ECHELON

NodeUtil and NodeLoad User's Guide

Load, configure, test, and diagnose LON® and LON/IP devices and routers using the NodeUtil and NodeLoad utilities

Overview

NodeUtil is a test and diagnostics tool for LON® and LON/IP devices and networks. NodeLoad is an application loader for LON and LON/IP devices. NodeUtil and NodeLoad are Windows console applications that you can use with Adesto's family of network interfaces and other OpenLDV compatible network interfaces, including the U10, U20, U60, and SmartServer IoT network interfaces, to load, configure, test, and diagnose LON and LON/IP devices, routers, and network interfaces.

Contents

Chapter 1 – Introduction

Welcome

NodeUtil is a test and diagnostics tool for LON® and LON/IP devices and networks. NodeLoad is an application loader for LON and LON/IP devices. NodeUtil and NodeLoad are Windows console applications that you can use with Adesto's family of network interfaces and other OpenLDV compatible network interfaces, including the U10, U20, U60, and SmartServer IoT network interfaces, to load, configure, test, and diagnose LON and LON/IP devices, routers, and network interfaces.

NodeUtil and NodeLoad work with Adesto's OpenLDV network driver, which is included with several Adesto software tools, such as the IzoT Commissioning Tool. You can also download the OpenLDV network driver royalty-free from www.echelon.com/downloads.

NodeUtil and NodeLoad are tools for experienced LON developers and integrators. Using NodeUtil or NodeLoad, it is possible to make a device inoperable, so that it no longer communicates with other devices in its network. Certain operations, such as modifying communications parameters to incorrect values, can cause an unrecoverable device failure which invalidates any device warranties.

Chapter 2 – Getting Started

System Requirements

The following are the minimum system requirements for NodeUtil and NodeLoad:

- 64-bit and 32-bit versions of Microsoft Windows 10, Windows 8.1, Windows 7, Windows Server 2016, and Windows Server 2012
- OpenLDV 5 (or newer) network driver
- 1GHz x86 or x64 processor or faster with SSE2 instruction set or newer (2 GHz or faster processor recommended)
- 2 GB RAM minimum plus the minimum Windows requirements for the selected version of Windows
- 10MB of available hard-disk space
- Adesto or other OpenLDV 5 (or newer) compatible network interface including the Echelon U10, U20, U60, and SmartServer IoT network interfaces
- Available USB 1.1, 2.0, or 3.0 port (if you are using a U10, U20, or U60 USB Network Interface)

Installing NodeUtil and NodeLoad

To download and install the NodeUtil and NodeLoad utilities, follow these steps:

1. Download and install version 5 or later of the OpenLDV network driver, if you do not already have it. You can download the latest OpenLDV network driver from www.echelon.com/downloads – enter **OpenLDV** in the Search box and select the latest version.

2. Configure your network driver, if required. See the documentation for your selected network interface for details.

3. Download NodeUtil. You can download NodeUtil from www.echelon.com/downloads – enter **NodeUtil** in the Search box and select the latest version. Starting with NodeUtil 3.51, NodeLoad is included in the download.

4. Unzip the contents of the file to a folder on your computer. For example, you can create a **NodeUtil** folder in your LonWorks folder, and unzip the NodeUtil zip package to this new folder. You run the **NodeUtil.exe** and **NodeLoad.exe** programs from this folder. Optionally, you can add the folder you select to the Windows path so that you can run NodeUtil or NodeLoad from any directory. If you have installed other Adesto tools, you can unzip the NodeUtil and NodeLoad utilities to your LonWorks **Bin** folder. The installation programs for many Adesto tools add the LonWorks **Bin** folder to your Windows path, which makes it easy to run NodeUtil and NodeLoad from any directory.

Chapter 3 – Using NodeUtil

Starting NodeUtil

To start the NodeUtil Node Utility, follow these steps:

1. Open a Windows Command Prompt. To open the Command Prompt, click the Search box in the Windows taskbar and enter **cmd**. Then, click the **Command Prompt** result.

2. If you installed NodeUtil in a directory that is not in your Windows path, use the **cd** (change directory) command set the current directory to the directory that contains the **NodeUtil.exe** file.

3. Enter **NodeUtil**, with appropriate command-line parameters (as described in the next section, *Command-Line Parameters*), to start the NodeUtil Node Utility. Alternatively, you can create a Windows shortcut to launch NodeUtil with your preferred command-line parameters.

When the NodeUtil Node Utility starts, it displays introductory text similar to the following text, followed by the NodeUtil main menu.

```
Echelon Node Utility Release 3.41
Vldv_open: C:\Program Files (x86)\LonWorks\system\NodeUtil LON1 NodeUtil
Received uplink local reset
Successfully installed TP/FT-10 network interface.
Welcome to the Echelon Node Utility application.
Activate the service pin on remote device to access it.
Enter one of the following commands by typing the indicated letter:
```

If NodeUtil cannot open the network driver, NodeUtil displays an error message similar to the following message, and exits.

```
Echelon Node Utility Release 3.41
Error initializing network interface (LON1): could not open network driver
```

In this case, check that the network driver is loaded correctly, that you have OpenLDV 5 or later installed, and that your network interface hardware is correctly configured.

NodeUtil Command-Line Parameters

The NodeUtil command has the following format:

> nodeutil *<command-line-parameters>*

where *<command-line-parameters>* include the following:

–D*<interface_name>*

> Specifies the name of the network interface to use. This parameter is required if your network interface is not named **LON1**, but is optional if you use **LON1**. If you specify an invalid network interface name, NodeUtil displays an error message. The network interface name is not case sensitive.

> To specify a SmartServer IoT or other edge server with an OpenLDV compatible RNI, use **x.***<Profile>*.*< SmartServer or i.LON name>*. For example, **-Dx.Default.SmartServerAtHQ** specifies the **Default** xDriver profile, with the **SmartServerAtHQ** edge server.

–A

Selects application mode. In this mode, NodeUtil does not configure the network interface, and does not accept service pin messages. That is, NodeUtil treats the network interface as an application device.

If you do not specify this parameter, NodeUtil assigns the network interface to a random subnet address in the zero-length domain with a node ID of 126 and sets the network interface to the configured state.

–B

Directs NodeUtil to use only the OpenLDV network driver, and bypass the IP-852 network interface driver (formerly called VNI), if installed. In this mode, other host applications cannot share a single network interface.

This option need not be specified because **–B** is a default parameter.

–H

Prevents NodeUtil from changing the configuration of the network interface.

–I<*input_file*>

Specifies that input to NodeUtil is an input text file instead of the keyboard. See *Using a Script File with NodeUtil* for more information.

–J

Prevents NodeUtil from querying self-identification (SI) data from a host-based device, such as a device using a network interface with an SMIP or ShortStack Micro Server. By using this parameter, you can query a host-based device that does not have an attached host processor.

–M

Specifies that an SMIP or ShortStack device is being loaded (see *Y – Download Neuron Executable*). If you plan to load an SMIP or ShortStack device that is not currently attached to its host processor, use the **–J** parameter in addition to the **–M** parameter.

–O<*output_file*>

Specifies that output from the NodeUtil Node Utility be copied to the specified file. You can use this parameter to create a script file that you can use with the **–I** parameter. See *Using a Script File with NodeUtil* for more information.

–S

Specifies slower transaction timers, which are suitable for use with PL-20 power line channels. By default, NodeUtil uses transaction timers that are suitable for FT channels.

–V

Specifies that NodeUtil start up in verbose mode. In this mode, NodeUtil displays all network interface messages in detail.

–Z<*domain_index*>

Specifies the domain index for NodeUtil to use for the local network interface. You can support two instances of NodeUtil on the same network interface by selecting a different domain index for each.

–?

Displays the command format, including a brief description of the command-line parameters, and exits.

The NodeUtil Main Menu

When NodeUtil starts, it displays introductory text followed by the following NodeUtil main menu:

```
Welcome to the Echelon Node Utility application.
Activate the service pin on remote device to access it.
Enter one of the following commands by typing the indicated letter:

F1-- Show Banner.
A -- (A)dd device to list.
C -- (C)lear the device list.
D -- Set the (D)omain of the network interface.
E -- (E)xit this application.
F -- (F)ind devices in the current domain.
1 -- Find devices in all (1)-byte domains.
2 -- ATM discovery.
G -- (G)o to device menu....
H -- (H)elp with commands.
L -- Display device (L)ist.
M -- Change device (M)ode or state.
P -- Send a service (P)in message.
Q -- (Q)uickly add device to list.
R -- (R)eboot 3150 device.
S -- Report device (S)tatus and statistics.
T -- (T)ransceiver parameters.
V -- Control (V)erbose modes.
W -- (W)ink a device.
X -- E(X)peditiously go to device.
Y -- Download multiple devices.
Z -- Shell out to command prompt.
[ -- Performance test.
] -- Performance test / response validation.
< -- Redirect input from a file.
> -- Redirect output to a file.
NodeUtil>
```

Type the indicated letter (or function key) to run the corresponding command. You can stop any command by pressing the **ESC** key.

The **G**, **M**, **R**, **S**, and **W** commands prompt you to enter an ID for the device that you want to address with the command. NodeUtil maintains a list of devices indexed by ID number. This device list initially contains only the network interface itself, which has an ID of 0. For most of these commands, the default device is the last device in the list. You can use any of the following methods to add devices to the NodeUtil device list:

- Attach the device to the network and activate its service pin. When NodeUtil receives the service pin message, it adds the device to the device list and then displays the device ID and the program ID of the device. Subsequent service pin messages only display the device ID. If the service pin message is received from a router, two entries are made in the device list; the first entry will be for the router near side (the side closest to the network interface), and the second entry for the router far side.

- Enter either the **Find devices in all (1)-byte domains** or the **(F)ind devices in the current domain** command. All unconfigured devices, and devices configured in any 1-byte domain, or in the configured domain of the network interface, as appropriate to the command selected, are added to the device list.

- Enter the **(A)dd device to list** command. You are prompted to enter the 6-byte Unique ID of the device you wish to add to the list. Enter the Unique ID in hex (for example, **9000011E28061102**).

For commands that require a Unique ID, you can reference the Unique IDs for the devices in the device list with "**@**<*n*>", where *n* is the device index in the device list. For commands with an iteration count, you can reference the next Unique ID in the device list with "**@@**". For example, if you have device indices 1, 2, 3 and 4 with Unique IDs A, B, C and D and go to device 2 and enter the following commands, the test will iterate through A, C, D, A, C, D.

```
[
6
0
q
3
10
n
4d
00cc4b @@ 51
```

For commands that require a filename, you can specify "**<**" to reference the last file used.

For request and response data, you can specify the following macros:

- **%1** – 8-bit device index
- **%2** – 6-byte Unique ID
- **%3** – 8-bit device index or-ed with 0x80

Use the (**G**)**o to device menu** command to display the device menu; see *The NodeUtil Device Menu*.

F1 – Show Banner

Displays the name and version of the NodeUtil Node Utility.

A – Add Device to List

Adds a device to the device list. The command prompts for the device's Unique ID. Enter the Unique ID as six hex values, optionally separated by spaces. For example, you can enter "**9000011E28061102**" or "**90 00 01 1E 28 06 11 02**". If NodeUtil can communicate with the device, it is added to the directory. If NodeUtil cannot communicate with the device, it displays the following error message: **Query status: Could not communicate with the node**.

C – Clear the Device List

Clears all the devices from the device list.

D – Set the Domain of the Network Interface

Displays, and optionally sets, the first domain table entry of the network interface. You can use this command to specify the domain before using the (**F**)**ind devices in the current domain** command (from the main menu).

When you enter the **D** command, you are prompted for the following parameters:

1. The domain ID length (0, 1, 3 or 6 bytes). Press the **ESC** key if you do not want to change the domain.

2. For domain ID length greater than zero, the domain ID values in hex, optionally separated by spaces. For example, enter "**0011AB**" or "**00 11 AB**" for a three-byte domain. The default domain ID is all zeroes.

3. The subnet ID for the network interface. The node ID in that subnet is fixed at 126 (decimal) or 7E (hexadecimal).

4. The authentication key to use for the primary domain table entry. Enter the key as hex bytes separated by spaces. The default key is 0xFF for all bytes.

5. Whether you are sure you want to make the domain change. If you reply **Y**, the new domain entry is written into the device; if you reply **N**, no change is made.

The second domain table entry of the network interface is always set to the zero-length domain for the purpose of receiving service pin messages.

You can change the authentication key to update network variables that have been configured as authenticated. An update to an authenticated network variable must be sent using a domain in which the target device is configured, using the correct authentication key.

E – Exit this Application

Exits the NodeUtil Node Utility. The device list is not saved.

F – Find Devices in the Current Domain

Finds all devices in the current domain. The current domain is set by the **Set the (D)omain of the network interface** command. The **F** command sends a domain-wide Query ID broadcast to all devices, and then queries all responding devices for their identification. Each new device is added to the device list. This command is useful when the service pins and Neuron IDs of the remote devices are not accessible. It discovers unconfigured and application-less devices, as well as devices that are configured in the domain of the network interface.

To query for devices configured in a specific domain, use the **Set the (D)omain of the network interface** command before the **(F)ind devices in the current domain** command. The **(F)ind devices** command broadcasts on the domain that you assign to the network interface. Changing the domain and subnet to the configured domain for the network is especially useful if your network contains routers. Normally, routers are set to the configured routing mode because that mode is the most bandwidth efficient. In the configured routing mode, routers pass messages only for configured domains. However, if the routers are set to the repeater mode, it might not be necessary to modify the domain and subnet node address of the network interface in order to find devices on domains, because the routers configured as repeaters pass valid messages sent on any domain.

1 – Find Devices in All 1-Byte Domains

Finds all devices in all one-byte domains. This command sends a domain-wide Query ID broadcast in each of the one-byte domains, and then queries all responding devices for their identification. Each new device is added to the device list. This command is useful when you do not know the domain of the devices in the network, and you are not able to physically access the devices in order to press the service pin. This command discovers unconfigured and applicationless devices, as well as devices that are configured in the domain of the network interface. This command uses the domain information stored in the network interface, and rewrites the domain information to the network interface when the command completes.

This command can take a long time to complete. You can interrupt the command by pressing any key on the keyboard. To query for devices configured in a specific domain, use the **Set the (D)omain of the network interface** command before the **(F)ind devices in the current domain** command.

G – Go to Device Menu

Displays the current device list, and allow you to switch to the device menu for a specific device. Enter the ID of the device (first column of the device list) that you want to address. After you enter the device ID, the prompt changes to **DEVICE:***x***>**, where *x* is the ID of the device. To return to the main menu, use the **(E)xit** command.

You can use the device menu to display and update various data structures within a device, and to interact with the device using network variables or application messages. For descriptions of the information displayed by the device menu commands, see the *FT 3120 / FT 3150 Smart Transceiver Data Book*, the *PL 3120 / PL 3150 / PL 3170 Power Line Smart Transceiver Data Book*, the *Series 5000 Chip Data Book,* or the *Series 6000 Chip Data Book*.

See *The NodeUtil Device Menu* for a description of the device menu and its commands.

H – Help with Commands

Displays the main menu. The main menu is also displayed if you enter an unrecognized command.

L – Display Device List

Displays the NodeUtil device list. You can add devices to the list using any of the methods described in *The NodeUtil Main Menu*. The list includes the network interface, devices, and routers that NodeUtil has discovered or that you have added. For each device (and for each side of a router), the NodeUtil device ID, Neuron ID, and program ID is displayed. The program ID is displayed in hexadecimal; it is also displayed in text if it is a text program ID.

M – Change Device Mode or State

Changes the device mode or state for one or all devices. If there is more than one device in the device list, you are prompted to enter a device ID, or to select all devices. The default is all devices, meaning that the mode or state that you select will be changed for all devices in the current domain. See *D – Set the Domain of the Network Interface* for more information about setting the domain.

See the device menu Change Device Mode or State command, *M – Change Device Mode or State*, for more information.

P – Send a Service Pin Message

Sends a service pin message using the NodeUtil network interface.

Q – Quickly Add Device to List

Adds a device to the NodeUtil device list without attempting to communicate with the device.

Use this command in environments that have a large number of devices that you manage with NodeUtil scripts (see *Using a Script File with NodeUtil*). The script can add the devices in the same order (so that they have the same device ID for each run of the script, thus aiding network testing). Also, for power line channels, adding many devices without the extra communications can be useful.

R – Reboot 3150 Device

Restores the contents of the on-chip EEPROM memory of an FT 3150 or PL 3150 Smart Transceiver to the state it had when the off-chip EEPROM was first installed.

This command restores the communications parameters, device state, buffer configuration, and on-chip EEPROM variables. It also re-initializes the domain, address, and network variable configuration tables. If the device was previously installed in a network, the reboot command removes it from the network.

This command does not operate on a FT 3120 or PL 3120 Smart Transceiver, a Series 5000 device, or a Series 6000 device.

You can use the **(R)eboot 3150 device** command on the network interface used to run NodeUtil, if it uses a FT 3150 or PL 3150 Smart Transceiver. In this case, the network interface is reconfigured after the reboot completes, so that it can continue to be used.

S – Report Device Status and Statistics

Sends a message to query the status of a specific device, and display the results. When you run this command from the main menu, you are prompted to enter a device ID.

See the device menu Report Device Status and Statistics command, *S – Report Device Status and Statistics*, for more information.

T – Transceiver Parameters

Displays the transceiver ID of the network interface, and to set the transceiver ID for network interfaces that have software-configurable transceiver IDs. For all network interfaces, this command also allows you to change the transceiver parameters for the network interface.

WARNING: You can make a device inoperable by changing the transceiver parameters to an invalid value. Changing the transceiver parameters to an incorrect value may invalidate any manufacturer warranty for the device.

See the device menu Transceiver Parameters command, *T – Transceiver Parameters*, for more information.

V – Control Verbose Modes

Enables or disables three verbose modes: for network interface messages, for LON network traffic, and for compressed UDP traffic. These modes are useful for debugging network problems. By default, network interface message verbose mode is off, network traffic verbose mode is on and UDP traffic is off.

If you enable verbose mode for network interface messages, NodeUtil displays the contents of all application buffers going to and coming from the NodeUtil network interface.

If you enable verbose mode for network traffic, any incoming message addressed to the application processor on the network interface is displayed.

If you enable verbose mode for UDP traffic, any incoming messages that are addressed to the network interface or use broadcast addressing will be displayed with decompression of the UDP header.

W – Wink a Device

Sends a wink message to a device or all devices. If the device has been programmed to respond to this message, it activates the wink task. You are prompted to enter a device ID for this command; if you accept the default (all devices), the wink message is sent as a broadcast to all devices configured in the current domain.

Y – Download Multiple Devices

Loads an application to multiple devices using the Image Upgrade Protocol (IUP). This command can only be used with devices that support IUP, which include devices based on the IzoT CPM 4200 Wi-Fi Module.

Z – Shell Out to Command Prompt

Opens a Windows command prompt without losing the device list. Type **exit** at the command prompt to return to the NodeUtil Node Utility.

[– Performance Test

Tests the performance of the channel and all devices in the device list.

You are prompted to provide the following information:

- **Number of iterations** – How many times to repeat the performance test before displaying the collected data. The default is once. For an indefinite number of iterations, specify 0. You can press the [key at any time to end the test.
- **Delay between messages** – How long, in milliseconds, to delay between sending messages to the device. The default is no delay.
- **Acknowledged, Unacknowledged, Request, or Repeated** – The type of message service to use for each message. The default is Acknowledged.
- **Retry count** – How many times to retry sending a message before marking it failed. The default for FT channels is three times; the default for power line channels is seven times.
- **Tx timer** – Transaction timer.
- **Use authentication** – Whether (Yes) or not (No) to use authentication for the messages. The default is No.
- **Message code** – The message code to send to the device. The default is 0.
- **Up to 100 bytes of hex data** – The data that you want to send to the device.

The result of the test is displayed at the conclusion of the test. For example, if the tests were successful, you receive **RESULT: No failures**. For an unsuccessful test, you receive a summary of the failures, such as **RESULT: 2 of 50 messages failed (4.00%)**.

You can also use the **Report device (S)tatus and statistics** command to see how the test affected the statistics collected by the device.

] – Performance Test / Response Validation

Tests the performance of the channel and the device with optional validation of responses. Parameters are the same as the "**[**" command. You are prompted to specify whether or not you want responses to be validated. If you specify response validation, you are prompted to enter the expected response.

< – Redirect Input from a File

Runs a command script. A command script is a text file that contains NodeUtil commands. A script file can use have any file extension, but a common extension is ".**scr**".

Commands that do not require a carriage return must be on their own line. Input that requires a carriage return must be on its own line. Network variable data is displayed (and entered) in hexadecimal.

You can run a script repeatedly, as needed.

See *Using a Script File with NodeUtil* for more information.

> – Redirect Output to a File

Redirects all NodeUtil output to a file. You are prompted to specify the file name. The default file name extension is **.log**. If the file already exists, you are asked if you want to **(O)verwrite** the file from the beginning, or **(A)ppend** output to the end of the file.

To stop redirection, enter a null file name.

You can use this command to save the output of any of the NodeUtil commands, and for creating script files. See *Using a Script File with NodeUtil* for more information.

The NodeUtil Device Menu

From the NodeUtil main menu, enter the **(G)o to device menu** command to display the current device list and switch to the following device menu for a specific device. Enter the ID of the device (first column of the device list) to select a device. After you enter the device ID, the prompt changes to **DEVICE:*x*>**, where *x* is the ID of the device. To return to the main menu, use the **(E)xit** command.

```
Enter one of the following commands:
F1-- Show Banner.
 A -- (A)ddress table.
 B -- (B)uffer configuration.
 C -- Application (C)onfiguration structures.
 D -- (D)omain table.
 E -- (E)xit this menu and return to main menu.
 F -- Configuration (F)iles.
 G -- (G)o to another device.
 H -- (H)elp with device commands.
 I -- Network variable al(I)as table.
 J -- (J)am network variable type.
 K -- Chec(K) Neuron executable.
 L -- (L)ist network variables.
 M -- Change device (M)ode or state.
 N -- (N)etwork Variable configuration table.
 O -- Device advanced (O)ptions.
 P -- (P)oll network variable.
 Q -- (Q)uickly send a message.
 R -- (R)ead device memory.
 S -- Report device (S)tatus and statistics.
 T -- (T)ransceiver parameters.
 U -- (U)pdate input network variable.
 V -- Control (V)erbose modes.
 W -- (W)rite device memory.
 X -- Create device interface ((X)IF) file.
 Y -- Download Neuron executable.
 Z -- Disas(Z)emble memory.
 $ -- Read device memory relative.
 = -- Signal strength.
 * -- Refresh memory.
 [ -- Performance test.
 ] -- Performance test / response validation.
 < -- Redirect input from a file.
 > -- Redirect output to a file.
 ! -- IP addresses.
DEVICE:1>
```

You can use the device menu to display and update various data structures within a device, and to interact with the device using network variables or application messages. For descriptions of the information displayed by the device menu commands, see the *FT 3120 / FT 3150 Smart Transceiver Data Book*, the *PL 3120 / PL 3150 / PL 3170 Power Line Smart Transceiver Data Book*, the *Series 5000 Chip Data Book,* or the *Series 6000 Chip Data Book*.

The following commands are not available for the NodeUtil network interface or for routers: F, I, J, K, L, N, P, U, X, Y, =, or *.

Type the indicated letter (or function key) to run the corresponding command. You can stop any command by pressing the **ESC** key. These commands operate on the selected device, which is displayed in the NodeUtil **DEVICE** prompt.

F1 – Show Banner

Displays the NodeUtil name and version.

A – Node Address Table

Displays or updates a specific address table entry or the entire address table for the selected device. The address table defines the network addresses to which the current device can send implicitly addressed messages and network variables. It also defines the groups to which the device belongs.

The columns of data displayed include:

- **Index** – Defines the index of the address table entry.
- **Type** – Defines the address type (for example, group address, broadcast, subnet/node, Unique ID, or unbound).
- **Domain** – Defines the domain to which the device belongs.
- **Mbr/Nod** – Defines the member ID of the device within the group for group addresses. The member ID is used in acknowledgments to allow the sender of an acknowledged multicast message to keep track of which nodes have responded.
- **Rpt tmr** – Defines the time interval (in milliseconds) between repetitions of an outgoing message for the repeated service.
- **Retries** – Defines the number of retries (0 to 15) for acknowledged, request/response, or unacknowledged-repeated service. The maximum number of messages sent is one more than the displayed number.
- **Rcv tmr** – Defines the receive timer (in milliseconds) for multicast (group) messages. If a message with the same transaction ID is received before the receive timer expires, it is considered to be a retry of the previous message.
- **Tx tmr** – Defines the time interval (in milliseconds) between retries for the acknowledged or request/response service. The transaction retry timer is restarted when each attempt is made, and also when any acknowledgment or response (except for the last one) is received.
- **Grp/Sbnt** – Defines the ID (0 to 255) of the group within the domain for group addresses. For managed networks, the group ID is allocated by the network management server. For Neuron ID addressing, a value of 0 indicates that the message should pass through all routers in the domain. For subnet/node addressing, a value of 0 is not a valid subnet ID. For broadcast addressing, a value of 0 indicates that the message should be delivered to all nodes in the domain.

After the entry is displayed, you can change the entry, one value at a time. For each value, the default value is the current setting. If you enter a non-valid value, the utility displays an error message (Value out of range) and displays the maximum allowable size, and you are prompted to enter a new value.

See the *ISO/IEC 14908-1 Control Network Protocol* standard for a description of the address table.

B – Buffer Configuration

Displays and changes the buffer configuration of the selected device. The columns of data displayed include:

- **Type** – Defines the buffer type.
- **Count** – Defines the current buffer count for the specified buffer type.
- **Size** – Defines the size of the specified buffer type.
- **Bytes** – Defines the number of bytes required for the specified buffer type (count times size).

The display also includes the total number of bytes required for the specified buffer configuration.

After displaying the current buffer configuration, you can change the configuration, one value at a time. For each value, the default value is the current setting. If you enter a non-valid value, the

utility displays an error message (Value out of range) and displays the maximum allowable size, and you are prompted to enter a new value.

After you update buffer values (or keep current values), the utility displays the new configuration, and you can accept or cancel the command before the new values are written to the device. If you configure the device to use more RAM than is actually available, the device will most likely crash:

- If the device is based on an FT 3120, PL 3120, or PL 3170 Smart Transceiver, it might not be recoverable.
- If it is based on an FT 3150 or PL 3150 Smart Transceiver, you might be able to recover it with the **(R)eboot** command. If the device is not recoverable with the reboot command, you can use the EEBLANK program (available from the Echelon Web site) to blank the EEPROM and reboot the default image.
- If it is based on an FT 5000 Smart Transceiver, Neuron 5000 Processor, FT 6000 Smart Transceiver, or Neuron 6000 Processor, you might be able to recover it by clearing the external non-volatile memory for the device. See the appropriate Series 5000 or Series 6000 Data Book for a description of how to clear the external non-volatile memory.

See the *Neuron C Programmer's Guide* for a description of the buffers.

C – Application Configuration Structures

Displays data from the read-only and configuration data structures of the selected device.

See the *ISO/IEC 14908-1 Control Network Protocol* standard for a description of the configuration data structures.

D – Device Domain Table

Displays a selected domain table entry or the entire domain table of the selected device. If the device uses DHCP the string DHCP is also displayed for domain index 0. You can also update the domain. You can set the authentication key even if the domain is not used.

You can enable and disable DHCP and Open Media Authentication (OMA) using the 'O' command.

If you prefer to display and update the devices address using IP format you may do so using the "!" command.

E – Exit this Menu and Return to Main Menu

Exits from the device menu and return to the main menu. The prompt changes to the **NodeUtil>** prompt.

F – Configuration Files

Reads the configuration file from the selected device and interpret the template. This command is supported only for devices that have a LonMark compliant Node Object functional block, which allows for direct memory read access to the configuration file. Enter the index of the **SNVT_address** network variable which points to the file directory. The template is checked for syntax errors.

G – Go to Another Device

Displays the current device list, and allow you to select another device for device menu operations. The prompt changes to **DEVICE:***x***>**, where *x* is the ID of the selected device.

H – Help with Device Commands

Display the device menus. The device menu is also displayed if you enter an unrecognized command.

I – Network Variable Alias Table

Displays or updates a selected network variable alias table entry or the entire network variable alias table of the selected device. Which columns of data displayed is dependent on whether or not the device supports the Extended Command Set (ECS) network management messages.

The columns displayed for devices that do not support ECS include:

- **Index** – Defines the index of the network variable alias table entry.
- **Selctr** – Defines the 14-bit network variable selector in the range 0 to 0x3FFF. Selector values 0x3000 – 0x3FFF are reserved for unbound network variables, with the selector value equal to 0x3FFF minus the network variable index. Selector values 0 – 0x2FFF are available for bound network variables. The input network variables on any one device must all have different selectors. For managed networks, the network management server allocates the network variable selectors.
- **Dir** – Defines the direction of the network variable, input or output.
- **Prio** – Defines whether the network variable is a priority network variable. It is specified by bind_info(priority | nonpriority) in the Neuron C declaration of the network variable.
- **Auth** – Defines whether the network variable uses authentication. For Neuron C applications, it is specified by bind_info(authenticated | nonauthenticated) in the Neuron C declaration of the network variable.
- **Addridx** – Defines the index into the address table for the network variable.
- Service – Defines the service used to deliver the network variable. For Neuron C applications, it is specified by bind_info(ackd | unackd_rpt | unackd) in the Neuron C declaration of the network variable.
- **TrnArnd** – Defines whether the network variable is a turnaround network variable, i.e. an NV that is bound to another network variable on the same device.
- **Primary** – Defines the index into the network variable configuration table.

The columns displayed for devices that do support ECS include:

- **Idx** – Defines the index of the network variable alias table entry.
- **Sel** – Defines the 14-bit network variable selector in the range 0 to 0x3FFF. Selector values 0x3000 – 0x3FFF are reserved for unbound network variables, with the selector value equal to 0x3FFF minus the network variable index. Selector values 0 – 0x2FFF are available for bound network variables. The input network variables on any one device must all have different selectors. For managed networks, the network management server allocates the network variable selectors.
- **Dir** – Defines the direction of the network variable, input or output.
- **Pri** – Defines whether the network variable is a priority network variable. It is specified by bind_info(priority | nonpriority) in the Neuron C declaration of the network variable.
- **Auth** – Defines whether the network variable uses authentication. It is specified by bind_info(authenticated | nonauthenticated) in the Neuron C declaration of the network variable.
- **Addr idx** – Defines the index into the address table for the network variable.
- **Svc** – Defines the service used to deliver the network variable. It is specified by bind_info(ackd | unackd_rpt | unackd) in the Neuron C declaration of the network variable.

- **Trn Ard** – Defines whether the network variable is a turnaround network variable, that is, bound to another network variable on the same device.
- **ECS Selector Types**
  - **Updt** –Defines how NV selection is performed for network variable updates, one of:
    - **Sel** – Use the selector only.
    - **Src** – Use the selector and source address. The source address is defined by the address table index.
    - **None** – No selection supported.
  - **Req** - Defines how NV selection is performed for network variable poll requests. The selection types are the same as for update selection.
  - **Resp** - Defines how NV selection is performed for responses to a network variable poll requests. The selection types are the same as for update selection.
- **SrcSel Only**. Indicates that the address index is used only for source selection only, and not for polling or updating remote network variables.
- **Rmt Auth** – Indicates whether to use authentication when sending network management messages such as NV fetch or NV update by index to the bound device.
- **R-B Indx** – Indicates whether to use read by NV index (as opposed to poll).
- **W-B Idx** – Indicates whether to use write by NV index (rather than NV update).
- **Pimry Nv**– Defines the index into the network variable configuration table.

See the *ISO/IEC 14908-1 Control Network Protocol* standard for a description of the network variable alias table.

J – Jam Network Variable Type

Display and optionally sets a network variable type. To change a network variable type, enter the index of the network variable, and then enter the SNVT index of the new type. This command can only be used on devices with writeable self-identification data. For device applications developed in Neuron C, you must specify the **snvt_si_eecode** or **snvt_si_ramcode** Neuron C compiler directive.

K – Check Neuron Executable

Compares the downloadable image file (**.nxe** or **.ndl** extension) contents on the computer with the application image loaded into the device. Any differences are displayed. This command does not work with Series 6000 devices.

L – List Network Variables

Displays the self-documentation and other network variable information available on the selected device, in Neuron C source format. You can save the information to an output file; if you do not specify a file, the displayed data is not saved.

If the device is LonMark compliant, the device and network variable self-documentation strings are interpreted and checked for errors.

M – Change Device Mode or State

Changes the selected device's mode or state. You can select one of the following modes or commands:

- **(R)eset** – Reset the LON stack.
- **o(N)line** – Change application mode to online.
- **O(F)fline** – Change application mode to offline.
- **(S)tate change** – Change device state to applicationless, unconfigured, configured, or hard offline.
- **(P)hysical reset** – force a hard reset of the device.

If the device is a router, you can also set the router type to configured, learning, bridge, or repeater.

N – Network Variable Configuration Table

Displays or updates a selected network variable configuration table entry or the entire network variable configuration table of the selected device. The columns displayed are the same as for the network variable alias table, except for the Primary column; see *I – Network Variable Alias Table*.

See the *ISO/IEC 14908-1 Control Network Protocol* standard for a description of the network variable configuration table.

O – Device Advanced Options

Sets the following advanced options for a device.

- **OMA** – Determines whether the device should use Open Media Authentication (OMA).
- **DHCP** - Determines whether the device should use DHCP for address assignment.
- **Protocol Mode** – Determines the LonTalk Services Protocol mode. One of:
    - o **Compatibility**. This mode is supported by legacy devices, Series 6050 devices, and .most other LON/IP platforms. It is not supported by FT 6010 devices.
    - o **Enhanced**. Supports 12-bit transactions. This mode is supported Series 6050 devices, FT-6010 devices, and other LON/IP platforms. It is not supported by legacy devices.
- **Network Management Authentication** – Determines whether network management authentication is on or off.

P – Poll Network Variable

Fetches the current value of a selected network variable, or of every network variable on the selected device. Enter the index of the network variable on the device, or press the ENTER key to fetch all network variables on the selected device. The network variable values are displayed in hexadecimal bytes, and can be further interpreted depending on length.

Q – Quickly Send a Message

Sends a LON message to the selected device. You can specify the service type (acknowledged, unacknowledged, request/response, or repeated), whether to use authentication, the message code, and the data to send.

R – Read Device Memory

Displays the memory contents of the selected device. You can specify the starting address in hexadecimal format and the number of bytes to read in hexadecimal. The memory contents are displayed in hexadecimal bytes.

Use the **Redirect output to a file** (**>**) command before the **(R)ead device memory** command to save the output to a file.

S – Report Device Status and Statistics

Queries the status of the selected device, and display the results.

See the *ISO/IEC 14908-1 Control Network Protocol* standard for a description of the Query Status command.

If the selected device is a router, additional information is displayed including the router type. For a description of the meanings of these router-specific values, see the *LonWorks Router User's Guide*.

You are given an opportunity to clear the status of the device, which clears the error counters, the reset cause, and the last error logged. Clearing the status can be useful before running the **Performance test** (**[**) command; see *[– Performance Test*.

T – Transceiver Parameters

Displays the transceiver (communications) parameters of the selected device. The information displayed depends on the transceiver type. For a description of these values, see the *FT 3120 / FT 3150 Smart Transceiver Data Book*, the *PL 3120 / PL 3150 / PL 3170 Power Line Smart Transceiver Data Book*, the *Series 5000 Chip Data Book,* or the *Series 6000 Chip Data Book*.

If the selected device is a network interface that supports changing the input clock, you are prompted to change the input clock rate by selecting from a menu. The choices are 5, 10, and 20 MHz. You then have an opportunity to change the transceiver parameters by selecting from a menu. The choices depend on the input clock rate.

For Series 5000 and Series 6000 devices, you are prompted to change the system clock rate along with the interface bit rate. The system clock can be set to 5, 10, 20, 40, or 80 MHz.

The NodeUtil Node Utility does not support changing the transceiver parameters for devices with input clocks less than 5 MHz.

If the device implements transceiver ID (XID) pins, you typically cannot change the interface bit rate or the transceiver type (differential, single-ended, or special-purpose mode). To change types of devices, you must first physically set the value read from the XID pins to 30 (decimal) or 1E (hexadecimal).

Selecting incorrect communications parameters could make it impossible to communicate with the device over the network:

- If the device is based on an FT 3120, PL 3120, or PL 3170 Smart Transceiver, it might not be recoverable.
- If it is based on an FT 3150 or PL 3150 Smart Transceiver, you might be able to recover it with the **(R)eboot** command. If the device is not recoverable with the reboot command, you can use the EEBLANK program (available from the Echelon Web site) to blank the EEPROM and reboot the default image.
- If it is based on an FT 5000 Smart Transceiver, Neuron 5000 Processor, FT 6000 Smart Transceiver, or Neuron 6000 Processor, you might be able to recover it by clearing the external non-volatile memory for the device. See the appropriate Series 5000 or Series 6000 Chip Data Book for a description of how to clear the external non-volatile memory.

U – Update Input Network Variable

Updates an input network variable on the selected device. Enter the index of the network variable (use the **(L)ist network variables** command or the **(P)oll network variables** command to determine the network variable index). The data from the last network variable update to the selected network variable is displayed.

At the prompt, enter the data to be sent to the network variable, in hexadecimal bytes. Separate each pair of hexadecimal digits with a space, and enter all the required data on one line. The data values for any bytes that you do not enter after the first character default to zero. For example, you can enter 01 00 00 or just enter 1 to set a **SNVT_switch_2** input network variable to the on state. Press **ENTER** without any characters to cancel the update so that the input value is not changed.

You can use the **Set the (D)omain of the network interface** command to configure the NodeUtil network interface with a domain and authentication key to match those of the destination device if the network variable has been configured as authenticated.

V – Control Verbose Modes

Enables or disables verbose modes. These modes are useful for debugging network problems. By default, network interface message verbose mode is off, and network traffic verbose mode is on.

If you enable verbose mode for network interface messages, NodeUtil displays the contents of all application buffers going to and coming from the NodeUtil network interface. Default is off.

If you enable verbose mode for network traffic, any incoming message addressed to the application processor on the network interface is displayed. Default is on.

If you enable verbose mode for udp traffic, any incoming UDP messages are displayed. Default is off.

If you enable verbose mode for application messages, statistics show up for message arrival counts and times. Default is off.

W – Write Device Memory

Writes to the memory of the selected device. You can specify the starting address in hexadecimal format. The contents of the memory address that you specify are displayed in hexadecimal. Enter a new value for the location in hex and then press the ENTER key, or press the ENTER key with no data to leave the value unchanged. To stop entering data, enter any value that is not a hexadecimal character, such as a period (.), and press ENTER.

If you change any of the data in memory, you are given the opportunity to update both the application and configuration checksums, or just the configuration checksum.

Important: The device should be in the applicationless state before you change any of the application image memory, otherwise the device will report an application checksum error. Similarly, the device should be in the applicationless or unconfigured state before you change any of the network image memory, otherwise the device will report a configuration checksum error.

X – Create Device Interface (XIF) File

Creates a version 3 device interface file (**.xif** extension) for the selected device. You can enter a file name for the XIF file. If the device has an alphanumeric program ID, this ID is the default filename. You can use the generated XIF file with any Echelon or third-party Open LNS tool, including the OpenLNS Commissioning Tool.

If output redirection is in effect, the current output file is closed before creating the XIF file.

Y – Download Neuron Executable

Downloads a Neuron application or system image file to the selected device. You can specify an application image file (**.ndl** or **.nxe** extension) that is generated by the IzoT NodeBuilder Software. If the device already contains an alphanumeric application ID, this name is the default for the application image file name.

In some cases you will be prompted if you want to update communication parameters. In general you should say yes to this. If you say no, the device could end up with unintended communication parameters or device clock rate. In other cases you will be asked if you want to bypass communication parameter validation. You should say no to this unless you really know what you're doing.

Recommendation: Use a **.ndl** file for downloading the device application. An **.ndl** file contains the application code and initial values for configuration network variables, whereas a **.nxe** file contains the application code only.

If the download is successful, you are asked if you want to set the device configured, online. Answer **yes** to this question only if the device already contains a valid network configuration for this application. By default, the device will be set to the unconfigured state. After setting the state, NodeUtil reads the state of the device and displays any errors.

For **.nxe** files, the **Y** command does not check that the application image file that you are loading is compatible with the device hardware configuration. The **Y** command does perform compatibility checking for **.ndl** files. Thus, before loading an application to a device using a **.nxe** file, verify that the application image file was linked using the correct device model and firmware version for your target device. For FT 3150 or PL 3150 devices, also verify that the application image file was linked using the correct memory map for the device.

If an incompatible application image file is loaded to a device, the device generally performs continuous watchdog resets. You might be able to recover it by downloading a compatible application image file. However:

- If the device is based on an FT 3120, PL 3120, or PL 3170 Smart Transceiver, it might not be recoverable.
- If it is based on an FT 3150 or PL 3150 Smart Transceiver, you can use the EEBLANK program (available from the Echelon Web site) to blank the EEPROM and reboot the default image.
- If it is based on a Series 5000 or Series 6000 processor, you might be able to recover it by clearing the external non-volatile memory for the device. See the appropriate Series 5000 or Series 6000 Data Book for a description of how to clear the external non-volatile memory.

If you attempt to set the device to the configured state after loading, but the device does not contain a valid network configuration for this application, then the Neuron firmware detects a configuration checksum error and sets the device to be unconfigured.

$ – Read Device Memory Relative

Displays the memory contents of the selected device. You specify the starting address and number of bytes to read in hex. You specify the starting address as a hex offset from the node statistics structure, node configuration structure, or flash memory origin. The memory contents are displayed in hexadecimal bytes.

Use the **Redirect output to a file** (**>**) command before the **(R)ead device memory** command to save the output to a file.

[– Performance Test

Tests the performance of the channel and the device.

You are prompted to provide the following information:

- **Number of iterations** – How many times to repeat the performance test before displaying the collected data. The default is once. For an indefinite number of iterations, specify 0. You can press the [key at any time to end the test.
- **Delay between messages** – How long, in milliseconds, to delay between sending messages to the device. The default is no delay.
- **Acknowledged, Unacknowledged, Request, or Repeated** – The type of message service to use for each message. The default is Acknowledged.
- **Retry count** – How many times to retry sending a message before marking it failed. The default for FT channels is three times; the default for power line channels is seven times.
- **Use authentication** – Whether (Yes) or not (No) to use authentication for the messages. The default is No.
- **Message code** – The message code to send to the device. The default is 0.
- **Up to 100 bytes of hex data** – The data that you want to send to the device.

The result of the test is displayed at the conclusion of the test. For example, if the tests were successful, you receive **RESULT: No failures**. For an unsuccessful test, you receive a summary of the failures, such as **RESULT: 2 of 50 messages failed (4.00%)**.

You can also use the **Report device (S)tatus and statistics** command to see how the test affected the statistics collected by the device.

] – Performance Test / Response Validation

Tests the performance of the channel and the device with validation of responses. Parameters are the same as the "**[**" command.

= – Signal Strength

Performs a series of tests of the signal strength for a device attached to a power line channel. The tests include both the Primary (normal) and Secondary (alternate) LonTalk paths. The displayed output includes a summary of packet errors, margin, and signal strength.

This command is not valid for devices that are not attached to a power line channel.

* – Refresh Memory

Performs a series of Refresh Memory network management commands on the device. Use this command to refresh the local non-volatile memory (on-chip EEPROM or external EEPROM or flash memory).

< – Redirect Input from a File

Runs a command script. A command script is a text file that contains NodeUtil commands. A script file can use have any file extension, but a common extension is "**.scr**".

Commands that do not require a carriage return must be on their own line. Input that requires a carriage return must be on its own line. Network variable data is displayed (and entered) in hexadecimal.

You can run a script repeatedly, as needed.

See *Using a Script File with NodeUtil* for more information.

> – Redirect Output to a File

Redirects all NodeUtil output to a file. You are prompted to specify the file name. The default file name extension is **.log**. If the file already exists, you are asked if you want to **(O)verwrite** the file from the beginning, or **(A)ppend** output to the end of the file.

To stop redirection, enter a null file name.

This command is useful for saving the output of any of the NodeUtil commands, and for creating script files. See *Using a Script File with NodeUtil* for more information.

! –IP Addresses

Displays the device's LON/IP and host IP addresses. You can also use this command to set the IP address and enable or disable DHCP.

The following columns are displayed:

- **LonTalk-IP Addr** – The LON/IP address. This address is the IP representation of the domain index 0 entry.
- **Host IP Addr** – The host IP address. Typically either blank (for devices that do not support mapped IP addressing) or the same as the LON/IP address. However, if the device is currently using mapped IP addressing instead of LonTalk derived IP addressing, this field will indicate the actual IP address used to communicate with the device..
- **DHCP** – Determines whether DHCP is used for address assignment
- **DomainID** – The LonTalk domain Id corresponding to the LON/IP address.
- **Subnet** – The LonTalk subnet ID corresponding to the LON/IP address.
- **Node** – The LonTalk node ID corresponding to the LON/IP address.

Using a Script File with NodeUtil

You can create script files to automate command entry for NodeUtil. You can use script files for unit testing, for system testing, and for manufacturing testers. If you know the commands (and command responses) that your script requires, you can create the script using any text or programming editor that can produce standard ASCII text files. You can also capture the output from running NodeUtil to a specified file by using the **–O** parameter (see *Command-Line Parameters*). You can then edit the output file (to remove text and output that are not commands) and use it with the **–I** parameter.

You can also use the **<** command to redirect input from a file for either main menu commands or device menu commands. You cannot use the < command within a script.

Commands that do not require a carriage return must be on their own line and input that requires a carriage return must be on its own line. Lines that start with a tab character are treated as comments by NodeUtil during scripting. Network variable and other data is entered in hexadecimal format.

A script may contain comments. Comments are lines starting with '**#**' and are ignored.

A script starting with "**#!**" is one that provides special instructions to NodeUtil. Following the "**#!**", you can specify either:

- **"count=<number>"** – Specifies that the script is run <number> times, where -1 means infinite. The default is one iteration.
- **"wait=<seconds>"** – Specifies that the script waits <seconds> seconds between runs. The default is zero seconds.

Normally you would be prompted for these inputs but the prompting is bypassed if you specify the instructions here.

For example, the following script would invoke a query status ('S') command 100 times with a 2 second pause between each invocation.

```
#! count=100 wait=2
S
N
```

Example 1

The following script adds a device based on a Unique ID, goes to that device, polls its first network variable, and updates its second network variable with a SNVT_switch value of on 100%. Run this script from the main NodeUtil menu.

```
A
9000011E28061102
G
1
P
1
U
2
C8 00
```

Example 2

The following script defines the domain and authentication key for a device that will be discovered and managed by a SmartServer IoT. The script also places the device in the configured, online state and enables network management authentication. Run this script from the NodeUtil device menu for the device.

Step 1. Set domain 0 to unused with OMA and set the first half of the 96-bit key.

```
d
0
y
u
112233445566
y
y
```

Step 2. Set domain 1 to the discovery clone domain and set the second half of the 96-bit key.

```
d
1
y
6
7a3340f1bcd2
1
1
y
665544332211
y
```

Step 3. Set the channel ID to 0 so that the SmartServer can discover this device.

```
q
q
y
6e
02000002040000
```

Step 4. Set the device to the configured state and online

```
m
s
c
m
n
```

Step 5. Turn on network management authentication.

```
q
q
y
6e
0200180104E8
```

Step 6. Exit to force user to issue another "g" command (to enter the key to talk to the device)

```
e
```

This script uses the "**112233445566**" string as the first half of the 96-bit key and the "**665544332211**" string as second half of the 96-bit key. Using a key of "**112233445566665544332211**" is better than not using authentication, but it is a weak key.

Chapter 4 – Using NodeLoad

Starting NodeLoad

To start NodeLoad, follow these steps:

1. Open a Windows Command Prompt. To open the Command Prompt, click the Search box in the Windows taskbar and enter **cmd**. Then, click the **Command Prompt** result.

2. If you installed NodeLoad in a directory that is not in your Windows path, use the **cd** (change directory) command set the current directory to the directory that contains the **NodeLoad.exe** file.

3. Enter **NodeUtil**, with appropriate command-line parameters (as described in the next section, *NodeLoad Command-Line Parameters*), to start NodeLoad. Alternatively, you can create a Windows shortcut to launch NodeLoad with your preferred command-line parameters.

If NodeLoad cannot open the network driver, NodeLoad displays an error message and exits. In this case, check that the network driver is loaded correctly, that you have OpenLDV 5 or later installed, and that your network interface hardware is correctly configured.

NodeLoad Command-Line Parameters

To load an application into one or more devices using NodeLoad, start NodeLoad from the Windows command prompt and specify the target with the following command-line parameters:

–B

Directs NodeLoad to use only the OpenLDV network driver, and bypass the IP-852 network interface driver (formerly called VNI), if installed. In this mode, other host applications cannot share a single network interface.

This option need not be specified because **–B** is a default parameter.

–C

Suppress the device initialization prompt after an application download is finished and leave the device unconfigured.

–D<*interface_name*>

Specifies the name of the network interface to use. This parameter is required if your network interface is not named **LON1**, but is optional if you use **LON1**. If you specify an invalid network interface name, NodeUtil displays an error message. The network interface name is not case sensitive.

To specify a SmartServer IoT or other edge server with an OpenLDV compatible RNI, use **x.**<*Profile*>**.**<*SmartServer or i.LON name*>. For example, **-Dx.Default.SmartServerAtHQ** specifies the **Default** xDriver profile, with the **SmartServerAtHQ** edge server.

–O<*output_file*>

Specifies that NodeLoad output be copied to the specified file. You can use this parameter to create a script file that you can use with the **–I** parameter. See *Using a Script File with NodeUtil* for more information.

–V

Specifies that NodeLoad start up in verbose mode. In this mode, NodeLoad displays all network interface messages in detail.

–A

Selects application mode. In this mode, NodeUtil does not configure the network interface, and does not accept service pin messages. That is, NodeUtil treats the network interface as an application device.

If you do not specify this parameter, NodeUtil assigns the network interface to a random subnet address in the zero-length domain with a node ID of 126 and sets the network interface to the configured state.

–B

Directs NodeUtil to use only the OpenLDV network driver, and bypass the IP-852 network interface driver (formerly called VNI), if installed. In this mode, other host applications cannot share a single network interface.

This option need not be specified because **–B** is a default parameter.

-E<*Windows_handle*>

Redirect the NodeLoad output string to another Windows application. The application uses the **WM_COPYDATA** message to send the output string to another application.   The receiving application can then access the output string from the **COPYDATASTRUCT** structure sent by the NodeLoad application. See the **WM_COPYDATA** documentation at msdn.microsoft.com for more information.

–H

Prevents NodeLoad from changing the configuration of the network interface if the network interface is configured.

–I<*input_file*>

Specifies that input to NodeUtil is an input text file instead of the keyboard. See *Using a Script File with NodeUtil* for more information.

–J

Prevents NodeUtil from querying self-identification (SI) data from a host-based device, such as a device using a network interface with an SMIP or ShortStack Micro Server. By using this parameter, you can query a host-based device that does not have an attached host processor.

–M

Specifies that an SMIP or ShortStack device is being loaded (see *Y – Download Neuron Executable*). If you plan to load an SMIP or ShortStack device that is not currently attached to its host processor, use the **–J** parameter in addition to the **–M** parameter.

–O<*output_file*>

Specifies that output from the NodeUtil Node Utility be copied to the specified file. You can use this parameter to create a script file that you can use with the **–I** parameter. See *Using a Script File with NodeUtil* for more information.

–S

Specifies slower transaction timers, which are suitable for use with PL-20 power line channels. By default, NodeUtil uses transaction timers that are suitable for FT channels.

–V

Specifies that NodeUtil start up in verbose mode. In this mode, NodeUtil displays all network interface messages in detail.

-Z<domain_index>

Specify 0-, 1-, 3-, or 6-byte domain ID.   Leave the argument empty for the zero-length domain. Examples:

-Z   -> zero-length domain

-Z14

-Z020812

-Z140923100523

-#

Force a larger buffer for faster downloads. NodeLoad will ignore this option if it is not safe.

-?

Displays the command format, including a brief description of the command-line parameters, and exits.

Example NodeLoad Commands

Following are three example NodeLoad commands:

- Open the LON1 interface, find a LON or LON/IP device, download an application file named **test.nei**, and set the transceiver parameters:

```
nodeload –Ltest.nei –Xtest.nei
```

- Open the LON2 interface, wait 3 seconds for a service message from a device, and set the transceiver parameters for the device:

```
nodeload –DLON2 –W3 –Xtest.nei
```

- Open the LON1 interface, specify the unique ID of a device to be loaded, and download the application:

```
nodeload –N050012AC1D00 –Ltest.nei
```

NodeLoad Result Strings and Codes

When NodeLoad completes, it returns a result string by the calling application to determine success or failure.

Errors that typically occur during setup, such as invalid network interface, invalid file format, invalid arguments, etcetera, will be printed before the final "Nodeload Result" message in order to aid in debugging during setup. The Nodeload Result message is formatted to make parsing relatively easy for calling an application on the production assembly line.

Passing Result

This result message indicates passing results and returns the NID of the loaded node.

```
Nodeload Result: Success; NID=050001020300
```

Failing Result

This **result** message indicates failing results and includes a code that specifies the failure.

```
Nodeload Result: Failed (error - #x)
```

Error Code Reference

Error codes for the NodeLoad utility (`error - #x`) are defined in the following table:

Error Code	Error Title	Description
1	No nodes found	The "`-Q`" switch was specified and no devices were found during the query.
2	Service pin timeout	The "`-W`" switch was specified and no service pin was received.
3	Comm failure	A communication failure occurred either on initial contact with the "`-N`" switch or subsequent contact in any mode.
4	File error	The NEI file was not found or was corrupted.
5	Reset failure	Nodeload could not successfully cause the node to reset. This could indicate an excessively long reset clause in the application.
6	Buffer failure	Nodeload could not successfully configure the buffers on the network interface.
7	State failure	The device could not be successfully put into the correct state at the end of the load.
8	Checksum failure	The device failed to set its checksum after the load.
9	Version mismatch	The device firmware version didn't match the firmware version that the NEI/NDL file was linked for.
10	Model mismatch	The device model number (i.e., part type) didn't match the model number that the NEI/NDL file was linked for.
11	Interface error	The network interface (e.g., U10) could not be communicated with.
12	Response error	The device reported a failing response. This could be due to an authentication key mismatch or a failure to write non-volatile memory.