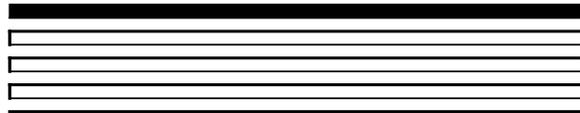


# LNS DDE Server User's Guide

Version 2.11



**ECHELON**  
Corporation



078-0170-01D

Echelon, LON, LONWORKS, LNS, LonTalk, Neuron, LONMARK, 3120, 3150, the LonUsers logo, the Echelon logo, and the LONMARK logo are registered trademarks of Echelon Corporation. LonPoint, LonMaker, and LonSupport are trademarks of Echelon Corporation.

Other brand and product names are trademarks or registered trademarks of their respective holders.

Smart Transceivers, Neuron Chips, LonPoint Modules, 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 or LonPoint Modules 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 AND YOU RECEIVE NO WARRANTIES OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY OR IN ANY COMMUNICATION WITH YOU, AND ECHELON SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Echelon Corporation.

Printed in the United States of America.  
Copyright © 1998-2002 by Echelon Corporation.

Model Number 39520

Echelon Corporation  
550 Meridian Ave.  
San Jose, CA 95126, USA

# Contents

<b>1</b>	<b>LNS DDE Server Overview</b>	<b>1-1</b>
	New Feature in Release 2.11	1-2
	New Features in Release 2.1	1-2
	Overview of LONWORKS® Networks	1-3
	Installing of a LONWORKS Network	1-3
	Monitoring and Controlling a LONWORKS Network	1-3
	Network Variables	1-3
	Configuration Properties	1-4
	LONMARK® Objects	1-5
	Application and Foreign Frame Messages	1-5
	The Role of the LNS DDE Server	1-5
	The Role of the LNS Network Interface	1-6
	Overview of DDE	1-6
	Types of DDE Services	1-7
	Execute Commands	1-7
<b>2</b>	<b>Getting Started</b>	<b>2-1</b>
	Requirements	2-2
	Installing the LNS DDE Server	2-2
	Installing and Configuring a Network Driver	2-6
	Starting the LNS DDE Server	2-6
	Using the LNS DDE Server	2-7
	Folder Pane	2-7
	Detail Pane	2-8
	Log Pane	2-8
	Menu Bar	2-9
	Toolbar	2-10
	Using the LNS DDE Server Examples	2-11
	Using the Example LNS Database	2-11
	Sequence of Operation	2-13
	Using the LNS DDE Example Database	2-13
	Using the Excel Example	2-14
	Using the InTouch Example	2-15
<b>3</b>	<b>Configuring the LNS DDE Server</b>	<b>3-1</b>
	Configuration Overview	3-2
	Server Settings	3-2
	DDE Server Settings	3-3
	Application Name	3-3
	Register DDE Share Name	3-3
	NetDDE Being Used	3-3
	Network Server	3-4
	Response Timeout (sec)	3-4
	Idle Timeout (sec)	3-4

Service	3-4
Run as a Windows NT Service	3-4
Write Messages to the Event Log	3-5
Message Logging Settings	3-6
Display Information Messages	3-6
Display Warning Messages	3-6
Display Error Messages	3-6
Display Trace Messages	3-6
Max Number of Lines	3-6
Scrolling Rate (ms)	3-6
Circular Log File	3-6
Max Log File Size	3-7
Log to WWLogger	3-7
LNS Object Server Settings	3-7
Format and Typefile Catalog Path Setting	3-7
Global Database Path Setting	3-8
Remote Settings	3-8
Remote Operation	3-8
Network Interface	3-9
Database Settings	3-9
Single User Mode	3-9
Trace Options	3-9
DDE Toolkit	3-10
Client Connection	3-10
Client Disconnection	3-10
Point Creation	3-10
Point Activation	3-11
Point Updating	3-11
Point Poking	3-11
Point Deactivation	3-11
Point Deletion	3-11
Execute	3-11
Formatting	3-11
Format Files	3-12
Type Files	3-12
Format Errors	3-12
Format Use	3-12
Format Search	3-12
Network Server	3-12
Launching	3-13
LNS Errors	3-13
Flow Control	3-13
Events	3-13
Update Errors	3-13
Service Pins	3-13
Attachments	3-14
Database Changes	3-14
License Settings	3-14
Current License	3-15
License Type	3-15
Restriction	3-15

Serial Number	3-16
Generate PC Key	3-16
Creating an Unlimited Mode License	3-16
Transferring Licenses	3-18
Network Settings	3-18
Network Name	3-19
Database Path	3-19
Authentication Key	3-20
Network Interface	3-20
Network Attached	3-20
Network Interface Name	3-20
Attach to Network when Exploring	3-20
Set Management Mode OnNet	3-20
Network Event Subscription	3-21
Default Data Reporting Settings	3-21
Polling Interval (1/10 sec)	3-21
Refresh Rate (1/10 sec)	3-22
Use Bound Updates	3-22
Use Priority Messages	3-24
Number of Retries	3-24
Reply Timeout (1/10 sec)	3-24
Exclusive Mode (TCP/IP)	3-25
Host Message Owner	3-25
Point Recovery Strategy	3-25
Message Options—Network Variables	3-25
Service	3-25
Bound Service	3-25
Bound Update Persistent	3-26
Message Options—Application Messages	3-26
Service	3-26
Authenticate Messages	3-26
Include Sequence Message	3-26
Creating Custom Topics	3-26
Topic Name	3-27
Network	3-27
Subsystem	3-27
Update Interval	3-27

<b>4 Addressing DDE Messages</b>	<b>4-1</b>
Address Components	4-2
Topics	4-2
Standard Topic Format	4-3
Custom Topic Format	4-4
Topic Types	4-4
Topic Types and the Folder Pane	4-6
Items	4-7

<b>5</b>	<b>Reading and Writing Network Variables</b>	<b>5-1</b>
	Specifying a Network Variable	5-2
	Specifying a Network Variable Topic	5-2
	Specifying a Network Variable Item	5-2
	Network Variable Formats	5-5
	Requesting a One-Time Network Variable Update	5-6
	Requesting a One-Time Network Variable Update Using Excel	5-6
	Requesting a One-Time Network Variable Update Using InTouch	5-8
	Receiving Ongoing Network Variable Updates	5-9
	Event Driven Updates versus Polling	5-9
	Monitoring Polled Output Network Variables	5-10
	Receiving Ongoing Updates Using Excel	5-10
	Receiving Ongoing Updates Using InTouch	5-11
	Sending a Network Variable Update	5-11
	Sending a Network Variable Update Using Excel	5-12
	Sending a Network Variable Update Using InTouch	5-13
	Timing Issues Related to Visual Basic Controls	5-14
<b>6</b>	<b>Reading and Writing Configuration Properties</b>	<b>6-1</b>
	Specifying a Configuration Property	6-2
	Specifying a Configuration Property Topic	6-2
	Specifying a Configuration Property Item	6-2
	Accessing Configuration Property User Formats	6-4
	Requesting a One-Time Configuration Property Update	6-5
	Receiving Ongoing Configuration Property Updates	6-6
	Sending a Configuration Property Update	6-6
<b>7</b>	<b>Receiving and Sending Application and Foreign Frame Messages</b>	<b>7-1</b>
	Specifying an Incoming Message Item	7-2
	Specifying a Request Message	7-3
	Specifying a Message Completion Event	7-5
	Specifying an Outgoing Application Message	7-7
	Specifying a Message Using Domain-Wide Broadcast Addressing	7-8
	Requesting a One-Time Update	7-10
	Requesting Ongoing Message Updates	7-11
	Sending a Message Update	7-11
<b>8</b>	<b>Using Properties</b>	<b>8-1</b>
	Properties	8-2
	Specifying a Property	8-2
	System Properties	8-2
	Device and Router Properties	8-3
	Properties Available in both Application Devices and Routers	8-4
	Properties Available only in Application Devices	8-6
	Properties Available in both Application Devices and in each side of Routers (NearSide and FarSide)	8-6
	LONMARK Object Properties	8-7

Point Properties	8-8
<b>Appendix A Resource Files</b>	<b>A-1</b>
LONMARK Resource Files	A-2
Type File	A-2
Format File	A-3
Alternate Formats	A-3
Language File	A-4
Functional Profile Template File	A-4
<b>Appendix B Installing the PCLTA-10/20 and PCC-10 Drivers</b>	<b>B-1</b>
PCLTA-10, PCLTA-20, and PCC-10 Adapter Installation	B-2
PCLTA-10, PCLTA-20, and PCC-10 Software Installation Procedure	B-2
Hardware Installation Procedure	B-4
PCLTA-10 Adapter Hardware Installation Procedure	B-4
PCLTA-20 Adapter Hardware Installation Procedure	B-5
PCC-10 Card Hardware Installation Procedure	B-5
PCLTA-10, PCLTA-20, and PCC-10 Diagnostic Procedure	B-6
<b>Appendix C WonderWare InTouch Client Example</b>	<b>C-1</b>
Introduction	C-2
InTouch Overview	C-2
InTouch Example Tutorial	C-3
InTouch Animation and I/O Tags	C-3
Network Variable Monitoring	C-8
Changing the Default Format	C-10
Display Unit Conversion	C-10
Using the InTouch DDE Link Wizards	C-11
Other Topic Types	C-14
<b>Appendix D License Order Form</b>	<b>D-1</b>
<b>Appendix E LNS DDE Server Software License Agreement</b>	<b>E-1</b>
Notice	E-2
LNS DDE Server Software License Agreement	E-2
License	E-2
Limited Warranty and Disclaimer	E-4
Limitation of Liability	E-4
Safe Operation	E-4
Language	E-5
General	E-5
<b>Appendix F LNS DDE Server Software License Transfer Agreement</b>	<b>F-1</b>
<b>Glossary</b>	<b>G1-1</b>



# 1

## LNS DDE Server Overview

This chapter defines the main concepts for a LONWORKS® network, the role of a network management tool in setting up a network, and the role of the LNS DDE Server in monitoring and controlling the network. It also provides a brief overview of DDE and a summary of the new features in Releases 2.1 and 2.11. Additional terms are defined in the *Glossary*.

---

## New Features in Release 2.11

Release 2.11 (release 2.1 with Service Pack 1 applied) of the LNS DDE Server includes adds date stamping to the log pane.

---

## New Features in Release 2.1

Release 2.1 of the LNS DDE Server includes the following new features:

- *Expanded Polling Capability* - When used with either an LNS fast network interface attached to a LonWorks channel, or an IP network interface attached to a LonWorks/IP channel, the LNS DDE Server can now poll up to 1,000 points simultaneously, instead of the maximum of 2 in the previous release. The result is a significant improvement in the number of points per second that may be polled.
- *Expanded Binding Capability* - When used with either an LNS fast network interface attached to a LonWorks channel, or an IP network interface attached to a LonWorks/IP channel, the LNS DDE Server now has an address table with 32,768 entries instead of the 16 entries in the previous release. This enables the LNS DDE Server PC to be a member of every group within a domain, eliminating many of the binding constraints of the previous release, and supporting many more bound connections.
- *LNS 3 Runtime* - Release 2.1 of the LNS DDE Server installs the LNS 3 Runtime. This provides significantly improved performance when using either an LNS fast network interface attached to a LonWorks channel, or an IP network interface attached to a LonWorks/IP channel. The performance improvements are a result of faster throughput through the LNS runtime and the network interface, and the increased capacities for polling and binding.
- *LonWorks/IP Channel Support* - Release 2.1 of the LNS DDE Server can be used to monitor and control remote full client networks over a LonWorks/IP Channel. This allows the LNS Server PC to be attached directly to an IP channel without requiring an LNS network interface card. This also increases performance for most monitoring and control applications. For example, an application that could previously poll no more than 35 points per second may be able to poll over 600 points per second from a LonWorks/IP client. A LonWorks/IP client can receive over 1200 updates per second for bound updates. Actual performance depends on your computer hardware and application.
- *Wonderware SuiteLink Support* -- The Wonderware SuiteLink protocol is now supported for accessing the LNS DDE Server, in addition to the Microsoft DDE protocol. This provides improved performance for Wonderware InTouch clients.
- *Multiple Network Support* - A single instance of the LNS DDE Server can now simultaneously open up to 100 LNS networks through a single LNS network interface. This includes both local and remote networks. This allows a single DDE application to provide an interface to multiple networks, either through a

single network interface, or through multiple network interfaces. To open multiple remote networks, you must configure the LNS DDE Server as a lightweight client. See *Remote Settings* in Chapter 3, *Using the LNS DDE Server*.

- *Updated Network Drivers* - New drivers are included for the PCC-10, PCLTA-10, and PCLTA-20 cards. These new drivers include new firmware for the network interface cards that reconfigures the cards as LNS fast network interfaces, providing significant performance improvements for LNS DDE Server applications.
- *Service Mode Operation* - It is now easier to run the LNS DDE Server as a Windows XP, Windows 2000, or Windows NT system service. This allows the LNS DDE Server to continue operating even as operators log out of and log onto a Windows XP, Windows 2000, or Windows NT computer. When running as a system service, the LNS DDE Server may be configured to interact with the user desktop, providing an easy-to-use interface for operators to view all points available to the LNS DDE Server. Alternatively, the LNS DDE Server may be configured to run hidden, preventing access by operators.

---

## Overview of LONWORKS Networks

A LONWORKS network consists of intelligent devices, also called *nodes*, connected by one or more communications channels that communicate with one another using the LONWORKS protocol. LONWORKS devices are programmed to send messages to one another in response to external events or messages that they receive. Two devices do not have to be attached to the same physical communications channel to communicate; however, they must be part of the same *network*. A network consists of all of the devices that can exchange LONWORKS messages, and the communications channels and routers that allow the devices to communicate.

---

### *Installing a LONWORKS Network*

Before the LNS DDE Server can monitor and control a network, the network must be installed using an LNS installation tool such as the LonMaker™ Integration Tool. Installation of a control network is the process of loading the unique network personality of each device into its internal tables. This process enables each device to communicate with other devices in the same system. Maintenance involves tasks such as adding new devices, replacing failed devices, and re-configuring the devices that share data with one another.

As the installation tool configures the devices on the network, it stores a copy of this information in the *LNS Server*. When installation is complete, the LNS Server contains a picture of the configuration of the entire network along with the names of all the devices and the network variables, configuration properties, and LONMARK objects (also called *functional blocks*) on the devices. The LNS DDE Server uses the information in the LNS Server to accomplish its tasks.

---

## Monitoring and Controlling a LONWORKS Network

Due to their high reliability, low cost, and flexibility, LONWORKS networks are used in many ways. For example, they can be used to instrument relay-controlled systems, saving the time, complexity, and cost associated with running wire from each point back to the monitoring station. They can also be used in conjunction with a central controller or PLC, for example, to provide a subsystem of smart sensors. LONWORKS networks are frequently used to create *peer-to-peer* systems where each device controls its own actions and shares information with its neighbors as needed to control the entire system.

When monitoring a LONWORKS network, the monitoring system can request status directly from a device, since each device is intelligent and knows its own status. Similarly, when controlling a LONWORKS network, a control device can send data directly to a controlled device. No master needs to be involved and no additional wires need to be run. Also, using the information in the LNS Server, devices can be addressed by the names assigned during installation. These names are much easier to remember than the complex addressing schemes used by hard-wired systems, and, since they are assigned in lock-step with the installation, they match what is actually contained in the system.

There are several methods for exchanging data between LONWORKS devices. These are network variables, configuration properties, and application and foreign frame messages. These are described in the following sections. LONMARK objects are also described since they provide a higher-level description of the interfaces to a device.

### Network Variables

The most common way for LONWORKS devices to communicate with one another is through *network variables*. Network variables are the logical inputs and outputs of a device. The types, functions, and number of network variables in each device are determined by the application code within the device. For example, a temperature sensor might have two network variables — a temperature reading output and an over-temperature alarm output. Some devices may also support dynamic network variables that can be added and deleted by an LNS network tool such as the LonMaker Integration tool. The host PC running the LNS DDE Server also supports dynamic network variables.

The actual data representation of a network variable is defined by its type. The type may either be a user defined network variable type (UNVT), or a standard network variable type (SNVT). SNVTs ensure common data representation and interoperation for all LONWORKS devices; using SNVTs is recommended whenever possible. A master list of standard network variable types is contained in the *LNS Utilities and LonMark Reference* help file included with the LNS DDE Server.

### Configuration Properties

Application configuration data is specified using *configuration properties*. Configuration properties may be implemented on a device as *configuration network variables* or as *configuration parameters*. Any device can use configuration network variables. Devices that comply with the *LONMARK Application Layer Interoperability Guidelines*, version 3.0 or later, may use configuration parameters.

Configuration parameters are stored in files on the device, and may be accessed using the LONWORKS file transfer protocol, or by direct memory read/write for Neuron Chip hosted devices. Configuration properties may be properties of the device, a LONMARK object, or a network variable.

The data representation and interpretation of a configuration property is defined by its type. The type may either be a user defined configuration property type (UCPT), or a standard configuration property type (SCPT). SCPTs are a standard in all LONWORKS devices; using SCPTs is recommended whenever possible. A master list of standard configuration property types is contained in the *LNS Utilities and LonMark Reference* help file included with the LNS DDE Server.

## LONMARK Objects

*LONMARK objects* are collections of associated network variables and configuration properties along with definitions of their associated behavior. LONMARK objects provide the basis for application layer interoperability.

The network variable and configuration property members of a LONMARK object, along with the interpretation of each, are defined by functional profiles. The functional profile may either be a user defined functional profile, or a standard functional profile. Standard functional profiles are defined and maintained by the LONMARK Interoperability Association.

## Application and Foreign Frame Messages

Application and foreign frame messages are a low-level method of exchanging data between LONWORKS devices. Each LONWORKS message must be given a unique *message code* for identification. The message code identifies the message as a network variable, application, foreign frame, network management, or network diagnostic message. Network variable and application messages are used for exchanging application data. Foreign frame messages are used by gateway applications. Network management and network diagnostic messages are used by the LNS Server. Application and foreign frame messages are less interoperable than network variables because all devices that need to share information must be designed with their message codes in agreement. The message codes are part of the application program and are not configurable.

Devices send and receive application and foreign frame messages using *message tags*. A message tag is an input or output port of a device. A device can send a message with any message code using any of its message tags. Similarly, a message with any code might arrive at any of the device's message tags. Establishing connections between device message tags is another task done by an LNS network tool during network installation.

---

## The Role of the LNS DDE Server

The LNS DDE Server exchanges network variable, configuration property, and application and foreign frame message data between LONWORKS devices and Windows applications. With the LNS DDE Server, any Windows application that can act as a DDE Client can monitor and control one or more LONWORKS networks.

That is, the application can observe the values of network variables, configuration properties, and application and foreign frame messages, and can change the values of them to affect the operation of the network(s). The LNS DDE Server uses the naming, addressing, and timing information stored in an LNS Server by a network tool such as the LonMaker Integration Tool.

There are many tools available for installing LONWORKS networks. Some of these are legacy tools that are not LNS compatible. However, the LNS DDE Server requires that the network installation be completed using an LNS installation tool. If you have a network that was installed with a network tool that was not based on LNS, you can automatically convert the network to LNS using the LonMaker Integration Tool, release 3.0 or later.

---

## The Role of the LNS Network Interface

An *LNS network interface* (also called a *network adapter*, *network services interface*, or *NSI*) is a hardware component with an associated network driver that provides the physical connection between a LONWORKS network and the PC running the LNS DDE Server. An LNS network interface is not required if your PC is running as a remote LONWORKS /IP Remote, LNS/IP remote (i.e. Lightweight), or NetDDE remote client. For the highest performance for workstation applications on a desktop attached to a LONWORKS/IP channel, you should use an IP network interface card such as an Ethernet adapter. For the highest performance for workstation applications on a desktop attached to a LONWORKS channel, you should use a PCLTA-20 PCI LonTalk Adapter as the LNS network interface (use a PCLTA-10 ISA LonTalk adapter if you do not have an available PCI slot). If you are using powerline products or an SMX transceiver, you can use the PCLTA-20/SMX PCI card. If you have a laptop attached to a LONWORKS channel with a PCMCIA-compatible PC Card slot, you can use the PCC-10 PC Card interface. For connecting to a LONWORKS channel remotely via phone lines and a pair of modems, or if your host does not have an available ISA, PCI, PC Card slot, you can use a SLTA-10 Serial LonTalk Adapter.

All PC-compatible LNS network interfaces are functionally equivalent; so your DDE client application does not need to be modified when the LNS network interface hardware changes. If you are not using a PCLTA-10, PCLTA-20, PCNSI, PCC-10, SLTA-10, or IP network interface card, check your documentation to see if your network interface is LNS-compatible.

---

## Overview of DDE

Dynamic Data Exchange (DDE) defines a standard way for Microsoft Windows applications to share information with one another. The following sections provide a very brief overview of DDE.

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* (this is different than a LONWORKS channel, which is the physical communications medium used by LONWORKS devices).

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) may ask the LNS DDE Server (the server) for the current fuel consumption from a flow sensor for use in an automated billing system. Or, an InTouch operator interface (the client) may tell the LNS DDE Server (the server) to change the state of a valve in a LONWORKS network in response to a request issued by the operator. The client application is always the application that establishes the conversation, irrespective of which way the data actually flows. The LNS DDE Server is always a server application.

---

## Types of DDE Services

A client uses different DDE services during the course of a DDE conversation. The DDE services discussed in this manual are the following:

<b>Initiate</b>	A request to start a new DDE conversation.
<b>Terminate</b>	A request to end a DDE conversation.
<b>Request</b>	A one-time request by the client for a specific piece of data from the server. This is also referred to as a <i>Cold</i> or <i>Manual Link</i> . The server does not notify the client of changes; the server only sends the item's value when requested by the client.
<b>Advise</b>	A request by the client for on-going updates for a specific piece of data from the server. There are two types of Advise requests.  <i>Hot</i> or <i>Automatic</i> Whenever the value of the item changes, the server automatically sends the client the new value.  <i>Warm</i> or <i>Notify</i> Whenever the value of the item changes, the server sends the client notification that the value has changed. If interested, the client requests the new value from the server.  <b>Note:</b> <i>The LNS DDE Server does not support warm (notify) links.</i>
<b>Poke</b>	A request from the client to the server to change the value of a specific piece of data.
<b>Execute</b>	A request from the client to the server to perform a command.

---

## Execute Commands

The LNS DDE Server provides the following commands via the DDE execute service. These commands are not case sensitive.

- [Pause] Temporarily stops the LNS DDE Server from sending updates to devices on the LONWORKS network, both updates to network variables and network variable polls. When paused, however, it will still receive network variable update messages from application devices and pass those to the client.
- [Resume] Reenables polling and outgoing update messages.

# 2

## Getting Started

This chapter explains how to install the LNS DDE Server, and how to start it with an existing network.

---

## Requirements

The LNS DDE Server requires a PC that meets the following requirements:

- Microsoft Windows XP, Windows 2000, Windows Me, Windows 98, or Windows NT 4.0 (Service Pack 3 or later is required for Windows NT; Windows XP or Windows 2000 recommended)
- Pentium 133 or faster (Pentium 200 or better recommended)
- 20 Megabytes free hard-disk space
- 64 Megabytes of RAM minimum (128 Megabytes minimum recommended)
- CD-ROM drive
- Mouse or compatible pointing device

**Note:** *The memory required is affected by the number of Windows applications that are running simultaneously. More than 64 Megabytes may be required if you are running many applications at the same time.*

The LNS DDE Server requires an LNS Server. The LNS Server may be on the same PC or a remote PC. The LNS DDE Server is distributed with the required LNS Server files.

---

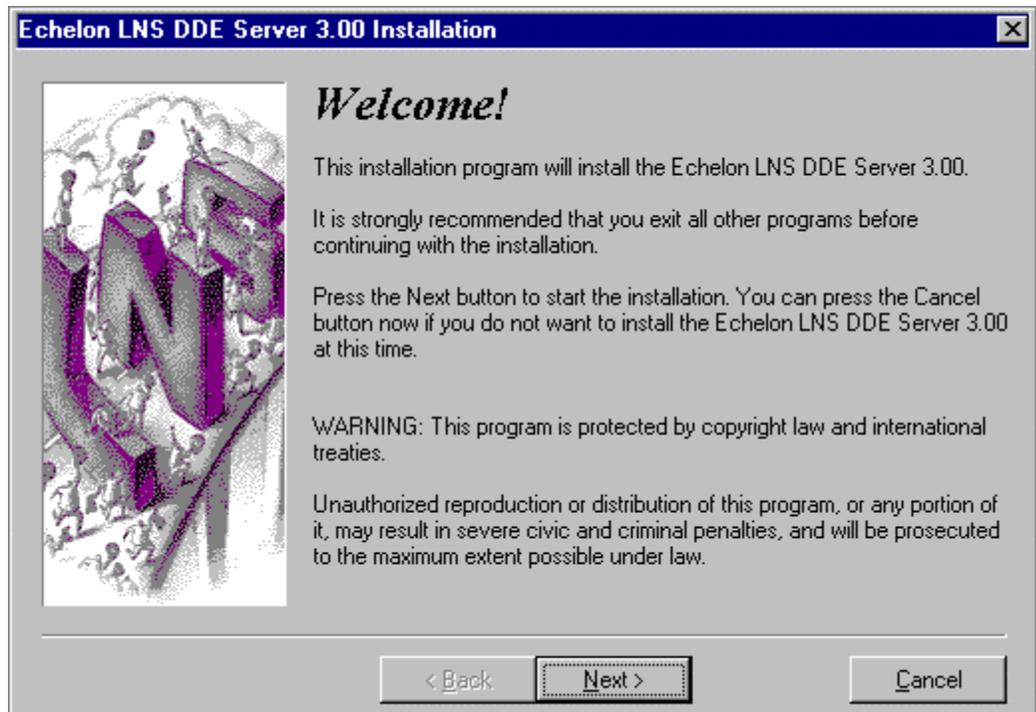
## Installing the LNS DDE Server

To install the LNS DDE Server, follow these steps:

- 1 If you are using Windows XP, Windows 2000, or Windows NT 4.0 and you do not have Administrator privileges, log out of Windows, then log on as a user who is a member of the Administrators group. Close all running Windows applications and disable any virus checking utilities.
- 2 If your copy of the LNS DDE Server was delivered on a CD-ROM, insert the CD-ROM into your CD-ROM drive. If the LNS DDE Server setup program does not start automatically, open the Windows Start menu and select Run. Browse to the Setup application on the LNS DDE Server CD and click Open, then click OK to run the Setup program.

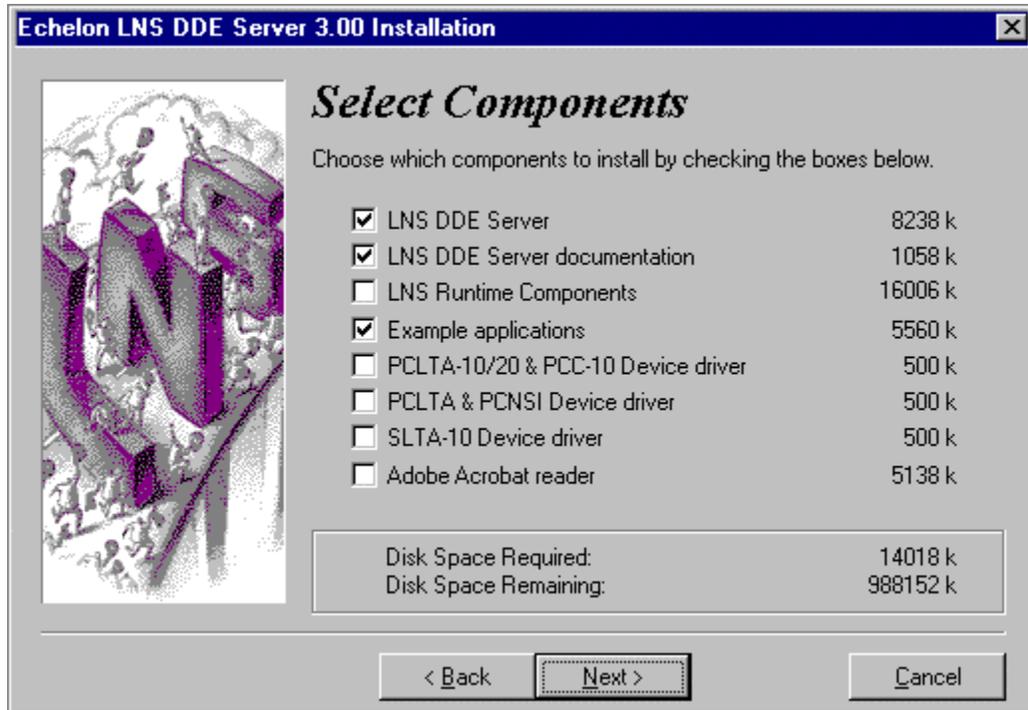
If you downloaded your copy of the LNS DDE Server, open the Windows Start menu and select Run. Browse to the Setup application that you downloaded and click Open, then click OK to run the Setup program.

The Setup program displays a series of dialogs that are described in the following steps. Some of the dialogs may not be displayed depending on the choices you make during installation. The first dialog is the Welcome dialog.



In the lower part of all the dialogs you will find navigation buttons. Use these buttons to go forward to the next dialog, go back to the previous dialog, or cancel the whole installation.

- 3 Click *Next* to continue with the next step. The License dialog is displayed.
- 4 Read the license in its entirety, and if you accept it, click *I Agree*. If you don't accept it click *I Do Not Agree* to terminate the installation without installing the LNS DDE Server. If you accept the license agreement, the Registration dialog is displayed.
- 5 Enter your name, company name, and LNS DDE Server serial number in the appropriate fields. The serial number is located on your LNS DDE Server case or envelope, or was provided to you when you downloaded the LNS DDE Server software. It is not necessary to enter the serial number to install the software, but when ordering an application key, a serial number must be supplied. Click *Next*. The setup program displays the Select Components dialog.



6 Select the LNS DDE Server components to be installed. The installation program automatically preselects some of the components based in part on what it finds already installed on your system. Some components may not be available for installation. When selecting or clearing components, the disk usage for the currently selected files is shown in the lower part of the window. Following is a description of each of the options:

- *LNS DDE Server.* Installs the LNS DDE Server application files and creates an Echelon LNS DDE Server program folder in your Windows Start menu. This option must be installed to use the LNS DDE Server.
- *LNS DDE Server documentation.* Installs LNS DDE Server help file and an Acrobat version of the *LNS DDE Server User's Guide*.
- *LNS Runtime Components.* Installs the LNS network operating system components required by the LNS DDE Server, including the LNS Server. These components are required. The setup program will make a determination of the proper setting of this option. This option also creates an Echelon LNS Utilities program folder in your Windows Start menu.
- *Example applications.* Installs Excel and InTouch examples. See *Using the LNS DDE Server Examples* for a description of the examples.
- *PCLTA-10/20 & PCC-10, SLTA-10, and PCLTA & PCNSI device drivers.* Installs drivers required for accessing an LNS network interface. Select the driver for the network interface hardware that you will be using on the PC with the LNS DDE Server. A driver is not required if you are using the LNS DDE Server as a LONWORKS/IP, LNS/IP, or NetDDE Client.
- *Adobe Acrobat reader.* Installs the Adobe Acrobat reader that can be used to read the Adobe Acrobat versions of the *LNS DDE Server User's Guide* and the *PCC-10, PCLTA-10, PCLTA-20, SLTA-10, and PCNSI* manuals.

Select the components to be installed and click Next. The Select Destination Directory window appears.

- 7 Choose a LONWORKS folder where you want the LNS DDE Server software installed. The software will be installed in an LNSDDE folder below the LONWORKS folder that you specify. If you have installed other LONWORKS software on this computer, be sure to use the same LONWORKS folder. The default folder is c:/LonWorks if you have not installed any other LONWORKS applications; the default is the last LONWORKS folder that you used if you have. Click Next. The Backup Replaced Files dialog is displayed.
- 8 The Backup dialog allows you to choose if you want to backup the LNS software. The installation program includes an uninstaller which is capable of undoing all changes and copying back the replaced files.

**Warning!** Backing up the software can lead to versioning issues under certain circumstances after uninstalling the LNS DDE Server; always choose the default not to back up the LNS software.

If you select the Yes option, the next dialog will allow you to select the folder where the replaced files will be placed. Enter a folder and click *Next*. You may receive a message stating Backup Already Exists Would You Like to Install that Directory Anyway? Choose Yes or No.

- 9 After making all required selections, the next dialog declares that the Setup program is ready to start installing the selected components. If you are unsure about the selections you have made, you can take this opportunity to use the *Back* button to review them. When you are sure everything is correct, press the *Next* button to commence installation.
- 10 If the example applications were selected in step 6, the setup program for the example applications will be started. Select the desired applications to install and click *Next*. Enter a destination folder for the examples and click *Next*.
- 11 If any drivers were selected in step 6, the setup programs for the selected drivers will start. Install the drivers as described in *Installing and Configuring a Network Driver*.
- 12 If the Adobe Acrobat installation was selected in step 6, the Acrobat setup program will start. Follow the instructions in the Acrobat setup dialog.
- 13 After the Setup program has completed, you may be prompted to restart your PC to complete the installation. If prompted, restart the computer.
- 14 Download the latest LNS DDE Server service pack from [www.echelon.com/lonmaker](http://www.echelon.com/lonmaker). At the time of publication of this manual, the latest service pack is LNS DDE Server 2.1 Service Pack 1. Follow the instructions provided with the LNS DDE Server service pack to install it.
- 15 Reboot when instructed.
- 16 Download and install the latest LNS 3 service pack. At the time of publication of this manual, the latest LNS 3 service pack was LNS 3 Service Pack 5A. You can

download the latest LNS 3 service pack from [www.echelon.com/lns](http://www.echelon.com/lns). Follow the instructions provided with the LNS service pack to install it.

- 17 If you selected a network driver in step 6, configure the network driver as described in *Installing and Configuring a Network Driver*, later in this chapter.

If you have not installed an application key for the LNS DDE Server, the LNS DDE Server will run for 30 days. After 30 days, it will continue to run for an hour at a time until you acquire an Application key as described in *Register Application Key* in Chapter 3.

---

## Installing and Configuring a Network Driver

The LNS DDE Server may communicate with a LONWORKS network through an LNS network driver and an LNS network interface or through an IP network driver and an IP network interface. Supported LNS network interfaces are the *i.LON 1000* Internet Server, PCLTA-10, PCLTA-20, PCC-10, PCNSI, and SLTA-10. The LNS DDE Server Setup application includes drivers for these network interfaces (except for the *i.LON 1000* . The drivers can be installed at the same time as the LNS DDE Server, or you may install them at a later time. It is not necessary to re-install the network driver if you already configured it for use with the LonMaker Integration Tool. Supported IP network interfaces include TCP/IP-compatible Ethernet adapters and PPP dial-up interfaces.

If you have already installed a PCLTA-10, PCLTA-20, PCC-10, or PCNSI driver, configure the driver. For the PCLTA-10, PCLTA-20, or PCC-10, be sure to install the driver software **before** you install the network interface hardware. If you have installed an SLTA-10 driver, configure the SLTA-10 hardware and driver as described in the *SLTA-10 User's Guide*, located in the SLTA-10 program folder. If you are using an *i.LON 1000*, install the *i.LON* software as described in the *i.LON* documentation. If you need to install a different driver, see the documentation for that driver for installation instructions.

---

## Starting the LNS DDE Server

As part of the installation process, the installation program will create a program folder named *Echelon LNS DDE Server* and place a number of shortcuts in it. One of the shortcuts points to the LNS DDE Server application. To access this folder, click the Windows *Start* button, point to *Programs*, then click *Echelon LNS DDE Server*.

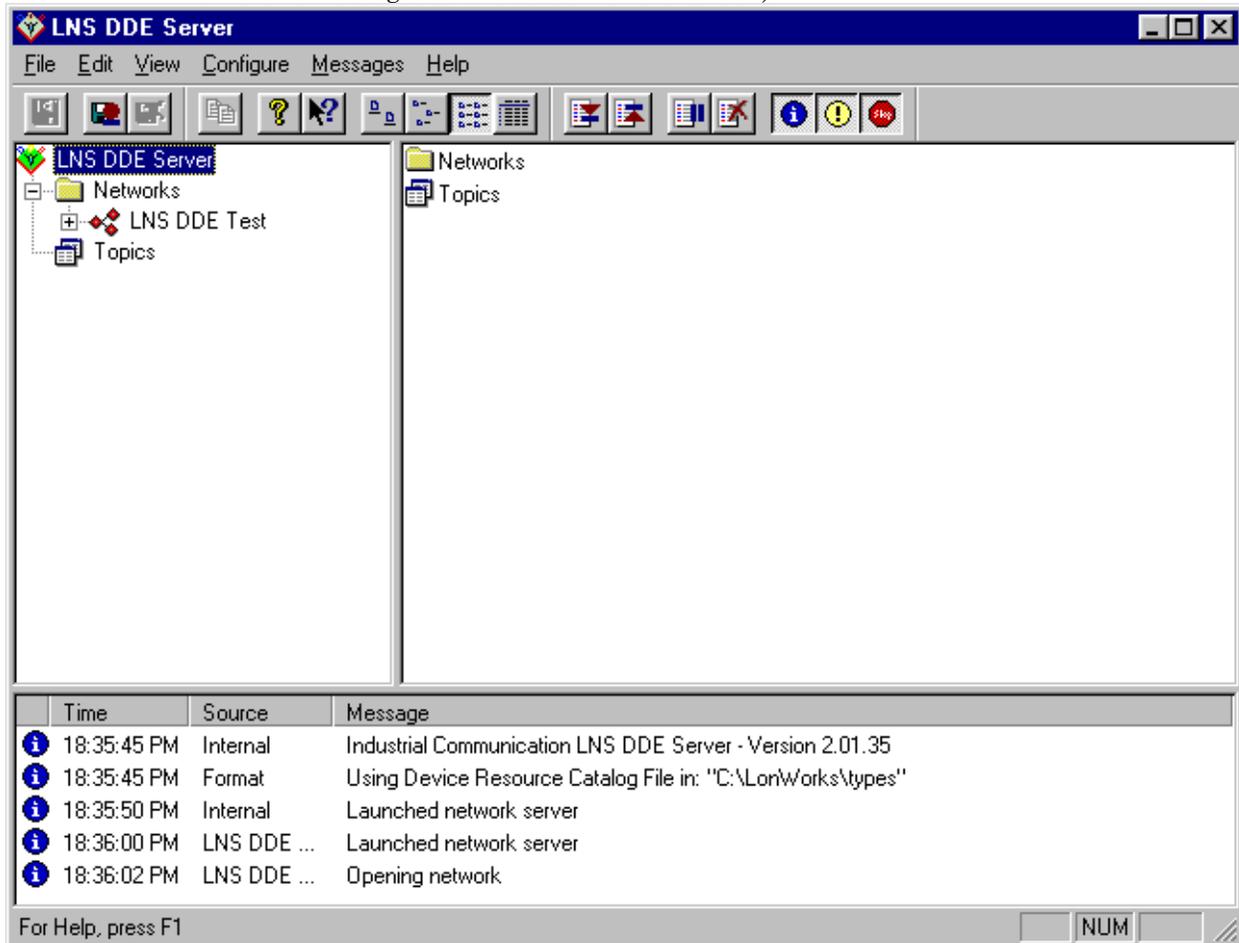
You can place a shortcut directly on your desktop, or you can make the LNS DDE Server start automatically when you log on. To make the LNS DDE Server start up automatically when you log on, copy the shortcut to your Windows *Startup* folder or, if using Windows XP, Windows 2000, or Windows NT 4, you can configure the LNS DDE Server to run as a service as described in *Run as a Windows NT Service* in Chapter 3.

To start the LNS DDE Server, select the LNS DDE Server shortcut. When started, the LNS DDE Server will display a window as described in the next section.

---

## Using the LNS DDE Server

When you start the LNS DDE Server, a main window similar to the following figure appears (you may see a different set of networks under the Networks folder and different messages at the bottom of the screen):



The main window is divided into three panes. On the left side there is a folder pane, on the right side there is a detail pane, and along the bottom a log pane. A menu bar is displayed across the top. The three panes and the menu bar are described in the following sections.

---

### Folder Pane

The folder pane on the left side of the main window allows you to navigate the networks, subsystems, devices, LONMARK objects, network variables, and configuration properties that are defined in an LNS Server. The top-level of the folder pane contains a Networks folder with all the currently available networks listed, as well as a Topics folder with all the pre-created topics.

To view the subsystems, devices, LONMARK objects, network variables, and configuration properties in a network, expand the Networks folder, then click the network name in the folder pane. When you click a network name, the objects

associated with the selected network will be listed in the detail pane on the right side of the window. You can simultaneously open any number of the networks listed in this pane.

---

## Detail Pane

The detail pane lists the objects that are associated with the currently selected folder on the folder pane. Depending on what type of folder is selected in the folder pane, the detail pane may list the networks, subsystems, devices, LONMARK objects, network variables, configuration properties, or topics belonging to the currently selected folder.

You can view the objects in the detail pane as large icons, small icons, or as a list with details. To change currently selected view, right-click somewhere in the detail pane and select a view from the shortcut menu, or select the desired view from the *View* menu on the LNS DDE Server menu bar.

---

## Log Pane

The log pane is displayed along the bottom of the LNS DDE Server main window. The log pane shows messages generated by the LNS DDE Server.

The log messages may be caused by DDE client applications, the underlying LNS software, or from internal activities in the LNS DDE Server. The log lists the severity, date, time, source, and message text for each message.

You can filter the events shown in the log by selecting *Messages/Log Display*. The selections made on the *Log Display* menu are temporary for the current session. To configure more permanent filter settings, select the *Server* command from the *Configure* menu on the menu bar. If messages are being logged to a file, the filter settings you use in the Log Pane do not affect the messages logged to the file.

If you select the *Trace* option from *Messages/Log Display*, the messages displayed in the Log Pane are controlled by the server configuration settings. See *Trace Options* in Chapter 3 for more information on the Trace option.

Use the Page Up and Page Down keys to scroll the log, or use the scrollbar on the right side of the pane. Right-click in the log pane and select the Top, Bottom, or Clear commands to move to the top or bottom of the log, or to clear the contents of the log. Alternatively, you can use the *Go to Top*, *Go to Bottom*, or *Clear Log* commands on the View menu or these same commands on the *Messages/Log Display* menu. You can stop automatically scrolling the log by selecting the *Freeze* command from the *Messages/Log Display* menu.

The number of lines shown in the log is limited. To view or set the limit, select the *Server* command from the *Configure* menu. The oldest lines will be removed from the log when the limit is reached.

---

## Menu Bar

The menu bar contains six menus. These are described in the following table:

### File

Save Log File	Saves the contents of the log file.
Exit	Terminates the LNS DDE Server.

### Edit

Copy Link	Copy link information for the currently selected object in the folder or detail pane to the clipboard.
-----------	--

### View

Toolbars/Log File	Toggles the display of the Log File toolbar, immediately below the menu bar on the left side.
Toolbars/Log Display	Toggles the display of the Log Display toolbar, immediately below the menu bar on the right side.
Toolbars/Miscellaneous	Toggles the display of the Miscellaneous toolbar, immediately below the menu bar in the center.
Status Bar	Toggles the display of the status bar at the bottom of the window.
Large Icons	Changes icon view to large-sized icons.
Small Icons	Reduces icon view to small-sized icons.
List	Changes icon view in the detail pane to a list
Details	Gives a description of each item in the Detail pane.
Go to Top	Go to the top of the log.
Go to Bottom	Go to the bottom of the log.
Clear Log	Clears all log entries.
Refresh	Refreshes the content of the folder and detail panes.

### Configure

Server	Opens a configuration dialog with tabs for the LNS DDE Server Settings, the LNS Object Server, Trace Options, and the LNS DDE Server License Settings.
Networks/ Configure All	Opens a configuration dialog with a tab for each network defined in the LNS Server (up to 50 networks).
Networks/ Import	Opens a configuration dialog on which you can name a network you want to import and configure the settings of that network.
Topics/ Configure All	Opens a configuration dialog with a tab for each custom topic.
Topics/ Create New	Creates a new custom topic.
Topics/ Delete	Deletes a custom topic.

### Messages

Log Display/ Information	Toggles displaying of information events.
Log Display/ Warnings	Toggles displaying of warning events.

Log Display/ Errors	Toggles displaying of error events.
Log Display/ Trace	Activates the tracing option.
Log Display/ Freeze	Stops automatic scrolling of the log.
Log Display/ Go to Top	Go to the top of the log.
Log Display/ Go to Bottom	Go to the bottom of the log.
Log Display/ Clear	Clear the contents of the log.
Log File/ Log to File	When enabled causes all log and trace messages to be written to a file on disk.
Log File/ Clear	Clears the active log file. Enabled only when there are messages present in the log file.
Log to WWLogger	Toggles sending log messages to the Wonderware WWLogger logging application. (if available)

## Help

Help Topics	Opens the LNS DDE Server help file.
About LNS...	Opens the About dialog.

---

## Toolbar

There are three toolbars: Log File, Log Display, and Miscellaneous. These toolbars contain buttons that activate certain menu options with a single mouse click.

### Log File

	Save Log File	Saves the current contents of the logfile.
	Log to File	Logs messages to a file.
	Clear Log File	Clears the active logfile.

### Miscellaneous

	Copy Link	Copy link information for the currently selected object in the folder or detail pane to the clipboard.
	About	Opens the About dialog.
	Help	Opens the online help file.
	Large Icons	Presents the icons in large-sized format.
	Small Icons	Presents the icons in small-sized format.
	List	Presents the log in a list format.
	Details	Provides a description of log listings.

### Log Display

	Bottom of Log	Scrolls to the bottom of the log.
	Top of Log	Scrolls to the top of the log.
	Freeze	Stops automatic scrolling of the log.
	Clear Message Display	Clears the log display.
	Display Information	This setting controls filtering of information messages for the log. Click this setting to display information messages in the log. The default setting is On.
	Display Warnings	This setting controls filtering of warning messages for the log. Click this setting to display warning messages in the log pane. This setting should always be enabled. The default setting is On.
	Display Errors	This setting controls error message filtering for the log. Click this setting to display error messages in the log pane. This setting should always be enabled. The default setting is On.

---

## Using the LNS DDE Server Examples

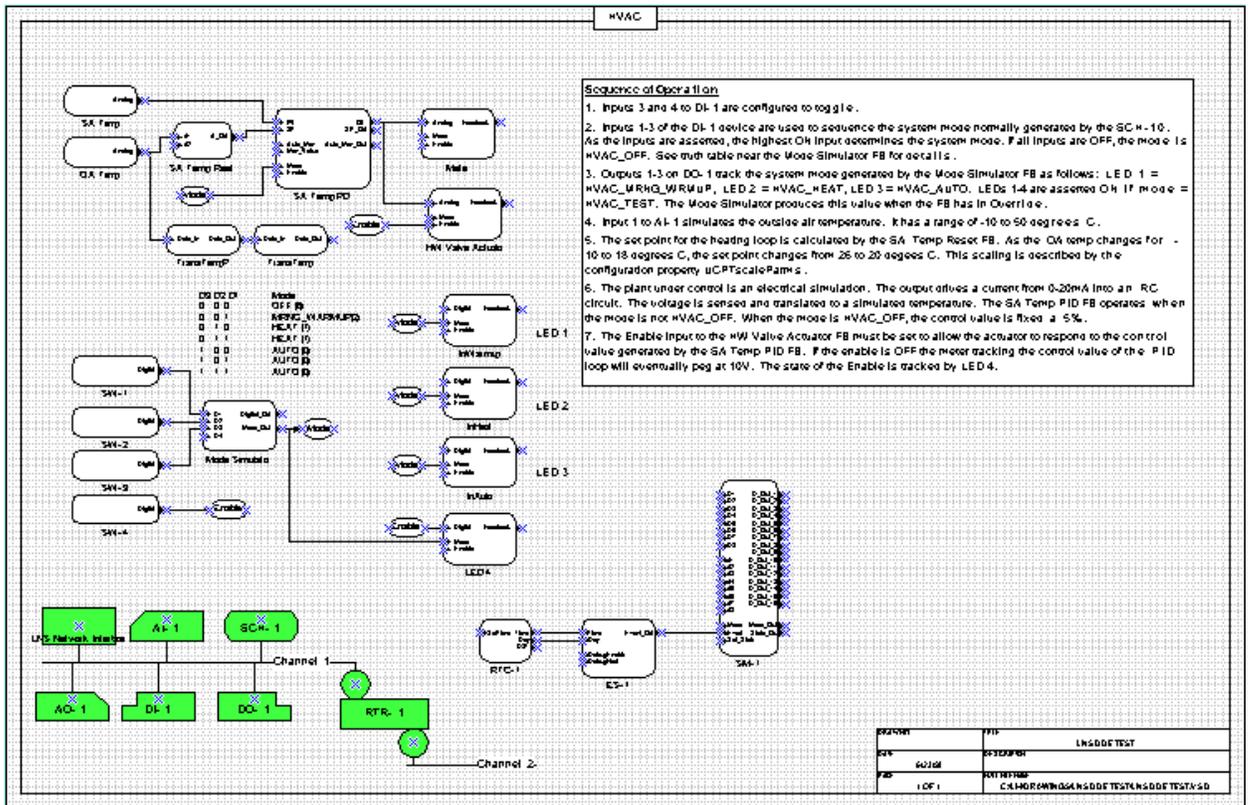
Two example applications are provided with the LNS DDE Server. They are an Excel 97 spreadsheet and an InTouch 7.0 application. These examples are installed if you select the `Example applications` option during the LNS DDE Server installation procedure. You can rerun the LNS DDE Server installation program and select the `Example applications` option if you did not initially install the examples.

Many of the examples in this guide come from these two examples.

---

## *Using the Example LNS Database*

The LNS database for both examples was created using the LonMaker Integration Tool. Two LonMaker backup files are provided: `LNS DDE Test V2.zip` and `LNS DDE Test V3.zip` in the `LNSDde` folder. These backup files each contain a LonMaker drawing and an LNS database for use with the Excel and InTouch examples. `LNS DDE Test V2.zip` contains a LonMaker database and drawing with version 2 LonPoint applications. `LNS DDE Test V3.zip` contains a LonMaker database and drawing with version 3 LonPoint applications. The following figure is the LonMaker drawing for the LNS DDE Test example.



The LNS DDE Test example includes 5 application devices and a router, which matches the hardware in a LonPoint Demo Kit. If you do not have LonPoint devices to use with the examples, you can still open the examples in engineered mode to see Excel and InTouch usage examples for the LNS DDE Server.

Following is a list of the 5 application devices defined in the LNS DDE Test example:

- **AI-1.** A LonPoint AI-10 Analog Input Interface Module. Used to monitor a simulated temperature and a temperature setpoint dial.
- **AO-1.** A LonPoint AO-10 Analog Output Interface Module. Used to control a simulated hot water valve actuator, and an analog meter monitoring the input to the valve.
- **DI-1.** A LonPoint DI-10 Digital Input Interface Module. Used to monitor 4 input switches. Three of the switches are used to control the system mode, simulating the output of a system mode generator. The fourth switch is used as a system enable.
- **DO-1.** A LonPoint DO-10 Digital Output Interface Module. Used to control 4 LEDs. Three of the LEDs are used to display the system mode. The fourth LED is used to display the system enable status.
- **SCH-1.** A LonPoint SCH-10 Scheduler Module. This device is not used in the example. Its operation is simulated by the mode simulator on the DI-1 device.

These five devices contain LONMARK objects that are shown as functional blocks on the LonMaker drawing. Each functional block has network variable inputs and outputs that are shown as triangles on the functional blocks. These network

variables are connected as shown on the LonMaker drawing. The following list summarizes the Sequence of Operations for the LNS DDE Test example.

## Sequence of Operation

1. Inputs 3 and 4 to DI- 1 are configured to toggle.
2. Inputs 1-3 of the DI- 1 device are used to sequence the system mode normally generated by the SCH-10. As the inputs are asserted, the highest ON input determines the system mode. If all inputs are OFF, the mode is HVAC\_OFF. See the truth table near the Mode Simulator functional block for details.
3. Outputs 1-3 on DO- 1 track the system mode generated by the Mode Simulator functional block as follows: LED 1 = HVAC\_MRNG\_WRMUP, LED 2 = HVAC\_HEAT, LED 3 = HVAC\_AUTO. LEDs 1-4 are On if mode = HVAC\_TEST. The Mode Simulator produces this value when the functional block is in Override.
4. Input 1 to AI- 1 simulates the outside air temperature. It has a range of -10 to 50 deg. C.
5. The setpoint for the heating loop is calculated by the SA Temp Reset functional block. As the OA temp changes from -10 to 18 degrees C, the setpoint changes from 26 to 20 degrees C. This scaling is described by the UCPTscaleParms configuration property.
6. The plant under control is simulated by a resistor-capacitor (RC) circuit. The output drives a current from 0-20mA into the RC circuit. The voltage is sensed and translated to a simulated temperature. The SA Temp PID functional block operates when the mode is not HVAC\_OFF. When the mode is HVAC\_OFF, the control value is fixed at 5%.
7. The Enable input to the HW Valve Actuator functional block must be set to allow the actuator to respond to the control value generated by the SA Temp PID functional block. If the enable is Off the meter tracking the control value of the PID loop will eventually peg at 10V. The state of the Enable is tracked by LED 4.

---

## Using the LNS DDE Example Database

To use either example, you must first load the LNS DDE Test database on your PC. If you are using the LonMaker Integration Tool, follow these steps to load the database:

1. If you have not already installed the LonPoint plug-in, run the LonMaker setup program from the LonMaker CD and install the LonPoint plug-in.
2. Start the LonMaker tool by selecting *LonMaker* from the Windows *Start* menu. The LonMaker design manager opens.
3. Click the Restore button. The LonMaker Restore dialog opens.

4. Click the Browse button. The Select Backup File dialog opens.
5. Browse to the LNS DDE Test V2.zip or LNS DDE Test V3.zip file in the LONWORKS LNSDde folder and click Open, then click OK. The Confirm Restore dialog opens.
6. Confirm the folders for the drawing and database files. To change the folders, click the Change Directory button. Once you have specified the desired folders, click OK. A dialog confirming that you want to open the restored drawing is displayed.
7. Confirm the LNS database path and click Next.
8. Click the Network Attached option if you have LonPoint devices to test the example with, otherwise deselect the Network Attached option to go into engineered system mode.
9. Select the default options for the remaining dialogs of the Network Wizard. Click Next for each dialog, and click Finish for the final dialog. A dialog suggesting that you resynchronize the drawing to the network is displayed.
10. Click No, since your LonPoint devices are physically different devices from those saved in the backup file.
11. If you are attached to the network and have LonPoint devices, select all of the application device shapes in the lower left of the LonMaker drawing, right-click one of the selected devices, and select the Replace command. Commission each of your devices, selecting all default options, except that you can bring each device up online instead of the default offline.

If you are using another installation tool besides the LonMaker tool, unzip the LNS database from the LNS DDE Test V2.zip or LNS DDE Test V3.zip file, and restore the database to your PC as described in the documentation for your installation tool. Once you have imported the LNS database, and optionally replaced the devices, you can exit your installation tool.

---

## *Using the Excel Example*

To use the Excel example, you will need Excel 97 or newer installed on your PC with the LNS DDE Server. You will also need a LonPoint Demo Kit to run the example, but you can open the example in engineered mode if you do not have LonPoint devices.

To open the Excel example, follow these steps:

1. Restore the LNS DDE Test drawing and database as described in the previous section.
2. Select the Microsoft Excel shortcut from the Example Application folder in the Echelon LNS DDE Server program folder.
3. If Excel displays a macro warning, click the Enable Macros button. A dialog confirming that you want to update links is displayed.
4. Click Yes to update links. A dialog confirming that you want to start the LNS DDE Server is displayed.

5. Click Yes to start the LNS DDE Server.
6. If a blank spreadsheet opens over the Excel example spreadsheet, close the blank spreadsheet.
7. Excel may time out while the LNS DDE Server starts. If it does, click on cell A7 (the Analog value), click anywhere within the cell formula in the formula bar, and press the Enter key. This will cause Excel to refresh all links.

The Excel example demonstrates monitoring for a number of analog and digital points, and one mode output. These points are all defined in the LNS DDE Test database.

The Excel example also demonstrates how an Excel button can be used to update a digital input. To change the input to LED 4, enter a new value in cell B17 and click the POKE Enable on LED 4 button. This will invoke an Excel macro to update LED 4 as described in Chapter 5. For example, to turn on the LED, enter a value of “100.0 1” in cell B17 and click the POKE Enable on LED 4 button. The LED 4 input is also connected to an Enable output, so you may see the LED flash briefly if you change its state.

---

## *Using the InTouch Example*

To use the InTouch example, you will need InTouch 7.0 installed on your PC with the LNS DDE Server. You will also need a LonPoint Demo Kit to run the example, but you can open the example in WindowMaker without an attached network if you do not have LonPoint devices. To open the InTouch example, follow these steps:

1. Import the LNS DDE Test drawing and database as described in *Using the Example LNS Database* earlier in this chapter.
2. Select the InTouch WindowMaker shortcut from the Example Application folder in the Echelon LNS DDE Server program folder.
3. To switch to runtime mode, click Runtime! on the InTouch menu. A dialog confirming that you want to start the LNS DDE Server is displayed.
4. Click Yes to start the LNS DDE Server.

See Appendix E for a description of how to use the InTouch example, and for a brief introduction to InTouch.



# 3

## Configuring the LNS DDE Server

This chapter discusses how to configure the LNS DDE Server.

---

## Configuration Overview

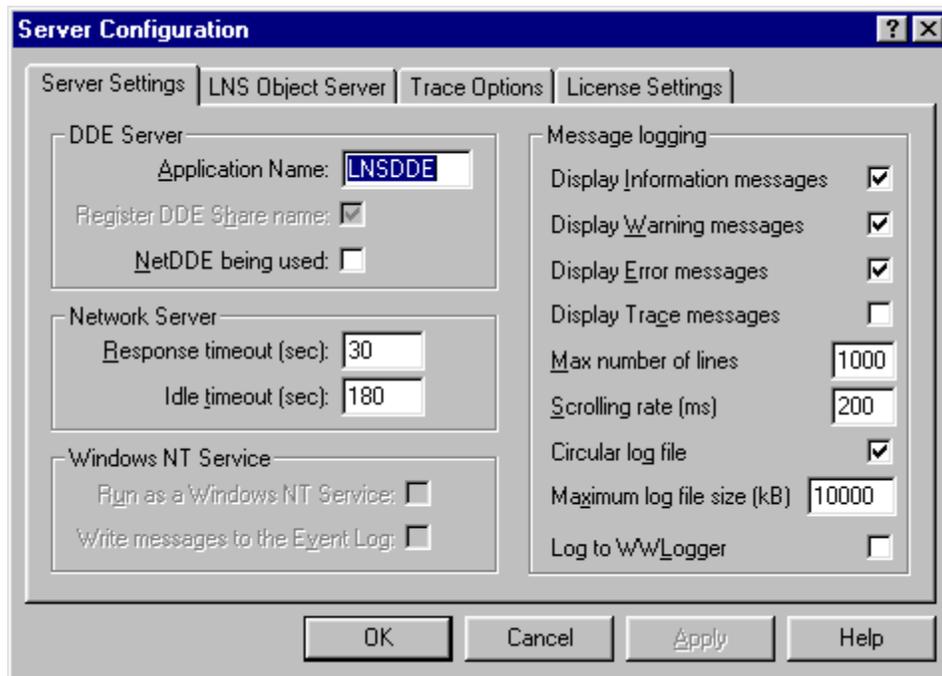
When the LNS DDE Server starts up the first time, it will run using its initial default configuration settings. The LNS Server determines some of these settings, so the initial defaults may have been modified if you have previously used other LNS applications. The LNS DDE Server will automatically detect and use LNS network interfaces and LNS networks on the local PC.

The following sections describe the LNS DDE Server configuration settings. Settings that are unique to the LNS DDE Server are stored entirely in the system registry of the local PC that the LNS DDE Server is operating on. Settings that affect the LNS Server are stored in the LNS Server database.

---

## Server Settings

To configure the LNS DDE Server, select the *Server* command in the *Configure* menu. You can also right-click the *LNS DDE Server* item at the top of the folder pane and select *Configure* from the shortcut menu. The Server Settings tab of the Server Configuration dialog appears:



The settings in this dialog are described in the following sections.

### Application Name

This setting specifies the DDE Application Name of the LNS DDE Server. This is not the filename on disk of the server. It is the name that DDE client applications need to use in order to be able to access the LNS DDE Server. The default setting is “LNSDDE”.

If this item is changed, the server will need to be stopped and restarted in order for the change to have an effect. A series of dialogs will help you through the restart process.

### Register DDE Share Name

By default, the LNS DDE Server creates a DDE share name specifically for itself. In some instances, this has caused conflicts with applications that set up application-specific shares. This option allows you to turn off the generation of DDE Share names by the LNS DDE Server.

### NetDDE Being Used

This setting influences how quickly the LNS DDE Server responds back to a DDE client when the client initially requests a value. Under normal circumstances the server may wait up to a minute before it responds back to a client that it is unable to deliver a valid value. When this option is checked, the server will report back immediately, usually with a timeout.

This option is provided for applications to allow them to bypass network delays when operating over a network using the NetDDE protocol. Although the server will initially report a timeout, the server will subsequently report an actual value when available, and most clients will handle this update correctly.

This option should be selected for DDE client applications using the DDEML API. The DDEML API requires immediate notification when setting up an advise, otherwise the DDEML API may disconnect the conversation if it takes more than 60 seconds for the LNS DDE Server to retrieve the point value. When this option is selected, the LNS DDE Server will send the client an immediate notification for each advise request made.

This setting should not be enabled for most networks, even when a client operates over a network using NetDDE.

See *Using NetDDE* in the LNS DDE Server Help File for more information.

## Response Timeout (sec)

This setting controls how long a time the LNS DDE Server will wait for a response from the LNS Network Server application. The LNS Network Server application runs in the background as a separate process and handles the communication between the LNS DDE Server and the LNS network database.

Some operations, particularly opening the network database and going OnNet, may take more than a few seconds to complete. The timeout value should take into account the maximum delay that may occur. The value is specified in seconds. The default is 30 seconds. This value may be set between 5 and 1000 seconds. If you have an SLTA-10 on a modem, specify at least 50 seconds.

The timeout for the LNS DDE server to open an LNS network database is 10 times the value in this field (i.e. if the response timeout is set to the default of 30 seconds, the network database open timeout is 5 minutes).

## Idle Timeout (sec)

The LNS Network Server is launched automatically by the LNS DDE Server when a network is opened. It runs in the background as a separate process. One instance of the LNS Network Server may be running at any time for each open network.

When an instance of the LNS Network Server is no longer being used, the LNS DDE Server will terminate it after the idle timeout timer expires. Keeping an LNS Network Server running idle for a longer time will speed up reconnecting to a network, but will decrease the amount of memory that is available for other applications. Valid timeouts are 5 seconds to 86400 seconds. The default is 180 seconds.

## Run as a Windows NT Service

The LNS DDE Server, when used on Windows XP, Windows 2000, or Windows NT 4.0, can run as a service. Windows XP, Windows 2000, and Windows NT 4.0 operating systems terminate standard applications when the current user logs off. Services can run continuously under this process, allowing you to run monitoring and control applications continuously, even as users are logging on and off. In addition, running as a service allows the LNS DDE Server to be configured to start up automatically when Windows XP, Windows 2000, or Windows NT 4.0 is started, and to be started without a user being logged on.

When running as a service, all DDE Clients, including those that run on the same computer as the service, must include the name of the computer as if the client was accessing the LNS DDE Server on another computer. In most DDE Client

applications, for example Microsoft Excel, this means prepending the computer name to the application name as in the example,  
=`\\MYCOMPUTER\LNSDDE|'MyNet.Subsystem 1.DevNV'|'MyDev.NV2'`. Always specify the computer name when the LNS DDE Server is running as a service.

While the LNS DDE Server is running as a service, it is not possible to start the LNS DDE Server as a normal desktop application. You also cannot start the LNS DDE Server as a service if it is already running as a desktop application. In both cases, the second instance of the server will discover that there is already another instance of it that is running. When this happens, a message is displayed indicating only one instance of the server is allowed. When running as a service, the LNS DDE Server can be configured to interact with the user desktop, providing the same interface as is available when running as a desktop application.

To configure the LNS DDE Server to run as a service, follow these steps:

1. Select *Server* from the LNS DDE Server's *Configure* menu. The Server Configuration Menu will open.
2. Check the *Run as Windows NT Service* option and click OK. The LNS DDE Server will be installed as a service that will start up automatically when the computer is restarted. To start running the LNS DDE Server as a service immediately, continue to Step 3.
3. Close the current instance of the LNS DDE Server.
4. Open a DOS prompt and enter the following command:

```
NET START LNSDDE
```

To stop the LNS DDE Server from running as a Service without restarting the PC, open a DOS prompt and enter the following command:

```
NET STOP LNSDDE
```

The *Run as Windows NT Service* option will be disabled if the current user does not have privileges to control services on the PC (i.e. Users with Administrator privileges or who have specifically been granted privileges to control services).

## Write Messages to the Event Log

When the LNS DDE Server is running as a service, the messages that are otherwise displayed in the logging pane may optionally be written to the Windows NT Event Log. The contents of the Event Log may be viewed with the Event Viewer application found in the Administrative Tools folder. By enabling this option the server will write the same messages that it otherwise displays in the logging pane to the Event Log.

**Note:** This option is only available if the *Run as Windows NT Service* option is set, and the LNS DDE Server is running on Windows XP, Windows 2000, or Windows NT 4.0.

### **Display Information Messages**

This setting controls filtering of information messages for the log. Click this setting to display information messages in the log. The default setting is On.

### **Display Warning Messages**

This setting controls filtering of warning messages for the log. Click this setting to display warning messages in the log pane. This setting should always be enabled. The default setting is On.

### **Display Error Messages**

This setting controls error message filtering for the log. Click this setting to display error messages in the log pane. This setting should always be enabled. The default setting is On.

### **Display Trace Messages**

This setting lets you view trace messages. Trace messages are stored in an internal program buffer. The default setting is Off.

### **Max Number of Lines**

This setting limits how many lines that the log may hold. When the limit is reached, any new log lines will cause the oldest log lines to be deleted. Valid limit values are 10 lines to 1000 lines. The default is 200 lines.

### **Scrolling Rate (ms)**

This setting controls the minimum time between updates to the log pane. The value is specified in milliseconds. Valid settings are 100 ms to 2000 ms. The default is 200 ms.

### **Circular Log File**

A circular log will partition the maximum log file size into equal lots and override the oldest messages with new messages when the log is full. Without a circular log, logging will stop when either the maximum log file size is reached or the disk containing the log file runs out of free space. The default setting is On.

## Max Log File Size

The log file size is user configurable, in kilobytes. The default log behavior is to stop logging messages when the maximum log size is reached. The minimum log size is 10KB, the maximum is 100,000KB.

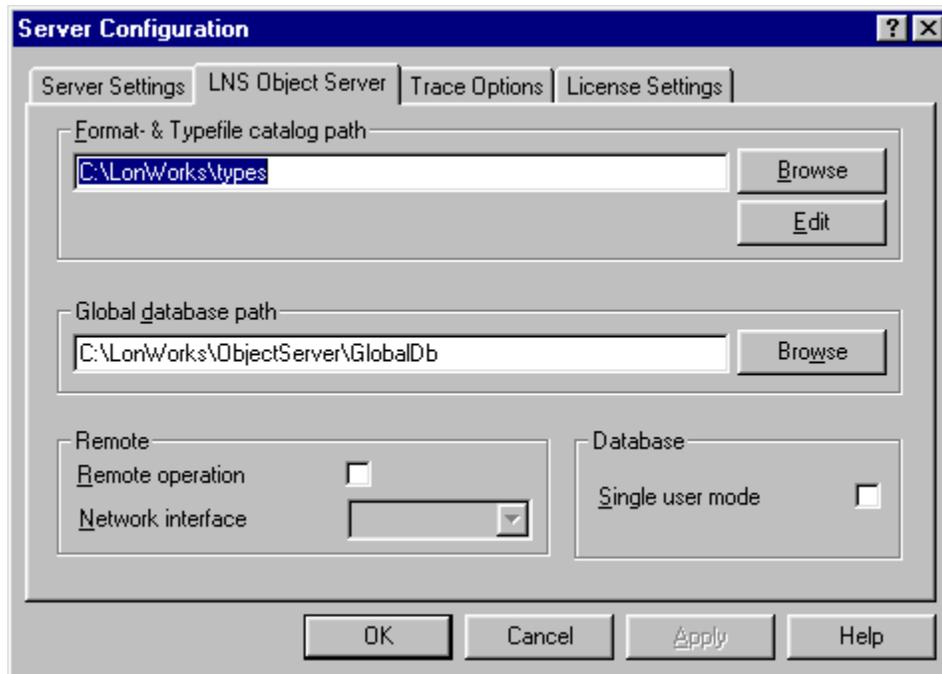
## Log to WWLogger

This setting controls if log messages displayed in the log pane will be forwarded to the Wonderware WWLogger application if it is installed and running. This option causes a slight performance penalty if the WWLogger application is running — there is no penalty if the WWLogger application is not installed or not running. The default for this setting is Off.

---

## LNS Object Server Settings

To configure the LNS Object Server, select the LNS Object Server tab. This tab is shown in the following figure.



The settings in this dialog are described in the following sections.

---

### *Format and Typefile Catalog Path Setting*

This setting specifies the folder that contains the LNS resource file catalog. This catalog identifies the resource files that are installed on your PC. This setting affects both the LNS Server and the LNS DDE Server.

To select another folder, either enter it manually or use the Browse button to search for one. The server validates the path entered when you press the *Apply* or *OK* buttons.

Use the *Edit* button to start the LNS Resource File Catalog Utility. This utility can be used to edit the catalog file. Folders containing format and other resource files may be added or removed from the catalog. You can combine changing the folder and editing the catalog. If you start the LNS Resource File Catalog Utility and the selected folder does not contain a catalog, a new and empty catalog is created automatically.

When changing the folder, or when adding or removing a folder from the catalog, you must stop and restart the LNS DDE Server for the change to take effect.

See Appendix A for more information on resource files.

---

## Global Database Path Setting

This setting specifies the folder where the LNS Server's global database is stored. This database contains information about the names and locations of all the networks that are managed by the LNS Server. The specified folder must exist and contain a valid LNS global database. In most cases this setting should never be changed from the default setting. Entering a folder that does not exist or that contains a corrupted database will render the LNS Server and LNS DDE Server inoperable.

Use the Browse button to locate the global database folder. To select another folder, either enter it manually or use the Browse button to search for one. The server validates the path entered when you press the *Apply* or *OK* buttons.

---

## Remote Settings

### Remote Operation

This setting specifies that the LNS Server is running on another PC; in this situation, the DDE Server runs as a *remote client*. There are two types of remote clients, full and lightweight. A full client runs on a PC other than the LNS Server and is attached to a LONWORKS or LONWORKS/IP channel (a channel which sends LONWORKS messages within IP packets). A lightweight client runs on a PC other than the LNS Server and is attached to an LNS/IP channel (a channel which sends LNS messages bundled within IP packets).

See the *LonMaker User's Guide* or the *LNS for Windows Programmer's Guide* for more information about lightweight and full clients.

In order to run as a remote full or lightweight client, the LNS Server must be running on a PC that is attached to the same network as the PC running the LNS DDE Server (but not necessarily the same channel). When remote client operation is enabled, all requests to look up network, subsystem, device, LONMARK object, network variable, and configuration property names are forwarded to the PC that

hosts the LNS Server. When running as a remote full client, network variable values are not read from or written to the LNS Server, instead, they are read from and written directly to the devices with the network variables. Configuration property updates are always managed through the LNS Server. You will typically see better performances with remote full clients on LONWORKS/IP channels than remote lightweight clients on LNS/IP channels. However, the LNS DDE Server can only access one remote network at a time when running as a full client but can access up to 100 networks when running as a lightweight client.

The LNS DDE Server can be attached to a LONWORKS, LONWORKS/IP or LNS/IP Channel. A LONWORKS/IP or LNS/IP Channel may extend over the Internet or a LAN.

To run as a remote lightweight client, follow these steps:

1. Open the Windows Start menu, point to Echelon LNS Utilities, and then click LNS Remote Client Configuration Utility. The LNS Remote Client Configuration Utility appears.
2. Create entries for each remote network to be accessed. Press F1 for help on using this utility.
3. Click Done. The LNS Remote Client Configuration Utility closes.
4. Start the LNS DDE Server.
5. Open the Configure menu and then click Server. The Server Configuration dialog opens.
6. Click the LNS Object Server tab.
7. Set Remote Operation.
8. Select TCP/IP for the Network Interface.
9. Click OK. The Server Configuration Dialog closes.
10. Quit and restart the LNS DDE Server. The networks that you added in step 2 appear in the Networks folder. You can open any or all of these networks at the same time.

To run as a remote full client, follow these steps:

1. If you are running the LNS DDE Server on a LonWorks/IP channel, create a channel that includes your computer and any i.LON 1000 Internet Servers using the i.LON Configuration Server. See the i.LON documentation for details. Then, open the Windows Control Panel and start **LonWorks/IP Channels**. Use this application to define your computer as a LonWorks/IP device. Press F1 for help on using this application.
2. Start the LNS DDE Server.
3. Open the Configure menu and then click Server. The Server Configuration dialog opens.
4. Click the LNS Object Server tab.
5. Set Remote Operation.
6. Select either the LonWorks/IP network interface that you created in step 1, or any other network interface other than TCP/IP.
7. Click OK. The Server Configuration Dialog closes.
8. Quit and restart the LNS DDE Server. The networks that can be accessed by your selected network interface appear in the Networks folder. You can only open one of these networks at a time.

## Network Interface

This property can only be set if the Remote Operation option is set. If the Remote Operation option is set, this field contains a list of all available network interfaces and a TCP/IP selection. If you are connecting to the LNS DDE Server as a Full Client, select the LNS Network Interface you will use. Using the LonWorks/IP Channels control panel in the Windows control panel, an Ethernet card can be configured as a LONWORKS/IP device, and this device can be selected as the network interface. If you are connecting to the LNS DDE Server as a Lightweight client, select TCP/IP.

---

## *Database Settings*

### Single User Mode

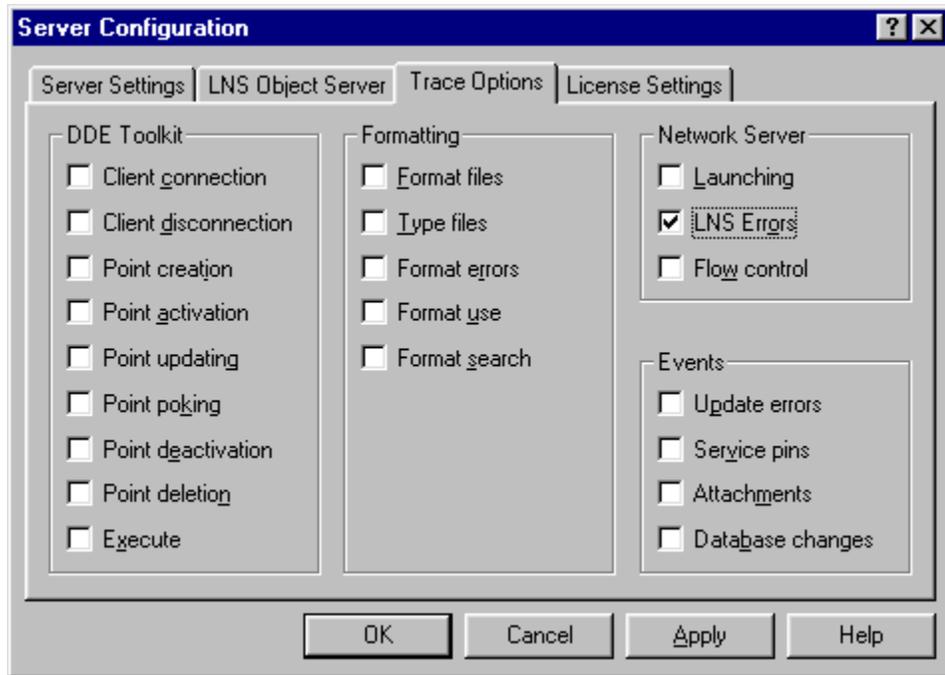
This setting specifies that the LNS Server is to run in single user mode. In this mode, only a single LNS application can run at a time. Do not enable this setting unless you are sure that only a single LNS application will be running at a time. The only advantage of single user mode is a very slight performance improvement.

---

## Trace Options

The LNS DDE Server contains several trace options that can be used to log internal events in the server. To change these options, open the Trace Options dialog (shown below) by selecting Configure/Server from the menu, and clicking the Trace Options tab.

By default all message types—information, warning, error, and trace—are written to the log file when writing to the log file is activated. In the log pane in the lower portion of the main window, only the message types selected in the Messages/Log Display menu are shown.




---

## DDE Toolkit

This section of the trace options concerns the interaction between the LNS DDE Server and the Wonderware I/O Server Toolkit (formerly the Wonderware DDE Server toolkit), which is the component of the LNS DDE Server that handles the DDE messaging between the client applications and the DDE Server. Many of these trace options produce output that is difficult to interpret without knowledge of the Wonderware I/O Server toolkit API. Some of the options produce a significant amount of output and should be enabled only when absolutely necessary.

### Client Connection

This option generates a trace message once for each unique topic that is opened. The message contains information about the topic name and the names used internally by the server to identify this topic. If more than one client application connects to the same topic, only the first connection will generate a message.

### Client Disconnection

This option generates a message whenever the LNS DDE Server closes a topic. The message contains the internal name.

### Point Creation

This option generates a message containing the internal names of the topic and its point, the item name string, and the type of data (discrete, integer, real, or string)

when a point is created. This option will generate a message for each point that is accessed. The server will not start reading the point until it has been activated.

## Point Activation

This option generates a message containing the internal names of the topic and point when a point is activated, and the result of the operation (typically “True”). The server does not start to read a point until it has been activated.

## Point Updating

This option generates a message each time the server internally updates a point. The message contains the internal names of the topic and point, and the value. The server uses report by exception exclusively. This means that if the update value is the same as the previous value, then the client or clients will not see the update.

## Point Poking

This option generates a message when a Poke (write) request is received from a client. The message contains the internal names of the topic and point, the value to write, and the result code of the request. The result code of the poke request only indicates that the value has passed a formatting test and that the poke has been added to the write request list.

## Point Deactivation

This option generates a message containing the internal names of the topic and point when a point is deactivated. The server will stop reading the point when it is deactivated.

## Point Deletion

This option generates a message containing the internal names of the topic and point, and the result of the deletion, when the point is deleted.

## Execute

This setting generates a message containing the contents of each Execute message and the internal topic name on which it was received. The server only responds to *Pause* and *Resume* Execute messages.

---

## Formatting

The trace options in this group generate messages on how the network data is interpreted and formatted by the server.

## Format Files

This option generates a message each time a format file is opened. The full path of the format file is included in the message. The server normally opens all registered format files during startup.

## Type Files

This option generates a message each time a type file is opened. The full path of the type file is included in the message. The server normally opens all registered type files during startup.

## Format Errors

This option generates a message when a format error occurs. The message contains information about the point in error, the internal name of the format file, the error code, and in some cases the string that caused the error.

A formatting error occurs when the formatted or binary data contains invalid data. The data is considered invalid when it does not conform to the constraints for the data type of the point.

## Format Use

This option generates one or two messages for each point (network variable, configuration property, or field thereof), that is accessed by a client application.

The first message contains information about the point name, field name (if applicable), format file, format specification handles, number of fields, and length (in bytes). A second message will be generated if a type has been explicitly specified by the client application by using the '-t' option.

## Format Search

This option generates a format search message when the server is unable to locate a format specification. The server will print out the program ID of the device containing the point (the point may be a network variable or configuration property), the scope selector, index, and any explicitly assigned format specification as registered in the LNS network database.

---

## Network Server

Communication between the LNS DDE Server and the LNS network operating system is handled by a separate process. This process is the LNS Network Server. The LNS Network Server is started, stopped, and controlled by the LNS DDE Server application. The LNS Network Server has no visible window.

One instance of the LNS Network Server runs for each network that is open. One additional instance is used by the server to list the available networks.

## Launching

This option prints additional information during startup of an LNS Network Server process. The information may be useful when there is a problem starting the LNS Network Server. In most cases, however, normal error reporting should be sufficient to determine the cause of a startup problem.

## LNS Errors

This option is not used. By default, all LNS errors that are not expected and handled will cause a warning or error message to be generated.

## Flow Control

This option generates a message when the request buffer from the LNS DDE Server and the LNS Network Server becomes full. A second message is generated when the LNS Network Server is ready to receive more requests.

---

## Events

The LNS network operating system produces a number of events that LNS applications, like the LNS DDE Server, can subscribe to. Some of the events are used internally by the LNS DDE Server.

## Update Errors

Update errors are generated by the LNS Data Server when it is unable to read a network variable. This option generates a message containing an internal index number (this number is used by the server to identify a network variable) and the handle of the network variable causing the update error.

The LNS DDE Server modifies the poll interval when update errors occur. When an update error occurs, the server will try to recover the point based on the Point Recovery Strategy setting. When the LNS Data Server can read the network variable, the LNS DDE Server will restore the poll interval. The LNS DDE Server also increments a counter for the number of update errors that have occurred.

## Service pins

This option generates a detailed message when a service pin message is received. The message contains the handle of the device that generated the service pin message, the Neuron ID, program ID, and location. The server uses this message to increment a counter of the number of service pins received from the device.

If this option is disabled, a summary message will be generated when a service pin message is received. The message contains only the device path (subsystem path and device name) if available or the Neuron ID of the device which sent the service pin message.

## Attachments

Attachment events are generated for devices that have ping enabled. You can use an LNS tool such as the LonMaker Integration Tool to enable and disable ping for a given device. The first time a device responds to a ping, an attachment message is generated if this option is enabled. When a device has not responded to three consecutive pings, a detachment message is generated if this option is enabled.

The server uses the attachment events to manipulate the poll interval for the network variables it accesses in the device. When a detachment occurs, the server will increase the poll interval to prevent network delays from happening due to excessive retries. After a detachment has occurred, the following attachment event will restore the normal poll interval.

Attachment events are displayed as information and warning messages. The trace message contains information about the internal name of the device for which an attachment event is generated, and whether it was an attachment or detachment.

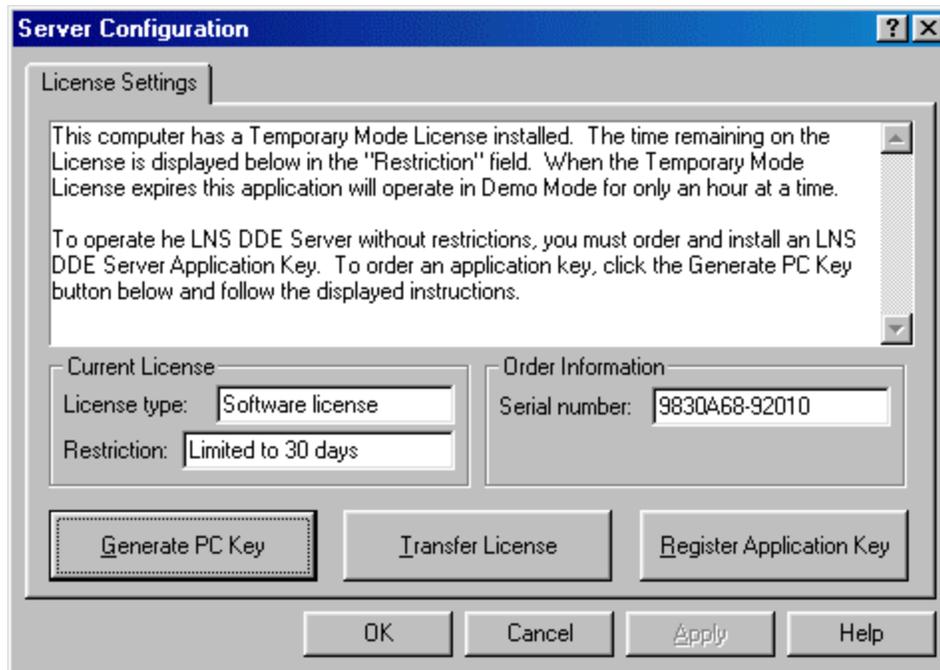
## Database Changes

This option generates a message when a change occurs in the LNS Server database. This event is not used by the LNS DDE Server. A database change occurs when an LNS tool adds or removes any of the following: a channel, a device, a router, or a subnet.

---

## License Settings

The License Settings dialog contains information regarding the current status of your LNS DDE Server license and allows you to register the product once a permanent license is purchased. This dialog appears as follows (the displayed text may be different depending on your LNS DDE Server supplier):



The LNS DDE Server has three licensing modes. The first is *Temporary Mode*. If there is no license installed on a PC, a Temporary Mode license is generated when the LNS DDE Server is first started. In Temporary Mode the LNS DDE Server will run on an otherwise unlimited basis for 30 days. You must acquire an application key as described in Register Application Key later in this chapter to continue to use the LNS DDE Server after the Temporary Mode license expires.

The second licensing mode is *Demo Mode*. In Demo Mode the LNS DDE Server will operate for approximately one hour at a time. You will be able to restart the LNS DDE Server when using Demo Mode, but you must close the LNS DDE Server application and restart it every hour. The LNS DDE Server will operate in Demo mode when the Temporary Mode license expires.

The third licensing mode is *Unlimited Mode*. In Unlimited Mode the LNS DDE Server will operate continuously without interruption. You must acquire an application key as described in *Register Application Key* to enable Unlimited Mode.

---

## Current License

### License Type

This field will display “No license” if you are running in Demo Mode. It will display “Software license” if you are running in Temporary Mode or Unlimited Mode.

### Restriction

This field shows any restriction placed on the installed license. If you are running in

Temporary Mode, this field will show the restriction placed on the license. The restrictions that can be placed on a license include duration and number of runs. If the license is restricted by duration, this field will say how many days that remain on the license.

If an unrestricted license is installed this field will say *None*.

---

## *Serial Number*

This field holds the serial number specified during installation of the LNS DDE Server. This number must be used when ordering a license. In case no serial number was specified during installation, or the serial number needs to be changed, type in the correct serial number in this field and click *Apply*.

The serial number is located on your LNS DDE Server case or envelope, or was provided to you when you downloaded the LNS DDE Server software.

---

## *Generate PC Key*

When you click this button, a window appears asking if you want to generate a PC Key. Click *Yes* to open a window which contains your PC Key and information on ordering an application key. This also copies the order form to the clipboard. You can paste this order form into an email, word processor, or notepad application (use email if available). If you need to get another copy of this form, click this button again and click *Yes* in the following window.

The PC Key is a unique number calculated from the unique properties of the PC the LNS DDE Server is installed on. This number must be specified when ordering a license.

When a PC Key is generated for the first time, a new Temporary Mode license is created. Any pre-existing Temporary Mode license is replaced. Only one additional Temporary Mode license will be created, even if you generate multiple PC keys.

**Note:** When a license is installed, or a restricted license expires, a new PC Key must be generated. When you order a license, specify the restriction shown in this field if a restricted license is installed.

---

## *Creating an Unlimited Mode License*

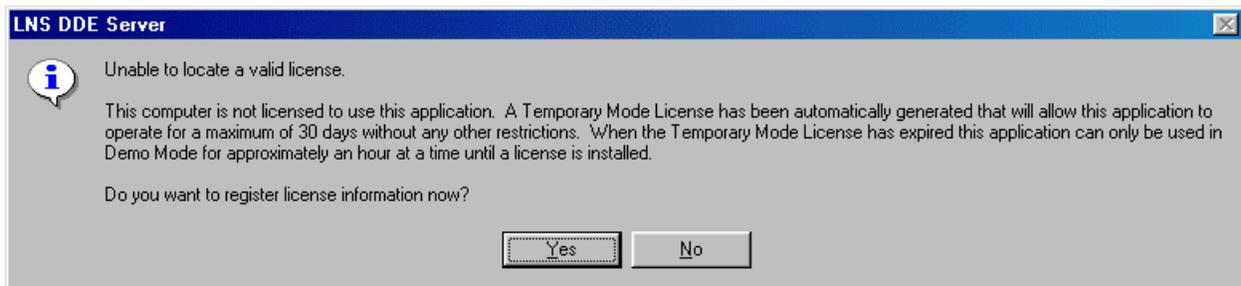
If you have not installed an application key for the LNS DDE Server, the LNS DDE Server will automatically generate a Temporary Mode license (see *License Settings*, earlier in this chapter). This Temporary Mode license will last for 30 days. If it runs out before you acquire and enter an Unlimited Mode license, you will only be able to run the LNS DDE Server in Demo mode.

An LNS DDE Server Application Key is required to enable Unlimited Mode. A separate application key is required for each PC that will be using the LNS DDE Server in Unlimited Mode, or that will be accessing a remote LNS DDE Server via

LONWORKS/IP, LNS/IP, NetDDE or other networking protocol. You may transfer an application key to a new PC as described in the next section. If you lose an application key due to a hardware or software failure, you may order a replacement key at no charge.

If you have installed and licensed a previous version of the LNS DDE Server, you may order an upgrade application key. An upgrade application key is purchased in an identical method to a standard application key, but at a lower cost. To order an application key, follow these steps:

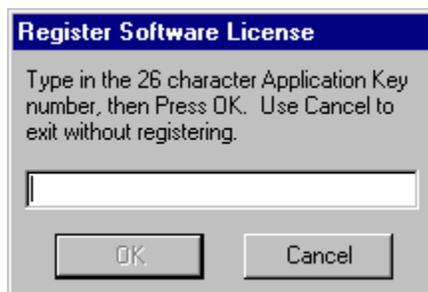
1. Start the LNS DDE Server. If you have not previously entered an application key, the following dialog displays:



2. Click Yes. The License Settings tab of the Server Configuration dialog opens. You can also open this tab at any time from within the LNS DDE Server main window by selecting the *Server* command on the Configure menu and clicking the License Settings tab.
3. If you have not previously entered your serial number, fill it in now in the Serial Number field and click Apply. You will not be able to enter an application key without a valid serial number.
4. Click *Generate PC Key*. A window opens asking you if you want to generate a PC Key. After you generate a PC Key, you will be granted a 30-day Temporary Mode license. This Temporary Mode license will replace any previously existing Temporary Mode license (i.e. the one generated when the LNS DDE Server was first opened). You must enter LNS DDE Server Application Key before the 30 days expire. If the license expires before you have a chance to enter the application key, the PC key will become invalid and you will have generate a new PC Key to order an application key. Click *Yes* to generate the PC Key.
5. A window opens containing the PC Key and informing you that an order form has been copied to the clipboard.
6. Open an email application, a word processor, or a notepad utility and paste the contents of the clipboard into a new document. Use email if possible since it eliminates transcription errors for the PC and application keys. If you don't have access to email or a printer use the order form in Appendix E of the *LNS DDE Server User's Guide*.
7. Fill in the blanks on the form. If this a new order, you must include your credit card or purchase order information. If you have pre-paid for the application key (for example if you have purchased the Model 37200-20 LNS DDE Server OEM

Edition), or if this is a replacement for a lost key, fill in the appropriate fields, and enter 0 as the price.

8. Email the form to the license administrator address at the top of the form, or if you do not have access to email, fax the form to the license administrator fax number at the top of the form. If you are ordering the application key through a distributor, the distributor may provide you with an alternate mailing address or fax number for the license administrator. Upon receipt of your order, the license administrator will provide you with an application key that enables Unlimited Mode for the specified PC. You cannot use this key with an LNS DDE Server installed on another PC.
9. When you receive your application key, click the `Register Application Key` button on the License Settings tab. The following dialog appears:



10. Type the 26 character key and click OK. If you received the application key by email, open the message with the key, copy the key to the clipboard, click the `Register Application Key` button, paste the key, and click OK. If you type in the application key correctly, Unlimited Mode is enabled and the fields in the License Settings dialog will be updated to reflect the change.

---

## Transferring Licenses

If you have created an Unlimited mode license as described in the previous sections, you can transfer the license to another PC by clicking the `Transfer License` button on the same computer, then on the destination computer. Follow the displayed instructions.

When you click this button, the License Transfer application starts. The License Transfer application is used to transfer a license from one computer to another. Only Unlimited Mode licenses can be transferred from or to a computer.

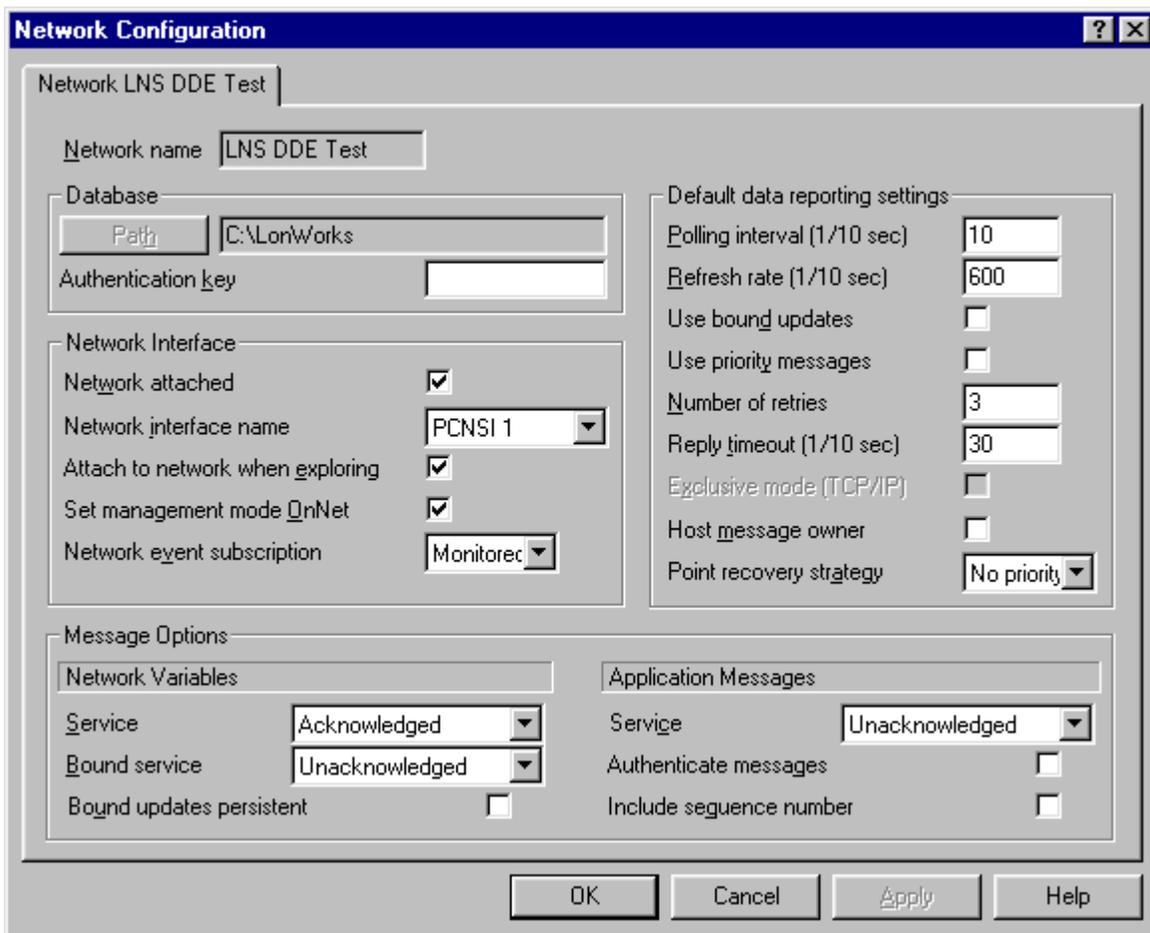
**Note:** *During the time that the License Transfer application is running, the License Settings dialog cannot be closed.*

---

## Network Settings

To configure the LNS DDE Server settings for an individual network, right-click the network name in the folder pane and select `Configure` from the shortcut menu. You can also open `Configure` menu, point to `Networks`, select `Configure All`, and then

select the tab corresponding to the network to be configured. In either case, the following dialog appears with one tab for each LNS Network Database on your PC:



The Network interface name and Set management mode OnNet settings can affect other LNS applications when you open those applications; all other settings apply only to the LNS DDE Server.

Click the OK or Apply buttons to enable your settings. The Cancel button allows you to void your changes. The Help button provides access to the online help files.

The settings in this dialog are described in the following sections.

## Network Name

This field displays the name of the network.

## Database Path

This field displays the pathname for the folder containing the network database.

## Authentication Key

This field allows you to enter the authentication key for the LNS network. This is required if you are accessing the network as a remote Full or remote Lightweight client and the network uses authentication. The authentication key is a hexadecimal value of 12 characters, for example "a327ff27ba24".

---

## Network Interface

### Network Attached

Select this setting if the PC is attached to the physical network and you want to communicate with the physical devices. The LNS DDE Server may be simultaneously attached to any number of local or remote networks. If this option is selected and you have more than one network interface in your PC, you must also select the name of the network interface in the `Network Interface Name` field.

The default is Attached. If the network is *not* attached, you can use only configuration properties. If you are communicating as a Lightweight client over an LNS/IP Channel (see *Remote Settings*, earlier in this Chapter), this option will not be available.

### Network Interface Name

If you are communicating over a LONWORKS or LONWORKS/IP Channel, this setting selects the network interface to be used to communicate with the devices in the network. If you are communicating as a Lightweight client over an LNS/IP Channel (see *Remote Settings*, earlier in this Chapter), this option will not be available.

### Attach to Network when Exploring

Select this setting to open a network when you click its name. If you do not turn on this setting, you can explore a network without being attached. The default setting is On. If you are communicating as a Lightweight client over an LNS/IP Channel (see *Remote Settings*, earlier in this Chapter), this option will not be available.

### Set Management Mode OnNet

This setting allows you to determine whether the management mode will be OnNet or OffNet. If you select OnNet, changes to configuration properties will be propagated to the physical devices of the network immediately. Network variables and LONMARK object changes are always immediately propagated. The default is On.

If you turn off OnNet mode (i.e. go OffNet), the changes you make to configuration properties will not be propagated until you reselect OnNet mode. This allows you to make many configuration property changes rapidly without having to wait for every change to propagate to the devices.

If you are communicating as a Lightweight client over an LNS/IP Channel (see *Remote Settings*, earlier in this Chapter), this option will not be available.

## Network Event Subscription

This setting controls how the LNS DDE Server subscribes to network variable address and connection change events. This can be set to one of the following:

- None*            The LNS DDE Server subscribes to no change events. If event subscription is disabled, LNS DDE Server startup is faster as no change events are subscribed to. This speed gain is made at the expense of possible data loss for the monitoring client if a network variable's binding changes. LNS Server overhead is reduced compared to the default, as no change notifications are sent to the monitoring client. Use this value if you have a static network. If this option is selected, you cannot use bound updates to monitor output network variables on the network.
- Monitored*        The LNS DDE Server subscribes to individual network variable change events. If individual event subscription is enabled, LNS DDE Server startup may be slow because a separate change event is subscribed to for each monitored network variable. The LNS Server's overhead is minimized, as change notifications are only sent for monitored network variables. The monitoring client should never lose data due to binding changes. Use this value for a dynamic network where devices and connections are frequently changed.
- All*                The LNS DDE Server subscribes to a single generic network variable change event. This is the default value. If generic event subscription is enabled, then the LNS DDE Server startup is faster as only one event subscription takes place. However, the LNS Server's overhead will increase when changes are made to the network due to the extra event notifications that are sent for every connection change. Use this value for a mainly static network.

---

## Default Data Reporting Settings

### Polling Interval (1/10 sec)

This setting specifies the default polling interval that the LNS DDE Server will use when polling unbound network variables, or the minimum update interval for bound network variables. Specify a value in tenths of a second. For example, a value of 5 specifies that the LNS DDE Server will poll every 0.5 seconds. When network variables are bound to the server, this setting specifies the minimum update rate between the LNS DDE Server and the DDE client. The poll interval may be between 1 and 32000 (.1 seconds and 53 minutes, 20 seconds). The default polling interval is 10 (1 second).

The network topology, type of network interface, and LNS DDE Server host PC performance will all affect the maximum number of points that can be polled per second. In early tests of the LNS DDE Server, version 2.1, a 600 MHz PC on a LONWORKS/IP channel connected to six TP/XF-1250 channels was able to poll over 600 points per second. Adjust polling intervals to ensure that you do not exceed the maximum throughput of your configuration. For example, a network variable in a sensor that monitors outside air temperature probably doesn't need to be polled more than once every 3 minutes. It is important to limit the polling demand to allow for temporary demand spikes, and allow bandwidth for timely delivery of outgoing updates.

You can override the default polling interval for an individual network variable with the `-i` switch.

## Refresh Rate (1/10 sec)

This setting specifies the default refresh rate that the LNS DDE Server will use when reading configuration properties. Unlike network variables, configuration properties are not read directly from a device. Instead they are retrieved from the LNS Server database. A value in the database may change due to an update by another LNS application.

Specify a value in tenths of a second. For example, a value of 100 specifies that the LNS DDE Server will refresh its configuration property values every 10 seconds. This value can generate significant network traffic when running remotely. Set this option to 0 to receive a one time only update.

This setting also applies to system, device, and LONMARK object properties. The default refresh rate is 600 (60 seconds).

## Use Bound Updates

When enabled, this setting causes a network variable to be created on the local PC, which is then bound to the network variable on the device to be monitored. A network variable is not created if the network variable to be monitored is already bound to the local PC. This occurs for every output network variable that is monitored.

Using bound updates will eliminate the need for polling for changes; this can be useful for network variables that change very infrequently but require immediate host notification when they do change (for example, network variables which indicate alarms). However, there are several drawbacks:

- When extensively using bound network variable updates, and using Acknowledged service, the network may quickly become saturated with retries if the PC that the LNS DDE Server runs on is disconnected from the network or experiences a fatal software exception. You can avoid this problem by using Unacknowledged or Repeated service. As long as the PC has power, the Network Interface will generate acknowledgements to a bound update. However, the

acknowledgements are not an indicator of reliable delivery should the LNS DDE Server and attached client applications not be running.

- If a network variable that is already connected to another device is monitored using bound updates, a group is created on the host. If unsuccessful in creating a bound connection, the LNS DDE Server will automatically revert to polling the network variable. **No errors or warnings are generated if the connection is not successful.**
- Each time a network variable is bound, the lifetime of the rewritable non-volatile memory on the device is decreased. If you use the `Bound Updates Persistent` option, this is not much of an issue. In most systems this is the desired option when using bound updates.
- Using bound acknowledged updates to the LNS DDE Server may produce uneven update rates under high traffic conditions due to resource limits on the host interface. For periods of 4 to 7 seconds or more, several devices may be blocked from updating the LNS DDE Server. Limit the number of bound connections for consistent update performance. The resource limits are less of an issue when using Unacknowledged service.

Unacknowledged bound updates cannot be used with network variables already using Acknowledged service.

- Shutting down the LNS DDE Server for a network with many bound connections to the host will take a long time while the LNS DDE Server is deleting all of these connections. Limit the number of bound connections to the host to improve shut down performance, or use the [Bound Updates Persistent](#) option to eliminate this issue.

For these reasons, the use of polled updates is typically recommended. If bound updates are used, the following guidelines should be followed:

- Use Unacknowledged (or Repeated) service. This will avoid a system saturation with retries in the event that the PC running the LNS DDE Server is turned off, crashes, or becomes disconnected. If an acknowledged connection connects the network variable to another device, the connection must first be defined to use Unacknowledged (or Repeated) service.
- Explicitly bind (i.e. establish a connection between the device network variable and network variable on the host) any network variables to be updated with bound updates using an LNS installation tool such as the LonMaker Integration Tool. The LNS DDE Server will automatically use bound updates for network variables that are explicitly bound. **If you explicitly bind a network variable to the host using an LNS installation tool, do not turn on bound monitoring for that network variable using the Network Configuration dialog or the `-b` option.**
- Use alias connections when binding a network variable to both the host and another network variable. This prevents creating a new group on the host for every network variable. Devices without alias tables allocated do not support this type of connection.

- Control the update rate of bound network variables. Network variables should generally only be bound if they will change infrequently. If there is a possibility that several network variables bound to the host will begin rapidly changing, slow the update rate to avoid saturating the network. For example, the update rate can be limited on many devices using the minimum send time standard configuration property type (SCPTminSendTime).

The default setting for Use Bound Updates is off. You can override the binding option for an individual network variable with the `-b` switch.

All temporary network variables and connections are removed automatically by the LNS DDE Server when the point is deleted, unless you have selected the `Bound Updates Persistent` option. Select this option if you have specified a large number of bound updates, otherwise it can take a long time to delete the points or exit the server.

Creating message tags on the host and binding to them is not supported by the LNS DDE Server.

## Use Priority Messages

When enabled, this setting specifies that the LNS DDE Server will use priority messages for outgoing application messages and network variable updates. The default setting is Off.

You can override the priority option for an individual network variable or application message with the `-p` switch.

## Number of Retries

This setting specifies the default number of retries to use for outgoing application messages and network variable updates that use the Repeated or Acknowledged service. When using the Acknowledged service, the number of retries specifies the maximum number of times that the LNS DDE Server will resend a network variable update or application message when an acknowledgement is not received within the time specified by the Reply Timeout setting. When using the Repeated Service, the number of retries specifies the number of times that every network variable update or application message will be repeated. The default number of retries is three (3).

You can override the number of retries for an individual network variable or application message with the `-r` switch.

## Reply Timeout (1/10 sec)

This setting specifies how long the LNS DDE Server will wait for an acknowledgement to an acknowledged update before retrying the operation. The value is specified in tenths of a second. For example, a value of 5 specifies that the LNS DDE Server will wait for 0.5 seconds before retrying an update. The default reply timeout is 30 (3 seconds).

## Exclusive Mode (Lightweight Clients)

This option is disabled for version 2.1 of the LNS DDE Server.

## Host Message Owner

This option is disabled for version 2.1 of the LNS DDE Server.

## Point Recovery Strategy

This setting determines the behavior when a device with one or more points loses communication the LNS DDE Server. Select one of the following:

- |             |   |
|-------------|---|
| None        | When there is an error communicating with a point, the LNS DDE Server scales back the polling interval to 60 seconds, to attempt to regain live data for each failed point. Once communication is reestablished, polling will continue at the normal rate.  |
| Priority    | When there is an error communicating with a device, the LNS DDE Server will stop monitoring all points on that device. The LNS DDE Server will send a ping to one point on the device every 60 seconds using priority messaging. Once communication is reestablished, all points on the device will be restored to automatic polling.     |
| No Priority | When there is an error communicating with a device, the LNS DDE Server will stop monitoring all points on that device. The LNS DDE Server will send a ping to one point on the device every 60 seconds using non-priority messaging. Once communication is reestablished, all points on the device will be restored to automatic polling. |

---

## Message Options—Network Variables

### Service

This setting selects the default messaging service to be used when updating input network variables on other devices from the LNS DDE Server. The available settings are *Acknowledged*, *Repeated*, or *Unacknowledged*. The default service is *Acknowledged*.

You can override the default messaging service with the `-s` switch.

### Bound Service

When a DDE point is created by default, it does not have a binding service. This setting selects the service that will be used for network variables created on the host network interface when they are automatically created by the `-b` option. The available settings are *Acknowledged*, *Repeated*, or *Unacknowledged*. The default binding service is *Unacknowledged*.

## Bound Update Persistent

Once the DDE Server shuts down, all temporary network variables and connections created when using bound updates are removed automatically when the point is deleted. To keep the temporary network variables and connections active, select this setting. Selecting this setting improves shut-down and start-up performance if you are using temporary network variables. The default setting is Off. See *Use Bound Updates*, earlier in this chapter, for more information.

---

## Message Options—Application Messages

### Service

The setting selects the default messaging service to be used when sending application messages from the LNS DDE Server to a device. The available settings are *Acknowledged*, *Repeated*, *Unacknowledged*, or *Request*. The default setting is *Unacknowledged*.

You can override the default messaging service with the `-s` switch.

### Authenticate Messages

When enabled, this setting specifies that the LNS DDE Server will use authenticated transactions for outgoing application messages that use the Acknowledged service. The default setting is Off.

You can override the authenticated option for an individual application message with the `-a` switch.

### Include Sequence Number

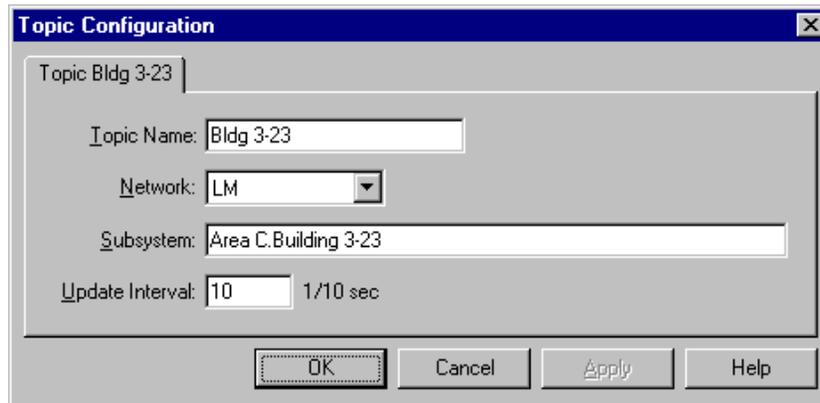
This setting specifies the default for whether the client will receive application message items from the LNS DDE Server even if the data has not changed from the last time this message was received from the network. If it is False, data is reported to the client only if it has changed. If it is True, a byte containing an incrementing counter will be appended to each incoming message. When the counter is turned on, the data received by the client contains the additional byte. The default setting is False. It may be overridden with the `-n` switch when the DDE transaction is started for the item.

---

## Creating Custom Topics

As described in the next chapter, the LNS DDE Server has a number of predefined topics that you use to identify the type of data that you are reading or writing. In addition to these predefined topics, you may create your own custom topics to simplify access to frequently used data. Custom topics are not required; however, they do provide a shorthand notation for the predefined topics.

To create a custom topic, open the *Configure* menu, point to *Topics*, then select *Create New*. Alternatively, you can right-click a subsystem in the folder or detail pane and select *Associate Topic* from the shortcut menu. The following dialog opens:



The settings in this dialog are described in the following sections.

## Topic Name

This setting specifies the name for the custom topic. This name is a component of the address used by DDE clients to access network variables or configuration properties as described in Chapter 4. The name is case insensitive.

## Network

This setting specifies the network with which to associate the topic. Select a network from the list.

## Subsystem

This setting specifies the full subsystem path with which to associate this topic. The subsystem path must be entered exactly as it is named in the LNS Server database. The case of this entry must match the case in the database.

The path is not validated until the topic is accessed by a client application. To ensure that the subsystem path is correct, open the network containing the subsystem, right-click the subsystem you wish to create a topic for, and select *Associate Topic* from the shortcut menu. Complete the topic configuration by specifying a topic name and a poll interval, then click *OK* or *Apply* to save the new topic.

## Update Interval

This setting specifies the default polling interval that the LNS DDE Server will use when polling network variables through this topic. For this topic only, this polling interval overrides the default polling interval specified for the network. The value is specified in tenths of a second. For example, a value of 5 specifies that the LNS DDE Server will poll every 0.5 seconds. The default value is 10 (1 second).

You can override the default polling interval for an individual network variable with the `-i` switch.

Click **Apply** to save the specified topic and specify additional topics. Click **OK** to save the specified topic and close the dialog.



# 4

## Addressing DDE Messages

This chapter describes how a DDE client specifies an address for a DDE message.

---

## Address Components

When a DDE client sends out a request to begin a DDE conversation, all DDE servers running on the PC see the message. In order to direct the message to a specified point on a particular server, the client addresses the message using a three-part DDE address with *application*, *topic*, and *item* components.

### Application

The name of the server to which the message is directed. For example, to address a message to the LNS DDE Server, the client may specify LNSDDE as the application. This name is case *insensitive*. By default, the DDE application name is LNSDDE for the LNS DDE Server, but this name can be changed on the Server Settings tab of the Server Configuration dialog described in Chapter 3.

### Topic

For the specified application, the type of information that the client is interested in receiving or sending. A server categorizes the services it provides by topic. The LNS DDE Server recognizes several topics depending upon your network configuration. The topic names and how they are constructed are described later in this chapter.

The names are made up of several individual parts, some of which are case sensitive, and others that are case insensitive.

### Item

Within the context of the specified topic, the name of the specific piece of information that the client wants to send or receive. For example, to tell the LNS DDE Server that it wants the value of the network variable named *status* on the device named *monitor*, the client specifies *monitor.status* as the item. The format of an item depends on the topic it is part of, and is in general, case-sensitive. This corresponds with the convention for device names, network variable names, configuration property names, and message tag names in an LNS database. The client application must be able to specify the device, network variable, configuration property, and message tag names with exactly the same case as they are defined in the LNS database.

---

## Topics

The topic is specified using either a standard or custom format. The standard topic format includes *Network\_name*, *Subsystem\_path*, and *Topic\_type* components. The custom topic format includes *Topic\_name* and *Topic\_type* components.

---

## Standard Topic Format

The standard topic format specifies a network, subsystem, and topic type. Use this format except when you are using custom topics as described in the next section. To use a standard topic format, specify a topic with the following three components:

*Network\_name.Subsystem\_path.Topic\_type*

Each component is separated from the other by a single period ("."). The *Subsystem\_path* component may consist of multiple subsystems, each also separated by a period. Following is a description of the three components (all three components must be specified):

<i>Network_name</i>	The name of the network as defined in the LNS database. The <i>Network_name</i> is case sensitive.
<i>Subsystem_path</i>	The full path to the subsystem containing the device to be accessed, as defined in the LNS database. The subsystem path in itself may consist of one or more components. For example, <i>Building-A54.Level-B.Section-K</i> is the subsystem path for a <i>Section-K</i> subsystem within a <i>Level-B</i> subsystem within a <i>Building-A54</i> subsystem. The <i>Subsystem_path</i> is not specified for the system topic. The <i>Subsystem_path</i> is case sensitive.
<i>Topic_type</i>	Specifies the type of information required. The topic type must be one of the following: <i>Netvar</i> , <i>DevNV</i> , <i>NV</i> , <i>LMNV</i> , <i>LMCP</i> , <i>NVCP</i> , <i>DevCP</i> , <i>DevNVCP</i> , <i>LMNVCP</i> , <i>LM</i> , <i>MsgTag</i> , <i>MT</i> , <i>Dev</i> , or <i>System</i> . These types are described later in this chapter under <i>Topic types</i> . The <i>Topic_type</i> is <u>not</u> case sensitive.

The LNS DDE Server folder and detail panes can be used to find the name of the network and subsystem path. If accessing a remote network, the LNS Server must be running on the host PC.

### Examples

Below are a few examples of correctly formatted topics:

*Network1.Subsystem 1.Netvar*  
*NetX.SubsystemY.SubsystemZ.MsgTag*  
*Building-A54.Level-B.Section-K.LMNVCP*  
*Network.System*

---

## Custom Topic Format

The custom topic format specifies a custom topic name and a topic type. Use this format with custom topics that you define. To use a custom topic format, specify a topic with the following two components:

*Topic\_name.Topic\_type*

Each component is separated from the other by a single period (.). Following is a description of the two components (both components must be specified):

<i>Topic_name</i>	The custom topic name that you have defined. When you create a custom topic name, you will specify the network name and subsystem path that the server is to use for the custom topic. The custom topic name must not contain any periods (.), and is <u>not</u> case sensitive.
<i>Topic_type</i>	Specifies the type of information required. The topic type must be one of the following: <i>Netvar</i> , <i>NV</i> , <i>DevNV</i> , <i>LMNV</i> , <i>LMCP</i> , <i>NVCP</i> , <i>DevCP</i> , <i>DevNVCP</i> , <i>LMNVCP</i> , <i>LM</i> , <i>MsgTag (MT)</i> , <i>Dev</i> , or <i>System</i> . These types are described later in this chapter under <i>Topic types</i> . The <i>Topic_type</i> is <u>not</u> case sensitive.

To create a custom topic, see *Creating Custom Topics* in Chapter 3.

## Examples

Below are a few examples of correctly formatted topic names:

```
TheTopic.Netvar
AnotherTopic.MsgTag
BldgA54_LvlB_SecK.LMNVCP
```

---

## Topic Types

There are many types of information in a LONWORKS network. When you are addressing a DDE message, you need to specify both the type and the name of the item that you are accessing. The topic type tells the LNS DDE Server where to look for the item in the LNS Server.

The topic types are grouped into the following six categories: Network Variables, Configuration Properties, LONMARK Objects, Devices, Application and Foreign Frame Messages, and System. In addition to this, Network Variables and Configuration Properties are further subdivided into several different types depending on the context. The following table describes the topic types.

Category	Topic Type	Description
Network Variable	<i>LMNV</i>	Network variables belonging to a LONMARK object. Use this type to access network variables belonging to a specific LONMARK object. The item name must specify both the device name and the LONMARK object name. See Chapter 5 for more information.
	<i>DevNV</i>	Network variables on a device. This type accesses network variables on a device that are <u>not</u> configuration properties and <u>not</u> members of a LONMARK object. See Chapter 5 for more information.
	<i>Netvar or NV</i>	All network variables on a device. Provided mainly for backward compatibility with the 16-bit LonManager DDE Server. A network variable accessed through this type may in fact be a configuration property or a network variable that belongs to a LONMARK object present in the device. Devices with LONMARK objects may have duplicate network variable names when accessed using this topic. The LMNV or DevNV topic should be used instead. See Chapter 5 for more information.
Configuration Property	<i>DevCP</i>	Configuration properties for a device. See Chapter 6 for more information.
	<i>LMCP</i>	Configuration properties for a LONMARK object. See Chapter 6 for more information.
	<i>LMNVCP</i>	Configuration properties for a network variable member of a LONMARK object. See Chapter 6 for more information.
	<i>DevNVCP</i>	Configuration properties for a network variable on a device. This type accesses configuration properties for network variables on a device that are <u>not</u> members of a LONMARK object. See Chapter 6 for more information.
	<i>NVCP</i>	Configuration properties for a network variable. As with the Netvar and NV topic types, any network variable may be specified. The network variable name may be ambiguous when specified using this topic type. See Chapter 6 for more information.
Application Messages	<i>MsgTag or MT</i>	Application and foreign frame messages. Used to receive application and foreign frame messages sent to the LNS DDE Server host and to write application or foreign frame messages. Application and foreign frame topics are not supported when accessing the LNS DDE Server as a Lightweight client. See Chapter 7 for more information.
LONMARK Object	<i>LM</i>	LONMARK objects. Used to access the properties of a LONMARK object on a device. See Chapter 8 for more information.

Device	<i>Dev</i>	Devices. Used to access a device's or router's properties. See Chapter 8 for more information.
System	<i>System</i>	System properties. For the <i>System</i> type <u>no subsystem</u> is specified. See Chapter 8 for more information.

## Topic Types and the Folder Pane

The example folder pane on the right illustrates how topic types are displayed by the LNS DDE Server.

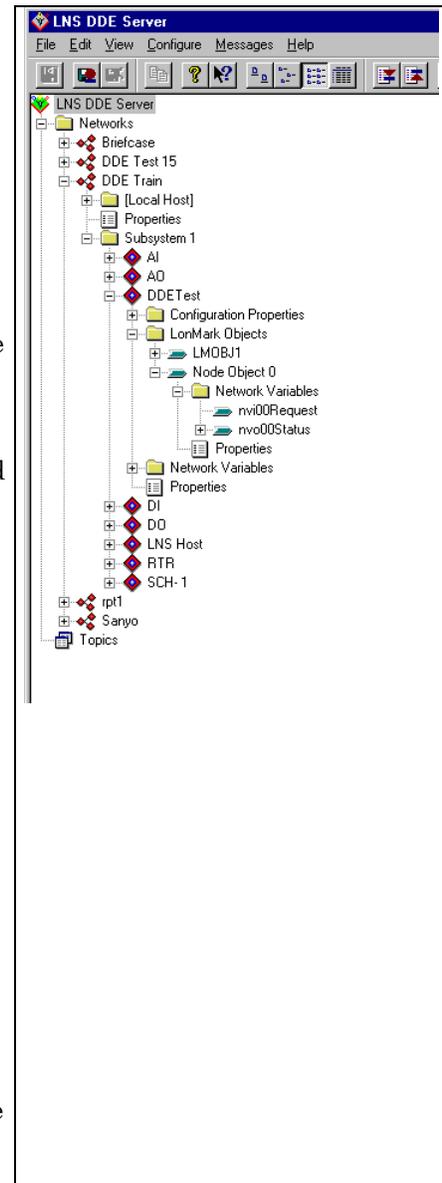
In the folder pane, a tree is displayed with the LNS DDE Server at the root, networks as branches off the LNS DDE Server, subsystems as branches off the networks, and devices as branches off the subsystems.

For example, the branches immediately below the Subsystem 1 subsystem in the example folder pane are the devices in Subsystem 1. Below each device is a set of type folders that contain the Network Variables, Configuration Properties, and LONMARK Objects in the device.

### Hierarchy

The folder pane displays the hierarchy of devices, LONMARK objects, network variables, and configuration properties. A LONMARK object is contained within a device. Network variables can be associated with a LONMARK object or with a device as a whole, so network variables can exist on the same level as LONMARK objects in the hierarchy, or one level below LONMARK objects. Configuration properties can be associated with network variables, LONMARK objects, or devices. Therefore, configuration properties can be on the same level of the hierarchy as LONMARK objects or network variables, or they can be on the level below network variables.

When type folders are contained inside other type folders, specify the topic type by concatenating the names of the types.



---

## Items

The *Item* is the last component of a DDE address. The topic type as described in the previous section determines the format of the item. The item format consists of one or more of the following components:

<i>Device_name</i>	Device name, as defined in the LNS database.
<i>LMO_name</i>	LONMARK Object name, as defined in the LNS database.
<i>NV_name</i>	Network variable name. See Chapter 5 for more information.
<i>CP_name</i>	Configuration property name. See Chapter 6 for more information.
<i>Field_name</i>	Field name within a network variable or configuration property.
<i>Msg_name</i>	Message tag name for application and foreign frame messages. See Chapter 7 for more information.
<i>Msg_code</i>	Message code for application and foreign frame messages. Valid ranges are 0-62 and 64-78. See Chapter 7 for more information.
<i>System_property</i>	System property name. See Chapter 8 for more information.
<i>Device_property</i>	Device property name. See Chapter 8 for more information.
<i>LMO_property</i>	LONMARK object property name. See Chapter 8 for more information.
<i>Parameters</i>	Optional parameters. See Chapters 5,6, and 7 for more information.

The components above are combined as shown in the following table to define an item. Components listed inside brackets ([ and ]) are optional.

<i>Dev</i>	Device_name.Device_property [Parameters]
<i>DevCP</i>	Device_name.CP_name[.Field_name] [Parameters]
<i>DevNV</i>	Device_name.NV_name[.Field_name] [Parameters]
<i>DevNVCP</i>	Device_name.NV_name.CP_name[.Field_name] [Parameters]
<i>LMNV</i>	Device_name.LMO_name.NV_name[.Field_name] [Parameters]
<i>LM</i>	Device_name.LMO_name.LMO_property [Parameters]
<i>LMCP</i>	Device_name.LMO_name.CP_name[.Field_name] [Parameters]
<i>LMNVCP</i>	Device_name.LMO_name.NV_name.CP_name[.Field_name] [Parameters]
<i>Netvar</i>	Device_name.NV_name[.Field_name] [Parameters]
<i>NV</i>	Device_name.NV_name[.Field_name] [Parameters]
<i>NVCP</i>	Device_name.NV_name.CP_name[.Field_name] [Parameters]
<i>System</i>	System_property [Parameters]
<i>MsgTag</i>	[Device_name.] msg_name [.Msg Code] [Parameters]
<i>MT</i>	[Device_name.] msg_name [.Msg Code] [Parameters]

---

## Examples

Below are a few examples of correctly formatted items:

<u>Type</u>	<u>Item</u>
Netvar	AI-1.nviTemp -i 100
DevNVCP	AI-1.AS_Analog_1.SCPTovrValue
LMNVCP	AI-1.AS-1.AS_Analog_1.UCPTobjectType
LM	AI-1.AS-1.Request.Override
Dev	AI-1.NeuronId
MT	lamp.msg_in.1

# 5

## Reading and Writing Network Variables

This chapter describes how to read and write network variables.

---

## Specifying a Network Variable

Monitoring and control applications will typically monitor input or output network variable values and allow an operator to update input network variable values. Network variables may be identified either by member name within a LONMARK object, or by name for a device. Devices with LONMARK objects may have multiple network variables with the same name, so using the LONMARK object name/member name is a more reliable way to access network variables on LONMARK-compatible devices. To access a network variable, you must select the naming method in the DDE Topic, and the specific network variable in the DDE Item. The NV and Netvar topics are provided for backward compatibility with applications developed for the 16-bit LonManager DDE Server.

---

## Specifying a Network Variable Topic

A network variable topic is specified using the standard topic format or custom topic format described in Chapter 4. The topic type will be one of the following network variable topics:

Category	Topic Type	Description
Network Variable	<i>LMNV</i>	Network variables belonging to a LONMARK object. Use this type to access network variables belonging to a specific LONMARK object. The item name must specify both the device name and the LONMARK object name.
	<i>DevNV</i>	Network variables on a device. This type accesses network variables on a device that are <u>not</u> configuration properties and <u>not</u> members of a LONMARK object.
	<i>Netvar</i> or <i>NV</i>	All network variables on a device. Provided for backward compatibility with the 16-bit LonManager DDE Server. A network variable accessed through this type may in fact be a configuration property or a network variable that belongs to a LONMARK object present in the device. Devices with LONMARK objects may have duplicate network variable names when accessed using this topic. The LMNV or DevNV topic should be used instead.

---

## Specifying a Network Variable Item

A network variable item identifies a device and a network variable or network variable field on the device. It also specifies a LONMARK object if the LONMARK object/member name method is used to identify the network variable. The format for specifying this information is the following:

```
device_name[.lm_name].nv_name[[.array_index]][.field_name] [parameters]
```

Each component of the Item format has the following meaning:

<i>device_name</i>	Required. The name of a device, as defined in the LNS database.
<i>lm_name</i>	Required if the LMNV topic is used. The name of the LONMARK Object as defined in the LNS database.
<i>nv_name</i>	Required. The name of the network variable on the device, as defined in the LNS database.
<i>array_index</i>	Required if the network variable is an element in an array of network variables, <i>array_index</i> is the array index in square brackets.
<i>field_name</i>	Optional. A field within a structured network variable. Fields may be within nested structures. Fields are only available for structured SNVTs and for user-defined types defined in a user resource file. The field name must be a parameter contained within a 'text' format definition in the format file for either a SNVT or a user-defined type. A master list of SNVTs is contained in the <i>LNS Utilities and LonMark Reference</i> help file included with the LNS DDE Server. For a user-defined structured data type, the field names are as defined in a user resource file. The field cannot be a union.
<i>Parameters</i>	Optional. Multiple parameters can be specified. For each parameter, zero or more spaces can separate the parameter and its argument.

The parameter options are:

<b>-a</b> <i>bool</i>	For a variable monitored by polling, this option specifies whether the poll messages should be sent with authenticated service. May be set to <code>True</code> or <code>False</code> . The default is the setting specified for the network in the Network Configuration dialog.
<b>-b</b> <i>bool</i>	For a variable monitored by polling, this option specifies whether to use bound updates. May be set to <code>True</code> or <code>False</code> . The default is the setting specified for the network in the Network Configuration dialog. This option only applies to output network variables on the monitored devices.
<b>-i</b> <i>nnnnn</i>	For a variable monitored by polling, this option sets the poll interval in tenths of a second, where <i>nnnnn</i> is a number between 1 and 33554431. The default is the setting specified for the network in the Network Configuration dialog.

- p** *bool* For a variable monitored by polling, this option specifies whether the poll messages should be sent with priority service. May be set to `True` or `False`. The default is the setting specified for the network in the Network Configuration dialog.
  
- r** *nn* Specifies the retry count to be used for request/response service used to poll network variables, where *nn* is a number between 0 to 15. The default is the setting specified for the network and is tool dependant.
  
- s** *ss* Specifies the protocol service to be used when updating the network variable from the LNS DDE Server. The available settings are *ACKD* (acknowledged), *UNACKD* (unacknowledged), or *UNACKD\_RPT* (repeated). The default is the setting specified for the network and is tool dependant.
  
- t** *format\_name* Specifies a format name to use for formatting the network variable as described in the next section. If this option is not specified, a format is selected by LNS. Use this option and specify a format name to change the format. The format name can be one of the five built-in formats (`INTEGER`, `DISCRETE`, `REAL`, `STRING`, or `RAW`), a format name from the standard resource file (for SNVTs and SCPTs), or a format name from any user resource file. If the format name is `RAW`, data will be returned in raw format.

When changing the data format using this parameter, the change is applied to the entire data value. This may cause unexpected results. For example, when reading a network variable which is a floating point value, and changing the format to an integer (`-t INTEGER`), the formatted value will be an integer interpretation of the entire floating point number, not just the integer part. The same thing applies when changing the data type to floating point (`-t REAL`) for an integer value. Make sure formats are compatible with the network variable value.

**Note:**

Floating point values must be between `-3.4E38` and `+3.4E38`.

---

## Network Variable Formats

Network variables have a default format stored in the LNS database. An LNS based network tool such as the LonMaker browser can change the default format for any network variable by changing the `DSFormatType` for the network variable in the LNS database. For example, the LonMaker browser allows a user to assign a custom format to any network variable by right-clicking the value and selecting the *Change*

*Format* command. Once the extended form is assigned to the network variable, the DDE client application does not need to specify the format type in the item name.

To override the default format, use the `-t` parameter described in the previous section. You may either specify a standard format, or specify a user format from the format file described in Appendix A, *Resource Files*, using the following syntax:

```
#<program id>[<selector>].<format name>
```

The fields are defined as follows:

`program id`      The program ID specified in the format file, in hex, without spaces or colons. This does not have to match the program ID for the device containing the network variable.

`selector`          The scope selector specified in the format file. The second line of the format file specifies a scope selector name that specifies which device types are covered by the format file. Translate this selector to a selector index using the following table:

<b>Scope Selector</b>	<b>Selector Name</b>
1	DEVICE_CLASS
2	DEVICE_CLASS and SUB_CLASS
3	MANUFACTURER
4	MANUFACTURER and DEVICE_CLASS
5	MANUFACTURER and DEVICE_CLASS and SUB_CLASS
6	MANUFACTURER and DEVICE_CLASS and SUB_CLASS and MODEL

The selector does not have to match the selector used by the functional profile template defining the LONMARK object(s) on the device.

`format type`      The format type name specified in the format file. The format type name must match a type name in the type file associated with the format file, or in a type file with a lower scope selector value.

`format modifier`    A modifier for the format type name that allows multiple formats to be specified for the same type.

For example, here is a sample user format file.

```
Set program_id 80:11:22:33:44:55:66:77;
Set selector DEVICE_CLASS;
//
UNVT_s_short: text("%d");
UNVT_s_long: text("%d");
UNVT_u_short: text("%d");
UNVT_u_long: text("%d");
//
```

The AI- 1.Node Object 0.Node\_Directory item would be reformatted as a UNVT\_u\_long, by using the -t option as follows:

```
AI- 1. Node Object 0.Node_Directory -t#8011223344556677[1].UNVT_u_long
```

This item specifies the AI- 1 device, the Node Object 0 LONMARK object, and the Node\_Directory network variable member of the LONMARK object. The format is specified as the format defined for program ID 8011223344556677, with a selector of 1, and a format name of UNVT\_u\_long.

---

## Requesting a One-Time Network Variable Update

To request a one-time update, your client application sends a DDE Request message to the LNS DDE Server. The LNS DDE Server responds with the requested value. The format of the request is application dependent. The following sections describe how to request a one-time update in Excel and InTouch.

---

### *Requesting a One-Time Network Variable Update Using Excel*

Three separate functions are required to request a one-time network variable update using Excel. They are the Excel DDEInitiate(), DDERequest(), and DDETerminate() functions. To call these functions, you must implement a Visual Basic for Applications (VBA) macro for each value to be requested. VBA is included with Excel.

The request macro will typically be called from an ActiveX control such as a button. To request a one-time network variable update using an Excel button control, follow these steps:

1. Create or open an Excel spreadsheet and select the worksheet that will contain the button.
2. If the Control Toolbox is not displayed, point to Toolbars on the View menu, and then select Control Toolbox.
3. Click the Command Button control on the Control Toolbox.
4. On the worksheet, drag the control to the size you want.
5. To set the properties for the control, right-click the control, and then select Properties on the shortcut menu.
6. To add VBA macro code to the control, right-click the control, and then select View Code on the shortcut menu. The VBA editor will be opened, and a new VBA function will be created for your button code.
7. Enter the following code:

```
Dim Channel As Long
```

```

Channel = DDEInitiate(Application, Topic)
Value = DDERequest(Channel, Item)
DDETerminate (Channel)

```

Following is a description of the parameters:

<i>Application</i>	The DDE application as a string. By default, this is "LNSDDE" for the LNS DDE Server.
<i>Channel</i>	A DDE channel number created by the DDEInitiate() function, as a long.
<i>Item</i>	A DDE item as a string, as described in <i>Specifying a Network Variable</i> earlier in this chapter.
<i>Topic</i>	A DDE topic as a string. For network variables, this will be an LMNV, DevNV, or NV topic as described in <i>Topics</i> in Chapter 4 and in <i>Specifying a Network Variable</i> earlier in this chapter.
<i>Value</i>	The returned value from the one-time update request. This will typically be assigned to a worksheet cell.

8. To return to Excel from the VBA editor, select Close and Return to Microsoft Excel on the File menu.
9. To exit design mode and enable the ActiveX control, click Exit Design Mode on the Control Toolbox.

## Example

```

Private Sub btnReadSwitch4_Click()
    Dim channel As Long

    channel = DDEInitiate("LNSDDE", "LNS DDE Test.HVAC.LMNV")
    Sheet1.Range("E21") = DDERequest(channel, _
        "DI- 1.SW- 4.Digital.State")
    DDETerminate (channel)
End Sub

```

---

## Requesting a One-Time Network Variable Update Using InTouch

The WRequest function is used to request a one-time network variable update using InTouch. To call this function, you must implement an InTouch action script for each value to be requested. The action script will typically be called from an InTouch button. To request a one-time network variable update using an InTouch button, follow these steps:

1. Open an InTouch application project in WindowMaker and open the window that will contain the button.
2. Click the Button tool on the Draw Object toolbar.

3. On the target window, drag the control to the size you want.
4. To set the properties for the button, double-click the button.
5. To add an action script to the button, click the Action button.
6. Select On Key Down as the Condition Type.
7. Enter the following code:

```
StatusTag = WWRequest (Application, Topic, Item, Value)
```

Following is a description of the parameters:

<i>StatusTag</i>	A tag containing the return value from the WWRequest function. The value is 1 for a successful request, 0 if the application is busy, and -1 for an error.
<i>Application</i>	The DDE application as a string or an InTouch tag. By default, this is "LNSDDE" for the LNS DDE Server.
<i>Topic</i>	A DDE topic as a string or an InTouch tag. For network variables, this will be an LMNV, DevNV, or NV topic as described in <i>Topics</i> in Chapter 4 and in <i>Specifying a Network Variable</i> earlier in this chapter.
<i>Item</i>	A DDE item as a string or an InTouch tag, as described in <i>Specifying a Network Variable</i> earlier in this chapter.
<i>Value</i>	A string or an InTouch tag that is going to hold the returned value from the one-time update request.

8. Click OK for the two open dialogs to return to the application window.
9. To exit design mode and enable the button, click Runtime! in the menu bar.

If a point no longer exists and you try to update it, you will receive a message in InTouch asking you if you want to start the server. Use the Trace option to determine whether or not the point exists.

## Example

```
StatusTag = WWRequest("LNSDDE", "LNS DDE Test.HVAC.LMNV", _
    "DI- 1.SW- 4.Digital.State", ResultTag)
```

---

## Receiving Ongoing Network Variable Updates

To receive ongoing updates for a network variable value, your client application sends a DDE Advise message to the LNS DDE Server. When you ask the LNS DDE Server to monitor a network variable, it looks in the LNS database to see if it is bound to the variable. If it is not bound to the variable, the LNS DDE Server begins to poll the variable at the specified frequency. The LNS DDE notifies your client whenever the variable's value changes. That is, the LNS DDE Server uses a report

by exception strategy; a new value is reported only upon changes in the variable's value.

If the LNS DDE Server is bound to an output network variable on a device and the LNS DDE Server has not yet received an update message from the variable, the LNS DDE Server polls the variable to get an initial value and then waits for event-driven updates for future values. The LNS DDE Server attempts to pass all these updates to your application as soon as the update arrives. It is possible that multiple updates will arrive before the LNS DDE Server is able to send an update to your application. In this case, only the most recent value is sent. To control the volume of data coming from the LNS DDE Server, your application can use the optional interval rate argument when specifying the item. For variables monitored via event driven updates, the `-i` interval rate option defines the maximum rate at which the LNS DDE Server will send updates to your application in tenths of a second. For example, an interval rate of 5 means that no matter how fast updates arrive, the LNS DDE Server will send updates to the DDE client no faster than once every  $\frac{1}{2}$  second.

The procedure used to receive ongoing updates via DDE is application dependent. The following sections describe the concepts and trade-offs of using event-driven versus polled updates in greater detail, and describes how to receive ongoing updates in Excel and InTouch.

---

## *Event-Driven Updates versus Polling*

In a LONWORKS network, devices communicate using either event-driven updates or polling. Most communication between devices controlling a system is done using event-driven updates by binding network variables together. With event-driven updates, each source device chooses when to send its information to the destination devices based on some event. It may send the data after a set period of time (for example, once every 0.5 seconds), it may wait until a significant change has occurred (for example, the measured value has changed 2%), or a combination of both. The destination devices do not know why the data was sent, or when it will be sent again.

The second method of communication between devices is polling. When using polling, a destination device requests (polls) the value of a variable and the source sends the variable to the destination. In some cases, destinations want to control when they receive information.

When installing the network and the monitoring system, you can either bind the LNS DDE Server host to a variable for event-driven updates or use polling. For example, consider monitoring an output network variable for an over-temperature alarm. With polling, your application has two choices. It can poll frequently (which ensures that it finds updates quickly, at the price of increased network traffic), or it can poll infrequently (which reduces network traffic at the price of a delay in update notification). Another solution is to bind the monitoring device to the update output network variable. This notifies your application of updates when they happen, without polling.

In most situations, polling is the appropriate monitoring method. For example, with binding you can only monitor output network variables. If you want to monitor the value of an input network variable, you must use polling. Polling an input network

variable that is driven by multiple output network variables (network variable fan-in) is a typical example of how bound monitoring is not applicable. Polling is also useful when you want to control the rate and volume of data being sent to your monitoring application.

---

## Monitoring Polled Output Network Variables

Device manufacturers can optionally specify network-related characteristics of the device's network variables. One characteristic that they can specify is that an output network variable is a polled output. This means that the variable will only send values when it is asked (i.e., when it is polled) — it will not generate event-driven updates. To monitor the value of a polled output variable using the LNS DDE Server, the LNS DDE Server must not be connected (i.e. bound) to the variable. If it is not connected, the LNS DDE Server will automatically poll the variable to read its value.

Output network variables on the LNS DDE Server host itself cannot be polled outputs (but they may be updated).

---

## Receiving Ongoing Updates Using Excel

Follow these steps to receive ongoing updates using Excel.

- 1 In the LNS DDE Server folder pane, navigate from the top level down through the hierarchy to the network variable level.
- 2 Right-click the network variable that you want to monitor and select Copy Link on the shortcut menu.

In Excel, right-click a cell on the spreadsheet and select Paste from the shortcut menu. This will paste a formula with the following format on the shortcut menu:

```
=application|topic!item
```

If the topic or item contains spaces, it must be surrounded by single quotes (!).

Now Excel will receive ongoing updates from the LNS DDE Server.

### Example

```
=LNSDDE|'Network1.subsystem1.LMNV'!'AI-1.NO-1.SetTime- i20'
```

Note that the parameters are part of the item name and must be enclosed in single quotes with the Item.

---

## Receiving Ongoing Updates Using InTouch

Follow these steps to receive ongoing updates using InTouch.

- 1 In the LNS DDE Server folder pane, navigate from the top level down through the hierarchy to the network variable level.
- 2 Right-click the network variable that you want to monitor, and choose Copy Link from the shortcut menu.
- 3 In InTouch WindowMaker, click the Wizards icon on the toolbar. Select I/O Analog Display from the Value Displays category, and click OK.
- 4 Place the object on the window and then double-click it. Select Paste Link In in the dialog and then click OK.
- 5 InTouch will now try to provide a tag-name based on the item. A dialog will inform you about it if it succeeds. Click OK to close the dialog and then either change the tag-name or click OK to use the tag-name provided by InTouch.

Now InTouch will receive ongoing updates from the LNS DDE Server.

---

## Sending a Network Variable Update

To change a network variable's value, your application sends a DDE poke message to the LNS DDE Server. The LNS DDE Server does not need to be bound to a variable to be able to change its value. How the LNS DDE Server processes the poke command depends on the direction of the variable.

If the network variable on the device is an input, the LNS DDE Server updates the variable on the device. If the input network variable is part of a connection that includes other input network variables, only the network variable specified in the item is affected. Outputs connected to the variable are not changed. Another device polling this variable will receive the new value.

If the network variable is an output on a device other than the host PC it cannot be updated.

If the network variable is an output on the LNS DDE Server host PC, all input network variables connected to the output network variable are updated

Input network variables on the LNS DDE Server host cannot be updated.

If you are sharing the network interface with another application, it is possible that the interface will be busy when the LNS DDE Server attempts to send the update. In this case, the LNS DDE Server will come back periodically and check to see if the interface is free to send the update. If your DDE client application or another DDE client application sends another update request for the same variable before the LNS DDE Server has sent the previous update, only the most recent update will be sent.

**Note:** If you are updating multiple fields of a network variable structure, you should set the `DDE_WriteDelay` point property before writing to the first field (see Point Properties in Chapter 8). This property sets a millisecond timer which begins counting down when a network variable value is updated using the LNS DDE Server. The new value will not be sent to the device until the timer expires. After the

timer expires, this property remains at 0 until explicitly reset to another value. Setting this property prevents a separate network variable update being sent on the network for each field updated. Set this property to a value such that all fields will be changed before the update is sent.

---

## *Sending a Network Variable Update Using Excel*

Three separate functions are required to send a network variable update using Excel. They are the Excel DDEInitiate(), DDEPoke(), and DDETerminate() functions. To call these functions, you must implement a Visual Basic for Applications (VBA) macro for each value to be requested.

The update macro will typically be called from an ActiveX control such as a button. To update a network variable update using an Excel button control, follow these steps:

1. Open an Excel spreadsheet and select the worksheet that will contain the button.
2. If the Control Toolbox is not displayed, point to Toolbars on the View menu, and then select Control Toolbox.
3. Click the Command Button control on the Control Toolbox.
4. On the worksheet, drag the control to the size you want.
5. To set the properties for the control, right-click the control, and then select Properties on the shortcut menu.
6. To add VBA macro code to the control, right-click the control, and then select View Code on the shortcut menu. The VBA editor will be opened, and a new VBA function will be created for your button code.
7. Enter the following code:

```
Dim Channel As Long  
Channel = DDEInitiate(Application, Topic)  
DDEPoke Channel, Item, Value  
DDETerminate (Channel)
```

Following is a description of the parameters:

<i>Application</i>	The DDE application as a string. By default, this is "LNSDDE" for the LNS DDE Server.
<i>Channel</i>	A DDE channel number created by the DDEInitiate() function, as a long.
<i>Item</i>	A DDE item as a string, as described in <i>Specifying a Network Variable</i> earlier in this chapter.
<i>Topic</i>	A DDE topic as a string. For network variables, this will be an LMNV, DevNV, or NV topic as described in <i>Topics</i> in Chapter 4 and in <i>Specifying a Network Variable</i> earlier in this chapter.

*Value*                      The value to be written to the network variable, as a worksheet cell reference.

8. To return to Excel from the VBA editor, select Close and Return to Microsoft Excel on the File menu.
9. To exit design mode and enable the ActiveX control, click Exit Design Mode on the Control Toolbox.

**Note:** If your Visual Basic application expects to update other network variables using the same channel, do not terminate the channel until the updates are received.

## Example

```
Private Sub btnUpdateLamp4_Click()  
    Dim channel As Long  
  
    channel = DDEInitiate("LNSDDE", "LNS DDE Test.HVAC.LMNV")  
    DDEPoke channel, "DO- 1.LED 4.Digital", Sheet1.Range("B17")  
    DDETerminate (channel)  
End Sub
```

---

## *Sending a Network Variable Update Using InTouch*

In most InTouch applications, updating a network variable requires the simple assignment of a statement to a tag that is declared to be an I/O tag.

To send an update from a script, use the **WWPoke** function. To call this function, you must implement an InTouch action script for each value to be requested. The action script will typically be called from an InTouch button. To send a network variable update using an InTouch button, follow these steps:

- 1 Open an InTouch application project in WindowMaker and open the window that will contain the button.
- 2 Click the Button tool on the Draw Object toolbar.
- 3 On the target window, drag the control to the size you want.
- 4 To set the properties for the button, double-click the button.
- 5 To add an action script to the button, click the Action button.
- 6 Select On Key Down as the Condition Type.
- 7 Enter the following code:

```
StatusTag = WWPoke (Application, Topic, Item, Value)
```

Following is a description of the parameters:

<i>StatusTag</i>	A tag containing the return value from the WWRequest function. The value is 1 for a successful update, 0 if the application is busy, and -1 for an error.
<i>Application</i>	The DDE application as a string or an InTouch tag. By default, this is "LNSDDE" for the LNS DDE Server.
<i>Topic</i>	A DDE topic as a string or an InTouch tag. For network variables, this will be an LMNV, DevNV, or NV topic as described in <i>Topics</i> in Chapter 4 and in <i>Specifying a Network Variable</i> earlier in this chapter.
<i>Item</i>	A DDE item as a string or an InTouch tag, as described in <i>Specifying a Network Variable</i> earlier in this chapter.
<i>Value</i>	A string or an InTouch tag specifying the value to be sent to the network variable.

- 8 Click OK for the two open dialogs to return to the application window. To exit design mode and enable the button, click Runtime! in the menu bar.

### Example

```
StatusTag = WWPoke("LNSDDE", "LNS DDE Test.HVAC.LMNV", _
    "DO- 1.LED 4.Digital", ValueTag)
```

---

## Timing Issues Related to Visual Basic Controls

If you receive the DDE exception "Timeout waiting for a DDE Response" when establishing an automatic link or making an explicit Link Request for a control, increase the LinkTimeout property.

If you observe that your Visual Basic control does not receive expected data changes, make certain that you have not terminated the DDE Link on the control before the data change has arrived.





# 6

## Reading and Writing Configuration Properties

This chapter describes how to work with configuration properties.

---

## Specifying a Configuration Property

Configuration properties may be associated with a device, LONMARK object, or network variable. For example, a configuration property associated with a LONMARK object for a sensor may define the processing for the sensor data; a configuration property associated with an output network variable may specify the override value for the network variable. To access a configuration property, you must select the type of configuration property in the DDE Topic, and the specific configuration property in the DDE Item.

---

## Specifying a Configuration Property Topic

The LNS DDE Server uses the configuration property topics to send and receive information about configuration properties. There are four configuration property topics: DevCP; DevNVCP; LMCP; and LMNVCP. Within the configuration property topic, the LNS DDE Server requires 2-4 pieces of information to be able to identify a configuration property. How much information it needs depends on what type of configuration property topic it is.

A configuration property topic is specified using the standard topic format or custom topic format described in Chapter 4. The topic type will be one of the following configuration property topics:

Category	Topic Type	Description
Configuration Property	<i>DevCP</i>	Configuration properties for a device.
	<i>LMCP</i>	Configuration properties for a LONMARK object.
	<i>LMNVCP</i>	Configuration properties for a network variable member of a LONMARK object.
	<i>DevNVCP</i>	Configuration properties for a network variable on a device. This type accesses configuration properties for network variables on a device that are <u>not</u> members of a LONMARK object.
	<i>NVCP</i>	Configuration properties for a network variable. As with the NV topic type, any network variable may be specified. The network variable name may be ambiguous when specified using this topic type.

---

## Specifying a Configuration Property Item

A configuration property item identifies a device and a configuration property or configuration property field on the device.

The format for specifying this information is the following:

```
device_name[.lm_name][.nv_name[[array_index]]].cp_name[.field_name] [parameters]
```

Where each component has the following meaning:

<i>device_name</i>	Required. The name of a device, as defined in the LNS database.
<i>lm_name</i>	Required if the LMCP or LMNVCP topic is used. The LONMARK Object name as defined in the LNS database.
<i>nv_name</i>	Required if the DevNVCP or LMNVCP topic is used. The name of the network variable on the device, as defined in the LNS database.
<i>array_index</i>	Required if the DevNVCP or LMNVCP topic is used and the network variable is an element in an array of network variables. <i>array_index</i> is the array index in square brackets.
<i>cp_name</i>	Required. The name of the configuration property, as defined in the LNS database.
<i>field_name</i>	Optional. A field within a structure configuration property. Fields may be within nested structures. Fields are only available for structured SCPTs and for user defined types defined in a user resource file. The field name must be a parameter contained within a <code>text</code> format definition in the format file for either a SCPT or a user-defined type. A master list of SCPTs is contained in the <i>LNS Utilities and LONMARK Reference</i> help file included with the LNS DDE Server. For user-defined structured data types, the field names are as defined a user resource file. For more information, see Appendix A, <i>LNS DDE Server Resource Files</i> . The field cannot be a union.
<i>parameters</i>	Optional. Multiple parameters can be specified. For each parameter, zero or more spaces can separate the parameter and its argument.

The parameter options are:

[ <i>index</i> ]	For a configuration property table, accesses elements of an array where <i>index</i> is a number between 0 and 999.
<b>-i</b> <i>nnnnn</i>	Controls the rate at which a configuration property value is checked for updates in the data base. This is specified in tenths of a second, where <i>nnnnn</i> is a number between 1 and 32000. The default is the setting specified for the network in the Network Configuration dialog.

`-t type_name`

Specifies a format name to use for formatting the configuration property as described in the next section. If this option is not specified, a format is selected by LNS. Use this option and specify a format name to change the format. The format name can be one of the five built-in formats (INTEGER, DISCRETE, REAL, STRING, or RAW), a format name from the standard resource file (for SNVTs and SCPTs), or a format name from any user resource file. If the format name is RAW, data will be returned in raw format.

When changing the data format using this parameter, the change is applied to the entire data value. This may cause unexpected results. For example, when reading a network variable which is a floating point value, and changing the format to an integer (-t INTEGER), the formatted value will be an integer interpretation of the entire floating point number, not just the integer part. The same thing applies when changing the data type to floating point (-t REAL) for an integer value. Make sure formats are compatible with the network variable value.

**Note:**

Floating point values must be between  $-3.4E38$  and  $+3.4E38$ .

---

## Accessing Configuration Property User Formats

Each configuration property type has a default format. An LNS network tool such as the LonMaker Integration Tool can change the default format for any configuration property. For example, the LonMaker browser allows a user to assign a custom format to any configuration property by right-clicking the value and selecting the Change Format command. To change the format, use the `-t` parameter described in the previous section. You may either specify a standard format, or specify a user format from a user format file as described in Appendix A, *Resource Files*, using the following syntax:

```
#<program id>[<selector>].<format name>
```

The fields are defined as follows:

<code>program id</code>	The program ID specified in the format file, in hex, without spaces or colons. This does not have to match the program ID for the device containing the network variable.
<code>selector</code>	The scope selector specified in the format file. The second line of the format file specifies a scope name that specifies which device types are covered by the format file. Translate this selector name to a scope selector using the following table:

<b>Selector Index</b>	<b>Selector</b>
1	DEVICE_CLASS
2	DEVICE_CLASS and SUB_CLASS
3	MANUFACTURER
4	MANUFACTURER and DEVICE_CLASS
5	MANUFACTURER and DEVICE_CLASS and SUB_CLASS
6	MANUFACTURER and DEVICE_CLASS and SUB_CLASS and MODEL

The scope selector does not have to match the selector used by the functional profile template defining the LonMark object(s) on the device.

`format type` The format type name specified in the format file. The format type name must match a type name in the type file associated with the format file, or in a type file with a lower scope selector value.

`format modifier` A modifier for the format type name that allows multiple formats to be specified for the same type.

---

## Requesting a One-Time Configuration Property Update

To request a one-time update, your application sends a DDE Request message to the LNS DDE Server. The LNS DDE Server responds with the requested value. This works almost exactly like requesting network variables as described in Chapter 5; the difference is how the item is specified and which topic type is used.

### Example

Here are two examples of application, topic, and item specifications for requesting a configuration property update.

Example 1:

```
Application    LNSDDE
Topic         NetworkDDE.HVAC.LMCP
Item         DO- 1.TransTemp.UCPTenumMap[ 7]
```

Example 2:

```
Application    LNSDDE
Topic         XD32.SubC.LMNVCP
Item         AO- 1.PID Controller[1].PID_SP_2.UCPTdefaultIn
```

See *Requesting a One-Time Update Using Excel* and *Requesting a One-Time Update Using InTouch* in Chapter 5, for Excel and InTouch examples.

---

## Receiving Ongoing Configuration Property Updates

Receiving ongoing configuration property updates is similar to receiving network variables as described in Chapter 5. However, for configuration properties, you need to specify the topic and item as described in *Specifying a Configuration Property* earlier in this chapter. See *Receiving Ongoing Updates* in Chapter 5 for an explanation.

The RefreshRate global option specifies the rate that the LNS DDE Server uses to check to see if a configuration property value has changed. When advising configuration properties, this is the maximum rate at which your application will see changes.

---

## Sending a Configuration Property Update

Sending a configuration property update resembles sending a network variable update. Again, the difference is how the item is specified and which topic type is used.

### Example

Here are two examples of what the application, topic, and item might look like when sending a control property update.

Example 1:

<i>Application</i>	LNSDDE
<i>Topic</i>	Testnet.Subsystem 1.LMCP
<i>Item</i>	DI-1.Switch3.UCPTonDelay.second

Example 2:

<i>Application</i>	LNSDDE
<i>Topic</i>	Sample.HVAC.LMNVCP
<i>Item</i>	AO.PID Controller[1].PID_SP_2.UCPTpresetTabFlt[ 4].useTable

See *Sending a Network Variable Update Using Excel* and *Sending a Network Variable Update Using InTouch* in Chapter 5, for InTouch examples.

# 7

## **Receiving and Sending Application and Foreign Frame Messages**

This chapter describes how to send and receive application and foreign frame messages.

---

## Specifying an Incoming Message Item

Application and foreign frame messages are typically used for proprietary, non-interoperable interfaces to LONWORKS devices. To receive application or foreign from messages from a device, use the MT topic and specify the item as follows:

```
[device_name.]msg_in[.msg_code] [parameters]
```

Each component of the item format has the following meaning:

<i>device_name</i>	Optional. Specifies the name of an application device, as defined in the LNS database. If not specified, the item denotes all incoming messages addressed to the LNS DDE Server host.
<b>msg_in</b>	Required. <b>msg_in</b> is a keyword, not a user-defined name. This keyword is case insensitive.
<i>msg_code</i>	Optional. Specifies monitoring of messages with the given <i>msg_code</i> , in decimal. Application message codes are from 0 to 62. Foreign frame message codes are from 64 to 78. Do not specify message codes outside of this range. If a message code is specified, then the data returned by the LNS DDE Server does not contain the code. If a message code is not specified, then the first byte of the data returned by the LNS DDE Server contains the received message code.
<i>Parameters</i>	Optional.

The parameter options are:

<b>-n</b> <i>bool</i>	<p>Specifies whether the client will receive message items from the LNS DDE Server even if the data has not changed from the last time this message was received from the network. Valid values are <code>True</code> and <code>False</code>. If <code>False</code>, data is reported to the client only if it has changed. If <code>True</code>, a byte containing an incrementing counter will be appended to each incoming message. If not specified, the default value specified in the Messaging Options configuration dialog is used.</p> <p>This option may not work as expected for a series of messages when some messages use <code>-nTrue</code> and some use <code>-nFalse</code>. It is recommended that you standardize on either using this option or not then set it globally using the <code>Include Sequence Number</code> option in the Network Configuration Settings dialog.</p>
-----------------------	---

**-t** *type\_name* Specifies a format name to use for formatting the message data as described in the next section. If this option is not specified, a format is selected by LNS. Use this option and specify a format name to change the format. The format name can be one of the five built-in formats (INTEGER, DISCRETE, REAL, STRING, or RAW), a format name from the standard resource file, or a format name from any user resource file. If the format name is RAW, data will be returned in raw format.

When changing the data format using this parameter, the change is applied to the entire data value. This may cause unexpected results. For example, when reading a network variable which is a floating point value, and changing the format to an integer (-t INTEGER), the formatted value will be an integer interpretation of the entire floating point number, not just the integer part. The same thing applies when changing the data type to floating point (-t REAL) for an integer value. Make sure formats are compatible with the network variable value.

**Note:**

Floating point values must be between  $-3.4E38$  and  $+3.4E38$ .

---

## Specifying a Request Message

When the LNS DDE Server sends a message using the REQUEST service type on the msg\_out item, it expects the destination device or devices to send response messages back in reply. If the request message was sent using one of the unicast (i.e. subnet/node) addressing modes (described later in this chapter), only one response message will arrive. If the request message was sent using a multicast addressing mode, one or more devices may respond.

Use the MT topic to receive response messages sent in response to an outgoing request from the LNS DDE Server. Within the MT topic, the LNS DDE Server uses the following item format for receiving response messages:

```
[device_name.]resp_in[.msg_code] [parameters]
```

Where each component of the item format has the following meaning:

*device\_name* Optional. Specifies the name of an application device, as defined in the LNS database. If not specified, the item specifies all incoming responses addressed to the LNS DDE Server host PC.

**resp\_in** Required. **resp\_in** is a keyword, not a user-defined name. This keyword is case insensitive.

*Msg\_code* Optional. Specifies monitoring of responses with the given message code, in decimal. If a message code is specified, then the data returned by the LNS DDE Server does not contain the code. If a message code is not specified, then the first byte of the data returned by the LNS DDE Server contains the received response code.

*Parameters* Optional.

The parameter options are:

**-n** *bool* Specifies whether the client will receive message response items from the LNS DDE Server even if the data has not changed from the last time this response was received from the network. May be set to `True` or `False`. If `False`, data is reported to the client only if it has changed. If `True`, a byte containing an incrementing counter will be appended to each incoming response. If not specified, the default value specified in the Messaging Options configuration dialog is used.

This option may not work as expected for a series of messages when some messages use `-nTrue` and some use `-nFalse`. It is recommended that you standardize on either using this option or not then set it globally using the `Include Sequence Number` option in the Network Configuration Settings dialog.

**-t** *type\_name* Specifies a format name to use for formatting the response. For response messages, the LNS DDE Server does not automatically determine the format. The format name can be one of the five built-in formats (`INTEGER`, `DISCRETE`, `REAL`, `STRING`, or `RAW`), a format name from the standard resource file, or a format name from any user resource file. If the format name is `RAW`, data will be returned in raw format.

When changing the data format using this parameter, the change is applied to the entire data value. This may cause unexpected results. For example, when reading a network variable which is a floating point value, and changing the format to an integer (`-t INTEGER`), the formatted value will be an integer interpretation of the entire floating point number, not just the integer part. The same thing applies when changing the data type to floating point (`-t REAL`) for an integer value. Make sure formats are compatible with the network variable value.

**Note:**

Floating point values must be between `-3.4E38` and `+3.4E38`.

See *Example 2* in *Requesting a One-Time Update* later in this chapter for an example of a request using this specification.

**Note:** To receive a response, you must establish an automatic link on `resp_in` in the DDE Client before sending the request on `msg_out` (i.e. the response message must be automatically forwarded to the client). Otherwise, the response will be discarded.

---

## Specifying a Message Completion Event

Completion events are used to determine the success or failure of outgoing messages sent by the LNS DDE Server. Completion events are not generated for messages sent using unacknowledged or repeated service. Completion events are generated for messages using the acknowledged or request/response service.

If the original message was sent to a single device using the acknowledged or request/response service, the message is considered a success if an acknowledgement or response is received from the device.

If the original message was sent to multiple devices and used acknowledged or request/response service, the message is considered a success if acknowledgements or responses are received from all of the destination devices. The acknowledged and request/response services are only supported when sending messages to 64 devices or fewer. If one or more acknowledgements or responses are not received within the configured number of retries and timeout interval, the message is considered to have failed. No information as to the identity of the non-responders is returned in a failure completion event.

Use the MT topic to receive completion events. Within the MT topic, the LNS DDE Server uses the following item format for completion events:

```
[device_name.]msg_complete[.msg_code] [parameters]
```

Where each component of the item format has the following meaning:

<i>device_name</i>	Optional. Specifies the name of an application device, as defined in the LNS database. If not specified, the item specifies all completion events.
<b>msg_complete</b>	Required. <b>msg_complete</b> is a keyword, not a user-defined name. This keyword is case insensitive.
<i>msg_code</i>	Optional. Specifies the completion status of outgoing messages sent with the given message code in decimal.
<i>Parameters</i>	Optional.

The parameter options are: as follows

**-n** *bool* Specifies whether the client will receive message response items from the LNS DDE Server even if the data has not changed from the last time this response was received from the network. May be set to `True` or `False`. If `False`, data is reported to the client only if it has changed. If `True`, a byte containing an incrementing counter will be appended to each incoming response. If not specified, the default value specified in the Messaging Options configuration dialog is used.

This option may not work as expected for a series of messages when some messages use `-nTrue` and some use `-nFalse`. It is recommended that you standardize on either using this option or not then set it globally using the Include Sequence Number option in the Network Configuration Settings dialog.

**-t** *format\_name* Specifies a format name to use for formatting the data in the incoming response. For response messages, the LNS DDE Server does not automatically determine the format. The format name can be one of the five built-in formats (`INTEGER`, `DISCRETE`, `REAL`, `STRING`, or `RAW`), a format name from the standard resource file (for SNVTs and SCPTs), or a format name from any user resource file. If the format name is `RAW`, data will be returned in raw format.

When changing the data format using this parameter, the change is applied to the entire data value. This may cause unexpected results. For example, when reading a network variable which is a floating point value, and changing the format to an integer (`-t INTEGER`), the formatted value will be an integer interpretation of the entire floating point number, not just the integer part. The same thing applies when changing the data type to floating point (`-t REAL`) for an integer value. Make sure formats are compatible with the network variable value.

**Note:**

Floating point values must be between  $-3.4E38$  and  $+3.4E38$ .

The completion code can be one of the following:

0 ( <code>LON_OK</code> )	The message succeeded.
7 ( <code>LON_MSG_FAILED</code> )	The message failed.
10 ( <code>LON_TIMEOUT</code> )	The network interface did not respond.

**Note:** To receive a completion event, you must establish an automatic link on `msg_complete` in the DDE Client before sending the request on `msg_out` (i.e. the

response message must be automatically forwarded to the client). Otherwise, the completion will be discarded.

See *Example 3* in *Requesting a One-Time Update* later in this chapter for an example of a request using this specification.

---

## Specifying an Outgoing Application Message

The LNS DDE Server uses the following item format for sending application and foreign frame messages:

```
[device_name.]msg_out[.msg_code] [parameters]
```

Where each component of the item format has the following meaning:

<i>device_name</i>	Optional. The name of an application device as defined in the LNS database.
<b>msg_out</b>	Required. <b>msg_out</b> is a keyword, not a user-defined name. This keyword is case insensitive.
<i>msg_code</i>	Optional. Specifies the message code, in decimal. Application message codes are from 0 to 62. Foreign frame message codes are from 64 to 78. Do not specify message codes outside this range. If a message code is specified, the item data contains just the message. If a message code is not specified, then the first byte of the item data must contain the message code.
<i>Parameters</i>	Optional.

The parameter options are:

<b>-a</b> <i>bool</i>	Specifies whether the message should be sent with an authenticated service. May be set to <code>True</code> or <code>False</code> . If not specified, the default value specified in the Messaging Options configuration dialog is used.  This option is ignored when used with broadcast messages; it applies only to unacknowledged or repeated service types.
<b>-p</b> <i>bool</i>	Specifies whether the message should be sent with priority service. May be set to <code>True</code> or <code>False</code> . If not specified, the default value specified in the Messaging Options configuration dialog is used.

- r** *nn* Specifies the retry count to be used for acknowledged and request/response service or the repeat count to be used for repeated service, where *nn* is a number between 0 to 15. If not specified, the default value specified in the Messaging Options configuration dialog is used.
  
- s** *service* Specifies the service type to use when sending the message. May be set to ACKD (acknowledged), UNACKD (unacknowledged), UNACKD\_RPT (repeated) or REQUEST (request/response). Messages being sent to more than 64 devices may not use the acknowledged or request/response services. If not specified, the default value specified in the Messaging Options configuration dialog is used.
  
- t** *format\_name* Specifies a format name to use for formatting the data in the outgoing message. For response messages, the LNS DDE Server does not automatically determine the format. The format name can be one of the five built-in formats (INTEGER, DISCRETE, REAL, STRING, or RAW), a format name from the standard resource file, or a format name from any user resource file. If the format name is RAW, data will be returned in raw format.

When changing the data format using this parameter, the change is applied to the entire data value. This may cause unexpected results. For example, when reading a network variable which is a floating point value, and changing the format to an integer (-t INTEGER), the formatted value will be an integer interpretation of the entire floating point number, not just the integer part. The same thing applies when changing the data type to floating point (-t REAL) for an integer value. Make sure formats are compatible with the network variable value.

**Note:**

Floating point values must be between -3.4E38 and +3.4E38.

See *Example 1* in *Sending an Application or Foreign Frame Message Update* later in this chapter for an example of how to send an application message.

## Specifying a Message Using Domain-Wide Broadcast Addressing

You will normally use a device name to specify the destination of an application or foreign frame message. For special applications, however, you may wish to broadcast a message for every device in a domain. The LNS DDE Server host PC must be a member of the domain specified.

You will use the MT topic to send domain-wide broadcast application or foreign frame messages. Within the MT topic, the LNS DDE Server uses the following item format:

**bcast\_d.msg\_out** [*.msg\_code*] [*parameters*]

Where each component has the following meaning:

<b>bcast_d</b>	Required. <b>bcast_d</b> is a keyword, not a user-defined name. This keyword is case insensitive.
<b>msg_out</b>	Required. <b>msg_out</b> is a keyword, not a user-defined name. This keyword is case insensitive.
<i>msg_code</i>	Optional. Specifies sending messages with the given message code, in decimal. Application message codes are from 0 to 62. Foreign frame message codes are from 64 to 78. Do not specify message codes outside this range. If a message code is specified, then the item data contains just the message. If a message code is not specified, then the first byte of the item data must contain the message code.
<i>Parameters</i>	Optional.

The parameter options are:

<b>-p</b> <i>bool</i>	Specifies whether the message should be sent with priority service. May be set to <code>True</code> or <code>False</code> . If not specified, the default value specified in the Messaging Options configuration dialog is used.
<b>-r</b> <i>nn</i>	Specifies the retry count to be used for acknowledged and request/response service or the repeat count to be used for repeated service, where <i>nn</i> is a number between 0 to 15. If not specified, the default value specified in the Messaging Options configuration dialog is used.
<b>-s</b> <i>service</i>	Specifies the service type to use when sending the message. May be set to <code>UNACKD</code> (unacknowledged) or <code>UNACKD_RPT</code> (repeated). If not specified, the default value specified in the Messaging Options configuration dialog is used.

**-t** *format\_name* Specifies a format name to use for formatting the data in the outgoing message. For response messages, the LNS DDE Server does not automatically determine the format. The format name can be one of the five built-in formats (INTEGER, DISCRETE, REAL, STRING, or RAW), a format name from the standard resource file, or a format name from any user resource file. If the format name is RAW, data will be returned in raw format.

When changing the data format using this parameter, the change is applied to the entire data value. This may cause unexpected results. For example, when reading a network variable which is a floating point value, and changing the format to an integer (-t INTEGER), the formatted value will be an integer interpretation of the entire floating point number, not just the integer part. The same thing applies when changing the data type to floating point (-t REAL) for an integer value. Make sure formats are compatible with the network variable value.

**Note:**

Floating point values must be between  $-3.4E38$  and  $+3.4E38$ .

See *Example 2* in *Sending an Application Message Update* later in this chapter for an example of how to send an application message.

---

## Requesting a One-Time Update

To receive an application or foreign frame message that is not a response to a request/response message, use the same procedure described for receiving network variables, except that you must specify the topic and item as described earlier in this chapter. Unlike network variables and configuration properties, the LNS DDE Server cannot poll a remote device to request the value of an application or foreign frame message; the LNS DDE Server can only receive application and foreign frame messages and must wait for the device to send a message to it. This means that if your application's request for an application message item is the first request for the item, the LNS DDE Server will send an empty string or numeric  $\emptyset$ . The LNS DDE Server will continue to return this initial data until the device sends a message for the item to the LNS DDE Server.

### Example

Here are three examples of what the application, topic, and item might look like when requesting an application message update assuming the network name is AppMsg Test, the application device name is switch, and the subsystem name is Subsystem 1.

**Example 1:**

<i>Application</i>	LNSDDE
<i>Topic</i>	AppMsg Test.subsystem 1.MT
<i>Item</i>	msg_in -t SNVT_count

**Example 2:**

<i>Application</i>	LNSDDE
<i>Topic</i>	AppMsg Test.subsystem 1.MT
<i>Item</i>	switch.msg_in

**Example 3:**

<i>Application</i>	LNSDDE
<i>Topic</i>	AppMsg Test.subsystem 1.MT
<i>Item</i>	switch.msg_in.1

See *Requesting a One-Time Update in Excel* and *Requesting a One-Time Update Using InTouch* in Chapter 5, for a description of how to request a one-time update.

## Requesting Ongoing Message Updates

To receive ongoing updates, use the same procedure described for receiving ongoing network variable updates in Chapter 5, except that you must specify the topic and item as described earlier in this chapter. When you ask the LNS DDE Server to monitor an application message, it looks in the LNS database to see if it can receive an application message from the specified device. If the LNS DDE Server cannot receive an application message from the specified device, the LNS DDE Server refuses to accept the request from your client (it returns a DDE negative acknowledgement, a NAK). Like network variables, the LNS DDE Server uses a *report by exception* strategy by default; only changes in a message's value are reported. This may be overridden with the `-n` switch when the conversation is initiated.

Unlike network variables, the LNS DDE Server can monitor application messages only if it is bound to the device sending the message or if the monitored device explicitly addresses messages to the LNS DDE Server host PC.

See *Receiving Ongoing Updates with Excel* and *Receiving Ongoing Updates with InTouch* in Chapter 5, for examples of receiving ongoing updates.

## Sending a Message Update

To send an application or foreign frame message, use the same procedure described for network variable updates in Chapter 5, except that you must specify the topic and item as described earlier in this chapter.

### Example

Here are three examples of what the application, topic, and item might look like when sending an application message update, assuming a network (AppMsgTest), an application device (switch), and a subsystem (subsystem1).

**Example 1:**

<i>Application</i>	LNSDDE
<i>Topic</i>	AppMsg Test.subsystem 1.MT
<i>Item</i>	msg_in -t SNVT_count

**Example 2:**

<i>Application</i>	LNSDDE
<i>Topic</i>	AppMsg Test.subsystem 1.MT
<i>Item</i>	switch.msg_in

**Example 3:**

<i>Application</i>	LNSDDE
<i>Topic</i>	AppMsg Test.subsystem 1.MT
<i>Item</i>	switch.msg_in.1

See *Requesting a One-Time Update in Excel* and *Requesting a One-Time Using InTouch* in Chapter 5, for a description of how to request a one-time update.

# 8

## Using Properties

This chapter describes the properties available for the different topic types.

---

## Properties

The LNS DDE Server provides access to additional properties besides the network variable, configuration property, and message values described in Chapters 5, 6, and 7. The available properties are dependent on the DDE topic. The following sections describe these properties.

---

### Specifying a Property

A property topic is specified using the standard or custom topic format described in Chapter 4. The topic specifies a category of properties, and the item includes the property name. The following table lists the topic type, item syntax, and example application, topic, and item specifications for each category.

Category	Topic Type	Item Syntax	Example Specifications
System Properties	System	<i>Property</i>	<i>Application:</i> LNSDDE <i>Topic:</i> LNS DDE Test.System <i>Item:</i> Point.InError
Device and Router Properties	Dev	<i>Device.property</i>	<i>Application:</i> LNSDDE <i>Topic:</i> LNS DDE Test.HVAC.Dev <i>Item:</i> AO -1.State
LonMark Object Properties	LM	<i>Device.lmo.property</i>	<i>Application:</i> LNSDDE <i>Topic:</i> LNS DDE Test.HVAC.LM <i>Item:</i> AO -1.SA Temp.PID.Request.Override
Point Properties	LMNV, DevNV, Netvar, NV, DevCP, LMCP, LMNVCP, DevNVP, or NVCP	<i>Device[.lmo].nv.property</i> <i>or</i> <i>device[.lmo][.nv].cp.property</i>	<i>Application:</i> LNSDDE <i>Topic:</i> LNS DDE Test.HVAC.LMNV <i>Item:</i> AI -1.SA Temp.Analog.DDE_ReadCount

---

### System Properties

System properties provide access to values that apply to a entire LNS network. The properties are accessed through the system topic, using the following syntax:

*Network\_name.system*

The *network\_name* is the name of the network as defined in the LNS database. It is case sensitive.

<i>Point.Count</i>	R	Number of points accessed in the network.  The count is not exact. The server keeps all points ever accessed in a single device in the point cache for as long as at least one point is still accessed in the same device. After the last point is no longer accessed, there is a delay of about 15 seconds until the device and all points belonging to it are deleted from the LNS DDE Server point cache, and the count of points is decreased.
<i>Point.Active</i>	R	Number of points currently being accessed in the network.  The active count is not identical to the total count of points accessed. The points marked active are the points that are currently being read from devices or the LNS database.

<i>Point.InError</i>	R	Number of points that cannot be read in the network.
<i>Point.WriteCount</i>	R	Number of write requests issued for this point.  The number of writes issued may not exactly match the aggregate count of updates issued by all clients for a point. The server will fold together all currently queued write requests for a single point. This also means that if two separate fields in a point are updated at nearly the same time, there is a good chance that the server will only need to issue one write request to update both fields.
<i>Point.WriteErrors</i>	R	Number of times a write operation has failed.  This count only applies to errors that occur after the server has accepted and queued an update request. An update with a value that the server immediately recognizes as invalid, is not counted.
<i>Point.UpdateErrors</i>	R	Number of times the LNS Server has reported an error when attempting to poll a network variable, or when some other problem occurred when trying to start polling a network variable or reading a configuration property.
<i>CP.RefreshRate</i>	R/W	This is the update interval for configuration properties as well as some system, device, and point properties.  Changing this value will only affect points that are accessed after the change, not points that are currently being accessed.
<i>ServicePin.Count</i>	R/W	Number of service pin messages received.
<i>ServicePin.Handle</i>	R/W	Handle for the device that last issued a service pin message.
<i>ServicePin.NeuronId</i>	R/W	Neuron ID for the device that last issued a service pin message. A string of twelve characters with the hexadecimal value of the Neuron ID of the device that issued the service pin message.
<i>ServicePin.ProgramId</i>	R/W	Program ID for the device that last issued a service pin message, represented as a hexadecimal string.
<i>ServicePin.Location</i>	R/W	Location string for the device that last issued a service pin message, represented as a hexadecimal string.
<i>Attachment.Found</i>	R/W	Number of attachment events for found devices. Tallies the number of devices that were found on the network after failing previously when pinged by the LNS Server. The Attachment.Found value will be less than or equal to the Attachment.Lost value unless the properties are written by the client application.  Attachment events are only generated for devices which are pinged by the LNS Server.
<i>Attachment.Lost</i>	R/W	Number of attachment events for lost devices. Tallies the number of devices that fail to respond to ping messages from the LNS Server.  Attachment events are only generated for devices which are pinged by the LNS Server.

---

## Device and Router Properties

Application device and router properties provide access to values and functions for individual devices on a network. Although the properties available for these two types of devices overlap to some extent, there are differences.

The list of supported properties below has been partitioned into three parts; common properties, application device only properties, and router only properties.

Some properties are stored in the LNS database, while others require the information to be fetched from the device. In case the server can't read a specific property from the device, the property is set to zero.

*Note: It is possible to have application devices and routers with the same name, in the same subsystem. When attempting to read a router property, and an application*

*device with the same name exists in the same subsystem, the value of the property for the application device will be returned. Change the name of either the router or application device to receive properties for each one.*

## Properties Available in both Application Devices and Routers

<i>Reset</i>	W	Issues a reset command to the device or router.
<i>Test</i>	R/W	<p>Tests an application device or router. The test consists of determining that the device can communicate on the network, and that the device information is consistent with what is registered in the LNS database for the device.</p> <p>The test must first be invoked by writing a value to this property. Any property can be written, but it is recommended you write a value which is not a possible return value such as the value 99. While the test is in progress, the written value will be returned as the value of this property. Once the test has been completed, the result of the test is returned as the value of this property. The result codes and their meaning are as follows;</p> <ol style="list-style-type: none"> <li>0 The device passed all applicable tests.</li> <li>1 Unable to communicate with the device. The device might not be attached to the network, powered off, or faulty.</li> <li>2 Unable to communicate with the device using Neuron ID addressing, and could not test the device using subnet/node ID addressing because the device has not been added or defined yet.</li> <li>3 Intermittent communication failure. Succeeded in communicating with the device using subnet/node addressing, but could not communicate with the device using Neuron ID addressing. Retry the test to see if the error persists.</li> <li>4 Unable to verify subnet/node configuration. Succeeded in communicating with the device using Neuron ID addressing, but could not communicate with the device using subnet/node addressing. Could not verify subnet/node configuration.</li> <li>5 Unable to communicate using subnet/node addressing. Succeeded in communicating with the device using Neuron ID addressing, but could not communicate with the device using subnet/node addressing. However, the device is configured with the expected subnet/node address. This error can occur if the device is unconfigured state, or might be caused by intermittent device or channel failures. Retry the test to see if the error persists.</li> <li>6 Neuron ID mismatch. Succeeded in communicating with the device using Neuron ID addressing. However, when using subnet/node addressing, the responding device contains a different Neuron ID. This failure indicates that multiple devices are configured with the same domain/subnet/node address.</li> <li>7 Domain ID mismatch. The domain ID in the device and the domain ID configured for the device does not match.</li> <li>8 Neuron ID mismatch. Unable to communicate with the device using Neuron ID addressing. However, the device that responded to the subnet/node addressed test message contains a different Neuron ID. It appears that the subnet/node address is configured in the responding device but not the tested device. One possible cause is that the tested device is supposed to replace the responding device, but the network images in both devices have not been updated yet.</li> <li>9 Subnet/node mismatch. The subnet/node address configured in the device does not match the network image. This result will be returned if the device's domain address has not been configured.</li> <li>10 Neuron ID not configured. This result will be returned if the device was added to the network, but without specifying a Neuron ID, and</li> </ol>

the device has not yet been commissioned.

- 11 Program ID mismatch. The ID of the program in the device does not match the program ID in the network image.
- 12 Unable to verify domain/subnet/node configuration. Succeeded in communicating with the device using Neuron ID addressing, but could not communicate with the device using subnet/node addressing. Because the device has authentication enabled, no attempt is made to verify that the device has been configured with the proper domain/subnet/node address because reading the address would result in transmitting the key over the network.
- 13 Authentication configuration mismatch. Authentication is enabled in the device, but disabled in the network image.
- 14 Authentication configuration mismatch. Authentication is disabled in the device, but enabled in the network image.
- 15 Authentication key mismatch. The authentication key in the device and the system's authentication key do not match.

**Note:** This test is only reliable if the device's network image is up to date. For example, if you made changes to a device while Offnet and tested it before going Onnet, the test may fail.

<i>Description</i>	R/W	Descriptive information about the device or router. The DDE Server will return only the first 95 characters of the description.
<i>Location</i>	R	The value of the specified device's or router's six byte location string as an ASCII string. The string may not be meaningful in ASCII format, and if a byte holds a zero value, the string will end there even if bytes that follow have a non-zero value.
<i>Location.hex</i>	R	The value of the device's or router's six byte location string as a hex string of twelve characters.
<i>Name</i>	R/W	The name of the device or router.
<i>PingClass</i>	R	The frequency with which a device is tested (pinged) by the LNS Server to determine if it is still attached to the network. The values returned by this property have the following meaning: <ol style="list-style-type: none"> <li>1 Mobile, frequently moving device – 1 minute ping.</li> <li>2 Temporary device – 2 minute ping.</li> <li>3 Stationary device – 15 minute ping.</li> <li>4 Permanent device – no ping.</li> </ol>
<i>ServicePin.Count</i>	R/W	Number of service pin messages received from the device.
<i>Attachment.Found</i>	R/W	Number of attachment events where the device was responding. Tallies the number of times contact was restored with the device after previously failing to respond to pings from the LNS Server. The Attachment.Found value will be less than or equal to the Attachment.Lost value unless the properties are written by the client application.  Attachment events are only generated for devices which are pinged by the LNS Server.
<i>Attachment.Lost</i>	R/W	Number of attachment events where this device failed to respond. Tallies the number of times contact was lost with the device after previously succeeding.  Attachment events are only generated for devices which are pinged by the LNS Server.
<i>Attachment.IsAttached</i>	R/W	True if the last attachment event for this device indicated the device was responding, False if it did not respond.

## Properties Available only in Application Devices

<i>Wink</i>	W	Winks the device, if supported by the device's application.
<i>SelfDocumentation</i>	R	The self-documentation string of a device.  The length of this string may be up to 1024 characters. However, the server limits the length to about 170 characters.
<i>Point.Active</i>	R	Number of points currently being accessed.
<i>Point.InError</i>	R	Number of points that cannot be read.
<i>Point.WriteCount</i>	R/W	Number of writes issued to this device.
<i>Point.WriteErrors</i>	R/W	Number of times a write operation has failed.  This count only applies to errors that occur after the server has accepted and queued an update request. An update request with a value that the server immediately recognizes as invalid is not counted.
<i>Point.UpdateErrors</i>	R/W	Number of times the LNS Server has reported an error to poll a network variable, or when some other problem occurred when trying to start polling a network variable or read a configuration property.

## Properties Available in both Application Devices and in each side of Routers (NearSide and FarSide)

<i>NeuronId</i>	R	The Neuron ID, stored as a 12 character string of hexadecimal values.
<i>ProgramId</i>	R	The Program ID, stored as a 8 character string of hexadecimal values.
<i>State</i>	R	The state of the device or router side. The values returned by this property have the following meaning; <ul style="list-style-type: none"> <li>0 Unable to retrieve the current state from the device. The device is not connected, powered off, or malfunctioning.</li> <li>2 Unconfigured. Application loaded, but not configured, or configuration invalid (checksum error).</li> <li>3 No application. The device has no application loaded, the application does not pass the checksum test, or there is a signature inconsistency.</li> <li>4 Configured online. The device has an application loaded, the application is running, and the configuration is considered valid.</li> <li>6 Configured offline. The device has an application loaded, the application is not running, and the configuration is considered valid.</li> <li>12 Soft offline. The device has an application loaded, is configured, and is soft offline. It will go on-line when reset or requested to go online.</li> <li>140 Bypass mode. The application confirmed the offline request, but is still running.</li> </ul>
<i>DetailInfo.ErrorLog</i>	R	The most recently logged firmware error.
<i>DetailInfo.LostMessages</i>	R	The number of times a packet was received and no application input buffer was available.
<i>DetailInfo.MissedMessages</i>	R	The number of times a packet was received and no network input buffer was available.

<i>DetailInfo.ModelNumber</i>	R	The type of protocol processor in the device. If the protocol processor is a Neuron Chip, this will be one of the following values:  0 Model 3150 8 Model 3120 9 Model 3120E1 10 Model 3120E2 11 Model 3120E3 12 Model 3120A20 13 Model 3120E5  A value of 128 or greater indicates a non-Neuron Chip protocol processor, such as a microcontroller running the Adept implementation of the LONWORKS protocol.
<i>DetailInfo.Reset Cause</i>	R	The cause for the last reset on the device.  0 No reset has occurred since the last time this status was cleared 1 The last reset occurred as a result of a powerup. 2 The last reset was caused by activation of the reset pin. 12 The last reset was caused by a watchdog timer timeout. 20 The last reset was caused by software.
<i>DetailInfo.TransactionTimeout</i>	R	The number of times an acknowledged or request/response message failed after all the retries. This number stops increasing when it reaches 65,535, and is set to zero when the device power cycles or when a Clear status command is issued to the device.
<i>DetailInfo.VersionNumber</i>	R	The version number of the protocol processor (typically the version of the Neuron Chip firmware) contained in a device.
<i>DetailInfo.XmitErrors</i>	R	The number of times the device received a packet with an invalid CRC, received a packet that was too short, received a packet too long for the device's input buffer, or saw a timeout. This number stops increasing when it reaches 65,535, and is set to zero when the device power cycles or when a Clear status command is issued to the device.
<i>DetailInfo.ReceiveTxFull</i>	R	The number of times the device or router received a packet for which there was no more room in the device's transaction database.

---

## LONMARK Object Properties

The LONMARK Object's properties provide access to individual LONMARK objects on LONMARK -compliant devices. LONMARK objects are not available on all devices, and some of the properties are optional for devices that have LONMARK objects. The device must have a LONMARK -certified (type 8) or LONMARK -compliant (type 9) program ID to use these properties.

<i>Description</i>	R/W	Descriptive information about the LONMARK object.
<i>FuncProfileDescription</i>	R	Descriptive comment of the functional profile for the LONMARK object.
<i>Name</i>	R/W	The name of the LONMARK object.
<i>Request.Normal</i>	R/W	Set to 1 to cause the object to be Enabled, and not in Override. Setting to 0 has no effect.
<i>Request.Disable</i>	R/W	Set to 1 to Disable an object. Set to 0 to Enable it.
<i>Request.SelfTest</i>	R/W	Set to 1 to execute an object self-test.
<i>Request.Override</i>	R/W	Set to 1 to place object in Override. Set to 0 to remove the Override condition.

<i>Request.Enable</i>	R/W	Set to 1 to Enable object. Set to 0 to Disable it.
<i>Request.RemoveOverride</i>	R/W	Set to 1 to remove the Override condition of an object. Set to 0 to place the object in Override.

---

## Point Properties

The point properties provide access to values that apply to individual network variables, configuration properties, and message tags. Some properties are not meaningful for configuration properties.

These properties are maintained entirely inside the LNS DDE Server. The properties can be used to change the current monitoring mode and other options that are otherwise controlled by specifying parameters in the item.

<i>DDE_Authenticated</i>	R	Authenticated messages. Does not apply to configuration properties.
<i>DDE_Direction</i>	R	Input (1) or output (0) type. Does not apply to configuration properties.
<i>DDE_Override</i>	R/W	Set to 1 to force the server to use your settings for the item.
<i>DDE_PollInterval</i>	R/W	Current poll interval in 1/10 second rate. Set to 0 to make the point update only once. Set to -1 to allow the server to select the current update interval automatically.
<i>DDE_Priority</i>	R/W	Priority. Does not apply to configuration properties.
<i>DDE_ReadErrors</i>	R/W	Number of times the server had issued a read and received an error from the LNS Server.
<i>DDE_Retry</i>	R/W	Number of retries. Does not apply to configuration properties.
<i>DDE_Valid</i>	R	Point value is valid. Set to 1 first time the point value is reported.
<i>DDE_ReadCount</i>	R/W	Number of times the server has issued a read. For network variables, this count only reflects the number of times the LNS DDE Server has requested the underlying LNS Data Server to start polling the network variable, not the actual number of polls issued. For other point types like configuration properties, the count is the actual number of times a request to read the point has been issued.
<i>DDE_WriteCount</i>	R/W	Number of times the point has been written to.
<i>DDE_WriteDelay</i>		<p>A millisecond timer which begins counting down when a network variable value is updated using the LNS DDE Server. The new value will not be sent to the device until the timer expires. After the timer expires, this property remains at 0 until explicitly reset to another value.</p> <p>This property should be used when a network variable structure is having multiple fields updated to prevent a separate network variable update being sent on the network for each field updated. Set this property to a value such that all fields will be changed before the update is sent.</p> <p>This value may be from 0 to 65535 ms. The default value of 0 indicates no delay.</p>
<i>DDE_WriteErrors</i>	R/W	Number of times the server has tried to write to a point but the operation has failed.
<i>DDE_UpdateErrors</i>	R/W	Number of times the LNS Data Server has reported an update error for the point.

# Appendix A

## Resource Files

This chapter discusses LONMARK resource files and how to use the Resource File Catalog Utility to add new resource files to the resource catalog.

---

## LONMARK Resource Files

The LNS DDE Server supports LONMARK resource files that define standard and user-defined network variable types, configuration property types, and functional profile templates (definitions of LONMARK object types). The LNS DDE Server is compatible with resource files that comply with version 3.2 of the *LonMark Application Layer Interoperability Guidelines*.

Standard network variable types (SNVTs) and standard configuration property types (SCPTs) are defined in the `STANDARD.TYP` file. Standard functional profile templates are defined in the `STANDARD.FPT` file. These standard types are defined by the LONMARK Interoperability Association. Formatting instructions for the standard types are defined in the `STANDARD.FMT` file. These files are updated periodically as the LONMARK Interoperability Association defines new standard types. The latest version of the standard resource files is available on the LONMARK website at <http://www.lonmark.org/products/guides.htm> under *SNVT and SCPT*.

Resource files also include language-dependent string files. The filename extension of the string file depends on the language being used. For example, the US English string file is named `STANDARD.ENU` and the UK English string file is named `STANDARD.ENG`.

In addition to the standard resource files, there may be multiple sets of user-defined resource files. (See the Device Resource File help file for more information about user-defined resource files.) These files are keyed by components of the program ID, using standard formats 8 (LONMARK certified device) and 9 (LONMARK compliant device). Since there may be many user-defined resource files, a standard resource catalog is used to manage and organize the resource files. This resource catalog may be modified using the LNS Resource File Catalog Utility. The resource catalog also may be updated automatically by a device manufacturer's software installation program. See the *LNS Resource File Catalog Utility User's Guide* under the *LonMark Reference* topic in the *LNS Utilities and LonMark Reference* help file for more information. A shortcut to this help file is in the Echelon LNS Utilities program folder on the Windows Start menu. Additional tools for developing resource files are available in the update area at [www.echelon.com/lonmaker](http://www.echelon.com/lonmaker).

The following sections describe the file types.

---

### Type File

The type files (e.g. `STANDARD.TYP`) contain information about network variable types (NVTs), enumeration types, and configuration property types (CPTs).

The LNS DDE Server uses the information in the type files in conjunction with the information in the format files to convert raw data to a scaled and formatted display, and to convert formatted data back to raw data. For example, a time-of-day reading can be converted to hours, minutes, and seconds. One or more sets of user type and format files can be used for conversion of additional, non-standard types. These sets of user type and format files are selected by matching with various parts of the program ID of the device containing the data being accessed by the LNS DDE Server.

For each network variable and configuration property type, the type file supplies the following information:

- The name of the type and the names for any fields within the type. These names are all programmatic names defined by the application developer.
- Indices to language-dependent strings for language-dependent names, comments, and units for the type, and any fields within the type.
- A minimum and maximum for each field in the type. The LNS DDE Server uses this information to check that values supplied by the user are within the valid range.
- For applicable fields, there are scaling factors that convert raw data into formatted data. By default, this transformation is an addition of zero and a multiplication by one. Certain fields, such as floating point numbers and enumeration values, do not use scaling factors.

**Note:**

The valid range for floating point values is  $-3.4E38$  to  $+3.4E38$ .

---

## Format File

The format files (e.g. `STANDARD.FMT`) define how the information contained in a network variable or configuration property should be displayed. The format file uses the information in the a file to obtain the data types and structures, and to convert enumeration types into strings.

The format specification and the network variable or configuration property type are related by the programmatic name key, for example “`SNVT_elapsed_tm`” or “`SCPT_location`”. For integer, floating point, and enumeration types, the formatting is almost always a simple conversion to an ASCII string. For structured types (for example, time-of-day expressed in hours, minutes, and seconds), the data will be formatted as individual fields. Each data type present in the format file must have a corresponding type defined in a user type file, although user type files may have additional types defined.

## Alternate Formats

*Alternate formats* are used to specify multiple formats for a single type. Alternate formats for a type are specified using the # symbol after the type name. For example, `SNVT_temp_f` (temperature in degrees celcius as a floating point number) has two formats, `SNVT_temp_f#SI` for *Système Internationale* units (i.e. Celsius) and `SNVT_temp_f#US` for US units (i.e. Fahrenheit).

Alternate formats are grouped into *alternate format classes*, where an alternate format class defines a set of related alternate formats. Alternate format classes are identified by two-character, upper-case, substrings within an alternate format specifier. Three standard alternate format classes are defined to support locale-specific formatting. The standard alternate formats are the following:

1. **SI** = *Système Internationale* measurement system format (also known as metric). These formats will be automatically selected if the operating system is configured to use *Système Internationale* units.

2. US = United States measurement system format. These formats will be automatically selected if the operating system is configured to use United States units.
3. LO = operating system (e.g. Microsoft Windows) locale-specific format. This will cause the data type to be displayed as configured in the operating system. For example, the operating system could display the date in MM/DD/YYYY format or DD.MM.YYYY format.

The SI and US classes have a special relationship: Any type that has at least one format defined with one of these classes must also have at least one format defined with the other one of these classes. Any format definition that specifies one of these classes cannot specify the other one—they are complimentary. Membership in these classes, together with the Windows system default locale, determines the runtime default format as described under *Default Format*.

Any format definition that contains the list-separator, date, or time locale-specific specifier must be a member of the LO alternate format class. This aids in ensuring backward compatibility and also make it obvious to the user that locale-specific data transformations will occur when using this format.

See the *LonMark Resource File Developers Guide* available on the LonMark website for more information.

---

## Language File

The language file defines all language-dependent strings. The language the file supports determines the extension of the language file. Two language string files are included with the LNS DDE Server, STANDARD.ENU for English (US), and STANDARD.ENG for English (UK). Each string reference from the type file is via an index key into the language file. Each language file uses the same index for a particular string, thus it is simple to switch from one supported language to another. Additional language files will be placed on the LONMARK web site as they are available.

---

## Functional Profile Template File

The functional profile template file defines functional profile templates. A functional profile template is a specification of a framework for a LONMARK object. The functional profile template specifies the required and optional network variable and configuration property components of a LONMARK object.

The actual implementation of a LONMARK object in a device may not contain some of the optional components. It may also contain additional network variable or configuration property components, these latter being referred to as manufacturer specific. Device manufacturers should provide functional profile template files for their user-defined functional profiles, and for their standard functional profiles that use any manufacturer-specific network variables or configuration property types.

The functional profile template file also defines default values, descriptive strings, and validation information for the network variable and configuration property components.





# Appendix B

## Installing the PCLTA-10/20 and PCC-10 Drivers

This appendix explains how to install the PCLTA-10, PCLTA-20, and PCC-10 cards to function as an LNS network interface for the LNS DDE Server.

---

## PCLTA-10, PCLTA-20, and PCC-10 Adapter Installation

The PCLTA-10, PCLTA-20, and PCC-10 installation software installs the files required by the Windows XP, Windows 2000, Windows NT 4.0, Windows Me, or Windows 98 operating systems to recognize the PCLTA-10, PCLTA-20, and PCC-10 adapters. Therefore, installation of the software **must** precede insertion of the adapter into a PC ISA (PCLTA-10), PCI (PCLTA-20), or PCMCIA PC Card (PCC-10) slot.

If you are installing a PCLTA-10 or PCLTA-20, once the software has been installed, you must shut down the PC, install the appropriate hardware as described in *Hardware Installation Procedure*, later in this chapter, and re-start. If you are installing a PCC-10, you can install the hardware without shutting down the PC; however, the computer must be rebooted between the installation of the driver software and the hardware. Then you must load the PCL10VNI (PCLTA-10 and PCLTA-20) or PCC10VNI (PCC-10) application into the hardware and run a diagnostic to ensure it is communicating correctly as described in *PCLTA-10, PCLTA-20, and PCC-10 Configuration Procedure*.

---

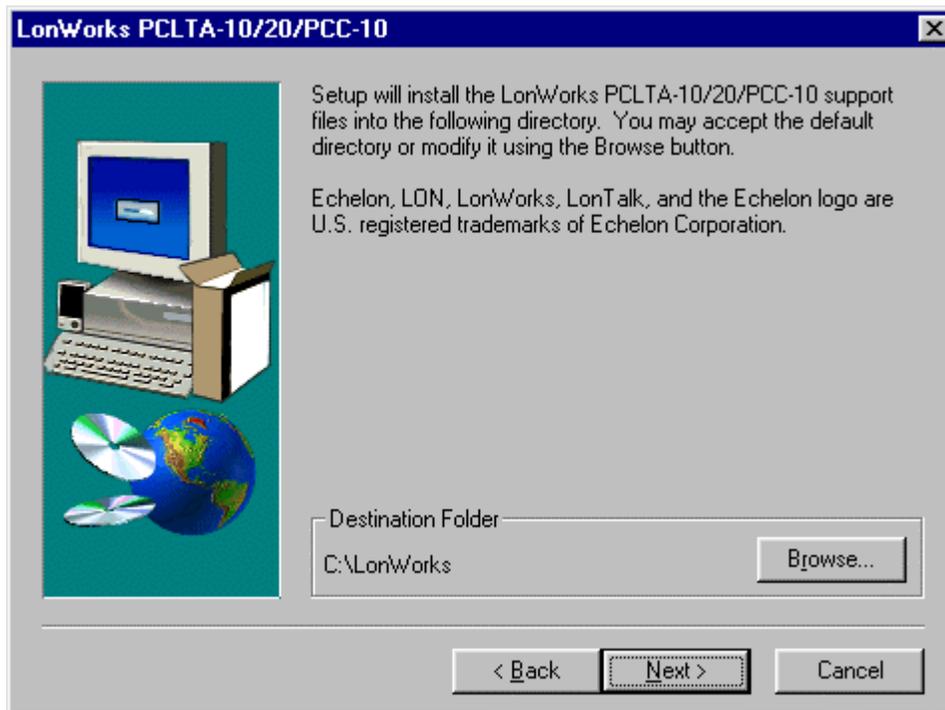
### *PCLTA-10, PCLTA-20, and PCC-10 Software Installation Procedure*

To install the PCLTA-10, PCLTA-20, and PCC-10 software follow these steps:

1. Start the installation procedure as described in *LNS DDE Server* in Chapter 2. Select the PCLTA-10/20/PCC-10 driver in step 6.
2. At the end of the installation procedure, the LNS DDE Server installation program will start the PCLTA-10, PCLTA-20, and PCC-10 driver installation program. The following Installation Language Options dialog opens.



3. Select the language you want to install and click **Next**.
4. A welcome window appears. Read the information on the welcome window and click **Next**. The software licensing agreement window appears.
5. If you agree with the terms of the agreement, click **Yes**. The following Choose Destination Directory dialog appears:



6. Choose the destination folder where you want to install the PCLTA-10, PCLTA-20, and PCC-10 drivers. You must choose the same folder you selected for the LNS DDE Server.
7. The installation program installs the software, then a dialog appears asking you if you want to modify your CONFIG.SYS file. This allows 16-bit Windows Applications to access the network interface. Click Yes or No. Select No if you not using any legacy 16-bit applications. This option is not required to use the LNS DDE Server or any LNS application.
8. A dialog appears asking if you want to install a DOS virtual device driver, which allows you to access the interface driver from DOS applications under Windows. Click Yes or No. Select No if you are not using any legacy DOS applications. This option is not required to use the LNS DDE Server or any LNS application).
9. A dialog appears which prompts you to view the Readme file. Select Yes to view the Readme file, and click Finish.
10. A dialog appears asking if you want to restart your computer. Click No.
11. Go on to *Hardware Installation Procedure*.

---

## Hardware Installation Procedure

The following sections describe the hardware installation procedures for the PCLTA-10 and PCLTA-20 adapters and the PCC-10 card.

---

### *PCLTA-10 Adapter Hardware Installation Procedure*



#### ESD Warning

This product contains devices which are sensitive to static electricity. Before installing or removing the PCLTA-10 adapter or the network cables, discharge any static electricity which may have accumulated to earth ground.

1. Turn off the PC and remove the power cord.
2. Open the PC case and locate an empty 16-bit slot with room for a half-length, half-height ISA-card. Remove the corresponding blank panel from the rear of the PC. Set aside the screw.
3. Insert the PCLTA-10 adapter into the slot, ensuring that the ISA bus connectors are fully mated and the slot in the rear panel mounting lug of the PCLTA-10 adapter is aligned with the threaded hole in the PC chassis.
4. Replace the screw to hold the PCLTA-10 adapter firmly in place.

5. Reinsert the power cord and then restart the PC. A *New Hardware Found* window will be displayed briefly when the operating system recognizes the PCLTA-10 adapter.

---

## *PCLTA-20 Adapter Hardware Installation Procedure*



### ESD Warning

This product contains devices which are sensitive to static electricity. Before installing or removing the PCLTA-20 adapter or the network cables, discharge any static electricity which may have accumulated to earth ground.

1. Turn off the PC and remove the power cord.
2. Open the PC case and locate an empty PCI slot. Remove the corresponding blank panel from the rear of the PC. Set aside the screw.
3. Insert the PCLTA-20 adapter into the slot, ensuring that the edge connectors are fully mated and the slot in the rear panel mounting lug of the PCLTA-20 adapter is aligned with the threaded hole in the PC chassis.
4. Replace the screw to hold the PCLTA-20 adapter firmly in place.
5. Reinsert the power cord and then restart the PC. A *New Hardware Found* window will be displayed briefly when the operating system recognizes the PCLTA-20 adapter.

---

## *PCC-10 Card Hardware Installation*

The PCC-10 card conforms to the Personal Computer Memory Card International Association's (PCMCIA) standard for hot plug-in. The PCC-10 card will not be harmed if it is inserted into, or removed from, a PC Card (PCMCIA) slot which conforms to this standard, whether the computer is on or off. In addition, the PCC-10 card is recognized as a UL (Underwriter's Laboratories) Listed Accessory and is designed to be used with UL Listed equipment.

Do not force the insertion of a PCC-10 card. The PCC-10 card can only be inserted one way into a PC Card slot. The keyed notches prevent the card from being inserted upside down. In the Windows 98 or Windows 95 environment, insertion of the PCC-10 card may cause the operating system to produce two brief tones: a low tone followed by a higher tone. Extracting the card will produce the tones in the reverse order: high, then low. If a device's property window is open in the System Control Panel, the tones will be produced after the window is closed. This confirms that the device is inserted correctly. Additionally, a PC Card icon may also appear in the status area to the right of the Windows taskbar.

If the computer was rebooted after installation of the software, insert the PCC-10 card into an open PC Card slot. Otherwise, reboot the computer before insertion.

The device driver for the PCC-10 card is not loaded until the first PCC-10 card is discovered. Likewise, when the last PCC-10 card is removed, the device driver is unloaded, thus freeing any system resources it was using.

Each PCC-10 card requires a single, dedicated interrupt request (IRQ) and four contiguous bytes of I/O address space starting on a modulo-4-based address.

Removal of a PCC-10 card while an application is using the card will result in a loss of communication with the device, which cannot be restored by re-inserting the card. Some applications will display unusual behavior, and will not properly function. Any application using the PCC-10 card must be restarted if a PCC-10 card has been removed to ensure proper operation of the device and software.

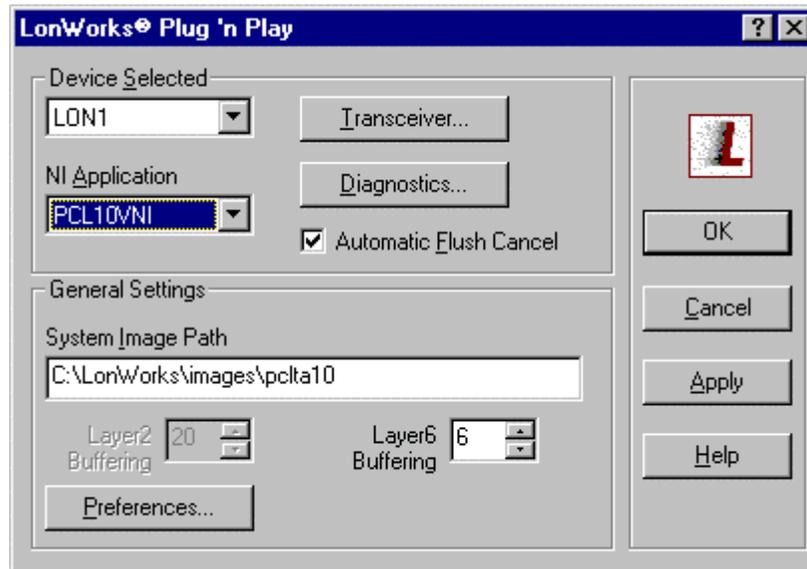
The first time a PCC-10 card is inserted into a running PC, a window will appear with the words "Echelon Corp.-PCC-10." Another window will appear stating that the Windows operating system is building a new database from the device information installed by the PCC-10 installation diskette. The PCC-10 can be configured when the PC has finished writing the device information.

---

## PCLTA-10, PCLTA-20, and PCC-10 Diagnostic Procedure

Once the software and hardware have been installed, you should verify that the adapter or card is operating correctly. To accomplish this, follow these steps:

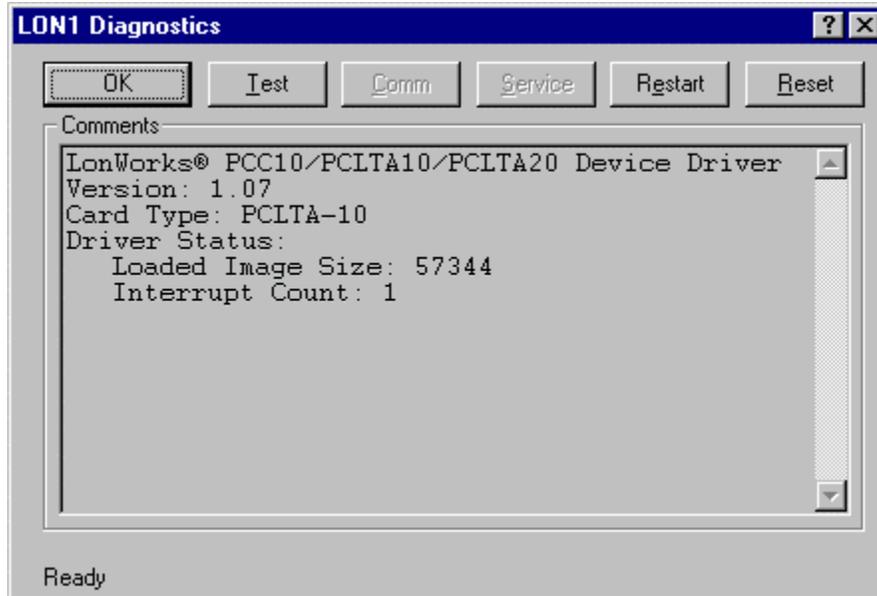
1. Open the *Start* menu, point to settings, then select the Control Panel.
2. Open the *LONWORKS Plug 'n Play* application on the control panel. The following window opens:



3. Ensure that the LONWORKS folder in the System Image Path matches the one used during software installation.
4. If you installed a PCLTA-10 or PCLTA-20 card, verify that PCL10VNI is selected in the NI Application list. If you installed a PCC-10 card, verify that

PCC10L7 is selected in the NI Application list. Click **Apply** whether or not you change the value.

5. Click the **Diagnostics** button. If the PCLTA-10/20 and PCC-10 software and hardware installation has proceeded correctly, the following dialog appears:



6. Refer to the PCLTA-10, PCLTA-20, or PCC-10 online help in the Echelon Drivers program folder if this dialog does not appear.
7. Click the **Test** button. The following statistics will appear if the card is functioning correctly (some of your values may be different):

```
-- Network Interface Status
--

CRC Errors: [00000]
TX Timeouts:[00000]
Lost (APP) Messages: [00000]
Missed (NET) Messages:
  [00000]
Node State: Unconfigured
Most recent error: 0
Reset Cause: External
```

If you receive an error message or your system hangs after clicking the **Test** button, uninstall the driver, reboot your PC, re-install the software, and try running this diagnostic again.

8. If the test passes, you know that the PC can communicate with the network interface card. The next step is to be sure that the interface card can communicate with a test device. Click the **Comm** button to display the following :

This procedure will configure the Network Interface for a zero-length domain. Do you want to proceed?

9. Click the OK button. The control panel enters a receive-ready state and displays the following message while waiting for a service pin message from another device on the network:

Now waiting for a service pin message.

10. Press the service pin on any device that can communicate with the network interface card. The control panel sends a diagnostic message to the device originating the service pin message using Neuron ID addressing. It will repeat this operation, referred to as a “ping,” once a second until you click either the OK or the Quit button (the Comm button becomes the Quit button). The pinging behavior should look something like this (your Neuron ID and Program ID may be different):

```
Received service pin, pinging node:
```

```
Neuron ID: 01 23 45 67 89 10
```

```
Program ID: A NODE
```

```
** Ping Passed **
```

```
** Ping Passed **
```

```
** Ping Passed **
```

```
...
```

This series of tests confirms that the network interface card is properly configured and can communicate with a device on the network.

11. Click OK to close the dialog, and click OK again to close the LONWORKS Plug N' Play application.

# Appendix C

## Wonderware InTouch Client Example

Wonderware's InTouch HMI development tool is one of the HMI development tools that supports the LNS DDE Server. The example described in this appendix is provided as a starting point to understand the basics about how to monitor and control various kinds of LONWORKS data using the InTouch software with the LNS DDE Server.

---

## Introduction

The InTouch example provides an overview of how to use the InTouch software with the LNS DDE Server. See *Using the InTouch Example* in Chapter 2 for a description of the requirements for using the example, and for installation instructions.

The following section provides a brief description of InTouch. You can skip this section if you are an experienced InTouch user.

---

## Intouch Overview

Wonderware's InTouch 7.0 is a Windows PC based supervisory HMI development tool included with Wonderware's Factory Suite 2000. InTouch enables rapid development of graphical monitoring and control applications that interact with a versatile offering of DDE based I/O servers. InTouch uses three integrated applications that perform the following functions.

- *InTouch Application Manager* - Is used to create, manage, and select HMI application projects.
- *InTouch WindowMaker* - Is the development environment used to create HMI applications.
- *InTouch WindowViewer* - Is the runtime environment for applications created in WindowMaker.

Normally, the HMI application developer runs the InTouch Application Manager to setup the framework for the application projects, and then spends their time in the WindowMaker and WindowViewer applications as the custom HMI is developed in iterative steps.

In InTouch, the tag dictionary is central to the operation of the application. The tag dictionary is a high performance database that is used to connect application data to graphic objects using animation links. The graphic objects provide the user's view to a system. There are four basic tag data types:

### **Discrete**

Used to represent 2-state conditions of either 0 (False, Off) or 1 (True, On).

### **Integer**

A 32-bit signed integer value between -2,147,483,648 and 2,147,483,647.

### **Real**

A floating (decimal) point value between  $-3.4e38$  and  $3.4e38$ . All floating point calculations are performed with 64-bit resolution, but the result is stored as a 32-bit value.

### **Message**

A text string that can be up to 131 characters

InTouch tags fall into three main category types. Memory tags are primarily data variables local to the PC running the InTouch WindowViewer application. I/O tags act as ports that move data to and from I/O servers and the running WindowViewer application. The LNS DDE server can be an I/O server for InTouch. Indirect tags are used to perform pointer-like operations to re-map tag data sources.

Using InTouch, the developer has the option of creating the HMI visualization elements from simple shapes, shapes grouped as cells that have one or more animation links, or using Wizards which are predefined widgets that speed development work but limit the object's actions to a set of predefined animation links. An animation link connects the value or a field of a tag to a specific behavior on the screen.

InTouch also provides a number of different script types. A script is a sequence of function calls, expressions, and simple control flow constructs that approximate a programming language procedure. Script types differ by the events which trigger the script's execution. For example, there are scripts associated with window opening, closing, and periodic execution while the window is open. Data change scripts are triggered when a tag value is changed. A condition script allows the developer to cause a script to run based on a defined condition such as a tag value exceeding a certain threshold level. The scripting features allow the HMI developer to implement part of the control strategy on the PC.

InTouch uses additional attributes of tags to connect to alarming, event logging, historical trending, and other key subsystems required in a full-featured HMI application.

---

## InTouch Example Tutorial

This section describes step by step instructions to navigate the example InTouch application.

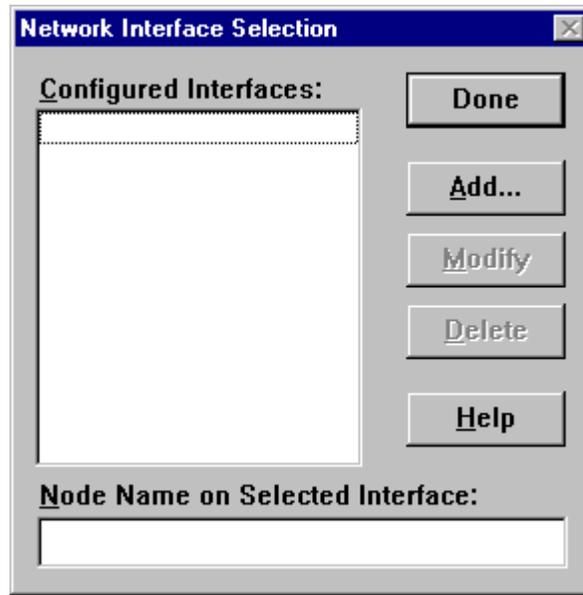
---

### *InTouch Animation and I/O Tags*

Objects in an InTouch HMI display are selected from a toolbar and placed on a window form. They come to life at runtime with animation links that bind the object's runtime behavior to a tag in the tag dictionary. To see how an object's animation behavior is defined and a single I/O tag is set up, follow these steps:

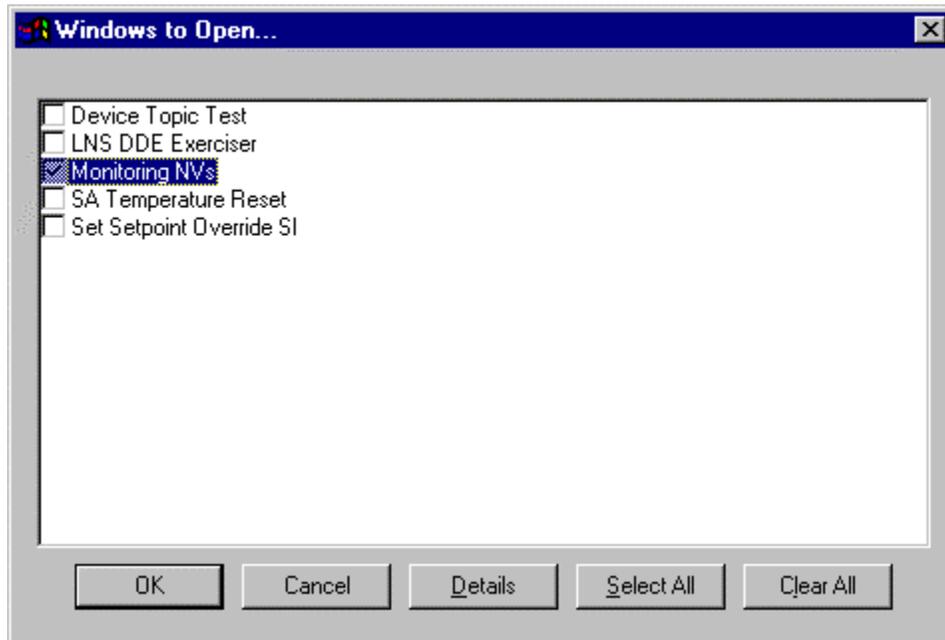
1. Open the InTouch example by selecting InTouch WindowMaker in the Example Applications folder, as described under *Using the InTouch Example* in Chapter 2.
2. If you are using a display resolution other than 1024x768, a dialog opens asking if you want to convert the example to your current display resolution. Click **Yes** if this dialog is displayed.

3. The following dialog opens:

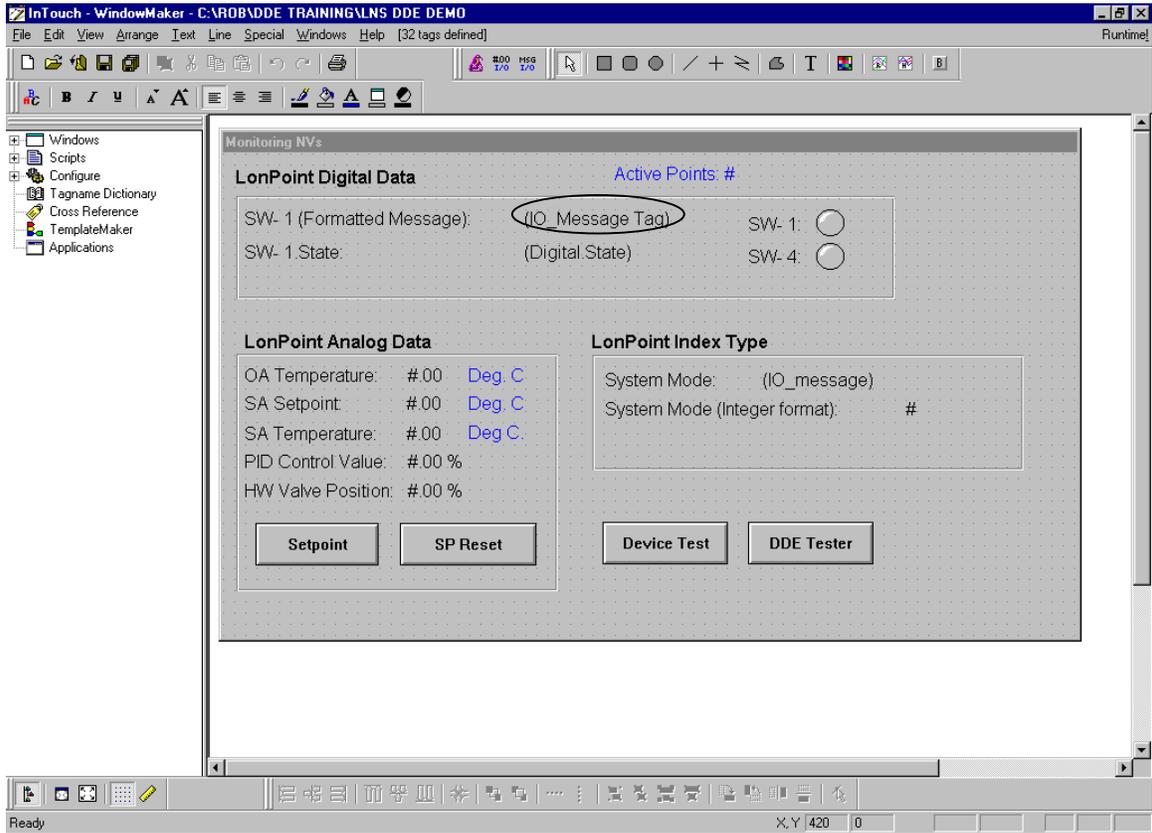


This dialog is associated with setting up NetDDE which is not used in this example.

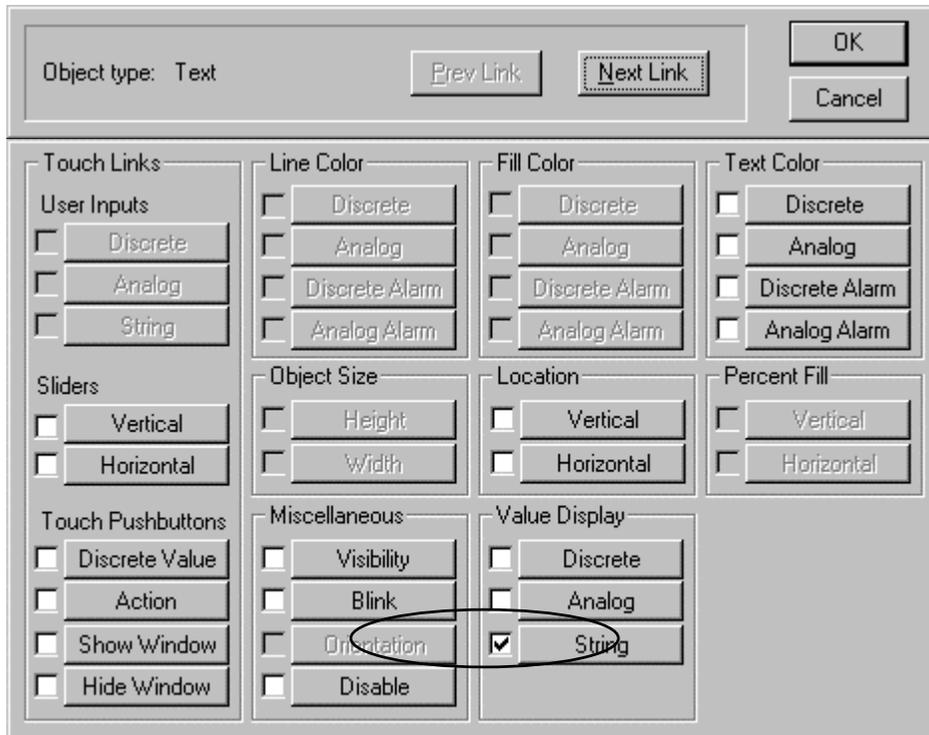
4. Click the Done button without modifying any fields in this dialog. The following dialog opens:



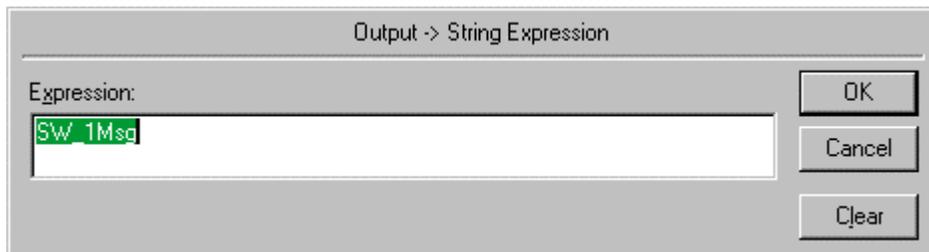
5. Select the Monitoring NVs window. Click OK. InTouch starts the WindowMaker application with the Monitor NVs window open for modification. The following dialog opens:



6. Double-click the IO\_Message tag label to show the animation link assigned to this object. The following dialog opens:



7. Select the String option to animate the label with a string value.
8. Click the String button of the Animation Link Dialog to see the expression that defines source of the string value when WindowViewer is running as shown in the following figure:



- Double-click the tag name in the `Expression:` edit box to see how this tag is defined. The following Tagname Dictionary dialog opens:

Click the `Type` button to the right of the `Tagname` to see the first key piece of information that is required to define the tag. In this case the tag is defined as an `I/O Message`. Once a tag is assigned a type (`Discrete`, `Integer`, `Real`, or `Message`), it will remain restricted to these foundation types. The first time a tag is defined, you can select from all the tag types that are supported by InTouch.

- The second key piece of information that is needed to connect an `I/O` tag to the `LNS DDE Server` is the `Access Name`. InTouch uses `Access Names` to define the `Application` and `Topic` portions of a `DDE link`. In this case we have defined the `Access Name` as `LNS5LMNV`. Click the `Access Name` button to show the following dialog:

These names were defined using the following convention: `LNS#<Topic Type>`. `LNS` is a shortened version of the application name, the `#` is the

handle for the Subsystem containing the targeted function block or device, and <Topic Type> is the topic type supported by the LNS DDE server.

11. Select LNS5LMNV and click the Modify button to review the definition of the Access Name using the following dialog:

The dialog box titled "Modify Access Name" contains the following fields and options:

- Access Name: LNS5LMNV
- Node Name: (empty)
- Application Name: LNSDDE
- Topic Name: InTouch DDE.HVAC.LMNV
- Which protocol to use:  DDE,  SuiteLink
- When to advise server:  Advise all items,  Advise only active items
- Buttons: OK, Cancel

In this dialog you define the Application and Topic portions of the DDE link. The LNS DDE Server only supports DDE as a protocol. The default, 'Advise only active items' option is selected. This option allows you to limit the polling of the application to the points currently monitored by active windows on screen and to points being logged or alarmed by InTouch. InTouch is very efficient in the way it manages DDE links. Once a point is created, InTouch does not destroy the link when the point is no longer in view. Instead, it deactivates the point causing polling to stop while leaving the data structures in place so the next access to the point will occur quickly. Other HMI applications may actually destroy the link when a point is no longer in focus (this is a key benefit of InTouch).

Close the dialogs just opened using the Cancel buttons to avoid any changes.

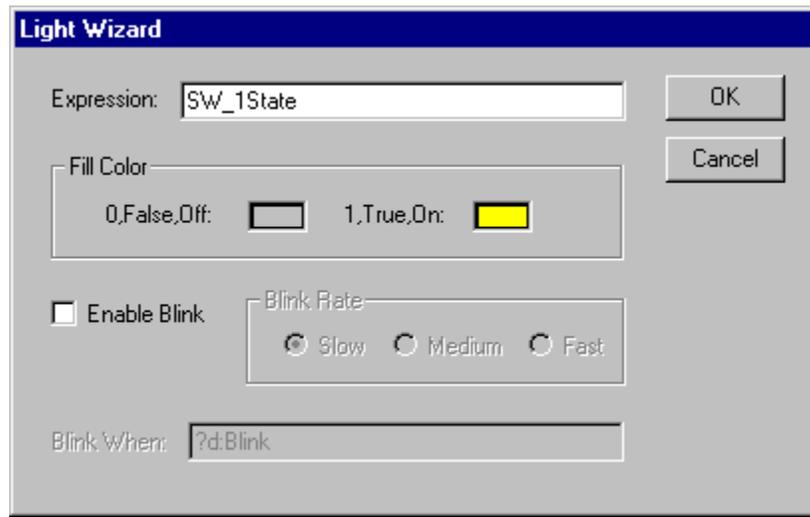
---

## Network Variable Monitoring

The Monitoring NVs window is the main window of the example application. It shows simple animation examples for monitoring the three most common network variable types used by the LonPoint system (and many other LONWORKS applications). The top frame shows digital data presented using three different techniques. The last section described how the formatted message view of a variable using the SNVT\_switch type is defined. The animated LED is a more visually appealing mechanism to present this data (see figure 9). In this case, a tag that looks at the state portion of the variable is defined.

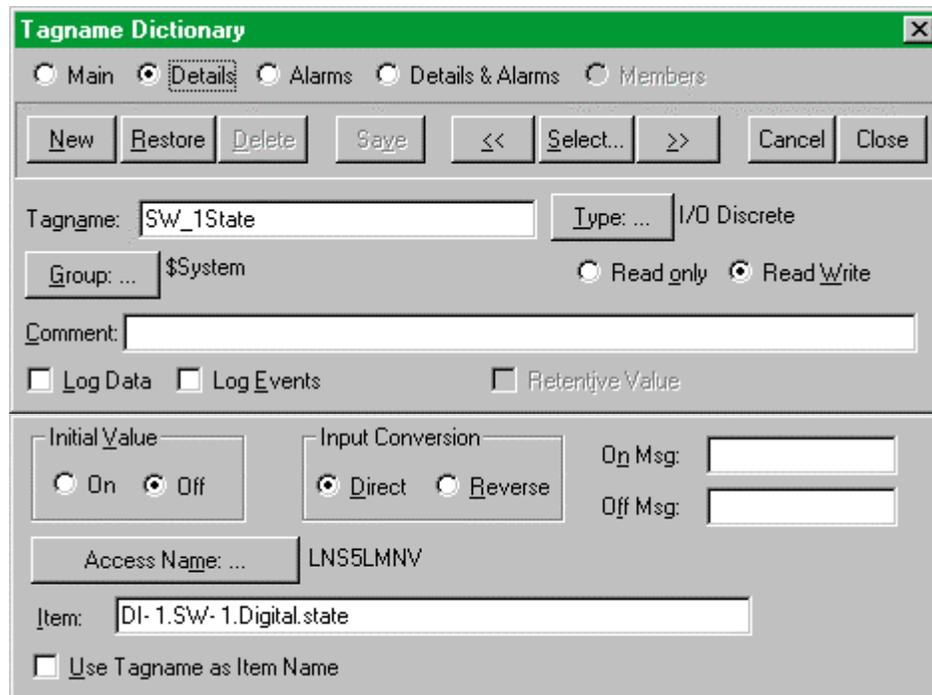
To see how an LED is configured in InTouch, follow these steps:

1. From the WindowMaker tool, double-click the LED to the right of SW- 1. The following dialog opens:



This object is an InTouch Light Wizard. Wizards are higher level objects that support only a predefined set of animation links. As seen in the figure 10, this object supports a discrete expression that is driven by the value of the SW\_1State tag.

2. Double-click the tag name in the Expression field to see the definition of this tag as shown in the following figure:



The item definition has an additional component beyond the network variable name that is the 'state' field of the SNVT\_switch typed variable.

---

## Changing the Default Format

Two views of the mode output of the mode simulator value are shown under LonPoint Index Type in the main window of the example (see step 5 under *InTouch Animation and I/O Tags*). The first form of the value uses the default format where the mode is reported as a string spelling out the enumerated value. The InTouch tag is declared as an I/O Message type, and the item description of the I/O tag is declared as follows:

```
DI-1.Mode Simulator.Mode_Out
```

The second form of the value uses an integer format where the mode is reported as an integer. The InTouch tag is declared as an I/O Integer, and the item description of the tag includes an additional flag to modify the format used to present the data as follows:

```
DI-1.Mode Simulator.Mode_Out -tInteger
```

The integer value representation is more appropriate if the HMI application needs to use the mode value to control the execution of a script.

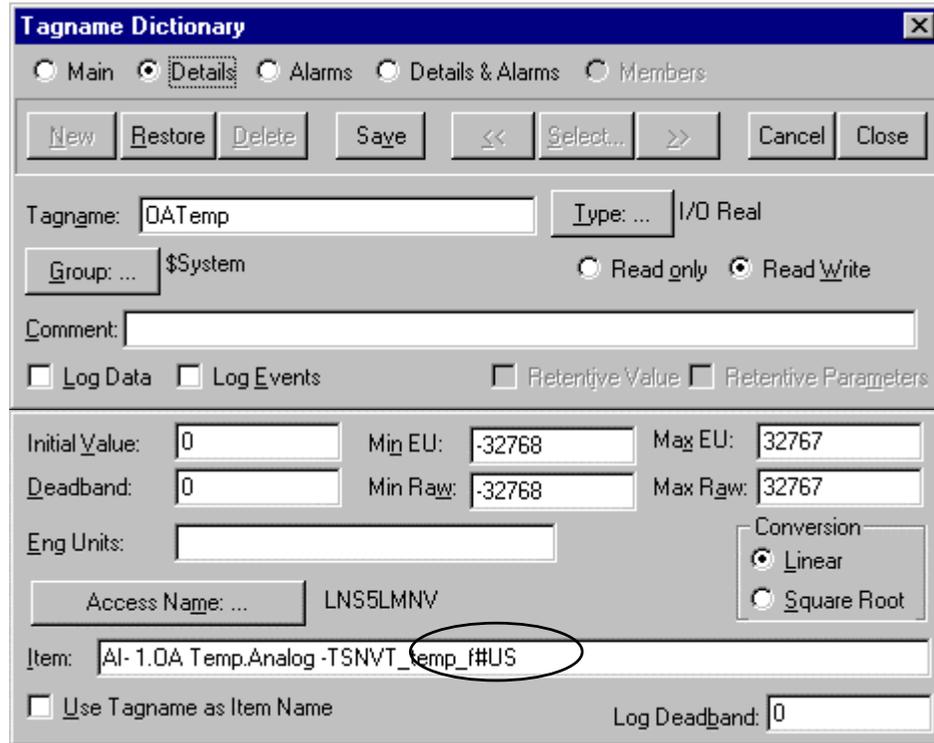
---

## Display Unit Conversion

The fields under LonPoint Analog Data in the Monitoring NVs window display points associated with the example control loop. These are temperature values that may be displayed as Celsius or Fahrenheit values. The default format is selected based on your Windows locale settings. To force the US or SI format to be used when the other format is the default, follow these steps:

1. From WindowMaker, double-click the '#.00' string to the right of the OA Temperature label.
2. Click on the Analog, Value Display animation link to determine the tag used for the animation link.
3. Double-click the tag name in the Expression field to open the tag edit dialog.

4. Modify the item to add the `-tSNVT_temp_f#US` option as shown in the following figure:



Substitute #SI for #US to force SI units.

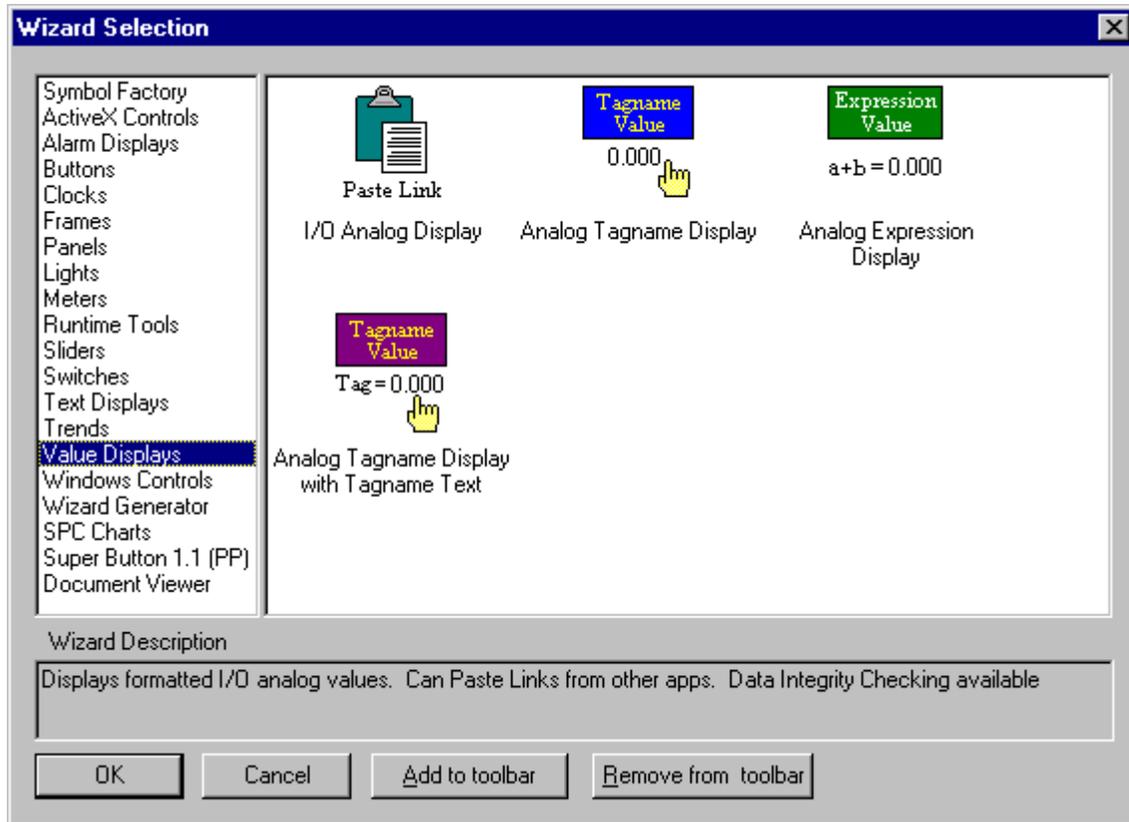
5. Save changes and close the open dialogs.
6. Click Runtime! to review the results.

---

## Using the InTouch DDE Link Wizards

The LNS DDE server supports copy/paste DDE link operations to speed the definition of I/O tags targeting the LNS DDE server. InTouch includes two wizards to support DDE copy/paste link operations. One is the I/O Analog display wizard and is suitable for displaying any integer or floating point value from the LNS DDE Server. The other is the I/O Message Display wizard and is suitable for displaying any value from the LNS DDE Server formatted as a text string. The following steps describe how to take advantage of this feature when defining I/O tags.

1. Click the Wizard icon in the WindowMaker tool bar (this icon is a small conical wizard's hat on the far left of the main drawing tool bar) to open the Wizard Selection dialog.
2. To start the I/O Analog Display wizard, select the Value Displays category on the left as shown in the following figure:



To open the I/O Message Display wizard, select the Text Displays category on the left.

3. Double-click the I/O Analog Display or I/O Message Display wizard. Click once at the desired position in the Monitor NV window. This will show a text object with a string '0.00' for the I/O Analog Display wizard or an 'I/O message' string for the I/O Message Display wizard. Double-click this object to see the following dialog for the I/O Analog Display wizard (or a similar dialog for the I/O Message Display Wizard):

**I/O Analog Value Wizard**

Application:

Topic:

Item:

Protocol:

OK

Cancel

Paste Link

If possible, this wizard will use an existing I/O Real Tag.  
If a Tag does not exist, however, one will be created:

When creating a tag, Use Item as Tagname     Allow Input

Tagname:

Value Display Format

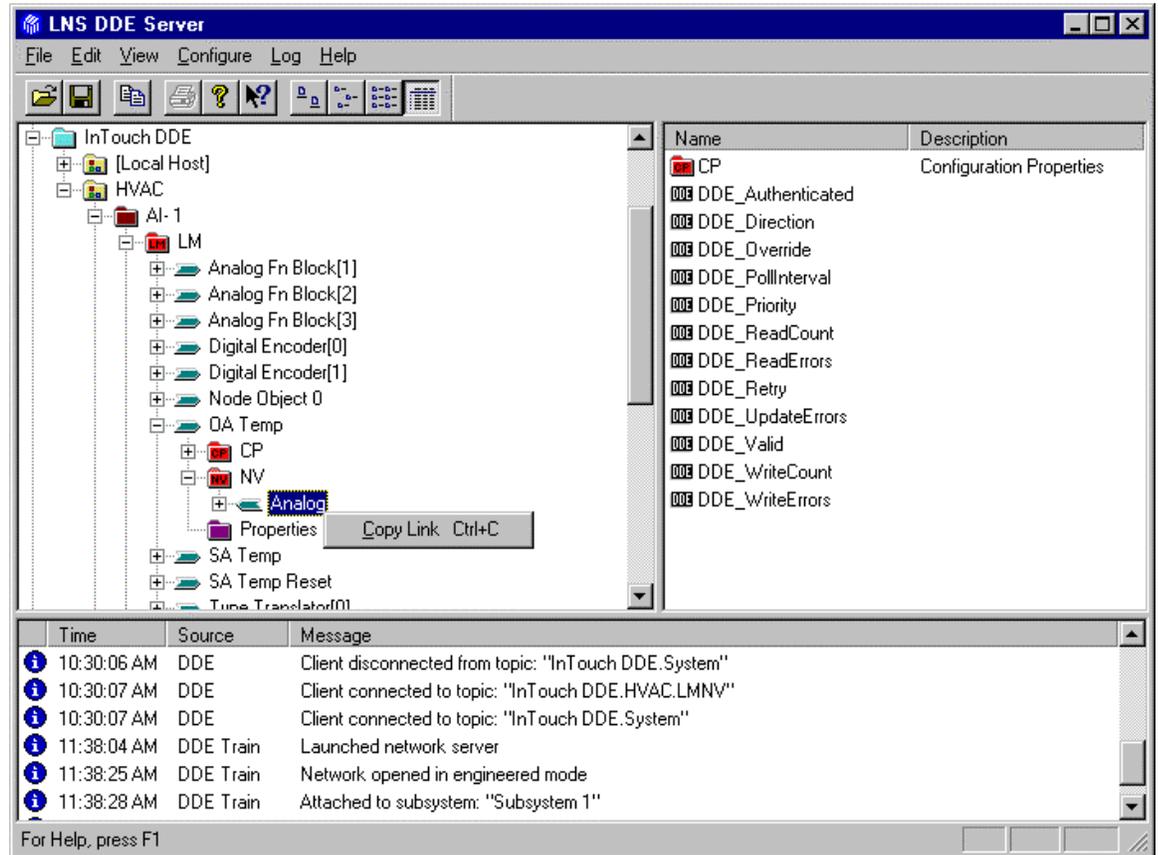
Number Format:

Data Integrity Checking:  Options...

Eng Units:

4. Click the LNS DDE Server button on the Windows taskbar to open the LNS DDE Server Window

5. Navigate to the OA Temp device Analog output value, right-click the Analog output, and select Copy Link from the shortcut menu as shown in the following figure:



6. Click the InTouch button on the Windows taskbar to return to the wizard.
7. Click the Paste Link button. This will automatically fill in the Application, Topic, and Item portions of the link definition. The wizard defines a default tag name. This field requires modification because InTouch tag names cannot support embedded spaces or period characters. Select the Allow Input option, and change the name by removing the invalid characters or entering a new tag name. If a suitable access name already exists, the wizard will use this access name for the tag definition.

Using the procedure just described prevents the tedious error prone exercise of completely specifying the item declarations resulting in a significant reduction in the time required to generate the tag definitions.

## Other Topic Types

Now that you are acquainted with the basic operation of InTouch and access to the tag dictionary, explore the remaining windows defined for the example HMI application to see access to configuration property data, LONMARK Object control, and Device object LNS DDE topic types. Table 1 provides a summary of what you will learn from each window. Consult InTouch help to understand how the scripting language works. Much of the interesting work

is contained in Action animation links attached to buttons placed in the window.

**Descriptions of other windows in the example HMI application.**

<b>Application Window</b>	<b>Point of Interest</b>
Device Topic Test	From this window you can issue test commands, monitor several device topic items, and cause Wink commands to be issued to the devices. This window shows control of a tag item definition using action scripts and the <code>IOSetItem()</code> function in <code>InTouch</code> . Accesses and control of DDE items targeting the <code>LMDEV</code> topic type are demonstrated in this window.
SA Temperature Reset	This window shows interaction with the <code>LMCP</code> topic type. The user input fields in this window show how to write to specific fields of a configuration property associated with a <code>LonPoint</code> analog functional block.
Setpoint Override	This window shows control of an item that uses the <code>LMNVCP</code> topic type, as well as control of the <code>LM</code> topic item to issue and cancel the <code>Override</code> state of a functional block.
LNS DDE Exerciser	This window allows the user to define DDE conversations, and to poke data to control various items. Action scripts behind the <code>Set Item</code> , and <code>Poke Value</code> reuse the I/O message tags using the <code>IOSetItem()</code> function call.



# Appendix D

## License Order Form

This appendix provides a printed copy of the order form for LNS DDE Server credits. See Chapter 3 for information on using and adding credits, and instructions for using this order form. The order form in this appendix is only for use when you cannot automatically generate an email or printed form as described in Chapter 3.

**Note:** Your version of the LNS DDE Server may use an alternate order form, or your LNS DDE Server license administrator may provide an alternate form.

TO: Echelon License Administrator  
Email: credits@echelon.com  
Fax: +1-408-328-3802 (please use email  
if available)  
(the fax number may change July 2001)

FROM: Name:  
Company:  
Phone:  
Fax:  
Email:

SUBJECT: Order for LNS(tm) DDE Server Application Key (Form  
2.1)

Please process the following order:

Quantity: 1  
Product: LNS DDE Server Application Key or  
Upgrade  
Model Number: [ ] 37200 Standard Key  
[ ] 37200-10 Prior Version Upgrade  
(See Note 1)  
[ ] 37200-UPGRADE LonManager(tm) Upgrade  
Key;  
LonManager DDE Server  
Serial Number must be  
entered here:  
[ ] 37200-PREPAID Prepaid Key; Identify  
Product with Prepaid Key  
below (see Note 2)  
[ ] Other

Price:

Note 1: The Model 37200-10 Prior Version Upgrade Key is  
only required for LNS DDE Server 2.0 installations that were  
never upgraded to Release 2.01. An upgrade key is not  
required for licensed Release 2.01 installations. The  
upgrade key will only work with copies of the LNS DDE Server  
that have been previously authorized with one of the other  
keys.

Note 2: A purchase order for Model 37200 is NOT considered  
a pre-paid key; if you have previously submitted a purchase  
order, check the 37200 option and enter the purchase order  
number below.

Following are the list prices. These prices are subject to  
change. Contact your distributor for current and local  
pricing. Your distributor may also provide you with an  
alternate mailing address for this order.

Model 37200	Standard Key	US\$795
Model 37200-10	Prior Version Upgrade	US\$320
Model 37200-UPGRADE	LonManager Upgrade Key	US\$395
Model 37200-PREPAID	Prepaid Key	No charge
Replacement for lost key		No charge
Backup key		No charge

The following serial number and PC key values are filled in automatically by the LNS DDE Server when you generate an application key order as described in the LNS DDE Server User's Guide. You should always use the order form generated by the software and send the form by email (or fax if email is not available) to ensure that there are no errors in transcribing the serial number and key values. When you use the software to generate the order form, you will have 30 days of unrestricted use of the software. This will give you adequate time to send in the order and to receive and install your application key.

LNS DDE Server Serial Number:

PC Key:

Select one of the following payment methods and fill in the information for the requested method:

- Credit Card
  - Type of credit card:
    - American Express
    - MasterCard
    - Visa
  - Name on the credit card:
  - Credit card number:
  - Credit card expiration date:
  - Credit card billing address:
    - Street 1:
    - Street 2:
    - City:
    - Providence/State:
    - Postal/Zip Code:
    - Country:
- Purchase Order
  - Purchase Order Number:
  - Authorized by:
  - Billing Address:
    - Street 1:
    - Street 2:
    - City:
    - Providence/State:
    - Postal/Zip Code:
    - Country:
- Pre-paid application key
  - Product ordered with pre-paid key:
  - Date that product was ordered:
  - Serial number:
- No charge replacement key. By submitting an order for a replacement key, you agree that you will use the key solely to replace an LNS DDE Server Application Key that was lost due to a PC hardware or software failure. To transfer an application key to a different PC, see the Transferring an Application Key topic in the LNS DDE Server help file. To order a replacement key, complete the following description:

Description of how the original application key was lost:

- [ ] No charge backup key. By submitting an order for a backup key, you agree that you will use the key solely to replace an LNS DDE Server Application Key that was lost due to a bitwise copy of a disk image with the LNS DDE Server software to a backup disk or PC. Limited to a single backup PC per licensed copy of the LNS DDE Server software. The LNS DDE Server cannot be operational on the backup PC at the same time it is operational on the primary PC.

Your application key will be sent to you by email or fax.

To receive a free manual with a new or upgrade key order, please complete the following shipping address (free manuals are not shipped for pre-paid or replacement key orders):

Attention:  
Shipping Address:  
    Street 1:  
    Street 2:  
    City:  
    Providence/State:  
    Postal/Zip Code:  
    Country:

# Appendix E

## LNS DDE Server Software License Agreement

This appendix provides a printed copy of the LNS DDE Server Software License Agreement.

---

## Notice

This is a legal agreement between you and Echelon Corporation ("Echelon"). YOU MUST READ AND AGREE TO THE TERMS OF THIS SOFTWARE LICENSE AGREEMENT BEFORE ANY SOFTWARE CAN BE DOWNLOADED OR INSTALLED OR USED. BY CLICKING ON THE "I AGREE" OR "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 CONDITIONS OF THIS SOFTWARE LICENSE AGREEMENT. IF YOU DO NOT AGREE WITH THE TERMS AND CONDITIONS OF THIS SOFTWARE LICENSE AGREEMENT, THEN YOU SHOULD EXIT THIS PAGE AND 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. INSTEAD, YOU MAY RETURN THE LICENSED SOFTWARE TO THE PLACE YOU OBTAINED IT FOR A FULL REFUND (IF APPLICABLE).

---

## LNS DDE Server Software License Agreement

Echelon Corporation ("Echelon") grants you a non-exclusive, non-transferable license to use the copy of the Licensed Software and accompanying documentation and any updates or upgrades thereto provided by Echelon according to the terms set forth below. As used herein, "Licensed Software" refers to the LNS DDE Server computer software and associated media, printed materials, and "online" or electronic documentation, including without limitation any and all executable files, tutorials, help files and other files, that accompany such software or are in the accompanying documentation; "Demonstration Mode" refers to a restricted mode of the LNS DDE Server where it will not operate for more than one hour at a time; "Temporary Mode" refers to an unrestricted mode of the LNS DDE Server that is limited to 30 days of use on a computer from the time the Licensed Software is first installed on the computer, or the first time an Application Key is ordered for the computer; "Unlimited Mode" refers to an unrestricted mode of the LNS DDE Server that is not limited to 30 days; and "Application Key" refers to a software key provided by Echelon that enables Unlimited Mode for a copy of the LNS DDE Server on a particular computer.

---

## License

**You may:**

- (a) use the Licensed Software on any number of computers in Demonstration Mode or Trial Mode,
- (b) use the Licensed Software on any number of computers in Unlimited Mode, provided that you have purchased and installed an Application Key for each computer to be operated in Unlimited Mode,
- (c) access the Licensed Software over a network using NetDDE or any other networking protocol, provided that you have purchased and installed a separate

Application Key for the Licensed Software for the server computer and every remote client computer,

(d) physically transfer an Application Key from one computer to another, provided that the Application Key is removed from the computer on which it was previously used and the Application Key is used on only one computer at a time,

(e) copy the licensed software as necessary for the uses expressly permitted above,

(f) transfer your rights under this Agreement to an end user of the Licensed Software; provided that (i) you require the transferee to execute both copies of the Software License Transfer Agreement included in this package; and (ii) you retain one (1) signed original thereof and furnish Echelon with a copy of same upon request. This right of transfer is exercisable on a one-time-only basis, and your transferee shall have no right whatsoever to further transfer any rights to the Licensed Software.

**You may not:**

(a) use an Application Key on more than one computer at a time,

(b) access the Licensed Software over a network using NetDDE or any other networking protocol (except as expressly permitted above),

(c) copy the Licensed Software (except as expressly permitted above), or copy the accompanying documentation,

(d) 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

(e) distribute, rent, transfer, or grant any rights in the Licensed Software or modifications thereof or accompanying documentation (except as expressly permitted above) 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, Application Key, accompanying documentation, and any copy made by you remain with Echelon. Unauthorized copying of the Licensed Software, Application Key, or the accompanying documentation, or failure to comply with the above restrictions, will result in automatic termination of this license and will make available to Echelon other legal remedies. You may make appropriate and truthful reference to Echelon and Echelon products and technology in your company and product literature; provided that you properly attribute Echelon's trademarks. No license is granted, express or implied, under any Echelon trademarks, trade names, or service marks.

---

## LIMITED WARRANTY AND DISCLAIMER

Echelon warrants that, for a period of ninety (90) days from the date of delivery or transmission and installation of an Application Key to you, the Licensed Software under normal use will perform substantially in accordance with the Licensed Software specifications contained in the documentation accompanying the Licensed Software. 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.

SOME STATES OR OTHER JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSIONS MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER RIGHTS THAT VARY FROM STATE TO STATE AND JURISDICTION TO JURISDICTION.

---

## LIMITATION OF LIABILITY

IN NO EVENT WILL ECHELON OR ITS SUPPLIERS BE LIABLE FOR LOSS OF DATA, LOST PROFITS, COST OF COVER OR OTHER SPECIAL, INCIDENTAL, PUNITIVE, CONSEQUENTIAL OR INDIRECT DAMAGES ARISING FROM THE USE OF THE LICENSED SOFTWARE OR ACCOMPANYING DOCUMENTATION, 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. IN NO EVENT SHALL ECHELON'S OR ITS SUPPLIERS' LIABILITY EXCEED THE AMOUNTS PAID FOR THE NETWORK INTERFACE. YOU ACKNOWLEDGE THAT THE AMOUNTS PAID BY YOU FOR THE NETWORK INTERFACE REFLECT THIS ALLOCATION OF RISK.

SOME STATES OR OTHER JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATIONS AND EXCLUSIONS MAY NOT APPLY TO YOU.

---

## SAFE OPERATION

YOU ASSUME RESPONSIBILITY FOR, AND HEREBY AGREE TO USE YOUR BEST EFFORTS IN, DESIGNING AND MANUFACTURING EQUIPMENT LICENSED HEREUNDER 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 NEURON CHIP, LONTALK PROTOCOL, NEURON CHIP FIRMWARE, LONWORKS NETWORK INTERFACES, 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 NEURON CHIP, LONTALK PROTOCOL, NEURON CHIP FIRMWARE, LONWORKS NETWORK INTERFACES, 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.

---

## 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.

---

## 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 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).

Echelon and LONWORKS are U.S. registered trademarks of Echelon Corporation.



# Appendix F

## **LNS DDE Server Software License Transfer Agreement**

This appendix provides a printed copy of the Software License Transfer Agreement.

---

# Software License Transfer Agreement

THIS SOFTWARE LICENSE TRANSFER AGREEMENT is entered into as of \_\_\_\_\_ [insert date], by and between \_\_\_\_\_ [insert company name of original purchaser] ("Assignor") and \_\_\_\_\_ [insert name of company purchasing the software] ("Assignee") for the \_\_\_\_\_ [insert name of software product being transferred] ("Software").

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 Restriction. Assignee acknowledges that Assignee shall have no right whatsoever to transfer, sell, or otherwise grant any rights to the Software or to assign any of its rights or obligations hereunder or under the Software License Agreement, and any such purported assignment or transfer shall be void.
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": \_\_\_\_\_  
Company Name  
Signed By: \_\_\_\_\_  
\_\_\_\_\_  
(Print Name)  
Title: \_\_\_\_\_  
Address: \_\_\_\_\_  
City,State/Province, Country, Postal Code  
Telephone: \_\_\_\_\_  
(Fax or email address)  
Date: \_\_\_\_\_

"ASSIGNEE": \_\_\_\_\_  
Company Name  
Signed By: \_\_\_\_\_  
\_\_\_\_\_  
(Print Name)  
Title: \_\_\_\_\_  
Address: \_\_\_\_\_  
City,State/Province, Country, Postal Code  
Telephone: \_\_\_\_\_  
(Fax or email address)  
Date: \_\_\_\_\_

For a copy of the appropriate Software License Agreement, call Echelon Corporation at 1-408-938-5200.

# Glossary

This glossary defines LONWORKS terms that you may encounter when using the LNS DDE Server.

<b>Application Device</b>	<p>The individual sensor, actuator, and controller devices that make up a network. Application devices contain an application program developed by the device manufacturer or system integrator. Typically, device manufacturers develop application programs using a LonBuilder Developer's Kit or NodeBuilder Development Tool.</p> <p>Some application devices use a LONWORKS network interface, and execute their application programs on a host processor. In this case, a Neuron Chip in the network interface may be used as a communications co-processor for the host processor. These are called host-based devices.</p>
<b>Application Message</b>	<p>A type of LonWorks message used to exchange low-level information between application devices. A <i>message code</i> within the message identifies the type of information contained within the remainder of the message. Interpretation of application message codes is application-dependent, and is generally not interoperable.</p>
<b>Configuration Parameter</b>	<p>Configuration properties implemented as memory blocks in an application device. The configuration parameter memory block may be read or written over the network using the LonTalk file transfer protocol or using direct memory read and write configuration network management messages.</p>
<b>Configuration Property</b>	<p>Configuration properties are high-level objects used to hold application configuration data. Configuration properties may be implemented as network variables or configuration parameters.</p>
<b>DDE</b>	<p>Dynamic Data Exchange. A standard protocol that Windows applications can use to exchange information with one another.</p>
<b>DDE Advise</b>	<p>A request by a DDE client for ongoing updates for a specific piece of data from a DDE server. There are two types of Advise requests.</p> <p><i>Hot or Automatic</i>            Whenever the value of the item changes, the server automatically sends the client the new value.</p> <p><i>Warm or Notify</i>            Whenever the value of the item changes, the server sends the client notification that the value has changed. If interested, the client requests the new value from the server. The LNS DDE Server does not support warm (notify) links.</p> <p><i>Cold or</i>                    The server does not notify a client of</p>

	<i>Manual</i> changes; the server only sends the item's value when requested by the client.
<b>DDE Client</b>	The originator of a DDE conversation. A DDE client application sends requests for information to a DDE server.
<b>DDE Conversation</b>	An exchange of information between two applications using DDE.
<b>DDE Execute</b>	A request from a DDE client to a DDE server to perform a command.
<b>DDE Initiate</b>	A request to start a new DDE conversation.
<b>DDE Poke</b>	A request from a DDE client to a DDE server to change the value of a specific piece of data.
<b>DDE Request</b>	A one-time request by a DDE client for a specific piece of data from a DDE server.
<b>DDE Server</b>	An application that responds to the requests of DDE client applications.
<b>DDE Terminate</b>	A request to end a DDE conversation.
<b>DDE Transaction</b>	A DDE client's request and the resulting response.
<b>Foreign Frame Message</b>	A type of LONWORKS message used to exchange low-level information between application devices, where that information contains encapsulated information from a non-LONWORKS network or device. A <i>message code</i> within the message identifies the type of information contained within the remainder of the message. Interpretation of foreign frame message codes is application-dependent, and is not interoperable.
<b>LNS Network Interface</b>	The hardware and its associated software network driver that connect a computer (such as the computer running the LNS DDE Server) to a LONWORKS network. An LNS network interface also manages communication with an LNS Server. Examples are the PCC-10, PCLTA-10, PCLTA-20, SLTA-10, and PCNSI. Also called a Network Services Interface (NSI).
<b>LNS Network Operating System</b>	The LNS network operating system consists of hardware and software that provides a broad range of services required by network tools and other network applications. These services include network management (installation, configuration, maintenance and repair), and system-wide monitoring and control.  The LNS DDE Server is an LNS application
<b>LONMARK</b>	The LONMARK Interoperability Association is an open international organization that develops guidelines and

international organization that develops guidelines and certification procedures for LONWORKS products. Devices that carry the LONMARK logo are able to interoperate even if they are made by different manufacturers. Only devices that adhere to the LONMARK guidelines and have passed a conformance review may carry the LONMARK logo.

<b>LONWORKS Message</b>	Application data contained within a LonWorks packet. LonWorks messages are used to exchange data between application devices on a LONWORKS network. A LonWorks message includes a <i>message code</i> that identifies the type of message. The interpretation of the remainder of the message depends on the message type specified by the message code. Also called a LonTalk message.
<b>LONWORKS Protocol</b>	The protocol used on LONWORKS networks to standardize communication between application devices. This protocol is defined by the EIA-709.1 standard. Also called the LonTalk protocol.
<b>LONWORKS System</b>	A set of tools and components for implementing control networks using the LonWorks protocol.
<b>Message Code</b>	A message type identifier within a LonWorks message. The message code identifies the message as an application message, foreign frame message, network variable message, network management message, or network diagnostic message.
<b>Message Tag</b>	Input and output ports that devices use to send and receive LonWorks messages.
<b>Neuron Chip</b>	A family of VLSI components that implements the LonWorks protocol. The Neuron Chip can manage I/O devices and execute user-written application code, or alternatively be used to create a network interface to a host processor.
<b>Network Services Interface (NSI)</b>	See LNS network interface.
<b>Network Variable</b>	High-level objects that application devices use to communicate with one another. The types, functions, and number of network variables in each device are determined by the application code within the device. Network variable update messages are a special type of LonWorks message.
<b>PCC-10 PC Card LonTalk Adapter</b>	The PCC-10 is a type II PC Card (formerly known as PCMCIA) LNS network interface. It has an on-board TP/FT-10 transceiver. Other types of transceivers are also supported by using external interface pods.
<b>PCLTA-10 PC</b>	The PCLTA-10 is a high performance ½ length ISA plug-n-

**LonTalk Adapter**

play compatible interface card that connects to a LONWORKS network. It has an on-board twisted-pair transceiver of one of the following types: TP/XF-78, TP/XF-1250, or TP/FT-10.

**PCLTA-20 PC  
LonTalk Adapter**

The PCLTA-20 is a high performance PCI plug-n-play compatible interface card that connects to a LONWORKS network. It has an on-board twisted-pair transceiver of one of the following types: TP/XF-78, TP/XF-1250, or TP/FT-10.

**SLTA-10 Serial  
LonTalk Adapter**

The SLTA-10 is an LNS network interface with an EIA-232 serial interface. The SLTA-10 is capable of communicating over a pair of modems, allowing the computer to be remote from the network for applications such as remote monitoring or diagnostics. It has an onboard twisted-pair transceiver of one of the following types: TP/XF-78, TP/XF-1250, or TP/FT-10.

