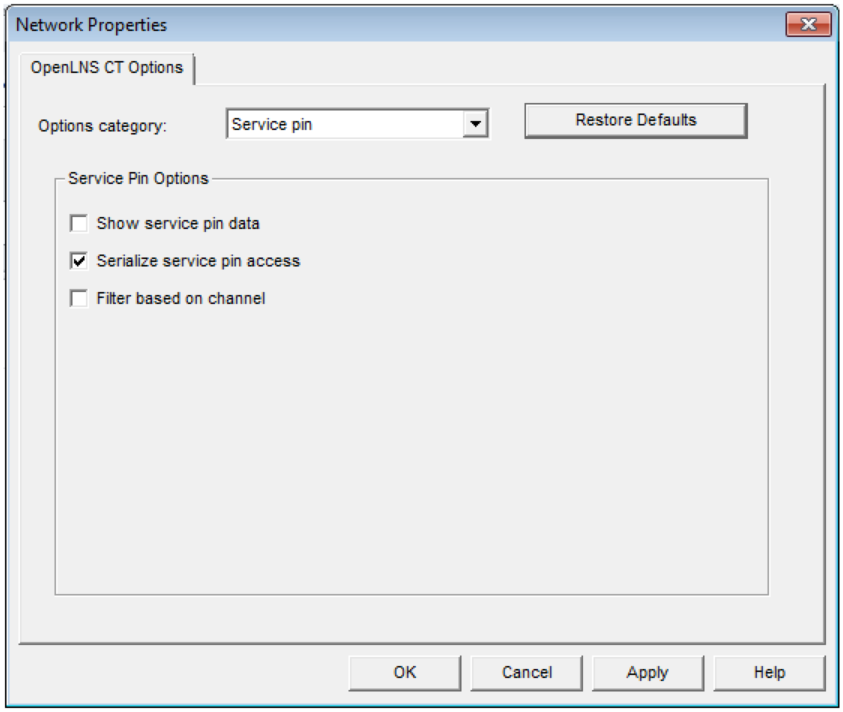
Update all Devices when Subsystem Name is Modified

Select whether OpenLNS CT updates the SCPTlocation configuration property and location field in an application device or router when the path or ID of its subsystem changes. You have three choices:

- **No**. Devices are not updated. You will need to resynchronize your OpenLNS CT drawing to update devices with the current subsystem path and ID.
- **Yes**. Application devices and routers are automatically updated to reflect the new subsystem path or ID.
- **Ask**. OpenLNS CT will prompt you whether to update devices. This is the default.

Scope: Current network, all OpenLNS CT computers

Service Pin Options



Options category: Service pin

Service Pin Options

□ Show service pin data

☑ Serialize service pin access

□ Filter based on channel

You can set options for using the service pin when commissioning a device.

Show Service Pin Data

Displays the service pin data received by OpenLNS CT when you press the device's service pin. This option is cleared by default.

You can change this option when you commission a device or router in the Commission Device/Router Wizard.

Scope: Current network, current OpenLNS CT computer

Serialize Service Pin Access

Limits service pin installation to only one installer looking for a service pin message at a time. This option is selected by default.

Clear this check box if multiple users are commissioning devices at the same time and are working on different channels connected by configured routers.

Select this check box if multiple users are commissioning devices at the same time and are working on channels connected by logical repeaters, or if multiple users are installing devices on the same channel.

Either setting can be used if you have a single user commissioning devices at a time on the network.

Scope: Current network, all OpenLNS CT computers

Filter Based on Channel

Ignores service pin messages not originating from the same channel on which the device being installed is located. This option is cleared by default.

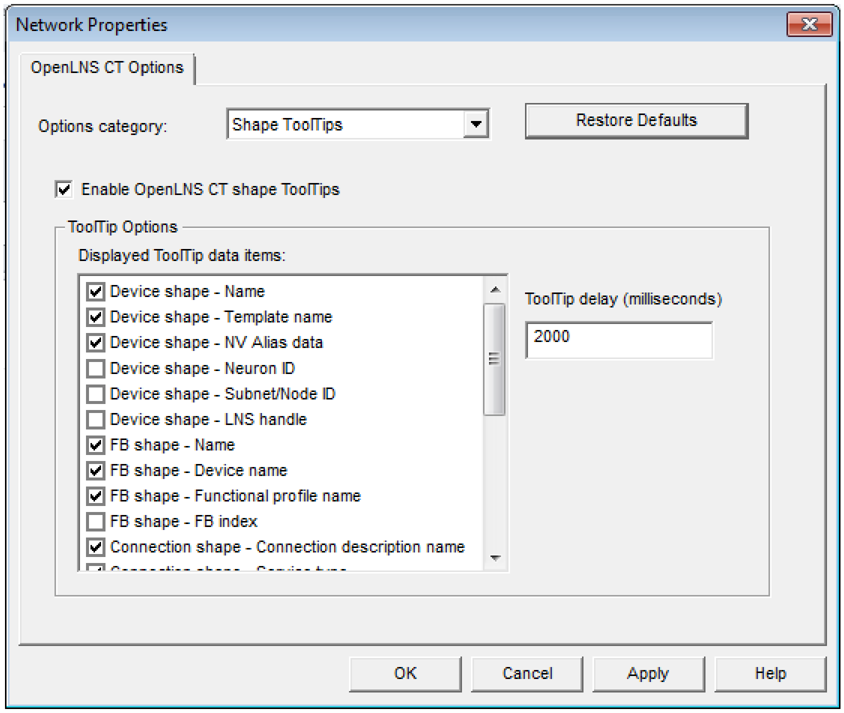
Select this check box if the network does not contain any logical repeaters.

Clear this check box if the network includes any logical repeaters.

Scope: Current network, all OpenLNS CT computers

Shape ToolTips Options

You can enable a ToolTip to appear when the mouse pointer is over an OpenLNS CT SmartShape in your OpenLNS CT drawing and select the type of data that opens in the ToolTip. Using a device for example, you can set a ToolTip to display any or all of the following data items when the mouse pointer is over its SmartShape: SmartShape name, template name, network variable alias data, Neuron ID, Subnet/Node ID, and OpenLNS handle.



Options category: Shape ToolTips

☑ Enable OpenLNS CT shape ToolTips

ToolTip Options

Displayed ToolTip data items:

☑ Device shape - Name
☑ Device shape - Template name
☑ Device shape - NV Alias data
☐ Device shape - Neuron ID
☐ Device shape - Subnet/Node ID
☐ Device shape - LNS handle
☑ FB shape - Name
☑ FB shape - Device name
☑ FB shape - Functional profile name
☐ FB shape - FB index
☑ Connection shape - Connection description name

ToolTip delay (milliseconds): 2000

The SmartShape ToolTips category contains the following options. The changes you make in the SmartShape ToolTips options category are applied to all networks on your computer.

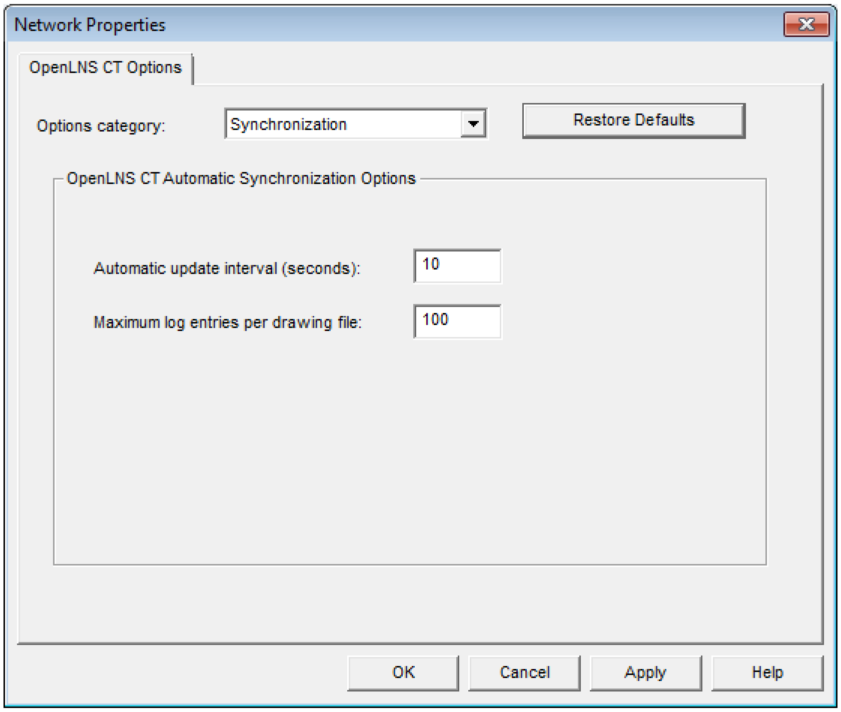
Enable OpenLNS CT Shape ToolTips — Displays a ToolTip when the mouse pointer is positioned over an OpenLNS CT SmartShape. This check box is selected by default.

Scope: All networks, current OpenLNS CT computer

ToolTip Delay — Specifies the number of milliseconds the mouse pointer needs to be positioned over an OpenLNS CT SmartShape for the ToolTip to be displayed. The default value is **2000** ms (2 seconds).

Scope: All networks, current OpenLNS CT computer

Synchronization Options



Options category: Synchronization

OpenLNS CT Automatic Synchronization Options

Automatic update interval (seconds): 10

Maximum log entries per drawing file: 100

The Synchronization category contains the following options.

Automatic Update Interval (Seconds)

Sets how frequently (in seconds) an OpenLNS CT client checks the OpenLNS CT Event Log and updates its copy of an OpenLNS CT drawing to incorporate any changes made by other clients. The default automatic update interval is **10** seconds.

Set the interval to 0 to disable automatic updates. You can then apply updates when you open a drawing, or you can manually apply updates by clicking **Add-ins**, pointing to **Synchronization**, and then clicking **Apply Events**.

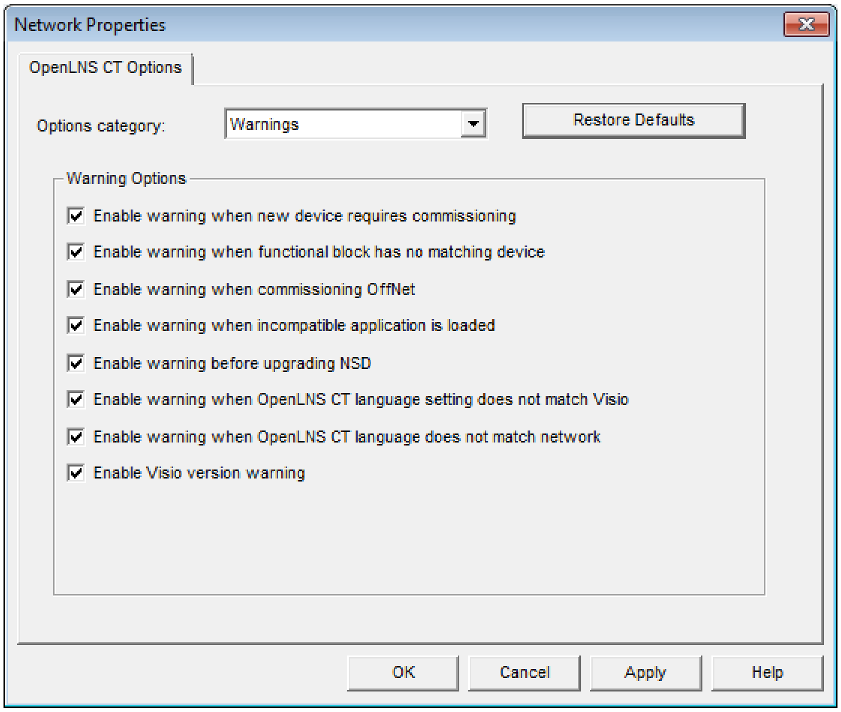
Scope: Current network, current OpenLNS CT computer

Maximum Log Entries per Drawing File

Sets the maximum number of entries to be stored in the OpenLNS CT Event Log. Once the limit is reached, older entries will be deleted to enable new entries to be logged. The default limit is **100** entries. To disable the event log, set the limit to **0**.

Scope: Current network, all OpenLNS CT computers

Warnings Options



Options category: Warnings

Warning Options

☑ Enable warning when new device requires commissioning

☑ Enable warning when functional block has no matching device

☑ Enable warning when commissioning OffNet

☑ Enable warning when incompatible application is loaded

☑ Enable warning before upgrading NSD

☑ Enable warning when OpenLNS CT language setting does not match Visio

☑ Enable warning when OpenLNS CT language does not match network

☑ Enable Visio version warning

The Warnings category contains the following options, which are all selected by default.

Enable Warning When New Device Requires Commissioning

Displays a message when you create a new device that needs to be commissioned.

Scope: Current network, current OpenLNS CT computer

Enable Warning When Functional Block has no Matching Device

Displays a message when you drag a functional block SmartShape onto a drawing but there is no compatible device on the drawing page. You will be prompted to select another subsystem.

Scope: Current network, current OpenLNS CT computer

Enable Warning When Commissioning OffNet

Displays a message when you commission a device or router while OpenLNS CT is OffNet. Commissioning while OpenLNS CT is OffNet sets the Neuron ID of the device SmartShape, but it does not affect the physical device.

Scope: Current network, current OpenLNS CT computer

Enable Warning When Incompatible Application is Loaded

Displays a message if you attempt to load an incompatible application image into a device.

Scope: Current network, current OpenLNS CT computer

Enable Warning Before Upgrading NSD

Displays a message when upgrading the OpenLNS network interface, which is represented by the network service device (NSD) SmartShape in the OpenLNS CT drawing.

OpenLNS CT upgrades the network interface when you upgrade from a version of OpenLNS CT before LonMaker Release 3, or if you are switching between a fast OpenLNS network interface (layer 2) and an OpenLNS network interface (layer 5). If you attempt to do either of these, OpenLNS CT will ask whether to upgrade the NSD. For best performance, perform the upgrade. If you are only temporarily using a different network interface, you may choose not to upgrade. In this case, communication with physical devices is limited until you reopen the network with the original network interface. If initially you choose not to upgrade the NSD and decide later to upgrade, right-click the NSD SmartShape, and then click **Upgrade NSD** on the shortcut menu.

When moving from an OpenLNS fast (layer 2) network interface to an OpenLNS (layer 5) network interface, all connections and monitor sets may not be created on the new network interface. If this happens, a dialog displays the items that will be lost and gives you an opportunity to cancel the upgrade.

Scope: Current network, current OpenLNS CT computer

Enable Warning When OpenLNS CT Language Setting does not Match Visio

Displays a message when the language set in OpenLNS CT and the currently installed language in Visio do not match.

Scope: All networks, current OpenLNS CT computer

Enable Warning When OpenLNS CT Language Setting does not Match Network

Displays a message when the languages set in OpenLNS CT and the OpenLNS CT drawing do not match.

This can occur if the installed languages are different between OpenLNS CT that created the OpenLNS CT drawing and the one opening it.

Scope: Current network, current OpenLNS CT computer

Enable Visio Version Warning

Displays a message if the version of Visio installed on your computer is incompatible with your version of OpenLNS CT.

This applies only to versions of Visio newer than Visio 2010 or Visio 2003. If an older version of Visio is installed on your computer and you do not also have Visio 2010 or Visio 2003 installed, OpenLNS CT will not run on your computer.

Scope: All networks, current OpenLNS CT computer

Appendix B
Glossary

This appendix provides definitions for terms commonly used with OpenLNS CT.

Glossary

Acknowledged Service

A service of the ISO/IEC 14908-1 Control Network Protocol that ensures a message was received by the addressee(s). After a configurable number of retries, failures are logged in a status register in the device, which can be accessed by network management tools.

Actuator

Any component that affects a physical variable of a system under control or indicates the values of system variables for human operators. Some examples are pumps, fans, heaters, alarm annunciators, and operator displays. This term may apply only to the component that converts electrical or pneumatic control signals into a physical force that causes a mechanical component such as a damper or valve to move, or it may apply to both components as a unit.

Ad-Hoc System Scenario

A control network installation scenario in which physical devices are configured and commissioned on-line as network data is entered. The network database is built simultaneously.

Alias Network Variable

An alias network variable behaves like a shadow copy of a network variable. An alias shares the value of the original network variable, but it maintains its own NV selector. In addition, an alias maintains its own address table index if it is an alias for an output network variable or polling input network variable. Alias network variables are used to overcome connection limitations of non-alias network variables.

Application Configuration

A process by which the application program in each physical device is tailored to the desired functionality by selecting the appropriate configuration properties. The OpenLNS network operating system) provides a platform for manufacturers to create easy-to-use graphical configuration interfaces, called *plug-ins*, that are compatible with any other OpenLNS-based network tool.

Application Device

A LONWORKS device that runs a LonTalk Application (OSI Layer 7). The application may run on a Smart Transceiver or Neuron Chip, in which case the device is called a *Neuron-hosted* device.

Application Image

Device firmware that consists of the object code generated by the device developer from the developer's application program and other application-specific parameters, including the following:

- Network variable fixed and self-identification data
- Device interface (XIF) data
- Program ID string
- Optional self-identification and self-documentation data
- Number of address table entries
- Number of domain table entries
- Number and size of network buffers
- Number and size of application buffers
- Number of receive transaction records
- Input clock speed of target Smart Transceiver or Neuron Chip
- Transceiver type and bit rate

Application Program

The software code in a LONWORKS device that defines how it functions. The application program, also referred to as the *application* or the *application layer*, may be in the device when you purchase it, or, for Neuron-hosted devices, you may load it into the device from application files (**.nld**, **.apb**, and **.nxe** extensions) using OpenLNS CT. The application program interfaces with the ISO/IEC 14908-1 protocol stack to communicate over the network. It may reside completely in the Smart Transceiver or Neuron Chip, or it may reside on an attached host processor (in a host-based device).

Applicationless

A state a device enters when it has determined that it does not have a valid application image. Program or hardware failure may cause a device to become applicationless. To restore an applicationless device, you must reload the application image into the device.

Attached

The state of OpenLNS CT when it is connected to a LONWORKS network.

Authentication

An ISO/IEC 14908-1 Control Network Protocol messaging service that ensures that a received message was sent by an authorized source. Authentication is implemented by distributing 48-bit or 96-bit keys to the participating devices when they are installed. For an authenticated message to be accepted by the receiving device, both the sending and receiving device must possess the same key.

The OpenLNS Server supports only a single authentication key for all authenticated devices; therefore, if you use 96-bit authentication keys, all devices that are authenticated must support 96-bit authentication.

Backbone Network

A high-speed network channel connecting several lower speed channels. Common backbone channels include IP-852 and TP/XF-1250 channels.

Backup

A .zip file containing a saved version of an OpenLNS CT drawing and/or OpenLNS network database. Backup files are used to protect against accidental file corruption or hardware failure, or to copy an OpenLNS CT network design from one computer to another.

Binding

Process of connecting network variables and message tags. Binding creates logical connections (virtual wires) between LONWORKS devices. Connections define the data that devices share with one another. Tables containing binding information are stored in the Neuron Chip’s EEPROM, and may be updated by OpenLNS CT.

Bit Rate

The rate in bits at which an ISO/IEC 14908-1 packet frame is transferred across the communication medium.

Bound Update

A connection between a monitored network variable and the OpenLNS CT Browser or an OpenLNS CT host. Bound updates are ideal for monitoring slowly changing network variables.

Changeable Type Network Variable

A network variable on a device that supports network variable type and size modification by a network management tool.

Channel

The physical media used by multiple devices to communicate. The ISO/IEC 14908-1 Control Network Protocol is media independent; therefore, numerous types of media can be used for channels: twisted pair, power line, fiber optics, IP, and RF, and other types.

Channel Segment

A portion of a channel. A single segment can be comprised of multiple sections connected by physical layer repeaters.

Channel Type

Channels are categorized into channel types, which are characterized by the device transceiver. Common channel types include the following: TP/FT-10 (ISO/IEC 14908-3 twisted pair free topology channel), TP/XF-1250 (high-speed twisted pair channel), PL-20 (ISO/IEC 14908-2 power line channel), FO-20 (ANSI/CEA-709.4 fiber optics channel), and IP-852 (ISO/IEC 14908-4 IP-communication).

Client/Server Architecture

In a LONWORKS network, the OpenLNS Server provides a set of directory, installation, management, monitoring, and control services for computers running OpenLNS CT and other OpenLNS tools, applications, and plug-ins (clients). The client/server architecture of OpenLNS allows multiple users running OpenLNS tools, applications, and plug-ins on separate computers to access the OpenLNS Server simultaneously. This means that managers, system integrators, installers, and maintenance personnel can all work on the same LONWORKS network at the same time.

Commissioning

The process in which OpenLNS CT determines and downloads network and application configuration data into a physical device. For devices whose application programs are not contained in ROM, OpenLNS CT can also download the application program into non-volatile memory in the device. Devices are usually either commissioned and tested one at a time, or commissioned and then brought online and tested incrementally.

Configuration Properties (CPs)

Configuration properties define the behavior of an application device by determining the manner in which data is manipulated and when data it is transmitted. Configuration properties can be applied at the device, functional block, or network variable level. Configuration properties determine the functions to be performed on the values stored in network variables. For example, a configuration property may specify a minimum change that must occur on a physical input to a device before the corresponding output network variable is updated.

Configured

A device state where the device has both an application image and a network image. This indicates that the device is ready for network operation.

Connection

The implicit addressing established during binding. A connection links one or more logical outputs (network variables or message tags) to one or more logical inputs. A connection may be represented with a connector SmartShape or a reference connection.

Connection Description

A set of properties, namely the message service type and addressing mode, that apply to a network variable connection.

Connector Shape

A single OpenLNS CT connector used to connect a pair of network variables (or message tags) within the same subsystem.

Controller

A high-level component that takes one or more sensor values and selects an appropriate setting on one or more actuators to achieve a desired function.

Control Network Protocol (CNP)

A communication protocol standard for control networks defined by the ISO/IEC 14908-1 standard. The CNP provides a standard method for devices on a LONWORKS network to exchange data. The CNP defines the format of the messages being transmitted between devices, and it defines the actions expected when one device sends a message to another. The protocol normally takes the form of embedded software or firmware code in each device on the network.

Custom Master Shape

A configured OpenLNS CT SmartShape or group of OpenLNS CT SmartShapes saved to a stencil. Custom master SmartShapes enable common devices and functions to be re-used easily. Custom master SmartShapes are used to speed up network design, as they already contain configuration data when they are added to an OpenLNS CT drawing. A custom device master SmartShape, for example, may contain the device template name and configuration property settings.

Data Point

A network variable, configuration property, or functional block state (enabled or in override) that OpenLNS CT can monitor and/or control.

Decommissioning

Process in which devices are temporarily uninstalled. Decommissioning devices is useful when optimizing, troubleshooting, or repairing a network. All the configuration data of a decommissioned device, including configuration properties and connections, are preserved in the OpenLNS network database. OpenLNS CT can also preserve the Neuron ID of a decommissioned device to speed up the recommissioning process.

Destination Address

The logical address contained in every LonTalk packet of the device or group of devices designated to receive the packet. The destination address can be the unique Neuron ID, the logical device address, a group address, or a broadcast address.

Device

A device that communicates on a LONWORKS network. A device may be an application device, network service device, or a router. Devices are sometimes referred to as *nodes* in LONWORKS documentation.

Device Interface (XIF)

The logical interface to a device. A device interface specifies the number and types of functional blocks; number, types, directions, and connection attributes of network variables; and the number of message tags. The program ID field is used as the key to identify each device interface. Each program ID uniquely defines the static portion of the interface. However, two devices with identical static portions may differ if dynamic network variables are added or removed, or if the types of changeable network variables are changed. Thus it is possible to have devices with the same program ID but different device interfaces.

Device Interface (XIF) File

A file that documents a device's interface with a network. The file can be a text file (**.XIF** extension), or it can be a binary file (**.XFB** extension).

Device Discovery

Commissioning process in which OpenLNS CT searches the network for unconfigured devices and matches them to device SmartShapes defined in the OpenLNS CT drawing.

Device-Specific Configuration Property

A configuration property that has values that can be modified independent of the network database. Changes made to a device-specific configuration property are not updated in the network database.

Device Template

A device template contains all the attributes of a given device type, such as its functional blocks, network variables, and configuration properties. You can create a device template by importing a device interface (XIF) file supplied by the device manufacturer, or by uploading the device interface definition from the physical device. A device template is identified by its name and its program ID. Both must be unique within a network—you cannot have two device templates with the same name or the same program ID in a single network.

Device Upgrade

The process in which a new application (**.nxe** or **.apb** extensions) is loaded into a Neuron-hosted device.

Disable

For a device, a device state in which the device application does not run even if the device is reset. For a functional block, a management command that causes output network variables to send their configured default values.

Domain

A logical collection of devices on one or more channels. Communications can only take place among devices configured in the same domain.

Domain ID

The top level of the LonTalk addressing hierarchy of domain/subnet/node. The domain ID can be 0, 1, 3, or 6 bytes long. When using the OpenLNS Server, the zero-length domain is reserved for the use of the OpenLNS Server and cannot be used as the system's domain.

Download

An installation process in which data, such as the application program, network configuration, and/or application configuration, is transferred over the network from the OpenLNS network database into a device.

Dynamic Data Exchange (DDE)

A standard protocol built into the Macintosh, Windows, and OS/2 operating systems. DDE enables two running applications to share the same data. For example, DDE makes it possible to insert a spreadsheet chart into a document created with a word processor. Whenever the spreadsheet data changes, the chart in the document changes accordingly. When applications share information with each other using DDE, they are said to be holding a DDE conversation. Each conversation has a well-defined beginning, middle, and end. To begin a conversation, one application, known as the client or destination application asks another application, known as the server or source application to open a communications channel. Once a conversation is established, the client can send and receive data from the server on the DDE channel. For example, an Excel spreadsheet (the client or destination) may ask the LNS DDE Server (the server or source) for the current fuel consumption from a flow sensor for use in an automated billing system. Alternately, an InTouch operator interface (the client or destination) may tell the LNS DDE Server (the server, or source) to change the state of a valve in a LONWORKS network in response to a request issued by the user interface.

Dynamic Functional Block

A functional block that is not pre-loaded on a device. Dynamic functional blocks are typically used by controllers that do not have a static interface. These controllers typically require the integrator to run a plug-in or other configuration software that defines the control algorithm and configures the controller. The controller manufacturer can use dynamic functional blocks to enable an OpenLNS CT user to add functional blocks to the controller using OpenLNS CT.

Dynamic Network Variable

A network variable that is added to a functional block after the device has been commissioned. Devices that support dynamic network variables include controllers and gateways with dynamic interfaces.

Engineered System Scenario

A control network installation scenario in which configuration data is collected into the network database when the system is being designed with OpenLNS CT and downloaded into the physical devices when the network is installed.

Enumerated Object

A network variable, configuration property, or functional block state that has enumerated types. Examples of enumerated objects include **SNVT_hvac_mode**, **SCPT_hvac_type**, and functional block states enable and override.

Fan-in Connection

A connection in which multiple output network variables (or message tags) are connected to a single input network variable (or message tag).

Fan-Out Connection

A multicast connection in which multiple input network variables (or message tags) are connected to a single output network variable (or message tag).

Free Topology

A connection scheme for a communication bus that removes traditional transmission line restrictions of trunks and drops of specified lengths and at specified distances, and terminations at both ends. Free topology allows wire to be strung from any point to any other, in bus, daisy chained, star, ring, or loop topologies, or combinations thereof. It only requires one termination anywhere in the network. This can reduce the cost of wiring by a factor of two or more.

Functional Block (FB)

A collection of network variables, configuration properties, and associated behavior that defines a desired system functionality. Functional blocks define standard formats and semantics for how information is exchanged between devices on a network.

Functional Block State

An indication whether a functional block is enabled or disabled, or whether it is in override mode.

Functional Profile

A specification that enables equipment specifiers to select the functionality they need for a system. A functional profile is a template for a type of functional block that defines mandatory and optional network variable and configuration property members along with their intended usage. A small number of standard functional profiles are available for generic devices such as simple sensor and actuators. Many industry-specific standard functional profiles are available for industry-specific applications. Industry-specific standard profiles are developed through a review and approval process, including a cross-functional review to ensure the profile will interoperate within an individual subsystem and also provide interoperability with other subsystems in the network. Device developers can also create custom manufacturer-specific profiles.

Group

A logical collection of devices within a domain. Unlike a subnet, devices are grouped together without regard for their physical location in the domain. The number of groups to which a device may belong is determined by the number of available address table entries. This number is set by the device application. Groups and group membership are defined by the OpenLNS Server during binding.

Group Addressing

A logical addressing mode in the ISO/IEC 14908-1 Control Network Protocol that allows a message to be sent simultaneously to a pre-configured group of devices. Each group has an 8-bit group ID. Each domain can have up to 256 groups defined.

Host

A device implementing layer 7 of the ISO/IEC 14908-1 Control Network Protocol. A host may be based on a Smart Transceiver or Neuron Chip, in which case it is called a *Neuron hosted* device. A host may be based on another processor, in which case it is called a *host-based* device. A host-based device uses a Smart Transceiver or Neuron Chip as a network interface to talk to the LONWORKS network.

Host Processor

The application processor for a host-based device. The host processor is typically a microcontroller, microprocessor, or computer.

Host-Based Device

A device in which the device application program runs on a processor other than a Smart Transceiver or Neuron Chip.

Hot Backup

A shadow copy of the OpenLNS network database made while one or more OpenLNS applications are accessing it.

Hub

The center of a connection, which is specified by the device handle and network variable index (or message tag index). Each connection is defined in terms of a hub and a set of targets. The hub must be either the only input or the only output network variable (or message tag) in the connection. For example, if the hub is an output network variable, all the other members in the connection must be input network variables.

Human-Machine Interface (HMI)

A network tool that provides services for human monitoring and supervisory control of systems. The HMI usually runs on a workstation or desktop/portable computer.

IP

Acronym for Internet protocol, the basic programming foundation that carries computer messages around the globe on the Internet. Sometimes called TCP/IP (Transmission Control Protocol/Internet Protocol), even though TCP is a protocol that runs on top if IP, and IP supports other protocols besides TCP.

IP-852 Channel

An IP-852 channel carries ISO/IEC 14908-1 packets enveloped in ISO/IEC 14908-4 packets. An IP-852 channel is a LONWORKS channel that uses a shared IP network to connect IP-852 devices and is defined by a group of IP addresses. These IP addresses form virtual wires that connect IP-852 devices so they can communicate with each other. IP-852 devices may include the OpenLNS Server computers, OpenLNS CT computers, IP-852 routers, and native IP-852 devices. An IP-852 channel enables a remote full client to connect directly to a LONWORKS network and perform monitoring and control tasks.

IP-852 Configuration Server

The IP-852 Configuration Server maintains an IP routing table database required for IP-852 channels. It is used to create, manage, and configure IP-852 channels and IP-852 devices.

IP-852 Network Interface

Software that enables IP-852 devices such as OpenLNS Server computers, OpenLNS CT computers, IP-852 routers, and native IP-852 devices to be attached to an IP-852 channel. An IP-852 network interface requires that the IP-852 Configuration Server be configured before trying to communicate with remote devices or remote computers. IP-852 routers include the SmartServer, i.LON 100 Internet Server, and i.LON 600 IP-852 Router.

IP-852 Router

An IP-852 router forwards ISO/IEC 14908-1 packets enveloped in ISO/IEC 14908-4 packets over an IP-852 channel. IP-852 routers include the SmartServer, i.LON 100 Internet Server, and i.LON 600 IP-852 Router.

LAN

Local area network. A LAN is a communications network that links a number of different workstations in the same area. The local area may be defined as the same building or campus of buildings. Using the LAN, individual workstations or computers can send messages and files to each other and to shared devices, such as printers, disk storage, and other computer systems. LAN performance is measured in the amount of data that can be transmitted and received, usually expressed as megabits or gigabits transmitted per second, so its critical factor is speed.

Local Client

An OpenLNS CT computer that is also running the OpenLNS Server.

LonScanner Protocol Analyzer

An ISO/IEC 14908-1 protocol analyzer for field use that provides LONWORKS manufacturers, system integrators, and end-users with a rich set of Windows-based tools to allow users to observe, analyze, and diagnose the behavior of installed LONWORKS networks. The LonScanner Protocol Analyzer includes three tools for network analysis and monitoring:

- Packet monitor tool
- Network traffic statistics tool
- Network diagnostics tool

LonMark International

A global membership organization created to promote and advance the business of efficient and effective integration of open, multi-vendor control systems utilizing ISO/IEC 14908-1 and related standards.

LonTalk File Transfer Protocol

An interoperable way for devices to share data files. The file types— 0, 1, and 2—are defined by the LonMark program for specifying configuration parameters.

LONWORKS Technology

The technology that allows for the creation of open, interoperable control networks that communicate with the Control Network Protocol. LONWORKS technology consists of the tools and components required to build intelligent devices and to install them in control networks.

Message Tag

Logical input and output ports that devices use to send and receive explicit messages. Message tags are used for standard data transfer mechanisms such as the LONWORKS file transfer protocol and the LONWORKS data log transfer protocol. In addition, they are used as a proprietary interface to devices such as factory calibration interfaces. Message tags can only be added to the virtual functional block of a device that supports message tags. A device always contains a **msg_in** tag and may contain declared message tags as well. Declared message tags are bi-directional (the device can both send and receive messages with them). The **msg_in** tag can only be used to receive messages. Generally, devices use network variables to communicate with one another since they are interoperable and produce more efficient code. Message tags are similar to network variables except that they do not have pre-defined types.

Monitor Set

A group of data points that OpenLNS CT uses to monitor and control network variables or message tags.

Monitored Connection

A connector SmartShape or reference connection on which network variable values are displayed and updated.

Network Configuration

The process of converting a network design into the data elements required by the Control Network Protocol in each device of the network. This includes:

- Assigning domain ID and logical addresses to all devices and groups of devices.
- Binding network variables to create logical connections between devices.
- Configuring the various Control Network Protocol parameters in each device for the desired features and performance, including channel bit rate, acknowledgement, authentication, priority service, etc.

Network Image

A device's network address and connection information. For Neuron Chip hosted devices, the device's network image is stored in EEPROM on the Neuron Chip. For host-based devices, the entire network image except the network variable configuration table is stored in EEPROM on the Neuron Chip.

Network Interface

A LONWORKS device that provides a layer 6 LonTalk interface to an external host computer such as a computer or a handheld maintenance tool. Network interfaces include IP-852 interfaces (SmartServer, i.LON 100 *e*3 Internet Server, and i.LON 600 IP-852 Router); U10/U20 USB network interfaces; and PCC-10 and PCLTA-10, 20, and 21 PCI network interface cards.

Network Service Device (NSD)

A device representing a local or remote client that connects the client to the LONWORKS network.

Network Services Interface (NSI)

The component in the OpenLNS architecture that provides the physical connection to the LONWORKS network, manages transactions with the NSS, and provides transparent remote access to the NSS.

Network Services Server (NSS)

The component in the OpenLNS architecture that processes network services, maintains the network database, and enables and coordinates multiple points of access.

Network Variable (NV)

Network variables allow a device to send and receive data over the network to and from other devices. Network variables are data items (such as temperature, the state of a switch, or actuator position setting) that a particular device application program expects to receive from other devices on the network (an *input network variable*) or expects to make available to other devices on the network (an *output network variable*).

Network Variable/Configuration Property Format

The network variable or configuration property format specifies how the raw data contained within a network variable or configuration property is translated for display or for use by an application. The format also specifies how data entered by a user or application is translated to the raw data to be transmitted on a LONWORKS network.

Network Variable/Configuration Property Types

A network variable or configuration property type defines the structure and contents of the object. A network variable type can be either a standard network variable type (SNVT) or a user-defined network variable type (UNVT). A configuration property type can be a standard configuration property type (SCPT) or a user-defined configuration property type (UCPT)

Neuron Chip

A family of VLSI components that implements the Control Network Protocol. The Neuron Chip can manage I/O devices and execute user-written application code, or alternatively it can be used to create a network interface to a host processor.

Neuron Firmware

Firmware required to implement the ISO/IEC 14908-1 Control Network Protocol, I/O drivers, and other system functions on a Smart Transceiver or Neuron Chip. This firmware is contained in the **VER***xxx* subdirectories of the LONWORKS **images** folder in files with the name **S*.NX***. Other custom firmware images are available for special purposes. Use of the Neuron Chip firmware is subject to terms and conditions defined in the NodeBuilder and Mini software license agreements and the LONWORKS OEM license agreement.

Neuron ID

A 48-bit number assigned to each Neuron Chip at manufacture time. Each Neuron Chip has a unique Neuron ID, making it like a serial number.

Network Explorer

An auto-hide window that displays a hierarchical view of your OpenLNS CT network and supports all of the commands available within the drawing pages. The Network Explorer can be used in an OpenLNS CT drawing as a navigational tool or as a network management tool.

Network Merge

Process in which one OpenLNS CT network design is merged into another existing network design.

Network Navigator

A dialog displaying a hierarchal view of the subsystems in a network that can be used to expand, select, and deselect items when viewing or configuring an OpenLNS CT network.

Network Recovery

Process in which a new OpenLNS CT drawing and OpenLNS network database are created from an existing LONWORKS network. Network recovery is useful for networks that were installed with legacy tools not based on OpenLNS, self-installed networks, and installed networks in which the OpenLNS network database has been lost

Node Address

A unique 15-bit logical identifier for each device in a domain. The node address consists of two parts: a 7-bit subnet address and an 8-bit Node ID. The Node ID is unique within the subnet.

Node ID

The third part of the LonTalk addressing hierarchy of domain/subnet/node. At installation time, each device is assigned a unique node ID within its subnet by the OpenLNS Server.

Node Object

A functional block that monitors the status of all functional blocks in a device and makes the status information available for monitoring by OpenLNS CT. A LONMARK-compliant device that has more than one functional block must have a Node object.

NodeBuilder Tool

A hardware and software platform used to develop applications for Smart Transceivers and Neuron Chips.

Non-Volatile RAM

Read/write processor memory that does not lose its contents when electrical power is removed. Two widely used semiconductor technologies are EEPROM (electrically erasable programmable read-only memory) and flash memory.

NV Selector

A 14-bit number used to identify connected network variables. When placing the network variable in a connection, the OpenLNS Server connection binder assigns a value representing that connection. All network variables in a given connection use the same network variable selector. The OpenLNS Server shares a network variable selector among connections if the connections share one or more network variables.

OffNet

A network management mode in which network configuration changes are stored in the OpenLNS network database, but not propagated to the devices on the network. To send the changes to the devices, you place OpenLNS CT OnNet. If OpenLNS CT is OffNet and attached to the network, you can still read and write network variables, and read configuration properties.

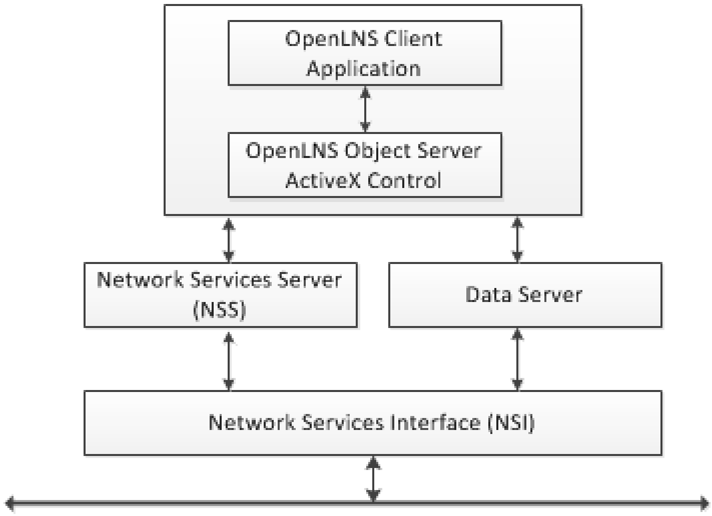
OnNet

A network management mode in which network configuration changes are propagated immediately to the devices on the network.

OpenLNS

A network operating system that provides services for interoperable LONWORKS installation, maintenance, monitoring, and control tools such as the OpenLNS Commissioning Tool. Using the services provided by the OpenLNS Server client/server architecture, tools from multiple vendors can work together to install, maintain, monitor, and control LONWORKS networks. The OpenLNS architecture consists of the following elements:

1. An OpenLNS client application, which is a software application that uses OpenLNS services to install, monitor, and control devices on a LONWORKS network. Multiple OpenLNS client applications can share the services of a single OpenLNS Server.
2. The OpenLNS Object Server ActiveX Control, which is a language-independent programming interface to access an OpenLNS Server.
3. The Network Services Server (NSS), which maintains an image of the network.
4. The Data Server, which provides services for monitoring and control.
5. The Network Services Interface (NSI), which is the physical interface to the network.


OpenLNS Client Application
OpenLNS Object Server ActiveX Control
Network Services Server (NSS)
Data Server
Network Services Interface (NSI)

OpenLNS Event

An action fired by an OpenLNS Server to notify an OpenLNS client application of a change to the OpenLNS network. OpenLNS CT uses OpenLNS events to keep it synchronized with other OpenLNS applications or plug-ins that are making changes to the network database.

OpenLNS Network Database

A database (also referred to as the network database), which includes the network and device configuration data for the devices in a LONWORKS network. The network database also contains extension records, which are application-defined records for storing application data.

OpenLNS Server

A software platform for creating an open control system that supports multiple applications and service providers by supplying the essential network services required for installing, operating, and maintaining LONWORKS control networks.

OpenLNS CT Browser

An OpenLNS plug-in that provides a table view of the network variables and configuration properties of selected devices and/or functional blocks. The OpenLNS CT Brower can be used to monitor and control the network variables and configuration properties in a network.

OpenLNS CT Device Manager

An OpenLNS dialog that lets you test, wink, disable, change the application state, and override application devices, functional blocks, and routers.

OpenLNS CT Drawing

An OpenLNS CT drawing contains the graphical representation of a LONWORKS network.

OpenLNS CT Event Log

A shared log in which OpenLNS CT stores every change made to an OpenLNS CT drawing by every OpenLNS CT user on the network. All OpenLNS CTs check the log when they open an OpenLNS CT drawing and periodically thereafter, and update their copies of an OpenLNS CT drawing based on the log entries. The OpenLNS CT Event Log enables multiple OpenLNS CTs to stay synchronized with the OpenLNS database and with each other, and it reduces instances in which an OpenLNS CT drawing needs to be manually resynchronized.

OpenLNS Commissioning Tool (CT)

An OpenLNS network tool that uses Visio as its graphical user interface. OpenLNS CT is used to design, commission, maintain, and document distributed control networks comprised of both LONMARK and other LONWORKS devices.

OpenLNS CT Layers

Separate views for the physical, logical, and data components in an OpenLNS CT network that can be used to hide and show parts of an OpenLNS CT drawing. The physical layer includes devices, routers, channels; the logical layer contains the functional blocks, network variables, and unmonitored connections; and the data layer consists of monitored connections. Subsystems and supernodes belong to both the physical and logical layers.

OpenLNS CT Network Design

An OpenLNS network database and an associated OpenLNS CT drawing.

OpenLNS CT Shape

A reusable drawing object related specifically to a LONWORKS device, connection, channel, or subsystem.

OpenLNS CT Style

A SmartShape style indicating the state of a device, functional block, connection, or network variable/message tag.

OpenLNS Plug-in

An OpenLNS application that implements the OpenLNS Plug-in API. OpenLNS plug-ins provide a standard way to extend and customize the functionality of OpenLNS applications and tools such as the OpenLNS Commissioning Tool. For example, OpenLNS plug-ins allow device manufacturers to provide customized software that simplifies the configuration, monitoring, or control of their devices that work seamlessly with the OpenLNS Commissioning Tool and other OpenLNS tools. OpenLNS plug-ins that are specific to a particular type of device are sometimes referred to as *device plug-ins*. OpenLNS plug-ins can also add new functionality to OpenLNS applications such as alarming, logging, and trending. The OpenLNS Plug-in API is compatible with the LNS Plug-in API, and LNS plug-ins are typically compatible with OpenLNS tools and application that support plug-ins.

Override

A management command used on a functional block that causes output network variables to send their configured override values. The override values are set using the configuration properties in the functional block containing the network variables. The override values are sent even if part of the network is not yet operating or there is no input to the functional block.

Passive Configuration Mode

A mode in which changes to configuration property values are manually propagated to one or more devices without affecting the network configuration of the devices. Passive configuration mode enables a plug-in to be used on devices installed by network tools not based on OpenLNS.

PCC-10

A type II PC (formerly PCMCIA) network interface card network that includes an integral TP/FT-10 transceiver. Other transceiver types can be connected to the PCC-10 via external transceiver pods.

PCLTA-10/20

A ½ size ISA network interface card.

Peer-To-Peer

A control strategy in which independent intelligent devices share information directly with each other and make their own control decisions without the need or delay of using an intermediate, central, or master controller. Because of the enhanced system reliability introduced by eliminating the master (a single point of failure) and the reduced installation and configuration cost inherent in peer-to-peer designs, LONWORKS technology supports but does not require a peer-to-peer control strategy.

Physical Medium

A communication environment that carries the modulated signals from sources to destinations in a network. LONWORKS technology supports many media types, including twisted pair, power line, fiber optic cable, radio frequencies, infrared, and coax.

Priority

An ISO/IEC 14908-1 Control Network Protocol service for messages sent from LONWORKS devices that allow a message priority access to a channel. Priority traffic is delivered before other traffic. Priority is a resource of the channel.

Protocol Analyzer

A tool that can read every packet on a LONWORKS channel. A protocol analyzer is different from a device containing the complete ISO/IEC 14908-1 Control Network Protocol stack in that it can receive every packet on the network, not just packets that are addressed to it. Protocol analyzers allow users to observe, analyze, and diagnose the behavior of installed LONWORKS networks. The LonScanner Protocol Analyzer includes the following tools for network analysis and diagnosis:

- Packet monitor tool
- Network traffic statistics tool
- Network diagnostics tool

Recommissioning

Process in which OpenLNS CT downloads network and application configuration data into a physical device that was previously commissioned with OpenLNS CT.

Reference Connection

An OpenLNS CT connection represented by a pair of identically named SmartShapes attached to both ends of the connection. Reference connection SmartShapes can be used to simplify an OpenLNS CT drawing and connect functional blocks and supernodes in different subsystems.

Remote Full Client

An OpenLNS CT computer that communicates with the OpenLNS Server running on a separate computer over a LONWORKS channel (an IP-852 or native LONWORKS channel). On a remote full client, network management tasks are routed through the OpenLNS Server, but monitoring and controlling tasks are sent directly to the network without going through the OpenLNS Server.

Remote Lightweight Client

An OpenLNS CT computer that communicates with the OpenLNS Server running on a separate computer using an OpenLNS/IP interface. On a remote lightweight client, network management, monitoring, and controlling tasks are all routed through the OpenLNS Server.

Repeated Service

A service of the ISO/IEC 14908-1 Control Network Protocol with which a message is sent more than once to help ensures message delivery to the addressee(s). The message is sent with a configurable number of retries. There is no acknowledgement from the addressee(s) that the message has been delivered.

Reset

Network management command that stops a device application, terminates all incoming and outgoing messages, sets all temporary settings to their initial values, and then restarts the device application. If a device was offline, it is placed online.

Resource File

A file included with a LONWORKS device that defines the components of the device interface to be used by OpenLNS CT. Defined components include functional blocks, network variables, and configuration properties implemented by the device application. Resource files allow for the correct interpretation and formatting of the data, and they are necessary for LONMARK certification of a device.

Replacement

Process in which a failed or obsolete device is replaced on a network and in an OpenLNS CT drawing with a new or upgraded device. OpenLNS CT preserves the device's functional blocks, network variables, configuration properties, and connections and transfers them into the new device.

Router

Routers manage network message traffic, extend the physical size of a channel (both length and number of devices attached), and connect different channel types. Unlike other devices, routers are always attached to two or more channels. Each router side can be addressed by its Neuron ID or by a subnet/node address. The side of the router closest to the OpenLNS Server is referred to as the near side; the other side of the router is referred to as the far side. Each router side can receive a packet, decide whether the packet needs to be transmitted, and transmit the packet on the other side's channel, if required. A router can be use in one of the following configurations:

- Repeater – all packets are forwarded
- Permanent Repeater – all packets are forwarded. Subnets can span permanent repeaters.
- Bridge – all packets in a given domain are forwarded
- Permanent Bridge – all packets in a given domain are forwarded. Subnets can span permanent bridges.
- Learning Router – packets are routed only for a given domain. The router starts as a bridge and reduces forwarding as it learns the topology. Learning routers are vulnerable to failures if configured devices are incorrectly moved within the topology.
- Configured Router – packets are routed only for a given domain. Configured routers forward packets based on tables that are configured by an OpenLNS Server. This is the most reliable and efficient form of router.

Self-Documentation

A mechanism that a device can use to provide descriptive information about itself. Self-documentation can be provided for the device's program and network variables. A provider of user-defined services and events may also support self-documentation for itself, its services, events, objects, and properties.

Sensor

Any component that is used to determine the condition or value of a physical system variable, or to accept commands from a human operator.

SLTA-10

A serial network interface with built-in twisted pair transceiver that connects to any host with an EIA-232 (formerly RS232) port. It can also connect to the host remotely using a modem.

Service Pin

An input/output pin on a Smart Transceiver or Neuron Chip that is used during installation to acquire the Neuron ID of the Smart Transceiver or Neuron Chip. When this pin is activated, the Smart Transceiver Neuron Chip sends a broadcast message containing its Neuron ID and program ID, which is called *service pin message*. The method used to activate the service pin varies from device to device. Examples of mechanical methods include activation via a push button or using a magnetic reed switch. By attaching one of the device's I/O pins to the service pin, the service pin can also be put under software control as long as the device is configured. For example, the device can ground the pin when the device is moved or when a predefined series of I/O occurs. The service pin can also be used to drive an LED that indicates the Smart Transceiver's or Neuron Chip's state. The service LED is solid on when the Smart Transceiver or Neuron Chip is applicationless, blinks slowly when the Smart Transceiver or Neuron Chip has an application and is unconfigured, is off when the Smart Transceiver or Neuron Chip has an application and is configured, and blinks once quickly each time the Smart Transceiver or Neuron Chip is reset.

Smart Connection Description (SCD)

A pre-defined connection description that OpenLNS CT automatically assigns to a connection being created or changed based on the number of target network variables, whether the target network variables are located on the same or different subnet, and the availability of network variable aliases.

Source Address

The logical node address of the transmitting device contained in every packet transmitted over a LONWORKS network.

Standard Configuration Property Type (SCPT)

A standard set of configuration property types defined by the LONMARK Association to facilitate interoperability. SCPTs are defined for a wide range of configuration properties used in many kinds of functional profiles, such as hysteresis bands, default values, minimum and maximum limits, gain settings, and delay times. SCPTs should be used in a LONWORKS network wherever applicable. In situations where there is not an appropriate SCPT available, manufacturers may define UCPTs for configuring their devices. See the Echelon or LONMARK website for a current list and documentation.

Standard Network Variable Type (SNVT)

A standard set of network variable types defined by the LONMARK Association to facilitate interoperability by providing a well-defined interface for communication between devices made by different manufacturers. SNVTs should be used in a LONWORKS network wherever applicable. In situations where there is not an appropriate SNVT available, manufacturers may define UNVTs for communicating with their devices. See the Echelon or LONMARK website for a current list and documentation.

Stencil

A collection of master SmartShapes that can be reused in Visio.

Structured Object

A network variable or configuration property that has a type with multiple fields. Examples of structured objects include **SCPTmaxSend Time** (has fields for day, hour, minute, second, and millisecond in the following format: **0 0:00:00.000**) and **SNVT_switch_2** (has fields for state, setting, and scene number in the following format: <*state*> **0.0 0**).

Subnet

A logical collection of up to 127 devices on the same channel within a domain. Up to 255 subnets can be defined within a single domain. Subnets cannot cross configured or learning routers.

Subsystem

A collection of application devices, routers, and functional blocks. With OpenLNS CT, each subsystem corresponds to one page within an OpenLNS CT drawing. Subsystems allow you to place application devices, routers, and functional blocks onto separate pages for organizational purposes. You can place a subsystem within another subsystem, allowing you to create a subsystem hierarchy for large networks. For example, a network may consist of HVAC, lighting, security, and operator subsystems. These may be further divided into subsystems for each floor, and each floor divided into subsystems for each room.

Supernode

A subsystem with a network variable interface. Supernodes enable a group of devices to be organized into logical units with selected network variables from the devices within the supernodes. This allows complex subsystem details to be hidden, reduces errors, and decreases the time required for network engineering and commissioning. A supernode's network variable interface may contain any network variable on any device found within the supernode or its nested subsystems.

Synchronized

The state of an OpenLNS CT network design when an OpenLNS CT drawing and the physical devices on the network match the OpenLNS network database.

Target

The destination of a connection. Each connection is defined in terms of a hub and a one or more targets connected to the hub. For network variable connections, the hub may only be a single input or output network variable in the connection. For example, if the hub is an output network variable, all the targets in the connection must be input network variables.

Terminator

Provides electrical termination for twisted pair channels. Properly installed terminators prevent signal reflections and improve communication reliability.

Test

A management command used to ensure that a device or functional block is operating and is configured correctly.

TP/FT-10

The free topology twisted pair LONWORKS channel type, 78kbps bit rate.

TP/XF-1250

A bus twisted pair LONWORKS channel type, 1.25Mbps bit rate.

TP/XF-78

A bus twisted pair LONWORKS channel type, 78kbps bit rate.

Transceiver

A device that is both a transmitter and a receiver for a communications channel. In a LONWORKS network, it is the component that physically connects a Smart Transceiver or Neuron Chip to its channel. The transceiver implements layer 1 of the ISO/IEC 14908-1 Control Network Protocol.

Unacknowledged Service

A service of the ISO/IEC 14908-1 Control Network Protocol where each message is sent once to the addressee(s). There is no acknowledgement from the addressee(s) that the message has been delivered.

Unattached

The state of OpenLNS CT when it is not connected to a LONWORKS network.

Unconfigured

A device state where the device has an application image, but no network image. The device must be configured before it can operate on the network.

User-defined Configuration Property Type (UCPT)

A non-standard data point used for configuration of the application in a LONWORKS device. UCPTs should be used only when there is no appropriate standard configuration property type (SCPT) defined. LONMARK-certified devices must have UCPTs documented in resource files according to a standard format, in order to allow the devices to be configured without the need for proprietary configuration tools.

User-defined Network Variable Type (UNVT)

A non-standard network variable type defined by the manufacturer of a LONWORKS device. UNVTs should be used only when there is no appropriate standard network variable type (SNVT) defined. LONMARK-certified devices must have UNVTs documented in resource files according to a standard format, in order to allow the devices to be interoperable.

Virtual Functional Block

A static functional block that that contains the network inputs and outputs for a device that are not part of other functional blocks on the device.

Wink

A management command used to identify a device on the network and verify that is communicating. Winking a device causes it to generate application-dependant audio or visual feedback such as a beep or a flashing service LED.

Appendix C
OpenLNS Software License Agreement

This appendix provides a copy of OpenLNS CT Software License Agreement.

OpenLNS CT Software License Agreement

Echelon OpenLNS Commissioning Tool (CT) Software License Agreement

I. NOTICE

This OpenLNS Commissioning Tool (CT) Software License Agreement (the "Agreement") is enterered between you ("You" or "Your") and Echelon Corporation ("Echelon"). YOU MUST READ AND AGREE TO THE TERMS OF THIS AGREEMENT BEFORE DOWNLOADING, INSTALLING, OR USING THE ECHELON OPENLNS COMMISSIONING TOOL SOFTWARE (THE "LICENSED SOFTWARE"). BY CLICKING THE "ACCEPT" BUTTON OF THIS AGREEMENT, OR DOWNLOADING THE LICENSED SOFTWARE, OR INSTALLING THE LICENSED SOFTWARE, OR USING THE LICENSED SOFTWARE, YOU HEREBY AGREE TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE WITH THE TERMS AND CONDITIONS OF THIS AGREEMENT, PLEASE EXIT THIS PAGE AND DO NOT DOWNLOAD OR INSTALL OR USE ANY LICENSED SOFTWARE. BY DOING SO YOU FOREGO ANY IMPLIED OR STATED RIGHTS TO DOWNLOAD OR INSTALL OR USE LICENSED SOFTWARE.

II. LICENSE GRANT

In consideration of Your agreement to the terms of this Agreement, Echelon grants You a non-exclusive, non-transferable, non-sublicensable license to use the Licensed Software during the Use Period and any updates or upgrades thereto provided by Echelon during the Maintenance Period according to the terms set forth below (the “License”).

III. DEFINITIONS

For purposes of this Agreement, the following terms shall have the following meanings:

- "Activate" and "Activation" mean the process of generating a Valid License for a Licensed Device. During Activation for your Licensed Device the Licensed Software will send information about the Licensed Software and the Licensed Device to Echelon. By using the Licensed Software you consent to the transmission of this information to Echelon.
- "Activation Key" means a unique identifier used to identify one or more Valid Licenses. The Activation Key is free for Trial Editions of the Licensed Software and must be purchased for all other editions.
- "Documentation" means the documentation included with the Licensed Software.
- "Licensed Device" means a computer, hardware partition, blade, or virtual machine with a Valid License for the Licensed Software.
- "Licensed Software" means the Echelon OpenLNS Commissioning Tool (CT), Echelon OpenLNS Server, and associated media, printed materials, and online or electronic documentation. The Licensed Software includes the Microsoft Visio software included with some editions of the Licensed Software.
- "Maintenance Period" means a time period during which you can install updates and upgrades for the Licensed Software at no additional charge. The initial Maintenance Period starts upon activation and continues for the Trial Period after activation for Trial Editions of the Licensed Software and starts upon purchase of an Activation Key and continues for one (1) year from time of purchase of an Activation Key for the Licensed Software for all other editions. Echelon may release updates that can be installed outside the Maintenance Period. The availability of free upgrades does not apply to the Visio software included with some editions of the Licensed Software.

- "Trial Period" is a 60-day period during which the Licensed Software can be used with limited functionality for a maximum of two networks, with a maximum of five devices per network, and a maximum run time of 60 minutes.
- "Use Period" means a time period during which you can use the Licensed Software. The Use Period is limited to the Trial Period for Trial Editions of the Licensed Software, and is unlimited for all other editions Licensed Software.
- "Valid License" means a License authorization generated by Echelon during activation that is specific to a single Licensed Device.

IV. PERMITTED USE

The License granted in Section II above permits you to:

(a) install and use the Licensed Software during the Use Period on one (1) primary Licensed Device (the "Primary Computer") for each Valid License for a Trial Edition, and for each pair of Valid Licenses for all other editions;

(b) remotely access the Licensed Software from another computer or other device provided there is a Valid License for each simultaneous user;

(c) install and use a second copy of the Licensed Software during the Use Period on one (1) portable Licensed Device (the "Portable Computer") for each pair of Valid Licenses, provided that (i) the second copy is for the exclusive use of the individual who is the primary user of the copy of the Licensed Software installed on the Primary Computer, (ii) the Licensed Software is not a Trial Edition, (iii) the Licensed Software are not used on the Primary Computer and its associated Portable Computer at the same time, and (iv) such installation and use otherwise comply with all the terms and conditions of the Agreement;

(d) install any updates or upgrades for the Licensed Software released during the Maintenance Period;

(e) extend the Maintenance Period for an additional year for one copy of the Licensed Software with a Valid License by purchasing an Echelon Maintenance Renewal product that Echelon designates covers the specific edition of the Licensed Software; once extended the Maintenance Period may be extended again by purchasing an additional Echelon Maintenance Renewal product that Echelon designates covers the specific edition of the Licensed Software; the Maintenance Period may only be extended during the Maintenance Period;

(f) install any updates that Echelon identifies as not requiring maintenance coverage at any time during the Use Period;

(g) copy the Documentation for internal reference use only by any person that has valid access to your Licensed Device or internal network, provided that You reproduce, unaltered, all proprietary notices on or in the copy;

(h) make one (1) copy of the Licensed Software in machine readable form solely for backup purposes, provided that You reproduce, unaltered, all proprietary notices on or in the copy;

(i) physically transfer a copy of the Licensed Software with a Valid License from one (1) computer to another no more than one time every 90 days or upon hardware failure of the Licensed Device, provided that such copy is removed from or inaccessible on the computer on which it was previously installed and such installation and use otherwise comply with all the terms and conditions of the Agreement; and

(j) transfer Your rights under this Agreement to an end user of the Licensed Software; provided that (i) You and the transferee execute two copies of the Software License Transfer Agreement included with the Licensed Software, or an equivalent agreement that you create, (ii) You identify all previous licensees of the Licensed Software in the Software License Transfer Agreement or your equivalent agreement, and (iii) You and the transferee each retain one (1) signed original thereof and furnish

Echelon with a copy of same upon request; this right of transfer is exercisable no more than one (1) time every year.

V. RESTRICTIONS

The License granted in Section 2 shall not permit you or allow you to permit others to:

(a) install the Licensed Software on more than one (1) Primary Computer per Valid License for a Trial Edition and more than one (1) Primary Computer and one (1) Portable Computer per Valid License for all other Editions;

(b) use the Licensed Software on more than one (1) computer at a time per Valid License;

(c) use the Licensed Software after the end of the Use Period for a Trial Edition;

(d) allow multiple users to use the Licensed Software on a Primary Computer at the same time;

(e) allow any individual other than the primary user to use the Licensed Software on the Portable Computer;

(f) copy the Licensed Software (except as expressly permitted above);

(g) update or upgrade the Licensed Software after the end of the Maintenance Period with the exception of updates that Echelon designates may be installed outside the Maintenance Period;

(h) extend the Maintenance Period after the end of the Maintenance Period;

(i) modify, translate, reverse engineer, decompile, disassemble or otherwise attempt (i) to defeat, avoid, bypass, remove, deactivate, or otherwise circumvent any software protection mechanisms in the Licensed Software, including without limitation any such mechanism used to restrict or control the functionality of the Licensed Software, or (ii) to derive the source code or the underlying ideas, algorithms, structure or organization from the software from the Licensed Software (except to the extent that such activities may not be prohibited under applicable law); or

(j) except for the limited rights granted above, distribute, rent, loan, lease, transfer or grant any rights in the Licensed Software or modifications thereof or the Documentation in any form to any person without the prior written consent of Echelon.

This License is not a sale. Title and copyrights to the Licensed Software, Documentation, and any copy made by You remain with Echelon or its suppliers. Unauthorized copying of the Licensed Software, or failure to comply withthis Section V, will result in automatic termination of this license and will make available to Echelon and its suppliers other legal remedies. You agree to indemnify, to the fullest extent permitted by law, Echelon and its directors, officers, agents, employees, members, subsidiaries, affiliates, and its suppliers (“Indemnitees”), for any third party claims, actions, proceedings, liabilities, loss, damage, cost, or expenses (including, without limitation, attorney’s fees), arising out of or in connection with Your failure to use of any Licensed Softwareas permitted under this Agreement.

VI. VISIO

The Microsoft Corporation (“Microsoft”) Visio product(s) has been integrated or pre-installed as part of specified editions of the Licensed Software and Echelon is obligated to pay royalties to Microsoft for licensed copies of these editions. You hereby give Echelon permission to disclose to Microsoft Your name and address, or the name and address of another individual within your organization, as such as been provided to Echelon, in connection with Echelon’s delivery of its royalty report to Microsoft. Such permission includes, if you are located outside of the United States, the transfer of this information to the United States. Microsoft has stated that it uses the information in Echelon’s royalty report for revenue calculation, internal revenue allocation, compliance and billing purposes. For more information about how Echelon handles personal information, see *www.echelon.com/company/legal/privacypolicy.htm*.

The functionality of the Microsoft Visio product as integrated in specified editions of the Licensed Software may differ from a non-integrated Microsoft Visio product and any issues concerning the functionality or performance of the Echelon OpenLNS Commissioning Tool and the integrated Microsoft Visio product should be directed to Echelon and not to Microsoft Corporation.

VII. TERMINATION

This License will continue until terminated by either party in writing. This license will automatically terminate at the end of the Use Period. Unauthorized copying of the Licensed Software or failure to comply with the above restrictions will result in automatic termination of this Agreement and will make available to Echelon other legal remedies. This license will also automatically terminate if You go into liquidation, suffer or make any winding up petition, make an arrangement with Your creditors, or suffer or file any similar action in any jurisdiction in consequence of debt. Upon termination of this license for any reason You shall destroy all copies of the Licensed Software after the License is terminated. Any use of the Licensed Software after termination is unlawful.

VIII. TRADEMARKS

You may make appropriate and truthful reference to Echelon, Echelon products and technology in Your company and product literature; provided that You properly attribute Echelon's trademarks in accordance with Echelon's trademark guidelines, available at *www.echelon.com/company/legal/copyright.htm*, and do not use the name of Echelon or any Echelon trademark in Your name or product name. No license is granted, express or implied, under any Echelon trademarks, or trade names.

IX. LIMITED WARRANTY AND DISCLAIMER

Echelon warrants that, for a period of ninety (90) days from the date of delivery or transmission to You, the Licensed Software under normal use will perform substantially in accordance with the Licensed Software specifications contained in the Documentation. Echelon's entire liability and Your exclusive remedy under this warranty will be, at Echelon's option, to use reasonable commercial efforts to attempt to correct or work around errors, to replace the Licensed Software with functionally equivalent Licensed Software, or to terminate this Agreement. EXCEPT FOR THE ABOVE EXPRESS LIMITED WARRANTIES, ECHELON AND ITS SUPPLIERS MAKE AND YOU RECEIVE NO WARRANTIES OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY OR IN ANY COMMUNICATION WITH YOU, AND ECHELON AND ITS SUPPLIERS SPECIFICALLY DISCLAIM ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT AND THEIR EQUIVALENTS. Echelon does not warrant that the operation of the Licensed Software will be uninterrupted or error free or that the Licensed Software will meet Your specific requirements. Echelon shall not be responsible for any errors or non-conformity in the Licensed Software resulting from Your failure to use the Licensed Software in conformance with this Agreement and the Documentation, or modifications of the Licensed Software by You.

X. LIMITATION OF LIABILITY

IN NO EVENT WILL ECHELON OR ITS SUPPLIERS BE LIABLE FOR LOSS OR CORRUPTION OF DATA, LOST PROFITS, LOSS OF CONTRACTS, COST OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY OR SERVICES OR OTHER SPECIAL, INCIDENTAL, PUNITIVE, CONSEQUENTIAL OR INDIRECT DAMAGES ARISING FROM THE SUPPLY OR USE OF THE LICENSED SOFTWARE, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY (INCLUDING NEGLIGENCE). THIS LIMITATION WILL APPLY EVEN IF ECHELON OR AN AUTHORIZED DISTRIBUTOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND NOTWITHSTANDING THE FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN. IN NO EVENT SHALL ECHELON'S OR ITS SUPPLIERS' LIABILITY EXCEED IN THE AGGREGATE ONE THOUSAND UNITED STATES DOLLARS (USD

$1000). YOU HEREBY AGREE THAT THE FOREGOING LIMITATIONS REFLECT THIS REASONABLE ALLOCATION OF RISK.

XI. SAFE OPERATION

YOU ASSUME THE SOLE RESPONSIBILITY FOR, AND HEREBY AGREE TO USE YOUR BEST EFFORTS IN, DESIGNING, MANUFACTURING, COMMISSIONING, ROUTING, MONITORING, AND CONTROLLING DEVICES USING THE ISO/IEC 14908-1 PROTOCOL TO PROVIDE FOR SAFE OPERATION THEREOF, INCLUDING, BUT NOT LIMITED TO, COMPLIANCE OR QUALIFICATION WITH RESPECT TO ALL SAFETY LAWS, REGULATIONS AND AGENCY APPROVALS, AS APPLICABLE. THE SMART TRANSCEIVER, NEURON CHIP, ISO/IEC 14908-1 PROTOCOL, NEURON FIRMWARE, SMARTSERVER, i.LON ROUTER, AND THE LICENSED SOFTWARE ARE NOT DESIGNED OR INTENDED FOR USE AS COMPONENTS IN EQUIPMENT INTENDED FOR SURGICAL IMPLANT INTO THE BODY, OR OTHER APPLICATIONS INTENDED TO SUPPORT OR SUSTAIN LIFE, FOR USE IN FLIGHT CONTROL OR ENGINE CONTROL EQUIPMENT WITHIN AN AIRCRAFT, OR FOR ANY OTHER APPLICATION IN WHICH THE FAILURE OF THE SMART TRANSCEIVER, NEURON CHIP, LONTALK PROTOCOL, NEURON FIRMWARE, SMARTSERVER, i.LON ROUTER, OR THE LICENSED SOFTWARE COULD CREATE A SITUATION IN WHICH PERSONAL INJURY OR DEATH MAY OCCUR, AND YOU SHALL HAVE NO RIGHTS HEREUNDER FOR ANY SUCH APPLICATIONS.

XII. COMPLIANCE WITH EXPORT CONTROL LAWS

You hereby agree to comply with all applicable export and re-export control laws and regulations, including the Export Administration Regulations ("EAR") maintained by the United States Department of Commerce. Specifically, You covenant that You shall not—directly or indirectly—sell, export, re-export, transfer, divert, or otherwise dispose of any software, source code, or technology (including products derived from or based on such technology) received from Echelon under this Agreement to any country (or national thereof) subject to antiterrorism controls or U.S. embargo, or to any other person, entity, or destination prohibited by the laws or regulations of the United States, without obtaining prior authorization from the competent government authorities as required by those laws and regulations. You agree to indemnify, to the fullest extent permitted by law, Echelon from and against any fines or penalties that may arise as a result of Your breach of this provision. This export control clause shall survive termination or cancellation of this Agreement.

XIII. LANGUAGE

The parties hereto confirm that it is their wish that this Agreement, as well as other documents relating hereto, have been and shall be written in the English language only.

Les parties aux présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise.

XIV. GENERAL

This Agreement shall not be governed by the 1980 U.N. Convention on Contracts for the International Sale of Goods; rather, this Agreement shall be governed by the laws of the State of California, including its Uniform Commercial Code, without reference to conflicts of laws principles. This Agreement is the entire agreement between us and supersedes any other communications or advertising with respect to the Licensed Software and accompanying documentation. If any provision of this Agreement is held invalid or unenforceable, such provision shall be revised to the extent necessary to cure the invalidity or unenforceability, and the remainder of the Agreement shall continue in full force and effect.

If You are acquiring the Licensed Software on behalf of any part of the U.S. Government, the following provisions apply. The Licensed Software and accompanying documentation were developed at private expense and are deemed to be "commercial computer software" and "commercial computer software documentation", respectively, pursuant to DFAR Section 227.7202 and FAR 12.212(b), as applicable. Any use, modification, reproduction, release, performance, display or disclosure of the Licensed Software and/or the accompanying documentation by the U.S. Government or any of its agencies shall be governed solely by the terms of this Agreement and shall be prohibited except to the extent expressly permitted by the terms of this Agreement. Any technical data provided that is not covered by the above provisions is deemed to be "technical data/commercial items" pursuant to DFAR Section 227.7015(a). Any use, modification, reproduction, release, performance, display or disclosure of such technical data shall be governed by the terms of DFAR Section 227.7015(b).

Appendix D
Software License Transfer Agreement

If you purchased a license for the OpenLNS CT software, you can transfer the license to another party. To do this, you and the other party must sign the agreement in this appendix.

Software License Transfer Agreement

THIS SOFTWARE LICENSE TRANSFER AGREEMENT is entered into as of :________________
(Date)

by and between "Assignor" ____________________________________
(Company name of original purchaser)

and "Assignee" ___________________________________
(Name of company purchasing the software)

for the transferred "Software" ___________________________________
(Name of software product being transferred)

1. Software License Agreement. Reference is made to the Software License Agreement(s) that Assignor entered into when Assignor purchased the Software ("Software License Agreement"), pursuant to which Echelon Corporation ("Echelon") granted Assignor a non-exclusive license to use the Software.
2. Assignment of Software License Agreement. For good and valuable consideration, the receipt and sufficiency of which are hereby acknowledged, Assignor hereby assigns to Assignee all of Assignor's rights under the Software License Agreement. Assignee hereby assumes all of Assignor's obligations under the Software License Agreement and agrees to be bound by all of the terms and conditions of the Software License Agreement. Once the Software has been installed, a copy of the software license agreement is available in the LICENSE.TXT file of the Software folder.
3. Transfer History. If Assignor acquired the Software as a result of a previous Software License Transfer Agreement, Assignor acknowledges that a complete and accurate record of previous transfers of the Software is attached to this Software License Transfer Agreement.
4. General. This Agreement shall be governed by the laws of the State of California without reference to conflict of laws principles. Echelon shall be deemed a third party beneficiary of this Agreement. Neither party shall have the right to modify or assign this Agreement without the prior written consent of Echelon.

IN WITNESS WHEREOF, the parties have executed this Software License Transfer Agreement as of the date first written above.

"ASSIGNOR":_________________________________	"ASSIGNEE": ________________
Company Name	Company Name
Signed By:	Signed By:
(Print Name)	(Print Name)
Title: ____________________________________	Title: ______________________
Address:____	Address:_______________
City,State/Province, Country, Postal Code	City,State/Province, Country, Postal Code
Telephone: ____________________________	Telephone: ______________
(Fax or email address)	(Fax or email address)
Date: ____________________________________	Date: ______________________

For a copy of the appropriate Software License Agreement, call Echelon Corporation at +1-408-938-5200.

www.echelon.com