DM-20/21 Device Manager User's Guide

Version 2.0

ECHELON®

Corporation

078-0173-01B

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.

Model Number 39510

Printed in the United States of America.

Echelon Corporation
4015 Miranda Avenue
Palo Alto, CA 94304, USA

Contents

1

DM-20/21 Overview

This chapter provides an introduction and overview for the DM-20 and DM-21 Device Manager embedded network management devices.

Introduction

The DM-20 and DM-21 Device Managers are embedded network management devices. The DM-20 (Model 43201) is designed for embedded applications in which it is mounted on a motherboard. The DM-21 (Model 43202) is a packaged assembly designed for installation in a suitable socket.

The DM-20/21 Device Managers provide the following features:

- Automatic installation, fault detection, and device replacement in a LONWORKS® network with up to 128 devices and one router per Device Manager;
- Autonomous operation without the use of a local PC once it has been commissioned with the LonMaker™ for Windows® Integration Tool;
- Records system activity using an internal Flash-based event log: no additional batteries are required;
- Front panel jack for simple network access;
- Cuts installation and service costs with two-piece design;
- Support for LONMARK 3.1 objects and configuration properties;
- Network variable and message tag support;
- Safety and regulatory agency approval (U.L. Listed, CE Mark, FCC).

Related Documentation

The following Echelon documents are suggested reading:

- *LONWORKS PCLTA-10 PC LonTalk Adapter User's Guide* (078-0159-01)
- *LONWORKS PCLTA-20 PCI LonTalk Adapter User's Guide* (078-0179-01)
- *LONWORKS PCC-10 PC Card User's Guide* (078-0155-01)
- *LonPoint Module Hardware and Installation Guide* (078-0167-01)
- *LonPoint Application and Plug-In Guide* (078-0166-01)
- *LonMaker for Windows User's Guide* (078-0168-01)
- *Junction Box and Wiring Guidelines for Twisted Pair LONWORKS Networks* (engineering bulletin 005-0023-02)
- *LONWORKS FTT-10A Free Topology Transceiver User's Guide* (078-0156-01)
- *LONMARK 3.1 Application Layer Guidelines*

A glossary of LONWORKS terms can be found at the end of this manual.

Device Manager

A Device Manager is an embedded network management device that can install and replace nodes on a LONWORKS network. It differs from conventional LONWORKS network management tools in that it does not make any network management decisions itself. Rather, it operates from a database created in advance using the LonMaker for Windows Integration Tool, which provides a rich set of design and configuration functions. What is left behind is a low cost Device Manager which manages the network based on instructions loaded from the tool.

Use of a Device Manager is ideal when it is not possible or desirable to leave a management PC on site, or where service personnel are not equipped with portable management PCs.

General Operation

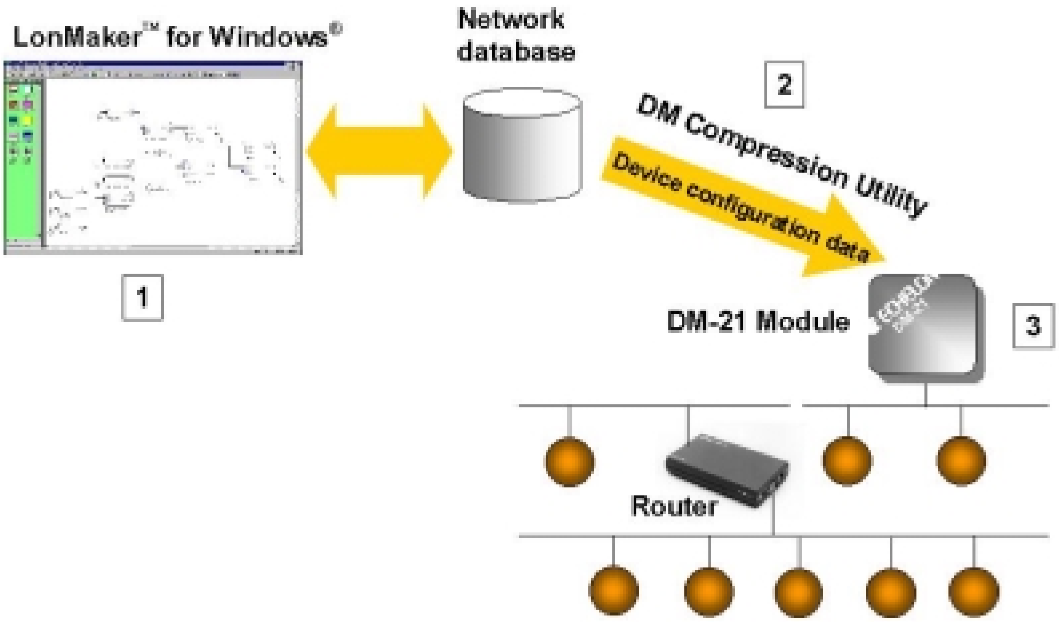


Figure 1.1 – Overview of Device Manager Installation

The diagram above illustrates the three steps necessary to use a Device Manager to manage a network:

1. The network is designed using the LonMaker for Windows Integration Tool;
2. A Loading Wizard is used to create a “Managed Device List” from the LONWORKS database created by the LonMaker tool. The Managed Device List is then loaded into the Device Manager;
3. The Device Manager is connected to the control network.

Once attached, the Device Manager will automatically locate and commission all the devices defined in the LonMaker drawing. No further user intervention is required to install or replace devices.

After commissioning, the devices on the network will be periodically queried (pinged) by the Device Manager. If a device fails to respond, an entry will be made in the Device Manager’s internal event log, and, optionally, a message can be sent to a monitoring device such as the LNS Alarm Dialer available at www.echelon.com/products/DM. The Device Manager's internal event log can be retrieved using a service tool to assist with diagnosing problems.

Periodically, the device manager will search the network for new devices. Any newly found or replacement devices will be integrated into the network as soon as they are detected. The Device Manager will automatically establish or, in the case of a replacement, re-establish all connections with other devices.

Allowable Topologies

A Device Manager can support single or dual channel networks. In a single channel network, the channel must be a TP/FT-10 free topology channel. In a dual channel network, the Device Manager must be attached to a TP/FT-10 channel, however the second channel (on the far side of the router) may be of any type of LONWORKS channel. The use of a single router is supported in this topology.

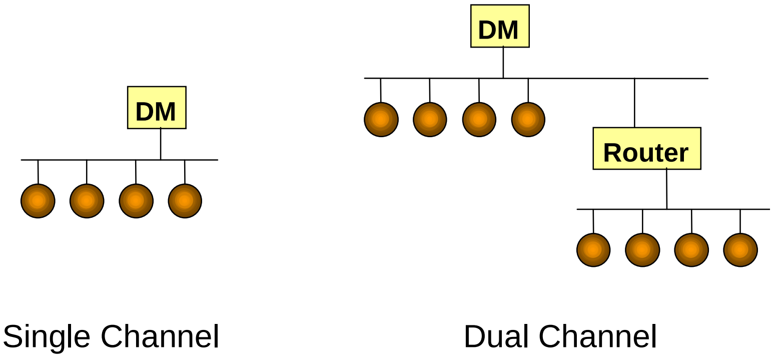


Figure 1.2 – Using a Device Manager in Single and Dual Channel networks

Using Subsystems

For large systems, it is possible to use multiple Device Managers to manage the network. For example a train might use one Device Manager per car, or a building might use one Device Manager per floor. Although several Device Managers are working together to manage a larger system, the same topology rules apply to each device manager.

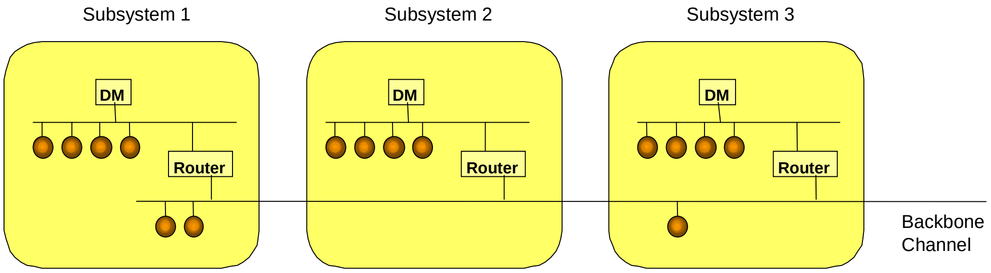


Figure 1.3 – Multiple Device Managers working together - Legal Topology

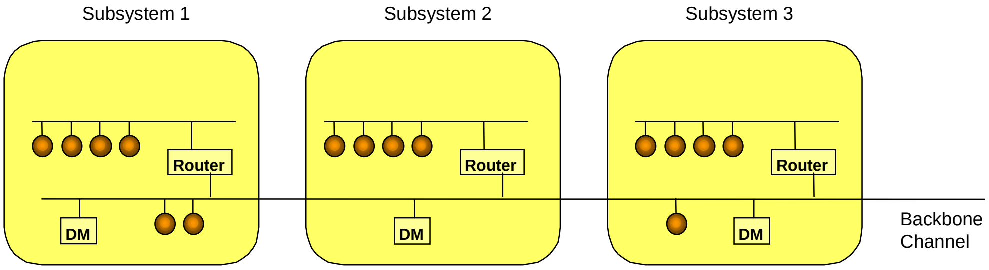
Figure 1.3 illustrates the use of three Device Managers working together to manage a large system. When a LonMaker drawing is made, all of the devices in the system are placed on the drawing (including the three routers and the three

Device Managers.) The drawing is divided into three subsystems where each subsystem contains a Device Manager, a router, and several devices. It is recommended that *all three* subsystems be loaded into *all* Device Managers using the Device Manager Loading Wizard (see Chapter 2). At runtime, each Device Manager is given a number (via its 4-bit subsystem selection input) representing which subsystem it will manage. The Device Manager will manage only its selected subsystem and will ignore any other devices it detects on the network. In essence, the Device Managers act as partners, each managing a piece of the network.

The reason for loading all three subsystems into all Device Managers is to permit Device Managers to be interchanged during routine maintenance or service. When a Device Manager is moved to a new location it will read a different value from the 4-bit input and automatically reconfigure itself to manage the specified subsystem.

In cases where the subsystem is so large that the resulting managed device lists cannot be loaded into every Device Manager due to memory limitations, only a single subsystem should be loaded into each Device Manager. This permits larger networks, but it will not be possible to interchange Device Managers; each Device Manager will work with only one managed device list. In this application there is no need to supply a 4-bit input to the Device Manager as it will be loaded with only a single subsystem.

Device Managers must be located on a network such that they communicate with a single router. For example, the topology shown in figure 1.4 is prohibited because the Device Managers cannot distinguish between local routers or the devices connected to the other channels.



1.4 – Prohibited Topology: Ambiguous Routers

Managing Variants Using Multiple Subsystems

LonMaker drawings [network designs] are generally organized into subsystems. Up to 16 subsystems may be loaded into a device manager. When reset, the device manager will select which subsystem to use based on an external 4-bit input. This allows you to load a single device manager with 16 network configurations.

The ability to select one of several possible network configurations is useful for designs where a single device manager might be placed in one of several possible network installations. For example, a modular photocopier might come in several configurations:

Photocopier only	10 nodes total

Photocopier + collator	13 nodes total
Photocopier + collator + stapler	15 nodes total
Photocopier + stapler	12 nodes total

For this example, the photocopier contains 10 nodes, the collator contains 3 nodes and the stapler contains 2 nodes.

To use a single device manager to handle any of the above configurations, follow these steps:

1. Create a LonMaker drawing that contains 4 subsystems: one subsystem that contains devices for just the photocopier, one subsystem that contains devices for the photocopier and collator, one subsystem that contains devices for the photocopier, collator and stapler, and finally, one subsystem that contains devices for the photocopier and stapler. Include a device manager in each subsystem.

2. Use the device manager plug-in to set *DM Pin Usage* to `Use IO pins as DB number`. (See Chapter 5 for details on using the device manager plug-in.)

3. Save your LonMaker design, and exit LonMaker.

4. Start the device manager Loading Wizard. See Chapter 5 for more information on the device manager Loading Wizard. Use the Loading Wizard's step-by-step process to select all four subsystems. Compress the subsystems. This creates a file known as the *Managed Device List*.

5. Continue the Loading Wizard's step-by-step process to load the Managed Device List into the device manager.

6. At runtime, supply a binary number to the device manager's subsystem selection pins. (See *Managed Device List Selection* in Chapter 4 for complete details.)

7. Based on this binary number, the device manager will reference one of the four subsystems when managing the devices it finds on the network. The device manager will manage only the devices defined in the subsystem that has been selected by the 4-bit input.

Note:

> If the binary number changes and the device manager is reset, it will apply a new subsystem and manage *that* set of devices. This allows the device manager to be attached to any of the four photocopier configurations discussed above.
>
> The devices on the network can be different in each case. The machine that contains 15 devices is not necessarily a superset of the machine that contains 10 devices.

A subsystem may contain any number of devices. The total number of devices is limited only by available Device Manager memory, and will vary based on the complexity of each device under management. As a benchmark, a device manager can manage up to 128 LonPoint® modules. LonPoint modules are used as a benchmark because they are considerably more complicated than the average small device. LonPoint modules use a full complement of network variables, LONMARK Objects, and configuration properties. Each LonPoint module requires about 2K of storage in the device manager.

Device Identification

In order for a Device Manager to distinguish between two different devices, there must be some means provided whereby each device can be uniquely identified. This is accomplished by using the device's Program ID and optionally its location string.

The Program ID is a hexadecimal string uniquely identifying the device's application (see the *LONMARK Application Layer Interoperability Guidelines* for more information).

The location string is a 6 byte string in the Neuron Chip's config database. Devices set their location string based on input from a keyed connector, DIP switch, rotary switches, smart card, or some other comparable identification mechanism. Devices that do not have a built in means of setting their location string may have their location string set manually using a tool designed to set location strings. See `www.echelon.com/products/dm` for an example of a PC application that can be used to set a device's location string. The location string may be left blank. If so, the Device Manager will match on Program ID only. If the location string is used, the location string in the Device Manager's Managed Device List must match the location sting in the device in order for the device to be discovered.

The Device Manager must be able to uniquely identify each device in the system by Program ID and, optionally, its location string.

Using #pragma run_unconfigured with the Device Manager

In order for a device to automatically set its own location string (i.e., via a keyed connector or DIP switch), the application within that device must be running. Typically, devices that are in the unconfigured state will not run any applications until a network management tool configures the device. This would create a problem for a system using Device Mangers when more than one device has the same Program ID, i.e., the same application.

This situation can be rectified by permitting devices to run while they are unconfigured. This is accomplished by including the following statement in the device's Neuron C program:

```
#pragma run_unconfigured
```

⚠

A device that uses the run_unconfigured pragma will run even when it is in the hard-offline state. If such a device attempts to transmit, it may do so using a suspect configuration state (hard-offline is used by LNS to prevent a device from transmitting while being reconfigured in power cycles and corrupts the device's memory during the process).

The application programmer must follow the convention that a node will not transmit while it is in the hard-offline state. This means that the application must trap the state change and verify that the node is not hard offline prior to any network transmission.

The run_unconfigured pragma is incompatible with most self-installation schemes, as it may cause a state semaphore error in the Neuron Chip error log, eventually causing the node to go unconfigured. This problem should not present itself in a system managed by a Device Manager unless the designer implements a self-installation feature above and beyond the functions performed by the Device Manager.

Device Manager Runtime Behavior

Once a LonMaker database has been compressed and loaded into the Device Manager and the Device Manager is reset, it will immediately begin managing the devices it discovers on the network as follows:

1. The Device Manager will self-install itself and select the subsystem it will manage based on the 4-bit input.
2. The Device Manager will go into “discovery mode” and try to locate each device defined in the subsystem. In multi-channel systems, the Device Manager will always try to locate the router first. If a router was defined in the subsystem, but is not physically present on the network, the Device Manager will not look for the other devices until the router has been discovered.
3. Devices will be commissioned by the Device Manager as they are discovered. From the perspective of the device, the commissioning process is indistinguishable from being commissioned by an LNS tool such as LonMaker for Windows. In essence, the Device Manager acts as a proxy for an LNS installation tool.
4. Once a device has been found and commissioned, the Device Manager will query (“ping”) the device to ensure the device is still operating. If a device fails to respond to a ping, the Device Manager reenters discovery mode and attempts replacement for the missing device (matching on Program ID and location)

   If devices are to be added to the network after the Device Manager has been commissioned, the Device Manager will continue to search for the devices that it has not yet discovered but which are expected to be present on the network based on the loaded database.

Discovery

A device is considered to be *discovered* after the Device Manager receives the device’s Neuron ID (either via the discovery process or service pin message) and successfully reads the device’s Program ID and location string. Upon discovery, an association is established between the discovered device’s Neuron ID and a record in the Device Manager’s Managed Device List. The discovery is recorded in the Device Manager’s internal event log and a value of `DMREP_DEVICE_DISCOVERED` is sent on the DM_report network variable.

If a newly discovered device matches an existing device that has already been discovered, the existing device will be pinged to see if it is still operating and has the correct location string. If it does, the newly discovered device will be ignored. If not, then the existing association of logical and physical device will be removed and the newly found device will be commissioned and added to the Managed Device List.

A device with an unknown Program ID will not be discovered; if found on the network, the device will be ignored, and no report made. However, devices with a known Program ID and an unknown location string will be discovered. If on a subsequent discovery operation the location string has changed to a known value (say due to configuration by a handheld tool), then the Device Manager will consider the device a new device and will discover it.

Automatic Commissioning

Automatic commissioning takes place if the device's Program ID and location string are known and the *auto-commission* bit in the Managed Device List is set. By default, all devices have their auto-commission bit set. **Echelon does not support alternative uses of this bit.**

Pinging Devices

Devices that have been commissioned are periodically pinged. Pinging consists of sending a query status diagnostic message and reading the device's location string. The Device Manager will then validate the device state, subnet/node address, and location string as follows:

- If the device fails to respond to a ping after the defined number of ping retries, the Neuron ID in the Managed Device List will be deleted, breaking the association between that device and the corresponding record in the Managed Device List. The DMEVNT_MISSING event will be logged in the Device Manager's internal event log and this value will be sent on the DM_Event network variable if its Async bit is set.
- If the device fails to respond to pinging and another matching device is discovered, the Neuron ID of the old device will be immediately deleted from the Managed Device List. The DMEVNT_MISSING event will be logged in the Device Manager's internal event log and this value will be sent on the DM_Event network variable if its Async bit is set. This is the same as above, except that a single failure causes this to occur since there is already a possible replacement.
- If the device's location string does not match the one in the Managed Device List, the Neuron ID will be immediately removed from the Managed Device List. The DMEVNT_MISSING event will be logged in the Device Manager's internal event log and this value will be sent on the DM_Event network variable if its Async bit is set. The physical device corresponding to the Neuron ID will be treated as though it is newly discovered.
- If the device's location string matches the location string in the Managed Device List, and the state is incorrect (but not applicationless), a DMEVNT_CONFIG_ERROR event will be logged in the Device Manager's internal event log and this value will be sent on the DM_Event network variable if its Async bit is set. The device will be re-commissioned.

When Devices Cannot Be Automatically Discovered

Devices cannot be discovered by the Device Manager if any of the following conditions are true:

- The device is already configured on a different domain from the domain specified for the Device Manager. In this case, the Device Manager can only discover the device if the device broadcasts a service pin message;
- The device is configured on the correct domain but has the wrong subnet **and** is on the far side of the router. In this case, the Device Manager cannot discover the device, even if the device broadcasts a service pin message;
- The router on that channel has not yet been discovered. Devices are not discovered until the router on that channel is discovered first;
- The device's location string does not match what was defined in the LonMaker drawing.

To avoid these situations, make sure that all devices on a network managed by a Device Manager module are unconfigured at installation time, and that any routers that are supposed to be present are installed.

2

Description of the DM-21 Base Plates

This chapter provides a description of the LonPoint Type 1 and Type 1D DIN Base Plate mounting enclosures for the DM-21 Device Manager (the DM-20 Device Manager is shipped as a printed circuit board and is designed for mounting on a motherboard).

The DM-21 Base Plate

The DM-21 module is installed in a LonPoint Type 1 Base Plate, revision 1A. Wiring connections to the DM-21 module are presented in Chapter 4.

The rear of the PCB assembly includes a polarized, shrouded, 14-pin interface connector that connects to a mating socket in a Type 1 Base Plate (for mounting on to 4" square electrical boxes) or Type 1D DIN Base Plate (for mounting to a 35mm DIN rail or a wall).

Network, power, and I/O wiring are connected to screw terminals located on a Type 1 Base Plate (figure 2.1) or Type 1D DIN Base Plate (figure 2.2), into which the DM-21 module is plugged; no wiring is ever connected directly to an Interface Module. A writing space is provided on the back of the Type 1 Base Plate for recording the date of installation or other pertinent information. Installers should take care to use a fine-point (smudge-free) pen. In Type 1/1D Base Plates, the associated module is affixed to the Base Plate by means of two 8-32, 3/8” screws (Echelon 205-0130-01) that are provided with each module.

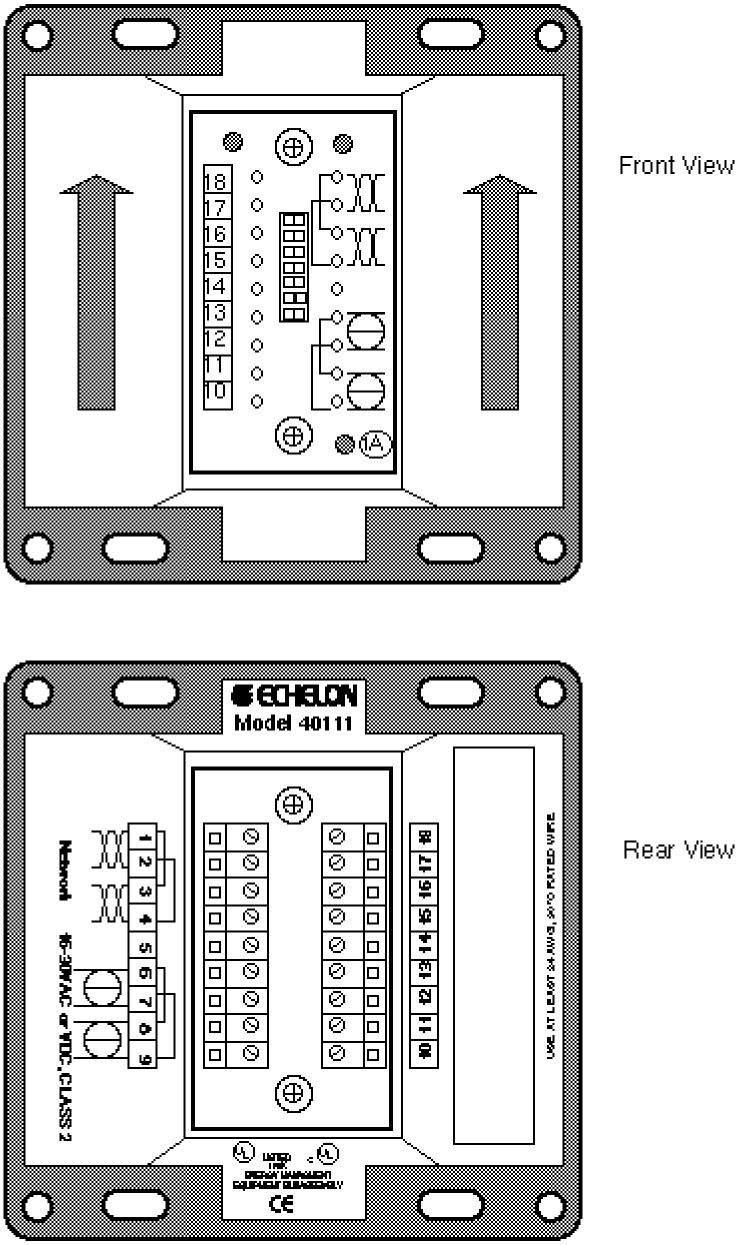


Figure 2.1 Type 1 Base Plate

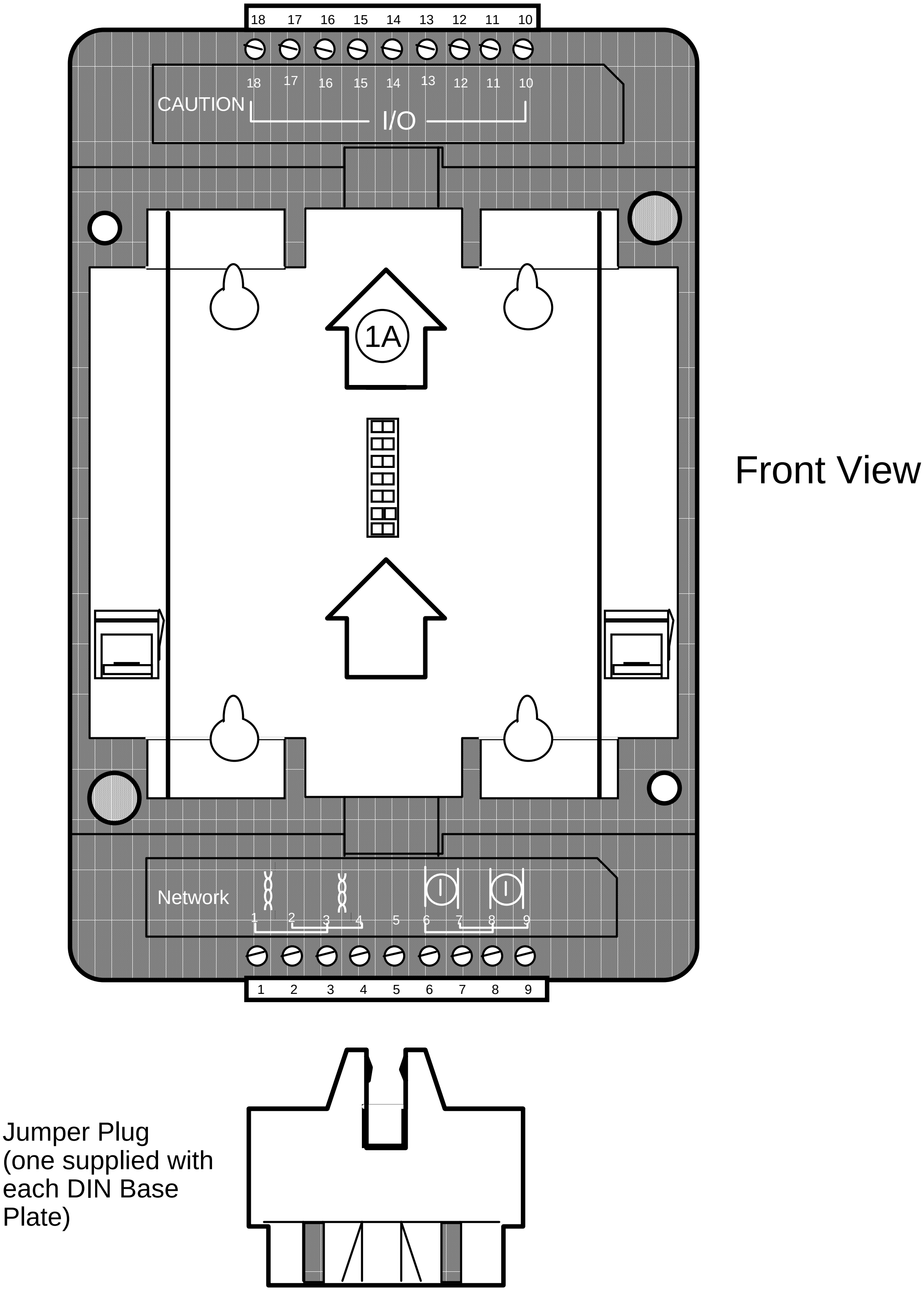


Figure 2.2 Type 1D Base Plate

The Type 1 Base Plate is mounted to a suitable US 4” square (figure 2.3), 2” deep (10.16cm x 5.08cm) electrical box, US double gang electrical box (figure 2.4), or an IP-65 (NEMA 4) enclosure.

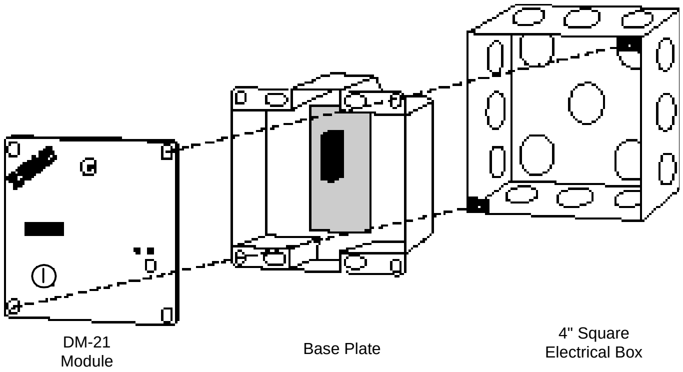


Figure 2.3 DM-21 Module 4” Square (2” Deep) Electrical Box Mounting Configuration

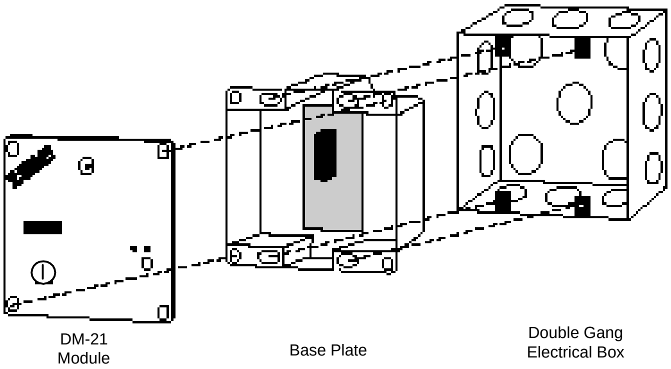


Figure 2.4 DM-21 Module Double Gang (2” Deep) Electrical Box Mounting Configuration

The Type 1D DIN Base Plate may be mounted to either a 35mm DIN rail or to a wall panel (figure 2.5). A jumper plug supplied with the Base Plate permits the power and network connections of several Type 1D DIN Base Plates to be interconnected without additional wiring.

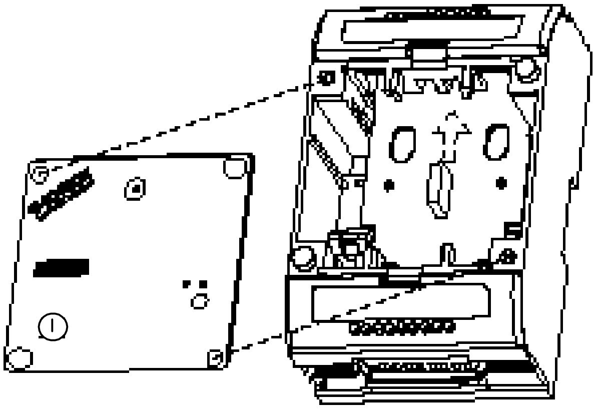


Figure 2.5 DM-21 Type 1D Base Plate Mounting Configuration

3

Network Cabling and Base Plate Installation

This chapter provides information about network, power, and input/output cabling for the DM-21 Module.

Network Cabling - TP/FT-10 Channel

The Device Manager Modules use Echelon's FTT-10A Free Topology Transceiver for network communications. This transceiver operates at 78 kilobits per second and is designed to support free topology wiring; it will accommodate bus, star, loop, or any combination of these cabling topologies using a twisted pair cable. By eliminating restrictions on the cabling topology, the installer is free to locate devices anywhere on the network cabling provided only that the maximum cabling distance limitations are observed. This capability simplifies system installation and makes it easy to add nodes should the network need to be expanded. Figures 3.1 through 3.5 present five typical network topologies.

In free topology cabling it is necessary to use one Echelon Model 44100 terminator. In bus topology cabling it is necessary to use two Echelon Model 44101 terminators. Terminators are required for proper network operation and must not be omitted.

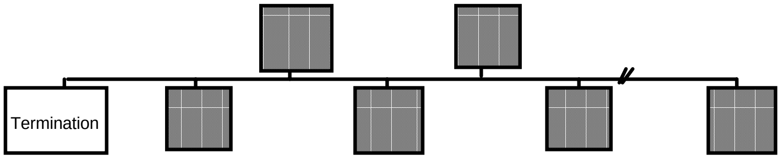


Figure 3.1 Singly Terminated Bus Topology (Model 44100 Terminator)

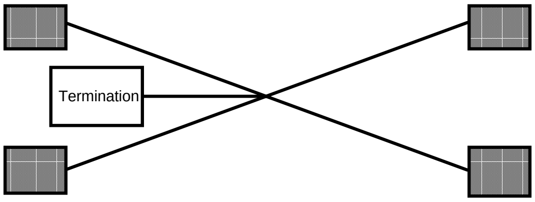


Figure 3.2 Star Topology (Model 44100 Terminator)

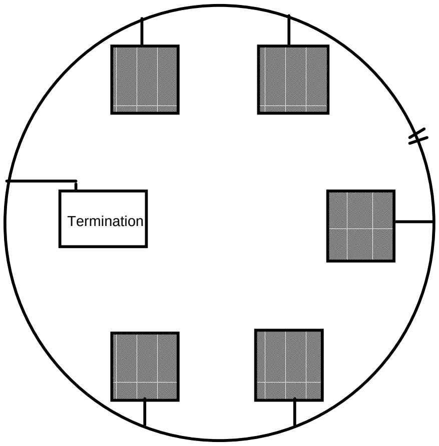


Figure 3.3 Loop Topology (Model 44100 Terminator)

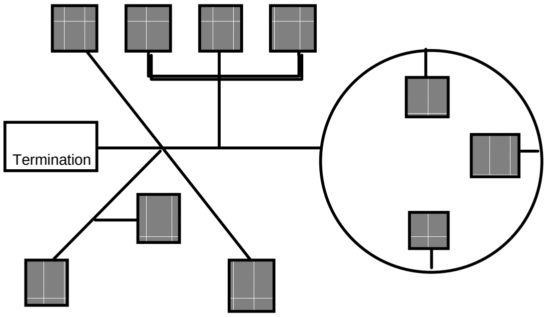


Figure 3.4 Mixed Topology (Model 44100 Terminator)

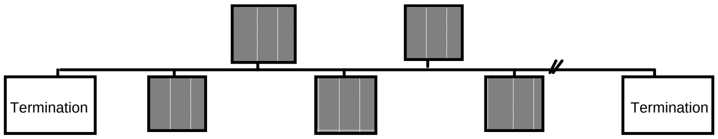


Figure 3.5 Doubly Terminated Bus Topology - Used for Very Long Cabling Distances
(Two Model 44101 Terminators)

A network consisting of devices using the FTT-10A transceiver is said to reside on a "TP/FT-10 channel." In some cases all of the devices will be connected to a single TP/FT-10 channel. A maximum of 64 devices (LonPoint modules, routers, PCLTA-10 and PCC-10 PC adapters, third-party LONMARK® devices) may be connected to any one channel. If more than 64 devices are to be used, or if it is necessary to add more cabling than is permitted on a single channel, then one or more model 42100 LPR-10 Routers (TP/FT-10 to TP/FT-10) would be placed in series with the network cabling.

See the *Junction Box and Wiring Guidelines for Twisted Pair LONWORKS Networks* (engineering bulletin 005-0023-02) and the *LONWORKS FTT-10A Free Topology Transceiver User's Guide* (078-0156-01), available at http://www.echelon.com/products/technical/manuals.asp, for more information about using the TP/FT-10 channel and FTT-10A transceiver.

I/O Cabling

The installer generally has wide latitude in the type of cabling selected for the I/O. It is good practice to ensure that all cabling is made from twisted pair wire, as this has the advantage of minimizing susceptibility to differential noise. Note when selecting the cabling that the Base Plate screw terminals accommodate wire gauges from 24AWG/0.5mm to 12AWG/2mm. In all cases, use at least 90°C rated wire.

Power Cabling

It is important to note that a maximum of 16 Amperes RMS at 24VAC RMS can be passed through the internal jumpers on the Power terminals of Type 1 Base Plates. This means that if power wiring is looped in and out of the power terminals of the Base Plates, the current load presented by all of the modules and any other devices powered by that circuit must be <16 Amperes, as shown in figure 3.6. If >16 Amperes must be supplied then the loop-through capability of the Base Plate power terminals must not be used and the power cabling should be run in parallel, as shown in figure 3.7. To calculate the load current, divide the total VA (sum of all devices powered by the power supply) by the power supply voltage. For example, if the total VA of all devices equals 100VA, and the power supply provides 24 VAC, then the load current is 4.16A.

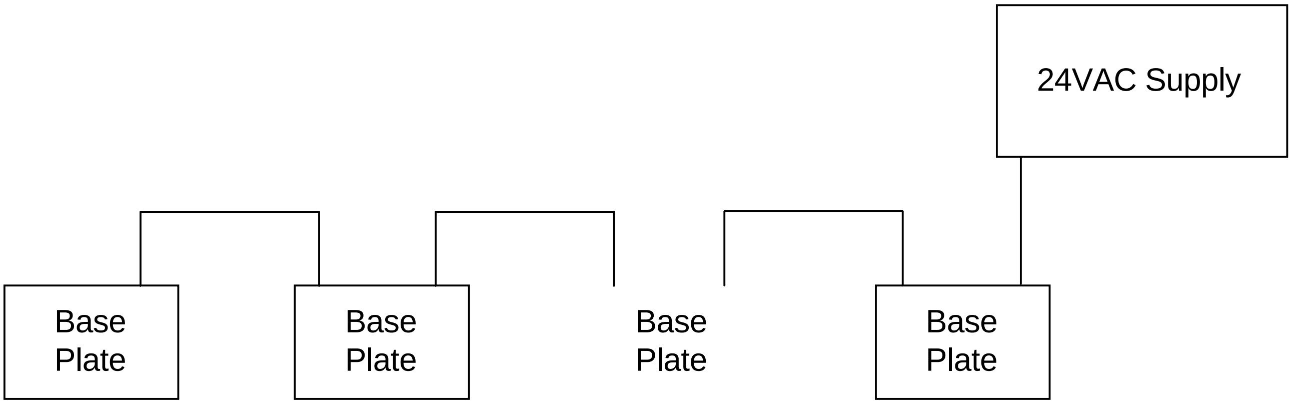


Figure 3.6 Looped-through Power Wiring - ≤16 Amperes at 24VAC

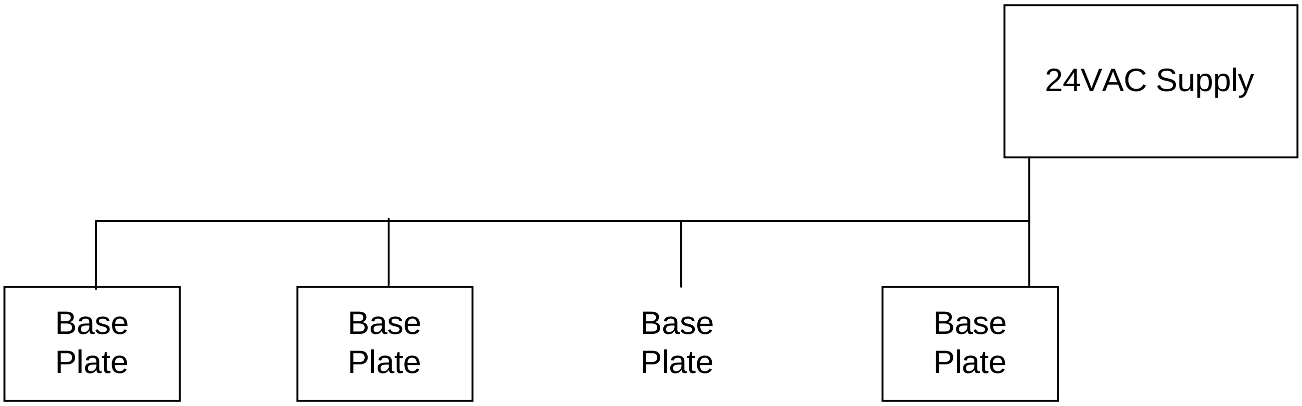


Figure 3.7 Parallel Power Wiring - >16 Amperes at 24VAC

As a rule of thumb, it is recommended that power and I/O cabling be separated by 18” (46cm) to prevent inadvertent pick-up of noise from the power circuit by sensitive analog I/O cabling. Power and network twisted pairs may be combined in a common cable if using an approved network cable.

The tables below present the wire gauge that should be used in order to deliver the specified power (total VA) across the distance shown. These tables only include wire gauges from 24AWG (0.5mm) to 12AWG (2mm) as these are the wire gauges supported by the Type 1 and Type 2 Base Plate screw terminals. For the supply cabling use at least 24AWG/0.5mm, 90° C rated wire.

To determine the total power that a power cable must carry, total the VA requirements for all of the devices that will be drawing power from that cable, including both modules and other devices.

To determine what gauge cable (in AWG or metric) will be required based on the total distance of the power cabling, refer to the tables 3.1 and 3.2 for cabling measured in AWG and in mm. If the wire gauge used is smaller than required, improper operation of one of more modules will result. All data in these tables assume an average wire temperature of +55°C.

Table 3.1 - Power Cabling Requirements in AWG

		Load (VA)											
Feet		10	15	20	25	30	40	55	75	100	130	180	200
35		22	22	22	22	22	22	22	22	22	22	22	22
40		22	22	22	22	22	22	22	22	22	22	22	20
55		22	22	22	22	22	22	22	22	22	22	20	20
65		22	22	22	22	22	22	22	22	22	20	20	18
70		22	22	22	22	22	22	22	22	22	20	18	18
75		22	22	22	22	22	22	22	22	22	20	18	18
90		22	22	22	22	22	22	22	22	20	20	18	18
100		22	22	22	22	22	22	22	22	20	18	18	16
105		22	22	22	22	22	22	22	20	20	18	18	16
115		22	22	22	22	22	22	22	20	20	18	16	16
135		22	22	22	22	22	22	22	20	18	18	16	16
145		22	22	22	22	22	22	20	20	18	18	16	16
150		22	22	22	22	22	22	20	20	18	16	16	16
155		22	22	22	22	22	22	20	20	18	16	16	14
165		22	22	22	22	22	22	20	18	18	16	16	14
185		22	22	22	22	22	22	20	18	18	16	14	14
215		22	22	22	22	22	20	20	18	16	16	14	14
230		22	22	22	22	22	20	18	18	16	16	14	14
235		22	22	22	22	22	20	18	18	16	14	14	14
250		22	22	22	22	22	20	18	18	16	14	14	12
C		22	22	22	22	20	20	18	16	16	14	14	12
295		22	22	22	22	20	20	18	16	16	14	12	12
300		22	22	22	22	20	18	18	16	16	14	12	12

Power Cabling Requirements in AWG (Continued)

		Load (VA)											
Feet		10	15	20	25	30	40	55	75	100	130	180	200
340		22	22	22	20	20	18	18	16	14	14	12	12
365		22	22	22	20	20	18	16	16	14	14	12	12
375		22	22	22	20	20	18	16	16	14	12	12	12
395		22	22	20	20	20	18	16	16	14	12	12	
400		22	22	20	20	18	18	16	16	14	12	12	
420		22	22	20	20	18	18	16	14	14	12	12	
470		22	22	20	20	18	18	16	14	14	12		
475		22	22	20	20	18	16	16	14	14	12		
500		22	22	20	18	18	16	16	14	12	12		
545		22	20	20	18	18	16	16	14	12	12		
580		22	20	20	18	18	16	14	14	12	12		
595		22	20	20	18	18	16	14	14	12			
630		22	20	18	18	18	16	14	14	12			
635		22	20	18	18	16	16	14	14	12			
750		22	20	18	18	16	16	14	12	12			
755		20	20	18	18	16	14	14	12	12			
795		20	20	18	16	16	14	14	12				
865		20	18	18	16	16	14	14	12				
945		20	18	18	16	16	14	12	12				
1000		20	18	16	16	16	14	12	12				
1010		20	18	16	16	14	14	12	12				
1190		20	18	16	16	14	14	12					
1200		18	18	16	16	14	12	12					
1260		18	18	16	14	14	12	12					
1375		18	16	16	14	14	12	12					
1500		18	16	16	14	14	12						
1590		18	16	14	14	14	12						
1645		18	16	14	14	12	12						

Table 3.3 - Power Cabling Requirements In mm

		Load (VA)											
Feet		10	15	20	25	30	40	55	75	100	130	180	200
23.0		0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
32.8		0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.65
36.1		0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.65	0.65
42.7		0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.65	0.80
45.9		0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.80	0.80
55.8		0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.65	0.80	0.80
59.1		0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.65	0.80	1.00
62.3		0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.65	0.80	0.80	1.00
75.5		0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.65	0.80	1.00	1.00
82.0		0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.80	0.80	1.00	1.00
88.6		0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.65	0.80	0.80	1.00	1.00
91.9		0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.65	0.80	1.00	1.00	1.00
101.7		0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.65	0.80	1.00	1.00	1.30
114.8		0.50	0.50	0.50	0.50	0.50	0.50	0.65	0.80	0.80	1.00	1.30	1.30
141.1		0.50	0.50	0.50	0.50	0.50	0.65	0.65	0.80	1.00	1.00	1.30	1.30
150.9		0.50	0.50	0.50	0.50	0.50	0.65	0.80	0.80	1.00	1.30	1.30	1.30
154.2		0.50	0.50	0.50	0.50	0.65	0.65	0.80	0.80	1.00	1.30	1.30	1.30
170.6		0.50	0.50	0.50	0.50	0.65	0.65	0.80	1.00	1.00	1.30	1.30	1.60
183.7		0.50	0.50	0.50	0.50	0.65	0.65	0.80	1.00	1.00	1.30	1.60	1.60
193.6		0.50	0.50	0.50	0.65	0.65	0.65	0.80	1.00	1.30	1.30	1.60	1.60
213.3		0.50	0.50	0.50	0.65	0.65	0.80	0.80	1.00	1.30	1.30	1.60	1.60
229.7		0.50	0.50	0.50	0.65	0.65	0.80	1.00	1.00	1.30	1.30	1.60	1.60
232.9		0.50	0.50	0.65	0.65	0.65	0.80	1.00	1.00	1.30	1.30	1.60	1.60
239.5		0.50	0.50	0.65	0.65	0.65	0.80	1.00	1.00	1.30	1.30	1.60	2.00
242.8		0.50	0.50	0.65	0.65	0.65	0.80	1.00	1.00	1.30	1.60	1.60	2.00
259.2		0.50	0.50	0.65	0.65	0.65	0.80	1.00	1.30	1.30	1.60	1.60	2.00
292.0		0.50	0.50	0.65	0.65	0.80	0.80	1.00	1.30	1.30	1.60	2.00	2.00
305.1		0.50	0.50	0.65	0.65	0.80	1.00	1.00	1.30	1.30	1.60	2.00	2.00
308.4		0.50	0.65	0.65	0.65	0.80	1.00	1.00	1.30	1.30	1.60	2.00	2.00
334.6		0.50	0.65	0.65	0.80	0.80	1.00	1.00	1.30	1.60	1.60	2.00	2.00
360.9		0.50	0.65	0.65	0.80	0.80	1.00	1.30	1.30	1.60	1.60	2.00	2.00

Power Cabling Requirements In mm (Continued)

		Load (VA)											
Feet		10	15	20	25	30	40	55	75	100	130	180	200
387.1		0.50	0.65	0.65	0.80	0.80	1.00	1.30	1.30	1.60	2.00	2.00	
390.4		0.50	0.65	0.80	0.80	0.80	1.00	1.30	1.30	1.60	2.00	2.00	
400.3		0.50	0.65	0.80	0.80	1.00	1.00	1.30	1.30	1.60	2.00	2.00	
413.4		0.50	0.65	0.80	0.80	1.00	1.00	1.30	1.30	1.60	2.00		
459.3		0.50	0.65	0.80	0.80	1.00	1.00	1.30	1.60	1.60	2.00		
469.2		0.65	0.65	0.80	0.80	1.00	1.30	1.30	1.60	1.60	2.00		
518.4		0.65	0.65	0.80	1.00	1.00	1.30	1.30	1.60	2.00	2.00		
554.5		0.65	0.80	0.80	1.00	1.00	1.30	1.30	1.60	2.00	2.00		
564.3		0.65	0.80	0.80	1.00	1.00	1.30	1.30	1.60	2.00			
587.3		0.65	0.80	0.80	1.00	1.00	1.30	1.60	1.60	2.00			
613.5		0.65	0.80	1.00	1.00	1.00	1.30	1.60	1.60	2.00			
626.6		0.65	0.80	1.00	1.00	1.30	1.30	1.60	1.60	2.00			
721.8		0.65	0.80	1.00	1.00	1.30	1.30	1.60	2.00	2.00			
734.9		0.65	0.80	1.00	1.00	1.30	1.30	1.60	2.00				
777.6		0.65	0.80	1.00	1.30	1.30	1.30	1.60	2.00				
784.1		0.80	0.80	1.00	1.30	1.30	1.60	1.60	2.00				
856.3		0.80	1.00	1.00	1.30	1.30	1.60	1.60	2.00				
918.6		0.80	1.00	1.00	1.30	1.30	1.60	2.00	2.00				
961.3		0.80	1.00	1.30	1.30	1.30	1.60	2.00	2.00				
1036.7		0.80	1.00	1.30	1.30	1.30	1.60	2.00					
1177.8		0.80	1.00	1.30	1.30	1.60	1.60	2.00					
1227.0		1.00	1.00	1.30	1.30	1.60	2.00	2.00					
1243.4		1.00	1.30	1.30	1.30	1.60	2.00	2.00					
1312.3		1.00	1.30	1.30	1.60	1.60	2.00	2.00					
1555.1		1.00	1.30	1.30	1.60	1.60	2.00						
1571.5		1.00	1.30	1.60	1.60	1.60	2.00						
1640.4		1.00	1.30	1.60	1.60	2.00	2.00						

LonPoint Type 1 Base Plate Installation

The DM-21 Module is installed in a Type 1 Base Plate, Model 40111. Figure 3.8 presents a detailed view of the front side of the base plate. The circuit board inside the Type 1 Base Plate includes a large number “1” or “1A” in the lower right corner, as viewed from the front of the base plate, designating it as a Type 1 Base Plate.

Warning

The DM-21 *requires* a Revision **1A** Type 1 Base Plate. The power supply internal to the DM-21 will be disabled if used with a revision 1 Base Plate. Be sure to use Revision **1A** *only*.

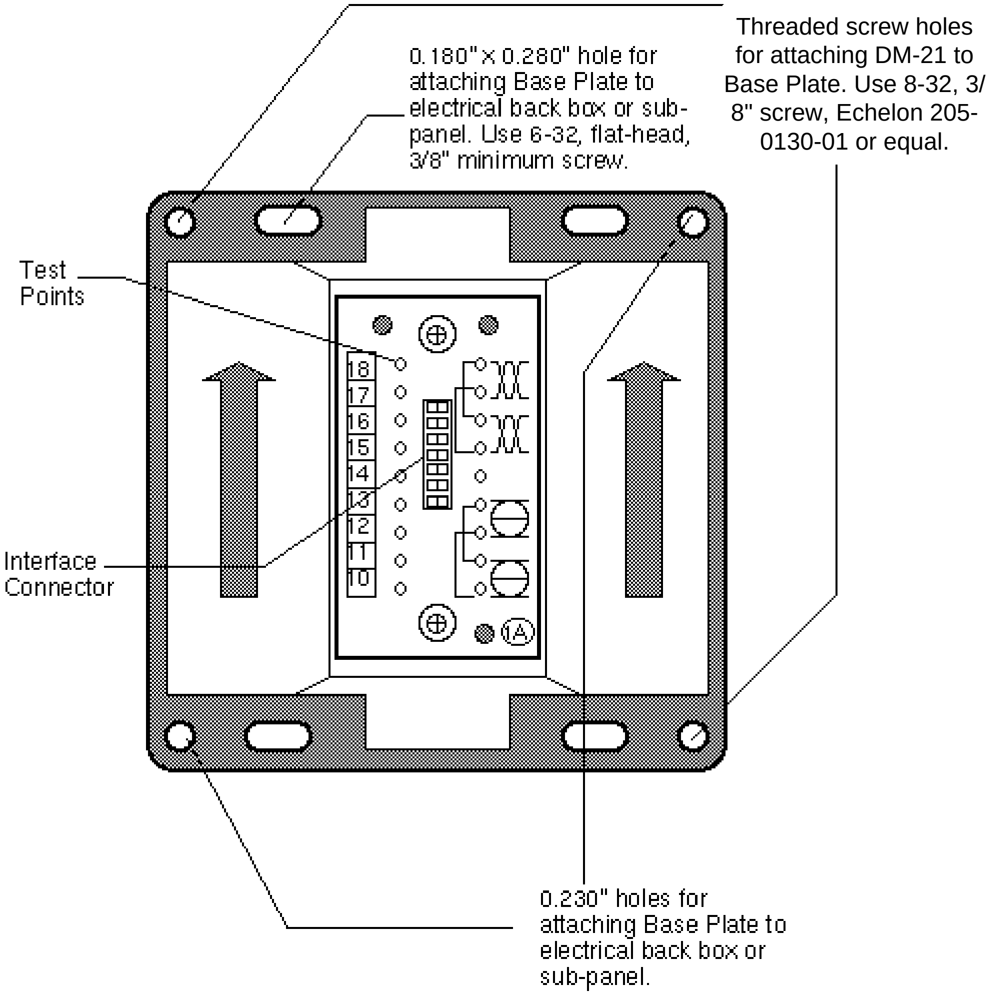

Threaded screw holes for attaching DM-21 to Base Plate. Use 8-32, 3/8" screw, Echelon 205-0130-01 or equal.
0.180" x 0.280" hole for attaching Base Plate to electrical back box or sub-panel. Use 6-32, flat-head, 3/8" minimum screw.
Test Points
Interface Connector
18
17
16
15
14
13
12
11
10
1A
0.230" holes for attaching Base Plate to electrical back box or sub-panel.

Figure 3.8 - Type 1 Base Plate - Front Panel

Figure 3.9 shows a detailed view of the rear side of the base plate, the side to which wiring connections are made. See *DM-21 Pinout* in Chapter 4 for more information about DM-21 wiring connections.

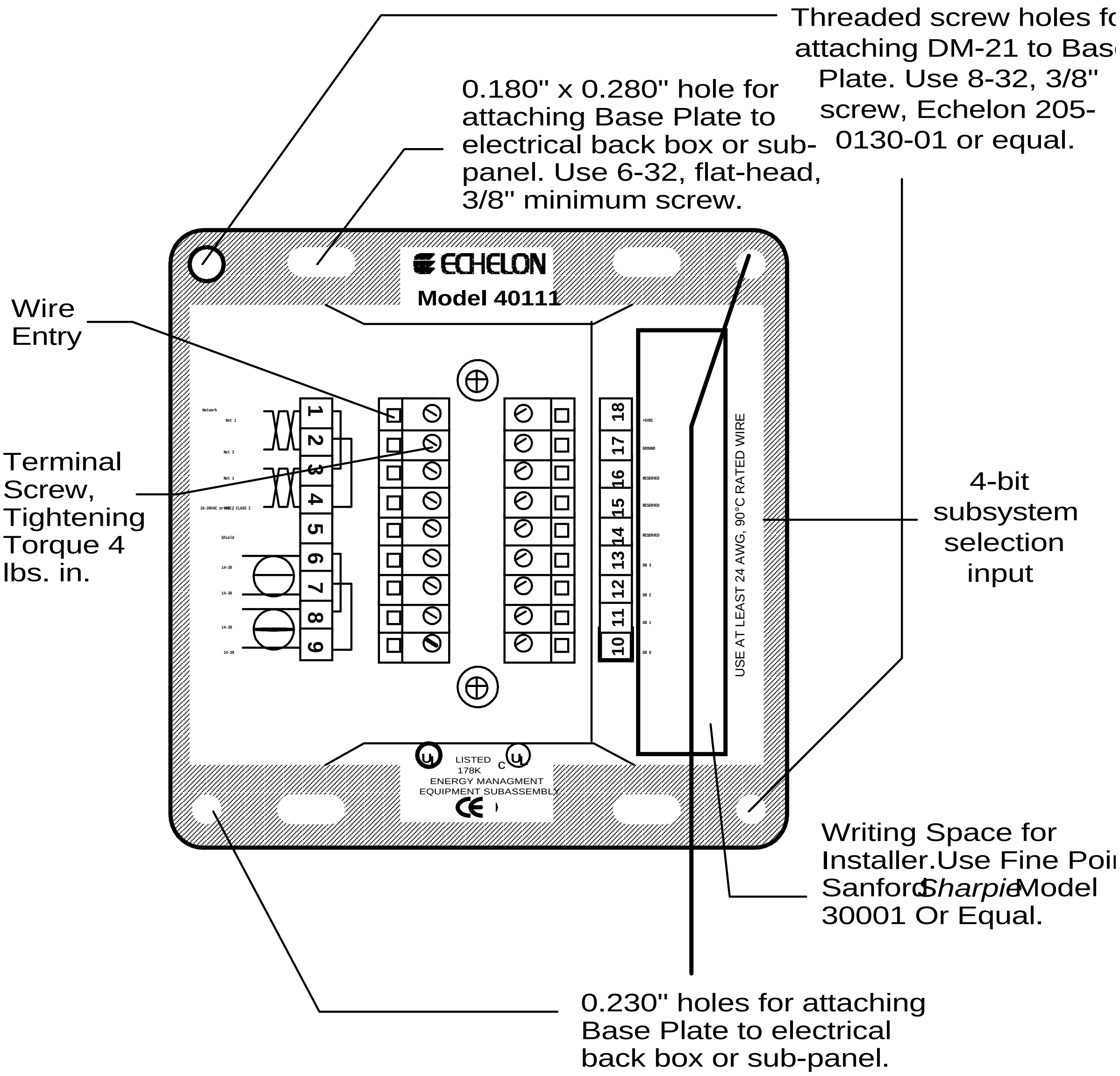


Figure 3.9 – Type 1 Base Plate - Rear Wiring Panel

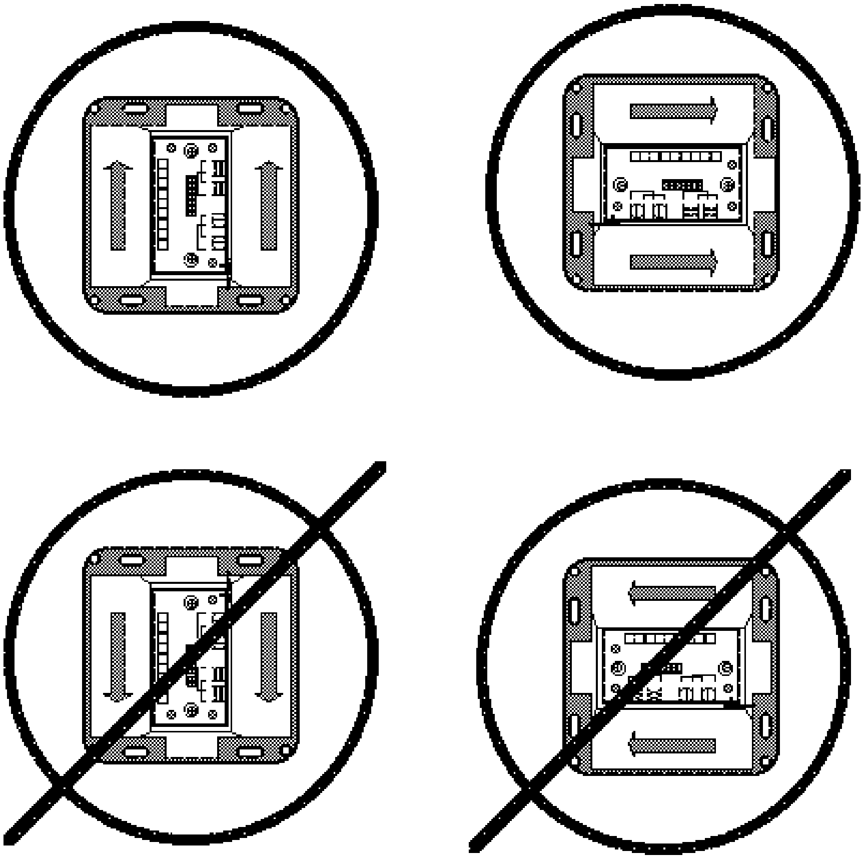


Figure 3.10 – Type 1 Base Plate Mounting Orientations

Connecting Wiring

Route network and power cabling into the electrical enclosure using suitable conduit fittings, bushings, or wire glands. Once the cabling has been brought into the electrical enclosure, leave a service loop of <6 inches (15cm) of cable to simplify wiring the Type 1 Base Plate. Dress the cables using tie wraps or tape to ensure that the cabling is not compressed or caught when the base plate is screwed into place.

Strip the cable jacket and wire conductors. The base plate screw terminals will accept 26AWG (0.4mm) to 12AWG (2.2mm) wire, which should be stripped to a length of 0.32” (8mm). Although not required, it may be useful to use a soldering iron to tin the stripped lengths of any stranded wires to prevent fraying and inadvertent contact with adjactent terminals. The screw terminal blocks on the rear panel of the Type 1 Base Plate are color coded to simplify wiring. The color coding scheme is shown below. The optimum tightening torque for the Type 1 Base Plate screw terminals is 4 lbs. in. (0.5Nm) maximum. The ideal screwdriver tip width is 0.125” (3mm).

Table 3.4 - Type 1 Base Plate Terminal Block Color Coding

Terminal Number	***Terminal Color***	***Function***
1 - 4	Orange	Network
5	Orange	Cable shield, if used
6 - 9	Black	Power
10 – 18	Green	4-bit subsystem selection input

Two sets of screw terminals are provided for both the power and network wiring connections. These connections are internally jumpered on the Type 1 Base Plate PCB to provide continuity of the network and power wiring, even if no DM-21 is installed, as shown in figure 3.11. This feature permits the base plate wiring to be pre-installed and the network and power circuits checked for continuity throughout the installation, before a single DM-21 is ever installed. This feature also prevents network and power interruptions as a result of hot-swapping modules during commissioning or service operations. Finally, providing two sets of screw terminals permits incoming and outgoing wiring to be landed at separate screw terminals without the need to insert more than one wire in any given screw terminal.

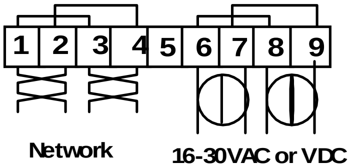
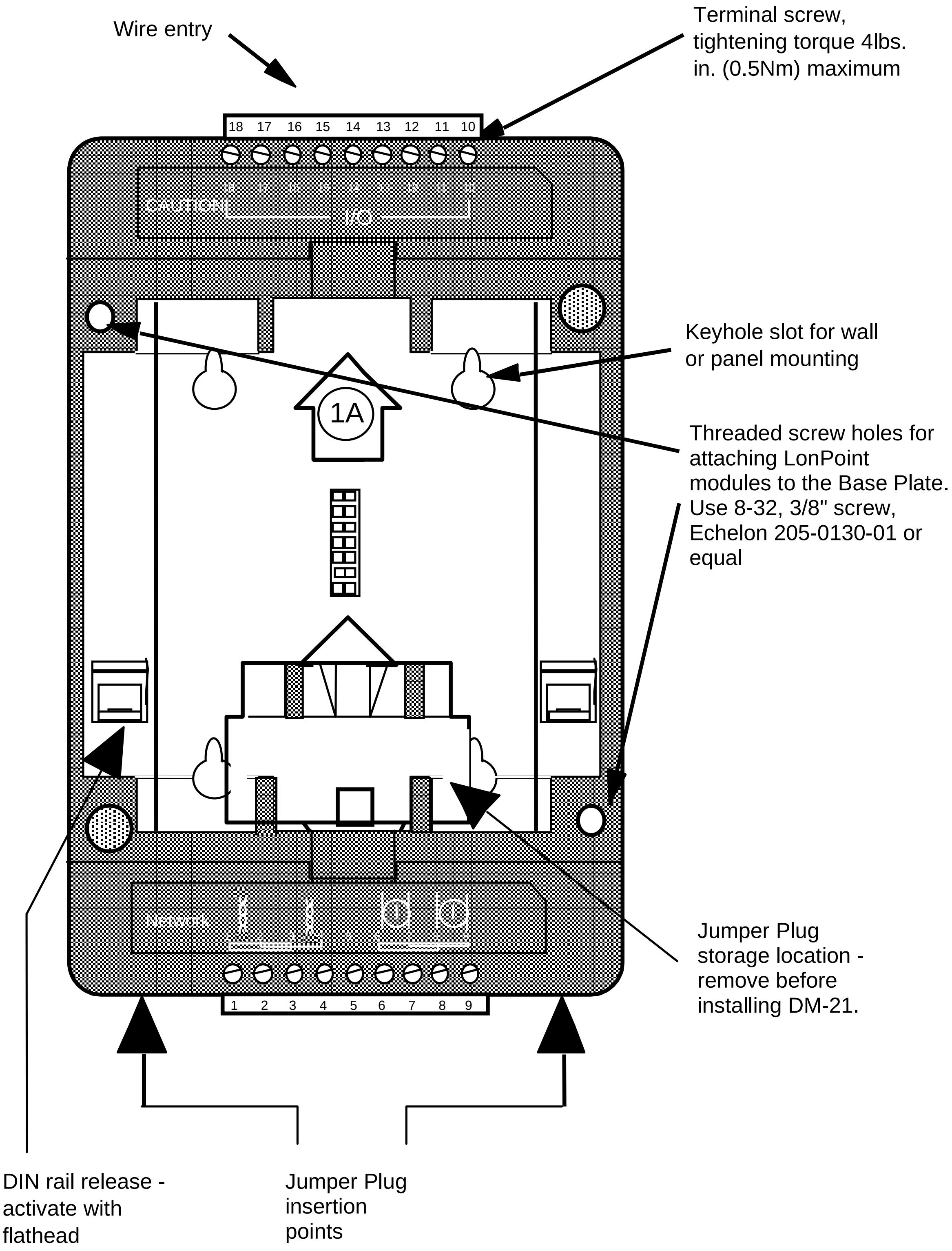


Figure 3.11 - Base Plate Power and Network Wiring Connections

The DM-21 uses pins 10 - 13 on a Type 1 Base Plate to supply a subsystem selection number. This is a binary number where Pin 10 represents the low-order bit. See Chapter 4 for more information about the DM-21 pinout.

Type 1D DIN Base Plate Installation

Figure 3.12 presents a detailed view of the front side of the base plate. The circuit board inside the Type 1D DIN Base Plate includes an arrow with the designation "1A," designating it as a Type 1 Base Plate with a Revision A circuit board. The 1D Base Plate is intended to be installed either on a 35mm DIN rail, or using the integral keyhole slots, on a wall or panel. Two arrows indicates the vertical orientation of the Base Plate that will provide the best viewing of the front panel legends.



Wire entry

Terminal screw, tightening torque 4lbs. in. (0.5Nm) maximum

Keyhole slot for wall or panel mounting

Threaded screw holes for attaching LonPoint modules to the Base Plate. Use 8-32, 3/8" screw, Echelon 205-0130-01 or equal

Jumper Plug storage location - remove before installing DM-21.

DIN rail release - activate with flathead

Jumper Plug insertion points

Figure 3.12 - Front Side of Type 1D DIN Base Plate

The base plate contains two integral DIN rail tabs that securely grab a 35mm DIN rail onto which the base plate is mounted. The base plate may be used on both 35mm x 7.5mm and 35mm x 15mm DIN rails. To release the Base Plate from the DIN rail, sequentially insert a flathead screwdriver into the DIN rail release tabs and gently pull away from the DIN rail (figure 3.13).

Four keyhole slots are provided for affixing the Type 1D DIN Base Plate to a wall or panel (figure 3.14).

The 1D Base Plate is supplied with a jumper plug that connects the power and network connections between adjacent base plates. The Jumper Plug is shipped inserted into the main cavity of the Base Plate, and can be removed by gently pulling the plug handle in the direction of the mounting arrow. The Jumper Plug **must** be removed before a DM-21 module can be inserted into the Base Plate.

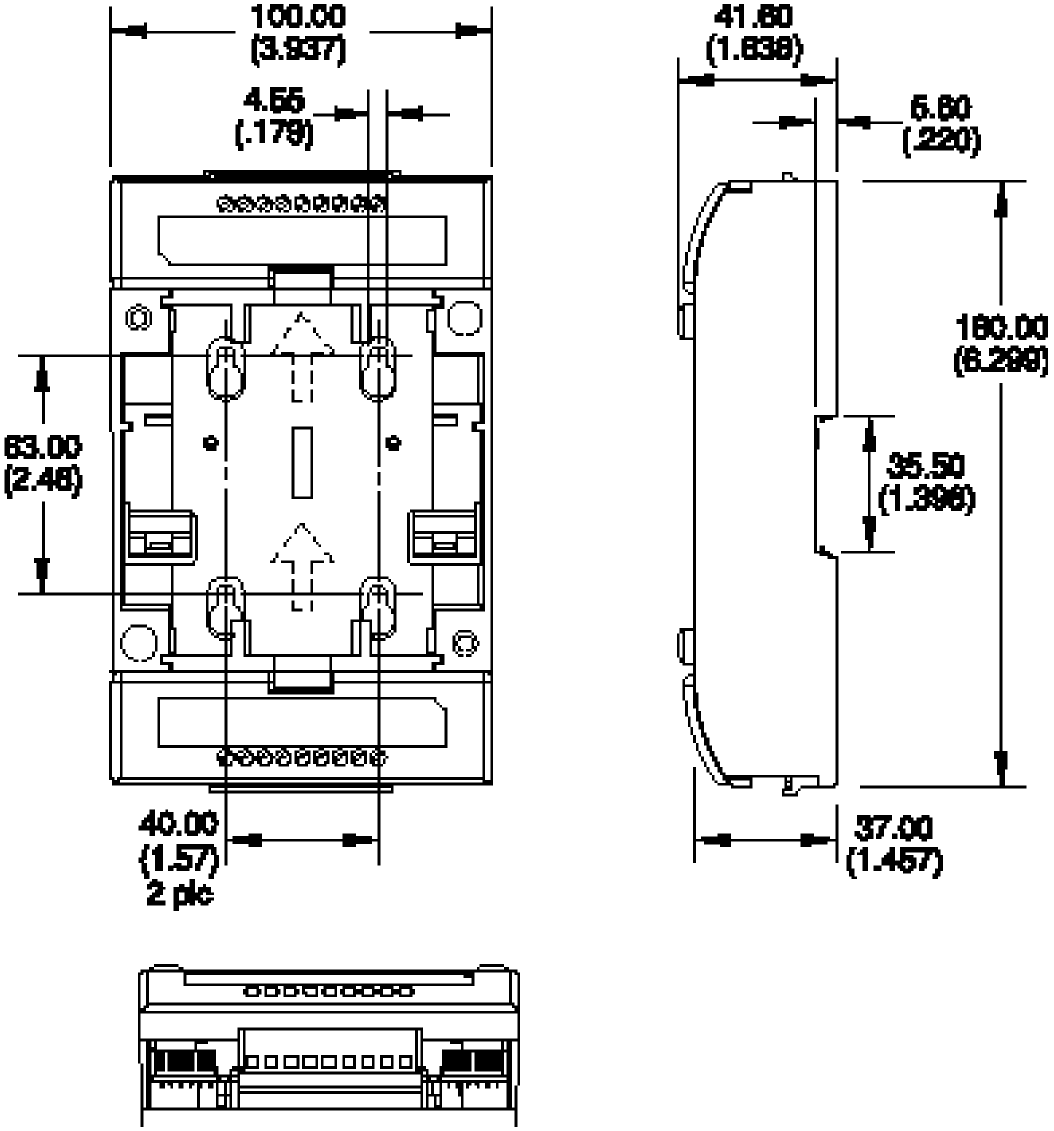


Figure 3.13 - DIN Rail Base Plate Dimensions (Dimensions are in mm (inches).)

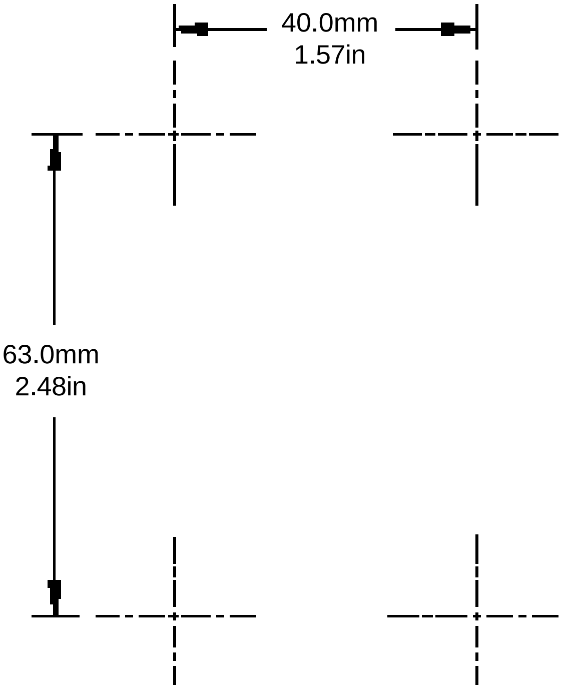


Figure 3.14 Mounting Hole Pattern Dimensions for Type1D Base Plates

Connecting Wiring

Route all network, power, and input/output cabling to the pertinent screw terminals located at the top and bottom of the Base Plate. It is good practice to separate the input/output cabling as much as possible from the network and power cabling, especially if low-level analog signals are being supervised.

Strip the cable jacket and wire conductors. The base plate screw terminals will accept 24AWG (0.5mm) to 12AWG(2.2mm) wire, which should be stripped to a length of 0.32" (8mm). Although not required, it may be useful to use a soldering iron to tin the stripped lengths of any strand wired to prevent fraying and inadvertent contact with adjacent terminals. Identifying screw terminal numbers are conveniently located both above and below each screw terminal. Symbols and/or language identifying the function of the screw terminals also are provided.

Note that terminals 1 and 9 are numbered from left to right, but terminals 10 to 18 are numbered from right to left.

The optimum tightening torque for the screw terminals is approximately 4 pounds (0.5Nm). The ideal flathead screwdriver tip width is 3/32" (2.5mm).

Table 3.5 Type 1D Base Plate Terminal Block Connections

Terminal Number	*Function*
1 -4	Network
5	Cable shield, if used
6 - 9	Power
10 – 13, 17, 18	4-bit subsystem selection input

Two sets of screws are provided for both the power and network wiring connections. These connections are internally jumpered on the Type 1D Base Plate PCB to provide continuity of the network and power wiring, even if no module is installed, as shown in figure 3.15. This feature permits the base plate wiring to be pre-installed and the network and power circuits checked for continuity throughout the installation, before a single module is ever installed. This feature also prevents network and power interruptions as a result of hot swapping modules during commissioning or service operations. Finally, providing two sets of screw terminals permits incoming and outgoing wiring to be landed at separate screw terminals without the need to insert more than one wire in any given screw terminal.

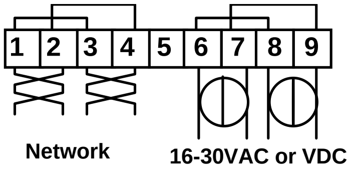
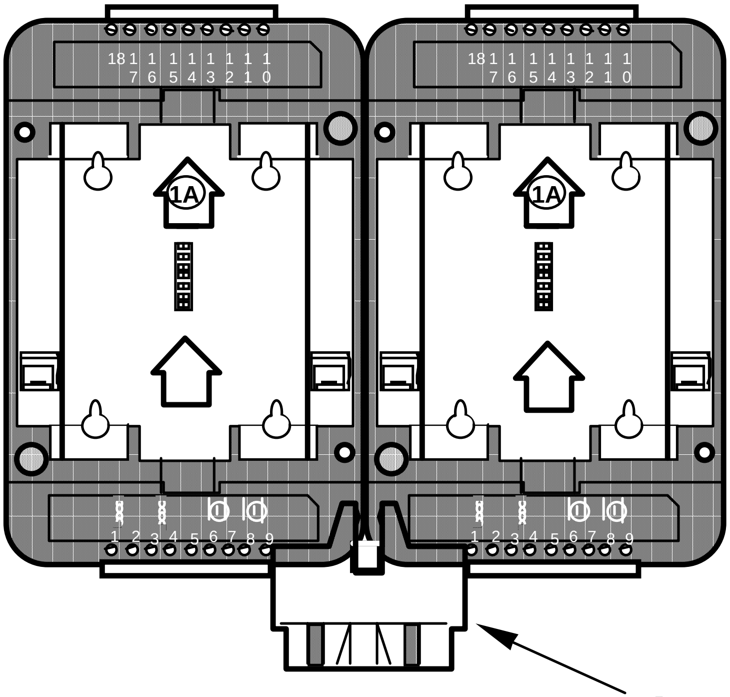


Figure 3.15 - Type 1D Base Plate Power and Network Wiring Connections

Connecting the Jumper Plug(s)

From time to time it may be necessary to mount two or more LonPoint Type 1D Base Plates adjacent to one another. In these instances, power and network wiring must be daisy-chained between each Base Plate. Routing power and network wiring between multiple base plates can be a laborious and time consuming task, and there is always the risk of a wiring error. For this reason, the Type 1D Base Plates are each supplied with a Jumper Plug and Jumper Plug connectors which simply and easily bus the power and network between adjacent Base Plates.

The Jumper Plug connectors are located on either side of the terminal block with connections 1 to 9 . The Jumper Plug connectors are wired in parallel with the network and power connections on the screw terminals. When two Base Plates are mounted adjacent to each other, with the mounting arrows pointing in the same direction, the Jumper Plug connectors will be aligned such that a Jumper Plug may be inserted into the two connectors, thus bridging power and network between the two Base Plates (figure 3.16).



Jumper Plug (x-ray view)

Figure 3.16 - Mounting Two Base Plates with Jumper Plug Connectors

When used with a Type 2D Router Base Plate, there are restrictions about the placement of the Type 2D Router Base Plates (and the routers they contain) relative to the adjacent Type 1D Base Plates. In all cases, any Type 2D Router Base Plates must be located to the left side of any Type 1D Base Plates (figure 3.17) with which they will share a common channel.

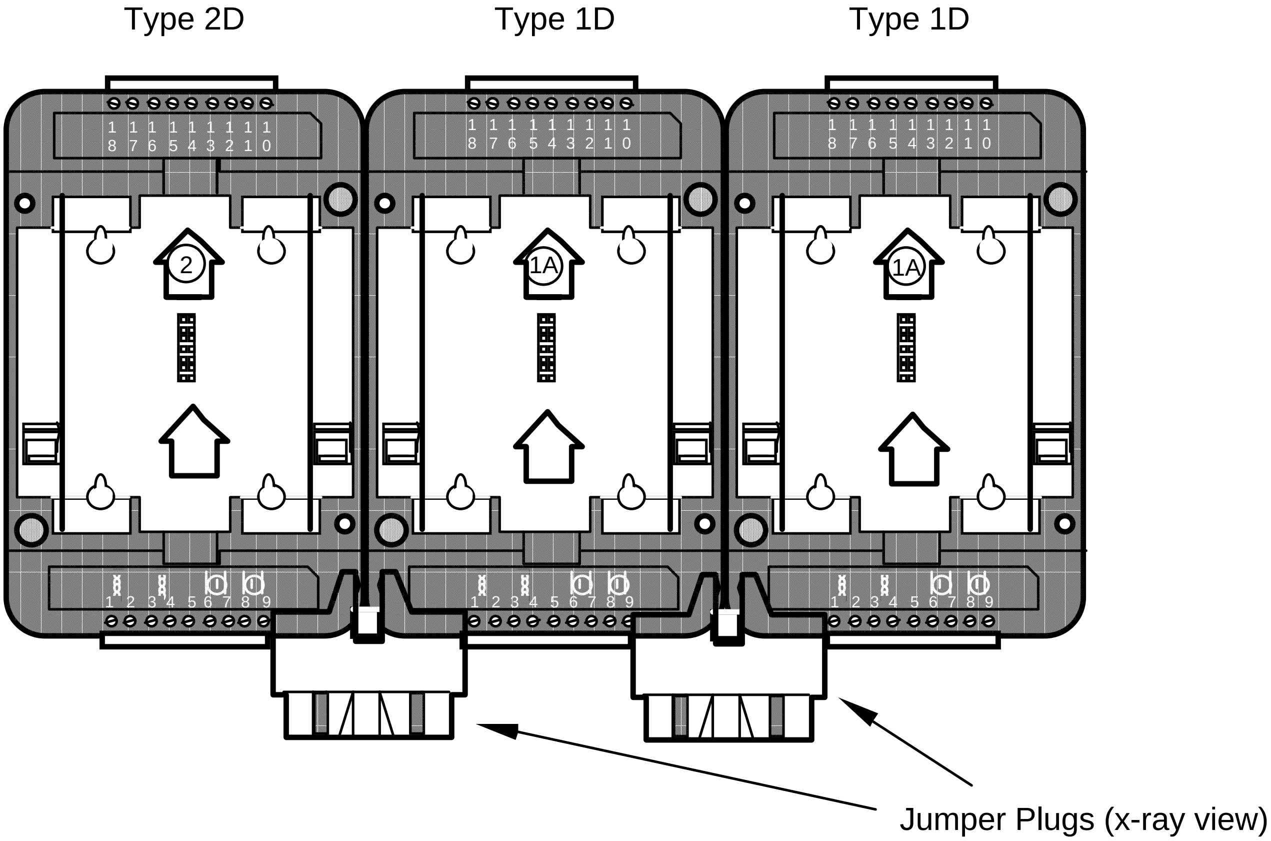


Figure 3.17 - Mixing Type 1D and Type 2D Router Base Plates

The reason for this limitation is that only the right Jumper Plug connector on a Type 2D Router Base Plate bridges the network connection; the left Jumper Plug connector on a Type 2D Router Base Plate bridges only the power connection. This arrangement permits several routers, each with a different channel type, to be mounted on a common DIN rail and share a common power supply. Figure 3.18 shows mounting configurations that are problematic and **must be avoided** because the router will be unable to communicate with the other LonPoint modules via the Jumper Plug.

If such a mounting arrangement must be used, then the base plates must be interconnected using wiring instead of the Jumper Plug.

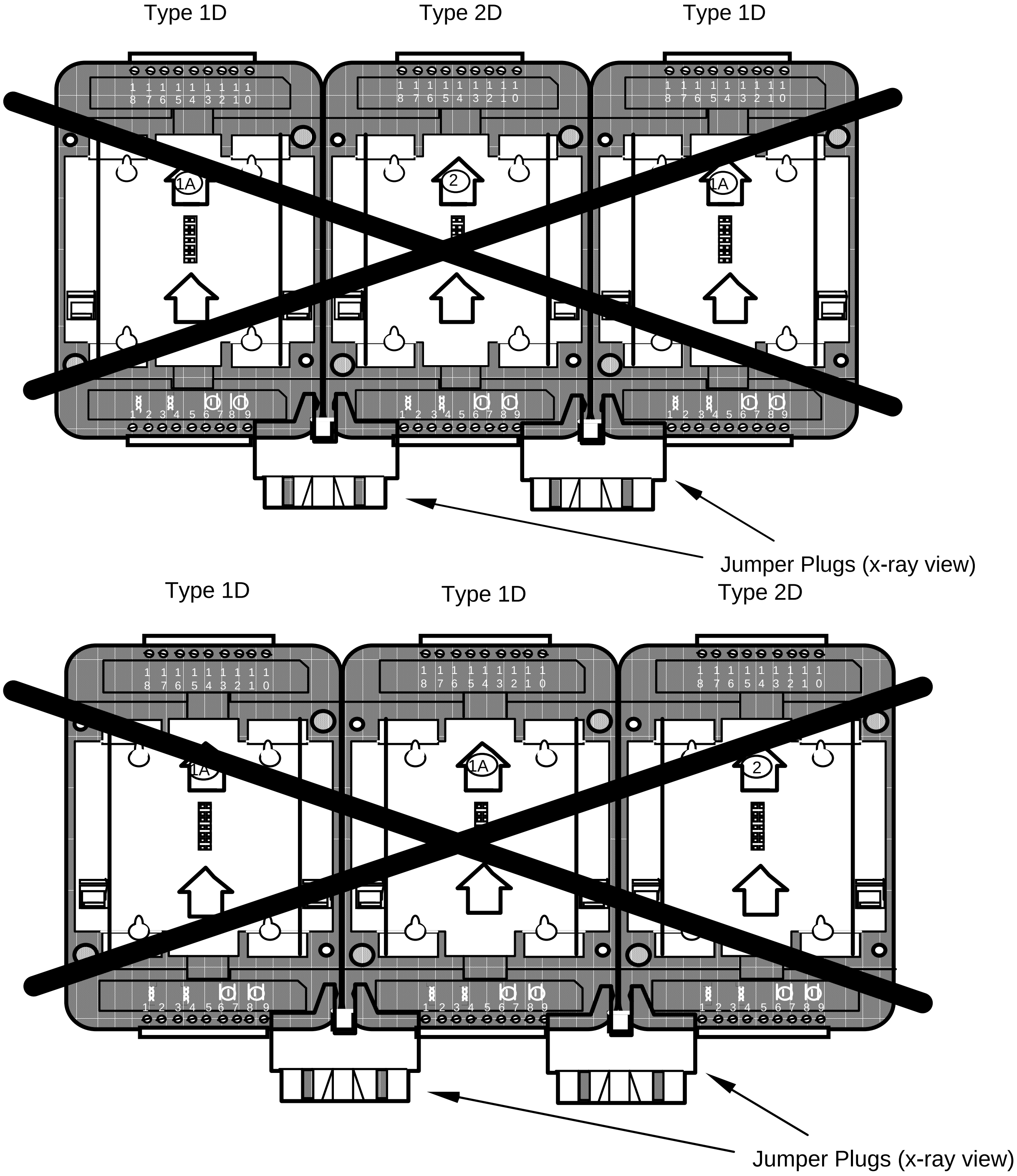


Figure 3.18 – Problematic Mounting Configurations when Using Routers and a Type 2D Router Base Plates

4

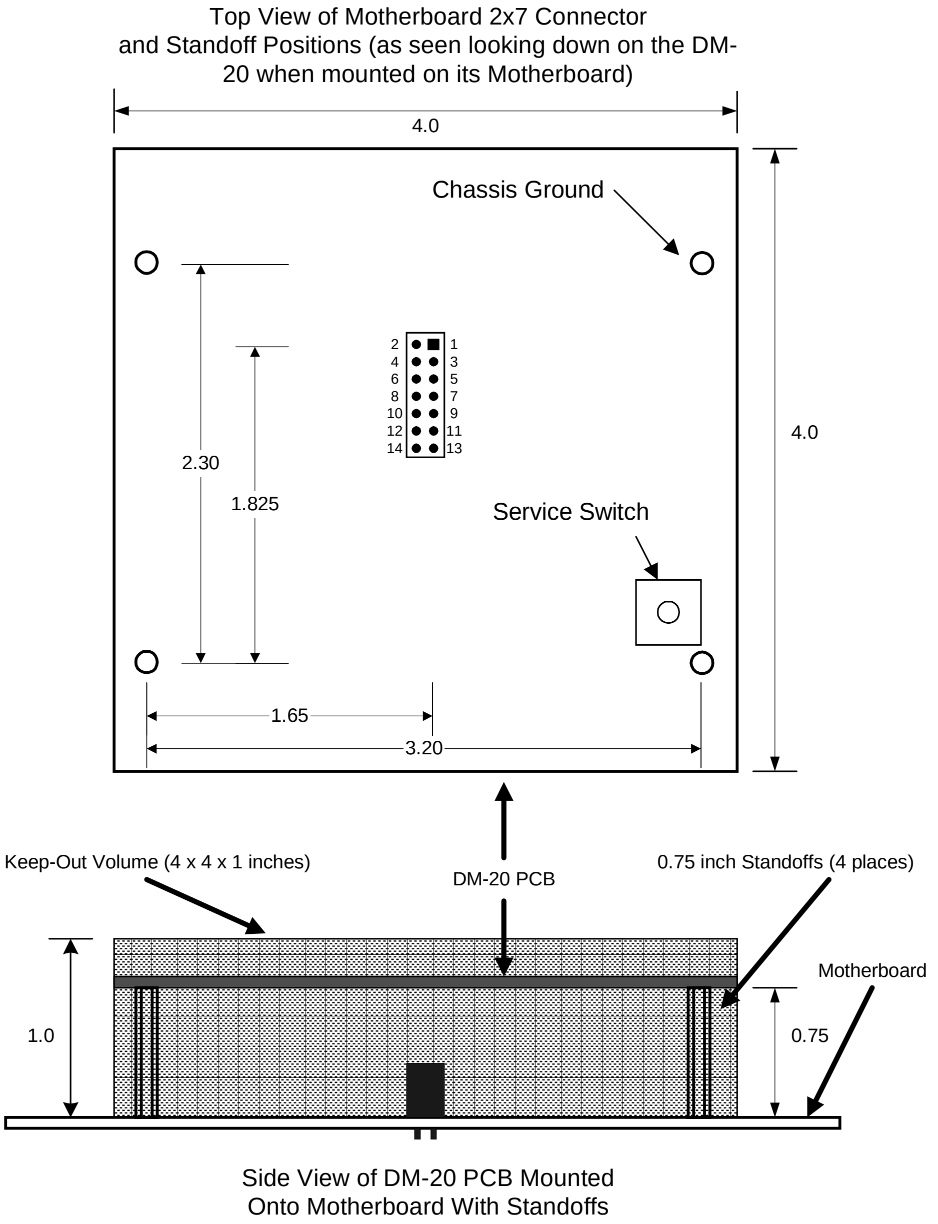
Device Manager Hardware Description and Installation

This chapter explains how to install the Device Manager hardware.

The DM-20 Device Manager Hardware Interfaces

Mechanical

Figure 4.1 shows a DM-20 module mounted onto a motherboard PCB (all dimensions are in inches):



Top View of Motherboard 2x7 Connector and Standoff Positions (as seen looking down on the DM-20 when mounted on its Motherboard)

Side View of DM-20 PCB Mounted Onto Motherboard With Standoffs

Figure 4.1 - DM-20 PCB Mounted onto Motherboard

The positions of the standoffs and 2x7 motherboard connector are shown in the top portion of the previous figure (as seen when looking down on the DM-20 when mounted on its motherboard PCB). All measurements are to the centers of the pins on JP1.

The DM-20 occupies a volume of 4 x 4 x 1 inches (10.2 x 10.2 x 2.5 cm), and other components should be kept out of this volume. The four standoffs are 0.75 inches long, and either #4 or #6 standoffs may be used. If the DM-20 assembly is conformally coated (this requires advance approval from Echelon to avoid voiding the product warranty) before mounting it on the motherboard, special care should be taken to not coat the four conductive screw holes on the DM-20 PCB that connect to the four standoffs.

Table 4.1 – DM-20 Mechanical Interface

Mounting holes	4
Screw size for mounting holes	#4 or #6
Motherboard standoff height	0.75 inches
Module connector	2 x 7 pin, 100 mil pitch male header
Motherboard mating connector	Samtec SSW-107-01-S-D-LL, 2x7 socket strip
Approximate module size	4 x 4 x 1 inches

Electrical

The grounding provided by the standoffs is an important part of the EMC design of the DM-20. The standoff nearest C410 is connected to the DM-20 module’s chassis ground, and the other three standoffs are connected to the DM-20 module’s logic ground. The DM-20 module’s chassis ground standoff must be connected via a low-impedance path to the chassis of the metal enclosure. The DM-20 module’s logic ground standoffs should be connected to the logic ground of the motherboard.

Pin Description

The DM-20 module has a single 2x7 connector. The pinout is described in table 5.2.

Table 4.2 DM-20 Pinout

JP1 2x7 Connector Pin Number	DM-20 Module Signal Type	Description
1	FTT-10A	Net1
2	+5V DC Power input	+5V±5% input @ 150 mA
3	FTT-10A	Net2
4	Ground	Ground
5	Not used	
6	Neuron ~Service	Neuron Service Pin signal
7 (Neuron IO)	Neuron IO	Reserved
8 (Neuron IO)	Neuron IO	Reserved
9 (Neuron IO)	Neuron IO	4-bit subsystem selection input (bit 2)
10 (Neuron IO)	Neuron IO	4-bit subsystem selection input (bit 3)
11	Do not connect	
12 (Neuron IO)	Neuron IO	4-bit subsystem selection input (bit 1)
13	Do not connect	
14 (Neuron IO)	Neuron IO	4-bit subsystem selection input (bit 0)

DM-21 Device Manger Hardware Interface

The front panel of the DM-21 module is shown in figure 4.2. A single Status LED is provided for the device.

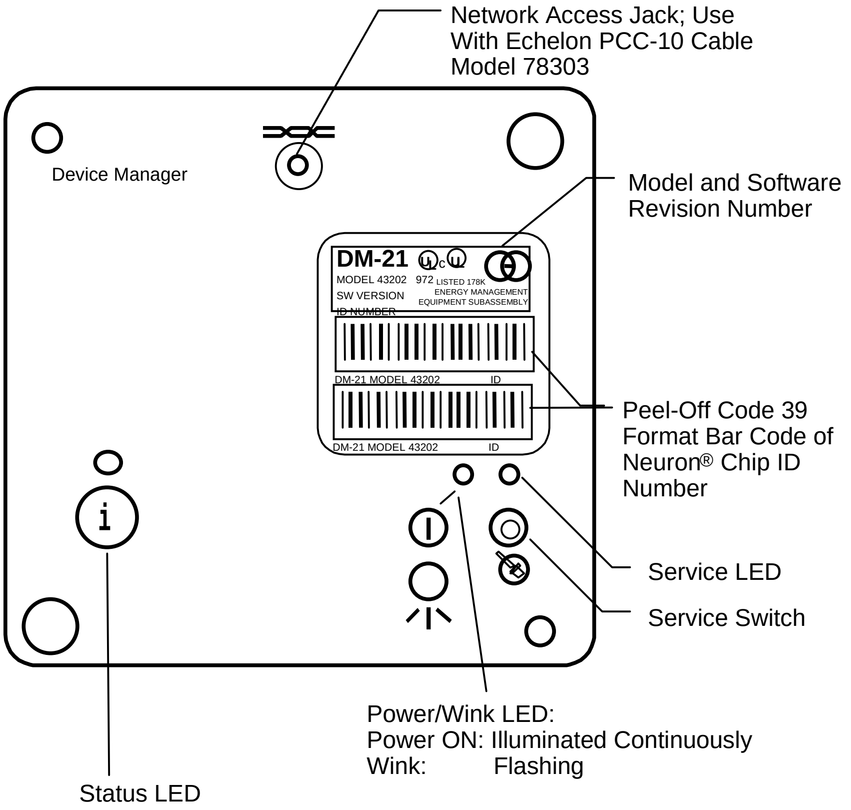


Figure 4.2 - DM-21 Module - Front Panel

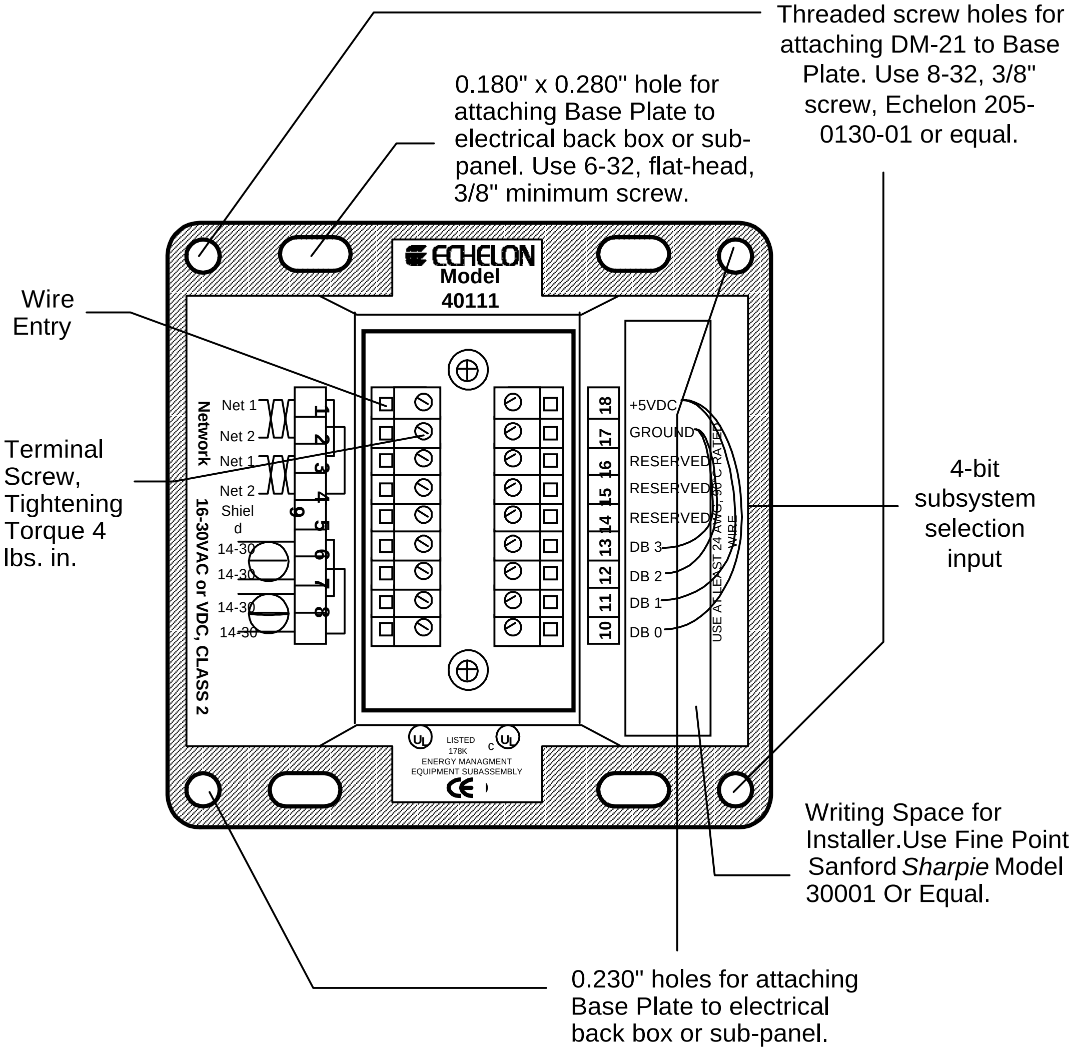
The DM-21 module is installed in a Revision 1A Type 1 Base Plate (see Chapter 3 for information on installing base plates).

DM-21 Pinout

Table 4.3 – DM-21 Pinout

Rear Panel Pin Number (DM-21 only)	DM-21 Signal Type	Description
1 & 3	FTT-10A	Net 1
2 & 4	FTT-10A	Net 2
5	No Connect	
6 & 8	16 - 30 VAC or DC Power input	16 – 30 VAC or DC power @ 2.4VA
7 & 9	16 - 30 VAC or DC Power input	16 – 30 VAC or DC power @ 2.4VA
10	Neuron IO	4-bit subsystem selection input (bit 0)
11	Neuron IO	4-bit subsystem selection input (bit 1)
12	Neuron IO	4-bit subsystem selection input (bit 2)
13	Neuron IO	4 bit subsystem selection input (bit 3)
14	Neuron IO	Reserved
15	Neuron IO	Reserved
16	Neuron ~Service	Neuron Service Pin
17	Ground	Ground
18	+5V DC Power outpu	+5V+/- 5% output @ 20mA

4-bit Subsystem selection Input Pins



Threaded screw holes for attaching DM-21 to Base Plate. Use 8-32, 3/8" screw, Echelon 205-0130-01 or equal.

0.180" x 0.280" hole for attaching Base Plate to electrical back box or sub-panel. Use 6-32, flat-head, 3/8" minimum screw.

Wire Entry

Terminal Screw, Tightening Torque 4 lbs. in.

4-bit subsystem selection input

Writing Space for Installer.Use Fine Point Sanford *Sharpie* Model 30001 Or Equal.

0.230" holes for attaching Base Plate to electrical back box or sub-panel.

Figure 4.3 – Type 1 Base Plate, Rear Wiring Panel

The DM-20/21 supports up to 16 subsystems. The subsystem is selected by setting a 4-bit binary input on the subsystem selection input. Subsystem selection input pins for the DM-21 are terminals 10, 11, 12, and 13 on the Type 1/1D back-plate, representing bits 0 to 3 respectively. Subsystem selection input pins for the DM-20 are pins 14, 12, 9, and 10 of JP1 representing bits 0 to 3, respectively. Tables 4.4 and 4.5 present the arrangement of jumpers to set the correct subsystem using the 4-bit subsystem selection inputs.

If only a single database is specified by the *DM Pin Usage* configuration property, the DM-20/21 will ignore the subsystem selection setting. The pins/terminals may then float high or low.

DM-20 Subsystem selection (2x7 DM-20 connector J1 pinout)

Table 4.4 – DM-20 Subsystem selection

	Pin 10	Pin 9	Pin 12	Pin 14
Subsystem 0	GND	GND	GND	GND
Subsystem 1	GND	GND	GND	+ 5VDC
Subsystem 2	GND	GND	+ 5VDC	GND
Subsystem 3	GND	GND	+ 5VDC	+ 5VDC
Subsystem 4	GND	+ 5VDC	GND	GND
Subsystem 5	GND	+ 5VDC	GND	+ 5VDC
Subsystem 6	GND	+ 5VDC	+ 5VDC	GND
Subsystem 7	GND	+ 5VDC	+ 5VDC	+ 5VDC
Subsystem 8	+ 5VDC	GND	GND	GND
Subsystem 9	+ 5VDC	GND	GND	+ 5VDC
Subsystem 10	+ 5VDC	GND	+ 5VDC	GND
Subsystem 11	+ 5VDC	GND	+ 5VDC	+ 5VDC
Subsystem 12	+ 5VDC	+ 5VDC	GND	GND
Subsystem 13	+ 5VDC	+ 5VDC	GND	+ 5VDC
Subsystem 14	+ 5VDC	+ 5VDC	+ 5VDC	GND
Subsystem 15	+ 5VDC	+ 5VDC	+ 5VDC	+ 5VDC

DM-21 Subsystem selection (Type 1/1D Base Plate Terminal Connections)

Table 4.5 – DM-21 Subsystem Selection

	Pin 13	Pin 12	Pin 11	Pin 10
Subsystem 0	GND	GND	GND	GND
Subsystem 1	GND	GND	GND	+ 5VDC
Subsystem 2	GND	GND	+ 5VDC	GND
Subsystem 3	GND	GND	+ 5VDC	+ 5VDC
Subsystem 4	GND	+ 5VDC	GND	GND
Subsystem 5	GND	+ 5VDC	GND	+ 5VDC
Subsystem 6	GND	+ 5VDC	+ 5VDC	GND
Subsystem 7	GND	+ 5VDC	+ 5VDC	+ 5VDC

Subsystem 8	+ 5VDC	GND	GND	GND
Subsystem 9	+ 5VDC	GND	GND	+ 5VDC
Subsystem 10	+ 5VDC	GND	+ 5VDC	GND
Subsystem 11	+ 5VDC	GND	+ 5VDC	+ 5VDC
Subsystem 12	+ 5VDC	+ 5VDC	GND	GND
Subsystem 13	+ 5VDC	+ 5VDC	GND	+ 5VDC
Subsystem 14	+ 5VDC	+ 5VDC	+ 5VDC	GND
Subsystem 15	+ 5VDC	+ 5VDC	+ 5VDC	+ 5VDC

Reset Circuit

The Device Managers incorporate an internal reset circuit. External resets to the Device Manager are accomplished by cycling power to the module.

Service Pin Circuit

The Device Managers incorporate a front panel service switch and yellow service LED. Additionally, ~Service is provided as an external signal. An external service switch and LED may be added provided that the total current draw does not exceed 20mA.

Power/Wink LED

The Device Managers incorporate a front panel combination Power/Wink LED that is driven by the Neuron 3150 Chip.

Status LED

The Device Managers provide a *status* LED that flashes in a predefined pattern to identify the health and activity of the module. The number of flashes of the status LED specifies the current state of the Device Manager as shown in table 4.6.

Table 4.6 – Status LED

5	The Device Manager object is disabled or the DM MODE has been set to disable discovery or commissioning.
4	No database is selected.
3	The Device Manager is commissioning itself.
2	One or more permanent devices have not been discovered/commissioned.
1	The Device Manager has determined that all permanent devices have been correctly installed (commissioned).
0	The Device Manager application is not running (offline, unconfigured).

The status interval is a 3-second cycle with a flash rate of 200 ms on, 200 ms off. For example, when all nodes are properly commissioned and the DM-20/21 is functioning properly, there will be one 200 ms pulse every 3 seconds.

Power Supply—DM-20

The DM-20 must be powered from 5VDC@150mA.

Power Supply—DM-21

The DM-21 must be powered from 16-30V AC or DC @ 2.4VA.

Installing the DM-20 Module

To install the DM-20 module, follow these steps:

1. Follow the node configuration procedure described in Chapter 5 to configure the software in the DM-20 module.

2. Ensure that the motherboard has been wired and installed correctly, and is securely fastened to its electrical enclosure. Ensure that the motherboard provides the proper signals to the DM-20 for subsystem selection.

3. Check for continuity on all network and power wiring. Ensure that the correct devices are connected to the network and that all devices can be uniquely identified by Program ID/Location String.

4. Make certain that all network channels are terminated with the correct number and type of Terminator.

5. Wear a properly-connected ESD protection strap while handling the DM-20 module.

6. Insert the DM-20 module into the motherboard by aligning the network access jack on the DM-20 with the mating connector on the motherboard. The DM-20 should be affixed using the proper .75 inch standoff. See *The DM-20 Device Manager Hardware Interfaces*, earlier in this chapter, for information on which standoffs should be connected to digital ground, and which standoff should be connected to chassis ground.

7. The DM-20 module cannot be hot-plugged: Ensure the motherboard power is off before attaching the DM-20 module.

8. Securely attach the DM-20 module to the motherboard.

Installing the DM-21 Module

To install the DM-21 module, follow these steps:

1 Follow the node configuration procedure described in Chapter 5 to configure the software in the DM-21 module.

2 Ensure that the Type 1/1D Base Plate has been wired and installed correctly, and is securely fastened to the electrical enclosure. Ensure that the proper wire jumpers have been installed for subsystem selection.

3 Check for continuity on all network and power wiring. Ensure that the correct devices are connected to the network and that all devices can be uniquely identified by Program ID/Location String.

4 Make certain that all network channels are terminated with the correct number and type of Terminator.

5 Insert the DM-21 module into the Base Plate by aligning the network access jack on the DM-21 with the arrow head on the Type 1 Base Plate (the arrow should point in the direction of the jack), and then pressing firmly to ensure that the module is properly seated in its mating Base Plate connector.

6 The DM-21 module can be hot-plugged: there is no need to disconnect power when installing the module. If power is present then the Power LED
will illuminate continuously. If power is not present then the Power LED will remain off until power is supplied to the module.

7 Securely attach the DM-21 module to the Base Plate by screwing the two Echelon 205-0130-01 screws (8-32 [M8], 3/8” [9.5mm]) into the threaded inserts in the Base Plate. Use approximately 4 lbs. in. (0.5Nm) torque on the screws.

5

Installing and Getting Started with the Device Manager Software

This chapter provides instructions for installing the Device Manager software. Also included are several typical network designs using the Device Manager.

Installing the Device Manager Software

Before the Device Manager software is installed, the LonMaker for Windows software and any other LONWORKS software to be used must be installed in a designated LONWORKS directory (typically C:\LONWORKS).

To install the Device Manager software, insert the CD into the CD-ROM drive. Navigate to the CD-ROM drive and double-click on `Setup.exe`. Follow the setup wizard's on screen instructions.

Creating a Device Manager Managed Network

Creating a Device Manager managed network has three basic steps.

1. Create the network design off-line using an LNS-based network design tool. The examples that follow describe the process using the LonMaker for Windows Integration Tool and Echelon's LonPoint devices; however, any tool based on LNS 2.0 or higher may be used, and any LONWORKS device may be substituted for the LonPoint modules.
2. Use the Device Manager Loading Wizard to convert the LNS database into a Managed Device List and load that Managed Device List into the Device Manager.
3. Physically install the devices that will be managed by the Device Manager. Be sure that all devices are in the unconfigured state and are powered. Attach the Device Manager to the network. When the Device Manager is powered it will reset, select a subsystem, and begin managing the network based on information extracted during step #2.

Single Subsystem Network with Unique Device Types (Example 1)

This example describes a simple network with two devices that have unique program IDs. In this example, Location Strings are not needed because the program IDs are enough to uniquely identify each device.

1. Start the LonMaker tool and create a new network. Do not attach to the network. Make sure that the Device Manager DM plug-in is registered. (See the *LonMaker for Windows Integration Tool User's Guide* for information on how to create networks using the LonMaker tool)
2. Select the *Domain* tab of the *Network Properties* dialog box to set the domain for this network. (The Network Properties dialog can be accessed by selecting *Network Properties…* from the *LonMaker* drop down menu.) Choose the domain value that will be used for the network. This value is used when servicing the network remotely.
3. From the *LonPoint Shapes* stencil drag and attach two devices (a LonPoint DI module, and a LonPoint DO module) to the default channel that the LonMaker tool provides upon start up.

4. From the *Device Manager Shapes* stencil drag and attach a Device Manager device to the default channel that the LonMaker tool provides upon start up.

5. From the *LonPoint Shapes* stencil drag and drop a *Digital Input* functional block to the LonMaker network drawing. **When prompted, be sure to specify the DI and not the DM as the source of the input.**

6. From the *LonPoint Shapes* stencil drag and drop a *Digital Output* functional block to the LonMaker drawing. When prompted, select the output from the only device that has outputs – the DO module.

7. Connect the input to the output. The drawing should look something like Figure 5.1.

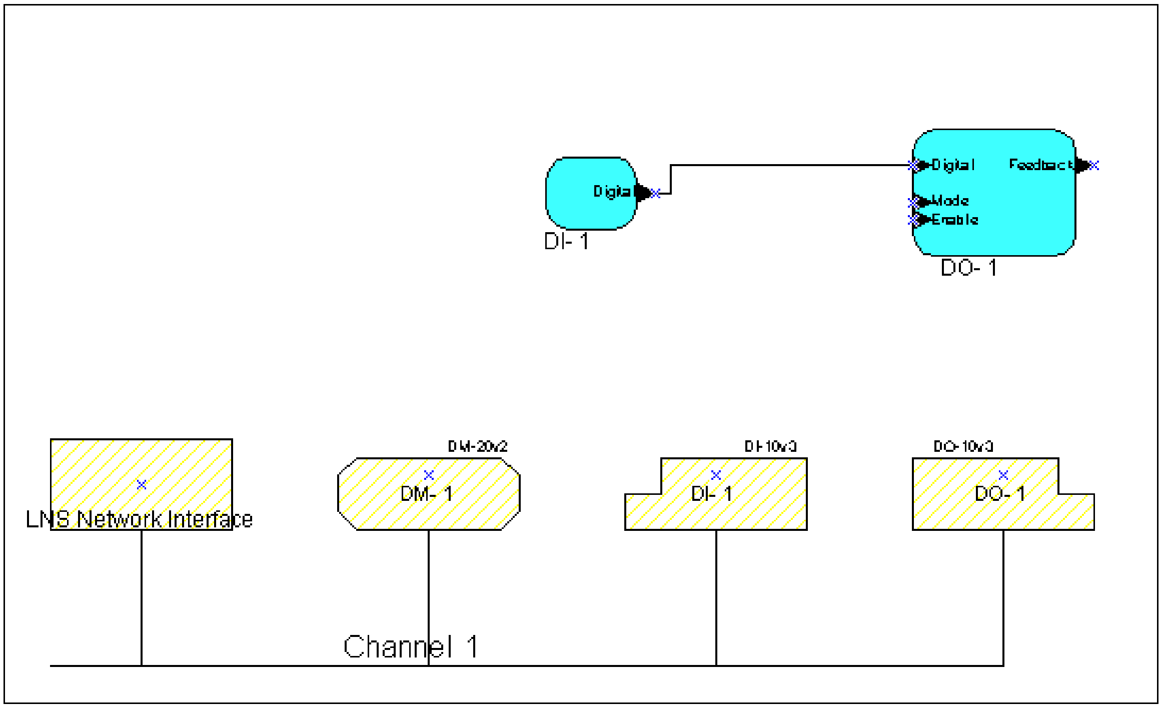


Figure 5.1 – LonMaker Network for Example 1

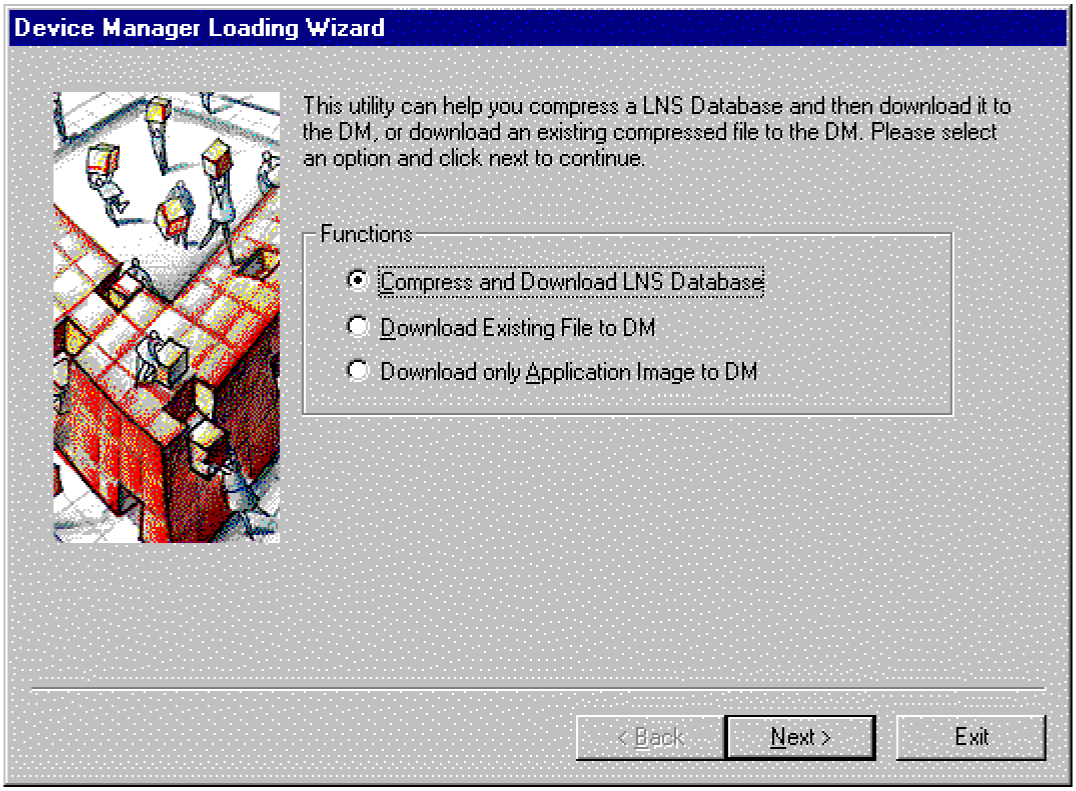
All the devices in the drawing exist in subsystem 1. When the loading wizard compresses the database, only subsystem 1 will be selectable; no other subsystems are specified in the drawing.

The Device Manager is part of the network and therefore must appear as a device in subsystem 1 since the devices it will manage are in subsystem 1.

Although the Network Service Device (NSD) appears in the subsystem, it will be ignored by the Device Manager.

All four devices in the network have different program IDs so it was not necessary to define the location string for any of the devices.

8. Start the Device Manager Loading Wizard by selecting it from START | PROGRAMS | Echelon Device Manager | Device Manager Loading Wizard. The window shown in figure 5.2 will appear.



Device Manager Loading Wizard

This utility can help you compress a LNS Database and then download it to the DM, or download an existing compressed file to the DM. Please select an option and click next to continue.

Functions

● Compress and Download LNS Database

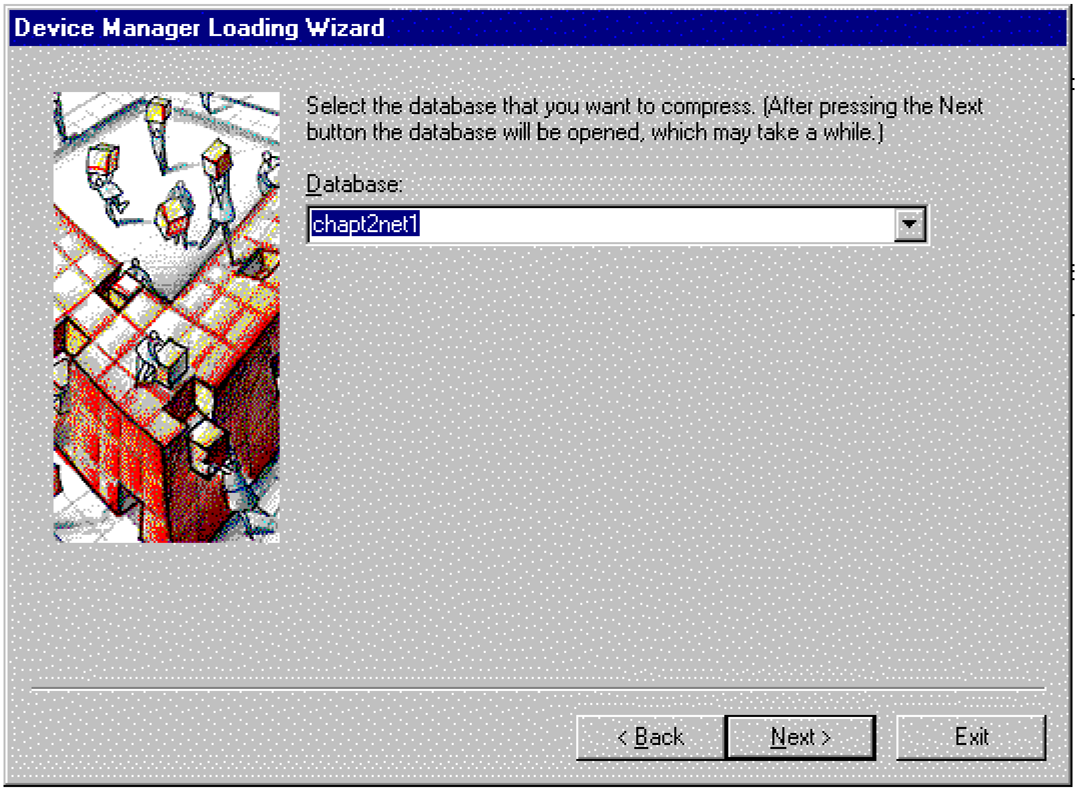
○ Download Existing File to DM

○ Download only Application Image to DM

Figure 5.2 – Device Manager Loading Wizard

From the window, an existing database may be compressed, an existing file may be downloaded to the Device Manager, or an application image may be downloaded to the Device Manager. To compress the database created in steps 1 through 7, choose the first option, *Compress and Download LNS Database.*

9. Click Next. The window shown in figure 5.3 will appear.



Device Manager Loading Wizard

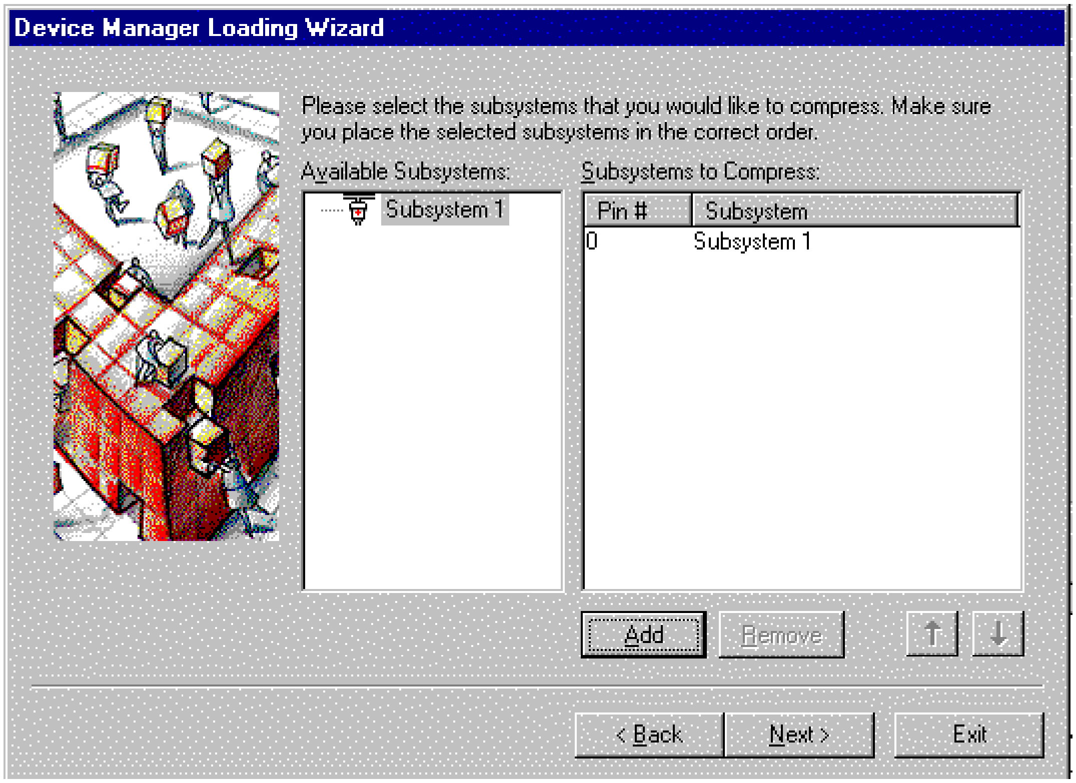
Select the database that you want to compress. (After pressing the Next button the database will be opened, which may take a while.)

Database: chapt2net1

Figure 5.3 – Select Database

The dropdown menu contains a list of all of the LONWORKS databases stored on the PC. Select the network database created in step 1 to compress and load.

10. Click Next. The window shown in figure 5.4 will appear.



Please select the subsystems that you would like to compress. Make sure you place the selected subsystems in the correct order.

Available Subsystems:

- Subsystem 1

Subsystems to Compress:

Pin #	Subsystem
0	Subsystem 1

Figure 5.4 – Select Subsystems

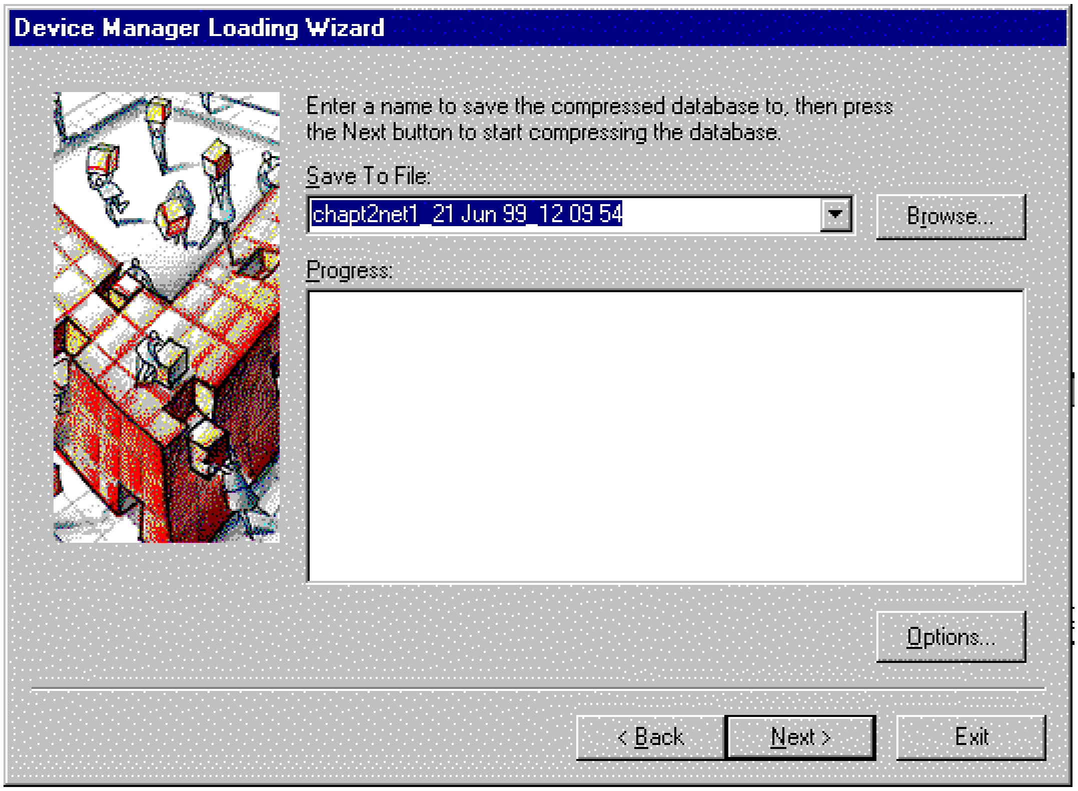
Select which subsystems are used to create the managed device list. For this simple network only one subsystem is listed in the left pane. In a more complex network the left pane is used to navigate the subsystem hierarchy to select the subsystems to be loaded.

11. Highlight the existing subsystem, and click Add.

The subsystem will be copied to the right pane indicating that it has been selected for compression and load. This first subsystem is assigned a PIN value of 0; setting the Device Manager's 4-bit subsystem selection input to zero at runtime selects this subsystem.

When the Device Manager is loaded with only a single subsystem it is not necessary to supply a 4-bit input. By default, the Device Manager's `DM_Pin_Usage` configuration property is set to *Use Subsystem Zero* which causes the Device Manager to select the first subsystem in its managed device list.

12. Click Next. The window shown in figure 5.5 will appear:



Device Manager Loading Wizard

Enter a name to save the compressed database to, then press the Next button to start compressing the database.

Save To File: chapt2net1_21 Jun 99_12 09 54

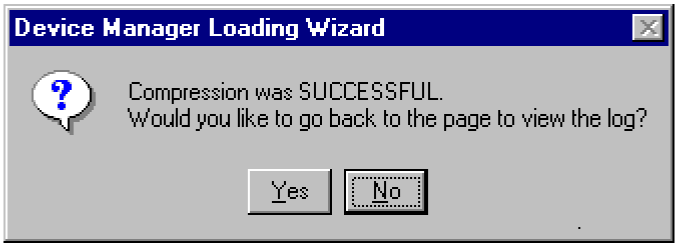
Progress:

Figure 5.5 – Select Database Name

Choose a name for the managed device list file. The Loading Wizard will suggest a name consisting of the database name and current date and time. By default this file will be saved in the LONWORKS `DM\CompressedDB` directory. This path can be changed by clicking on the *Browse…* button.

For convenience, the file is saved on the hard disk so that it can be re-used later if necessary (see step 8 in this example). The database may also be compressed each time a Device Manager is loaded, however, using the saved file can save time in a production environment.

13. Click Next. The Loading Wizard will now compress the selected database. The progress of the compression will be displayed in the *Progress* pane. When compression has been completed the dialog box in figure 5.6 indicates success or failure.

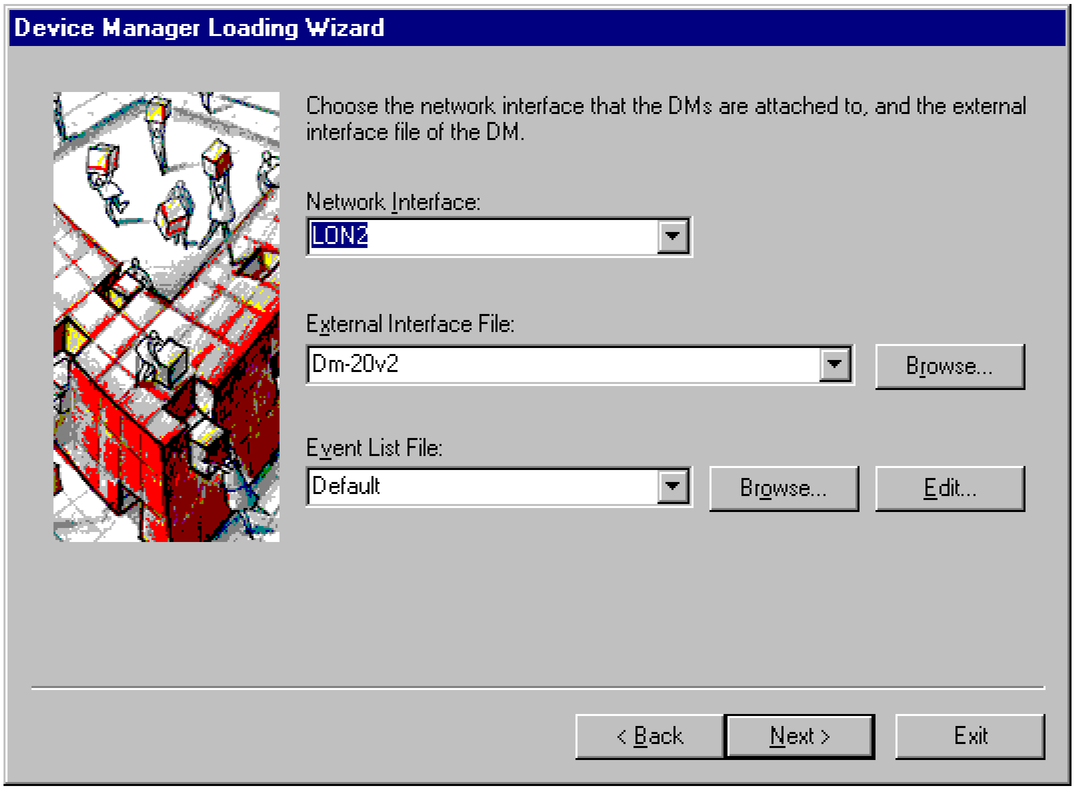


Device Manager Loading Wizard

Compression was SUCCESSFUL.
Would you like to go back to the page to view the log?

Figure 5.6 – Confirmation Dialog

14. Click No to move past the dialog box. The window in figure 5.7 will appear:



Device Manager Loading Wizard

Choose the network interface that the DMs are attached to, and the external interface file of the DM.

Network Interface: LON2

External Interface File: Dm-20v2

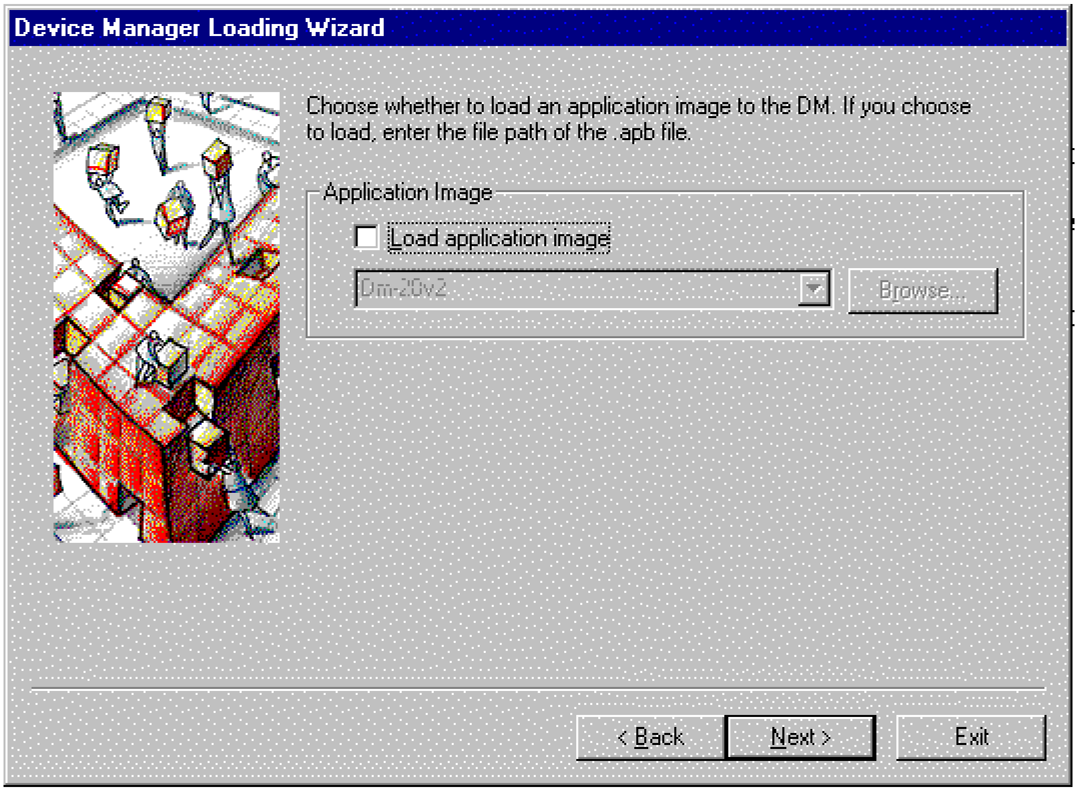
Event List File: Default

Figure 5.7 – Select Network Interface/XIF

This window allows the compressed database to be loaded into a Device Manager. Select the LONWORKS communication interface and the External Interface File (.XIF extension) for the Device Manager version being used. The Loading Wizard will automatically suggest using the XIF that corresponds with the latest version of Device Manager firmware on the PC. This should not be changed unless a Device Manager with an older firmware version is being loaded. It is recommended that the Device Manager be loaded with the latest firmware version.

This window also allows the Event List used by the Device Manager to be modified (see Chapter 7). Use the Default event list file for this example.

15. Click Next. The window in figure 5.8 will appear.



Device Manager Loading Wizard

Choose whether to load an application image to the DM. If you choose to load, enter the file path of the .apb file.

Application Image

□ Load application image

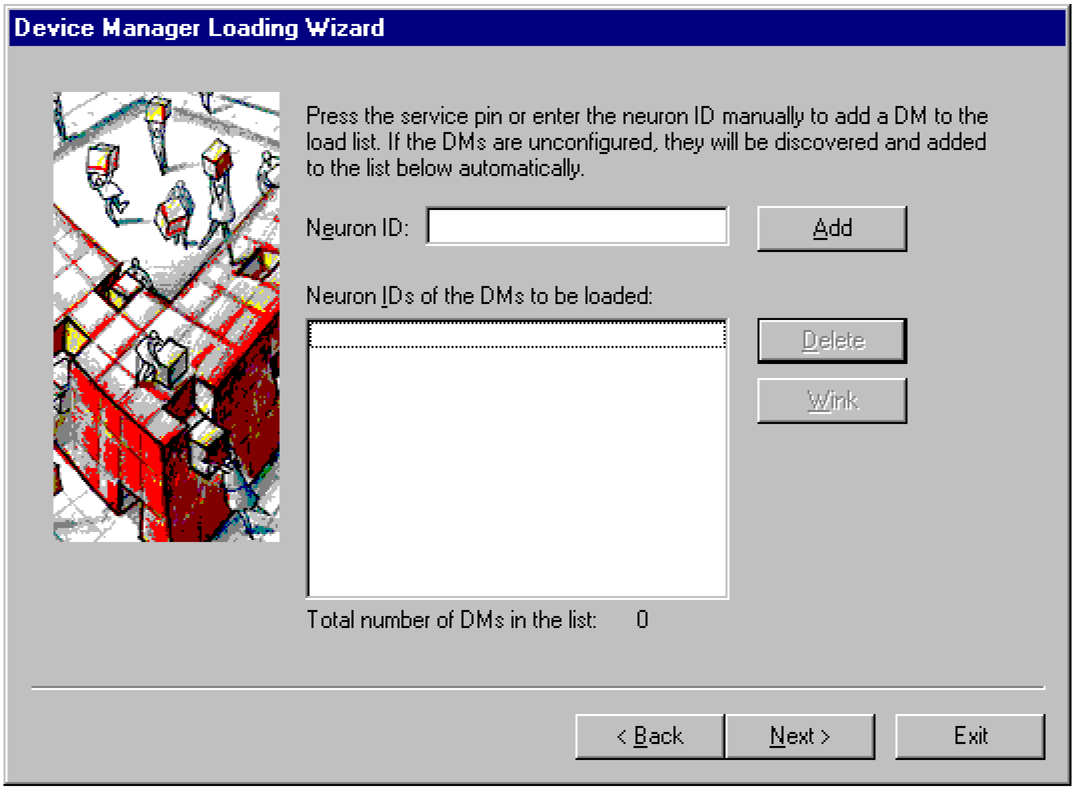
Dm-20v2

Figure 5.8 – Load Application Image

Device Managers are shipped with the firmware pre-loaded, however, this option can be used to load a different firmware version into the Device Manager should this be desirable.

For this example, deselect the check box, as the goal is to load the compressed file and not the firmware.

16. Click Next. The window in figure 5.9 will appear:



Press the service pin or enter the neuron ID manually to add a DM to the load list. If the DMs are unconfigured, they will be discovered and added to the list below automatically.

Neuron ID:

Neuron IDs of the DMs to be loaded:

Total number of DMs in the list: 0

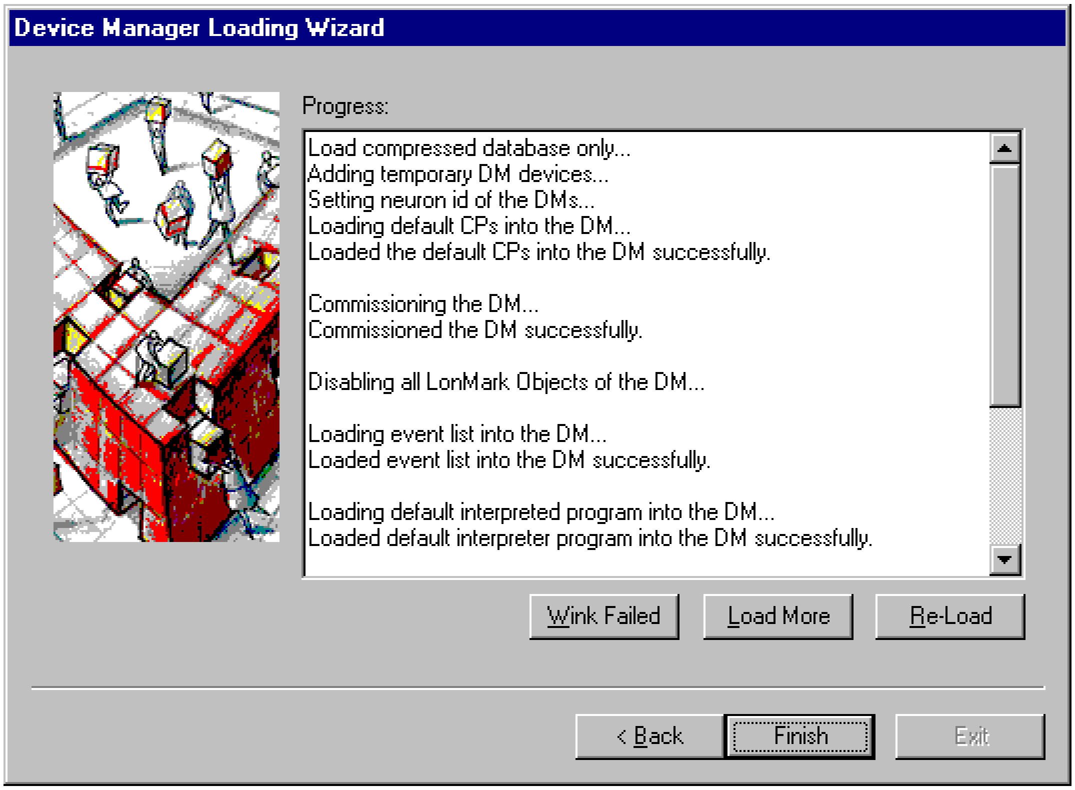
Figure 5.9 – Press Service Pin

The Neuron ID of the Device Manager can be entered manually or acquired by pressing the service pin on the Device Manager. Each time the service pin on a Device Manager is pressed its Neuron ID will be added to the list box. Likewise if a Neuron ID is entered manually and the *Add* button is clicked the Neuron ID will be added to the list box.

Multiple Neuron IDs may be entered to load the compressed LONWORKS database into several Device Managers. The loading wizard will then load the list of collected Neuron IDs into each Device Manager.

Pressing the service pin on a Device Manager will add its Neuron ID to the list box.

17. Click Next. The window in figure 5.10 will appear:



```
Load compressed database only...
Adding temporary DM devices...
Setting neuron id of the DMs...
Loading default CPs into the DM...
Loaded the default CPs into the DM successfully.

Commissioning the DM...
Commissioned the DM successfully.

Disabling all LonMark Objects of the DM...

Loading event list into the DM...
Loaded event list into the DM successfully.

Loading default interpreted program into the DM...
Loaded default interpreter program into the DM successfully.
```

Figure 5.10 - Status

It may take several minutes to load a Device Manager depending on the size of the managed device list and whether the Device Manager's application image is also being loaded. Loading progress is displayed by the loading wizard. When all Device Managers are loaded, additional Neuron IDs can be collected, the existing Device Managers can be reloaded, or the loading wizard can be exited.

18. Click Finish to exit the loading wizard.

19. Attach the Device Manager to the LONWORKS network. The Device Manager will automatically begin finding and installing the nodes defined in the LonMaker network drawing.

Testing the Example Device Manager

Test the example Device Manager using the following steps:

1. Attach the LonPoint DI-10 and DO-10 modules to the network. Connect LEDs to the DO-10 module and switches to the DI-10 module.

2. Supply power to the DI-10 and DO-10 modules. Ensure that both modules are in the unconfigured state, i.e., the service LEDs are blinking. If the service LEDs are not blinking, force the LonPoint modules into unconfigured states

by pressing and holding the service switches for 10 seconds (see the *LonPoint Module Hardware & Installation Guide* for more details).

3. Attach and power the Device Manager to the network.

4. Note that the Device Manager's status LED blinks 4 or 5 times per interval. 4 blinks means that no database is selected. 5 blinks means that the Device Manager object is disabled or the DM MODE has been set to disable discovery or commissioning (see *Status LED* in Chapter 4 for more information about this LED). When the Device Manager has completed network installation, the status LED will blink only once per interval. The DI-10 and DO-10 modules reset as the Device Manager commissions them.

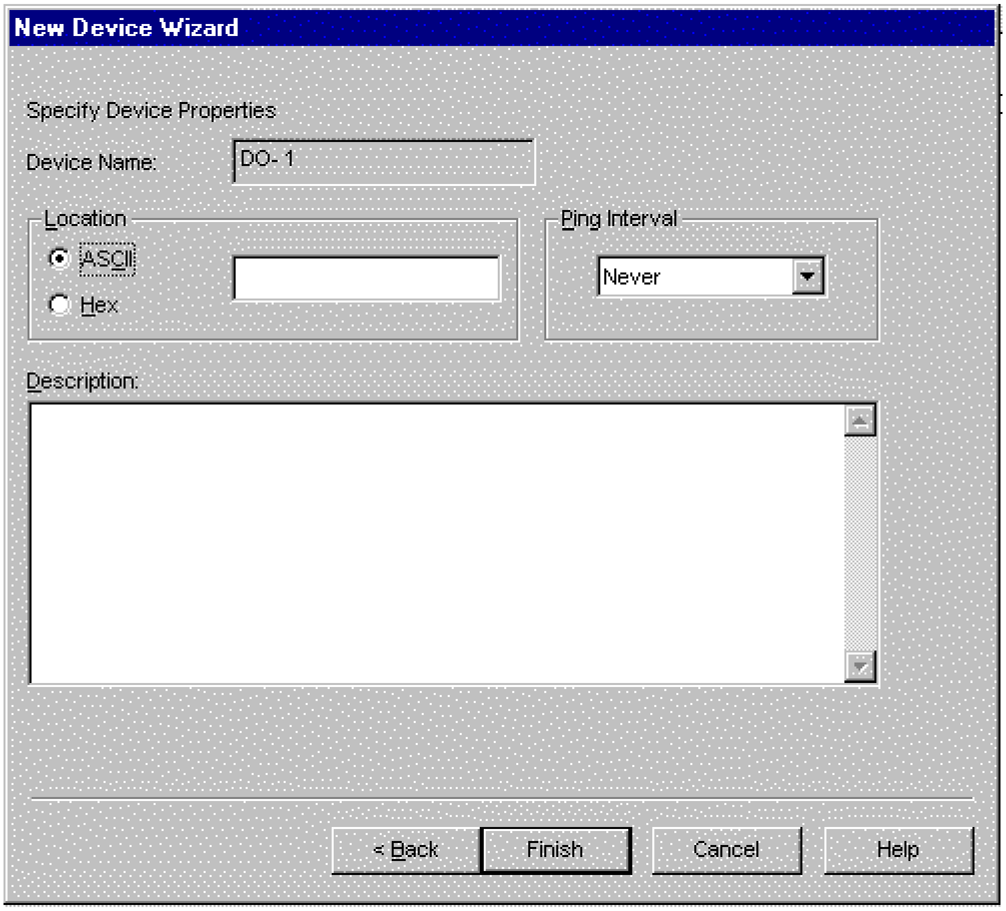
Single Subsystem Network with Ambiguous Devices (Example 2)

In this example a simple network will be created with three devices: a LonPoint DI-10 and two LonPoint DO-10 modules. Since the two DO-10 modules have identical Program IDs, location strings must be used to differentiate between the two DO-10 devices. For the purposes of this example, it is assumed that the reader has completed example 1 above.

1. Start the LonMaker for Windows Integration Tool and create a new network. Do not attach to the network. Make sure that the DM plug-in is registered.

2. Set the domain for the network.

3. From the *LonPoint Shapes* stencil drag and attach the three devices (a LonPoint DI-10 and two LonPoint DO-10s) to the default channel that the LonMaker tool provides upon start up. As each shape is dropped the LonMaker tool's device wizard can be used to set the Location String for the device using the window shown in figure 5.11.



New Device Wizard

Specify Device Properties

Device Name: DO- 1

Location

● ASCII

○ Hex

Ping Interval

Never

Description:

Figure 5.11 – LonMaker Device Creation Wizard

A device's location string may be at most 6 characters in length. If a 4 character location string is entered, the LonMaker tool will fill the remaining characters with the HEX value 00. Choose a 6 character ASCII location for each of the LonPoint DO-10 modules, e.g., "LEFTDO" and RGHTDO".

4. From the *Device Manager Shapes* stencil drag and attach a DM-21 Device Manager to the default channel that the LonMaker tool provides upon start up.

5. From the *LonPoint Shapes* stencil drag and drop two *Digital Input* functional blocks in the LonMaker network drawing (use Digital Input[0] and Digital Input[1] from the DI-10 device).

6. From the *LonPoint Shapes* stencil drag and drop a *Digital Output* functional block to the drawing (use Digital Output[0] from the DO-10 device labeled LEFTDO).

7. From the *LonPoint Shapes* stencil drag and drop a *Digital Output* functional block to the LonMaker drawing (use Digital Output[0] from the DO-10 device labeled RGHTDO).

8. Connect the Digital Input[0] to LEFTDO's functional block, and Digital Input[1] to RGHTDO's functional block.

The drawing should look like figure 5.12.

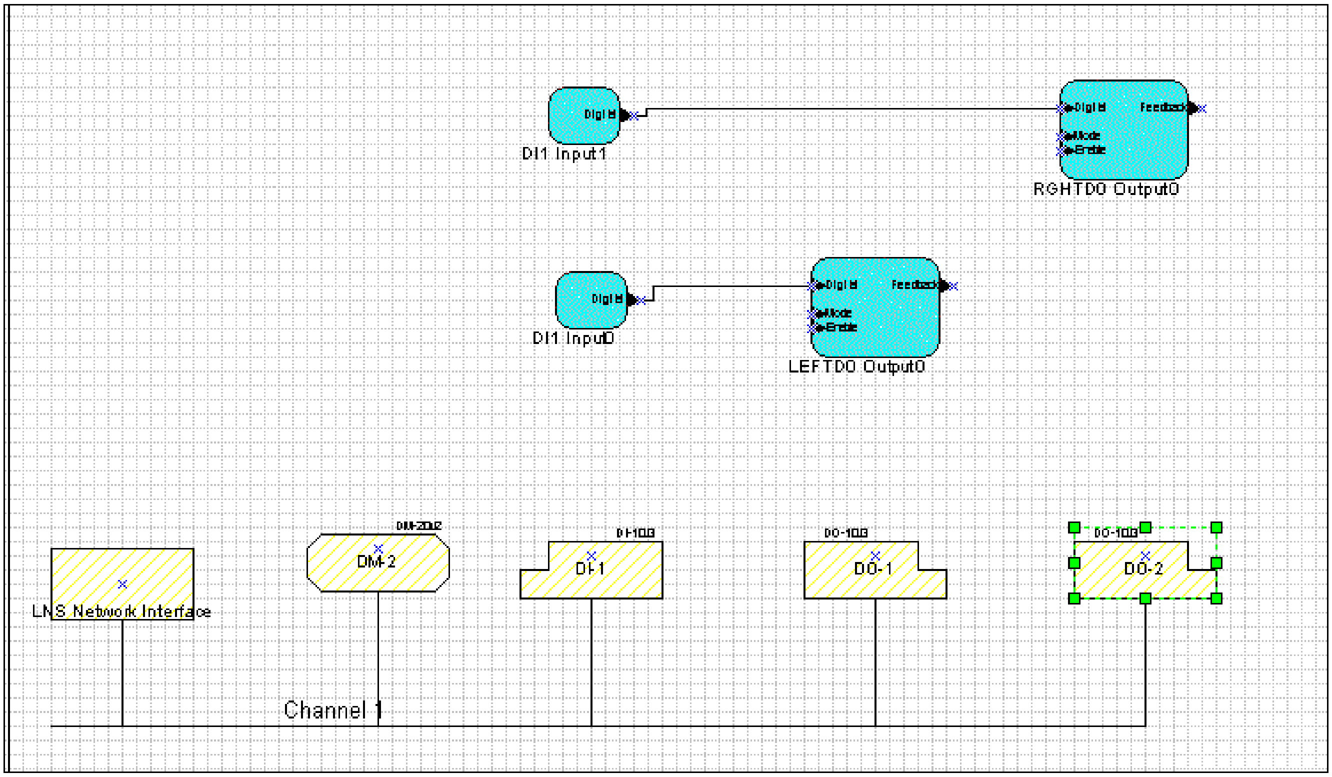


Figure 5.12 – LonMaker Network Drawing for Example 2

9. Use the Loading Wizard to load the database into a Device Manager (following steps 8 to 18 in Example 1, earlier in this chapter).

10. Physically install the LonPoint DI-10 and DO-10 modules.

11. Using the demonstration location string setting tool available on the Echelon web-site (http://www.echelon.com/products/dm), or an equivalent tool, set the location strings of both DO-10 modules.

12. Attach the Device Manager to the network. The Device Manager will find the DI-10 and both DO-10 modules, query the locations of the DO-10 modules and commission the network.

- Unlike Example 1, Example 2 contains two devices of the same type, namely the two LonPoint DO-10 modules. To differentiate between the two DO-10 modules, the Device Manager relies upon the location string present in the DO-10 module to match the location string specified in the drawing.

- An example of such a tool that sets individual location strings and runs on a PC is available at http://www.echelon.com/products/dm.

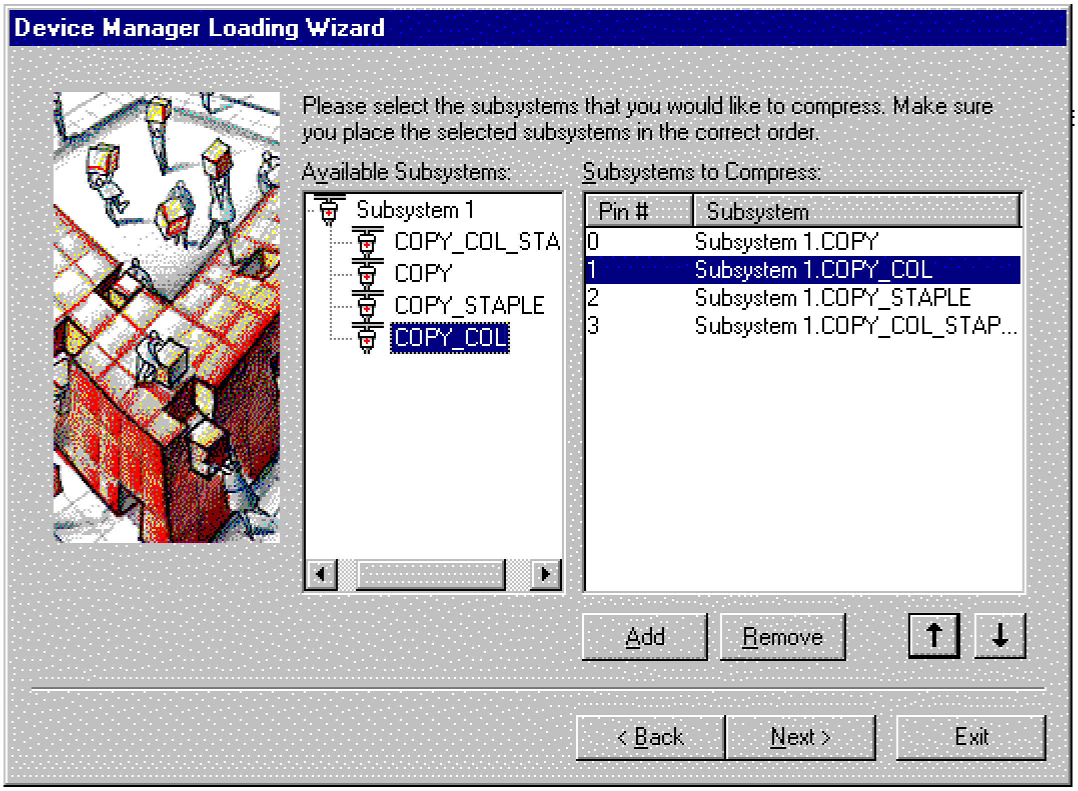
Multi Subsystem Network Using a Single Device Manager (Example 3)

This example consists of 4 subsystems each with varying numbers of devices and builds on the photocopier example described in Chapter 1.

1. Download the example photocopier network from www.echelon.com/products/dm. This network is not intended to be operable, however, it is sufficient to illustrate the use of the Loading Wizard when designing a network containing multiple subsystems.

2. Use the LonMaker for Windows Integration Tool to restore the archived network. Browse the network and observe the subsystem hierarchy.

3. Using the Device Manager Loading Wizard, include all four subsystems in the managed device list as shown in figure 5.13.



Device Manager Loading Wizard

Please select the subsystems that you would like to compress. Make sure you place the selected subsystems in the correct order.

Available Subsystems:

Subsystems to Compress:

Pin #	Subsystem
0	Subsystem 1.COPY
1	Subsystem 1.COPY_COL
2	Subsystem 1.COPY_STAPLE
3	Subsystem 1.COPY_COL_STAP...

Figure 5.13 – Subsystem Selection

Note which PIN is associated with each subsystem. When the Device Manager is installed, set the correct 4-bit subsystem selection input on the Device Manager's input pins. Reorder the subsystems using the arrow buttons if desired.

4. Continue compressing and loading the Device Manager as shown in steps 8 to 18 in Example 1, earlier in this chapter.

- Examples 1 and 2 use single subsystems so the 4-bit subsystem selection input to the Device Manager was not necessary at runtime for subsystem selection. Example 3 has 4 subsystems, each containing a Device Manager and all the nodes required for configuration of the photocopier. Depending on the desired photocopier configuration, a different subsystem is selected using the 4-bit subsystem selection input.

- On power-up, the Device Manager will first check its *DM Pin Usage* configuration property to determine whether subsystem 0 or a different subsystem should be used. In Example 3, the *DM Pin Usage* configuration property was set to *Use IO pins as DB number* so that the Device Manager would select a subsystem based on the 4-bit subsystem selection input. More information on *DM Pin Usage* is available in the Device Manager plug-in description below.
- See Chapter 4 for more information on configuring the 4-bit subsystem selection input.

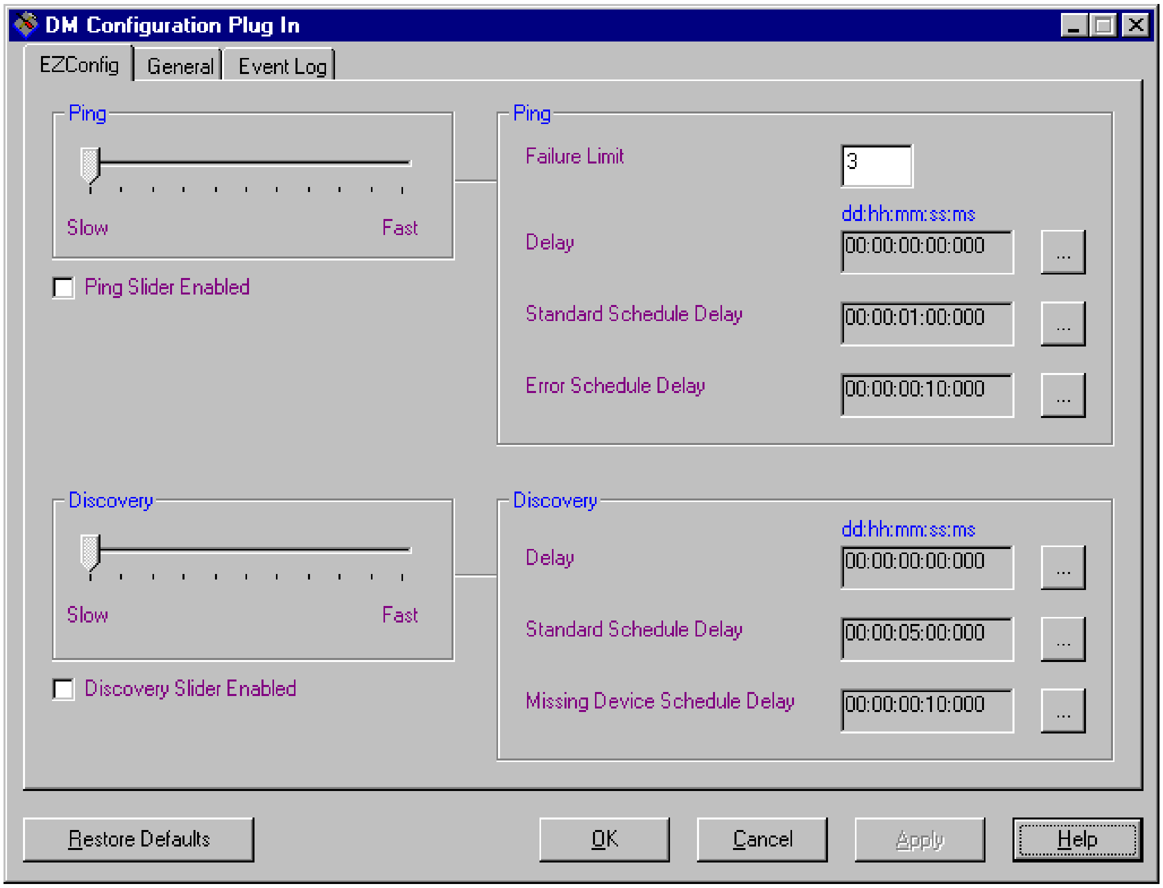
Configuring the Device Manager

The Device Manager has many configuration properties that are exposed through the Device Manager plug-in. The Device Manager plug-in allows parameters to be set that control (among other things) how often and how quickly the Device Manager will try to discover and ping devices when the Device Manager is attached to the network. The available properties are grouped by function and presented in a tabular format.

To use the Device Manager plug-in right click on the Device Manager device shape and select *Configure* from the shortcut menu. This will open a three-tab dialog box with the heading DM Configuration Dialog. Ensure that the plug-in was registered when the network was created in step 1; if not, it may be registered now by using the Register Plug-In tab of the Network Properties dialog box. Open the Network Properties dialog box by clicking on *Network Properties* menu option from the *LonMaker* menu. The three tabs of this plug-in are described in the following sections.

EZConfig

The `EZConfig` tab appears in figure 5.14.



DM Configuration Plug In

Ping

Slow — Fast

□ Ping Slider Enabled

Ping

Failure Limit: 3

dd:hh:mm:ss:ms

Delay: 00:00:00:00:000

Standard Schedule Delay: 00:00:01:00:000

Error Schedule Delay: 00:00:00:10:000

Discovery

Slow — Fast

□ Discovery Slider Enabled

Discovery

dd:hh:mm:ss:ms

Delay: 00:00:00:00:000

Standard Schedule Delay: 00:00:05:00:000

Missing Device Schedule Delay: 00:00:00:10:000

Figure 5.14 – DM Configuration Plug In, EZConfig Tab

This tab is used to set the speed at which the Device Manager tries to discover and ping devices using the following properties:

Ping and Discovery Slide Bars

Use the slide bars to set aggressive (fast) or relaxed (slow) discovery and ping intervals or specify specific values for the various configuration properties.

If the `Slider Enabled` checkbox is selected, the corresponding `Delay` values will be set based on the position of the slider. If this checkbox is cleared, the `Delay` values can be set manually (see below). Generally, it is better to use the slide bars and allow the plug-in to set reasonable values than to enter specific values.

Ping: Failure Limit

This function sets the number of times that a logical device must fail to respond to ping before the Device Manager forgets about the device's Neuron ID. 0 means the Device Manager will always remember the device's Neuron ID. Forgetting a Neuron ID means that the Device Manager will start a discovery interval to find a replacement for whatever device is now considered missing.

Ping: Delay

This parameter is used to throttle the pinging (testing) of devices. It represents the delay between pinging individual devices. Click the ... button to set this property.

Ping: Standard Schedule Delay

Under normal circumstances, this parameter controls the delay imposed after pinging all devices on the network before starting to ping again. If ping errors have been detected, the *Ping: Error Schedule Delay* parameter controls this behavior instead. Click the ... button to set this property.

Ping: Error Schedule Delay

This parameter controls the delay imposed before pinging recommences after attempting and failing to ping all devices on the network. If no ping errors occurred in the pinging cycle, the *Ping: Standard Schedule Delay* parameter controls this behavior instead. Click the ... button to set this property.

Discovery: Delay

This parameter is used to throttle the discovery of devices. It represents the delay between discovery messages on a per device basis (in other words, how soon the Device Manager tries to discover another device after discovering a device). Click the ... button to set this property.

Discovery: Standard Schedule Delay

Under normal circumstances, (all required devices have been discovered) this parameter controls the delay imposed after discovery has been completed and before a new discovery cycle is started. If a specified device does not exist, the *Discovery: Missing Device Schedule Delay* parameter controls this behavior instead. Click the ... button to set this property.

See Temporary Devices, in Chapter 8 for more information on how to designate a device as not required.

Discovery: Missing Device Schedule Delay

If one or more required devices are missing, this parameter controls the delay imposed after discovery is complete before starting a new discovery. If all devices have been installed, the *Discovery: Standard Schedule Delay* parameter controls this behavior instead. Click the ... button to set this property.

See Temporary Devices, in Chapter 8 for more information on how to designate a device as not required.

Restore Defaults

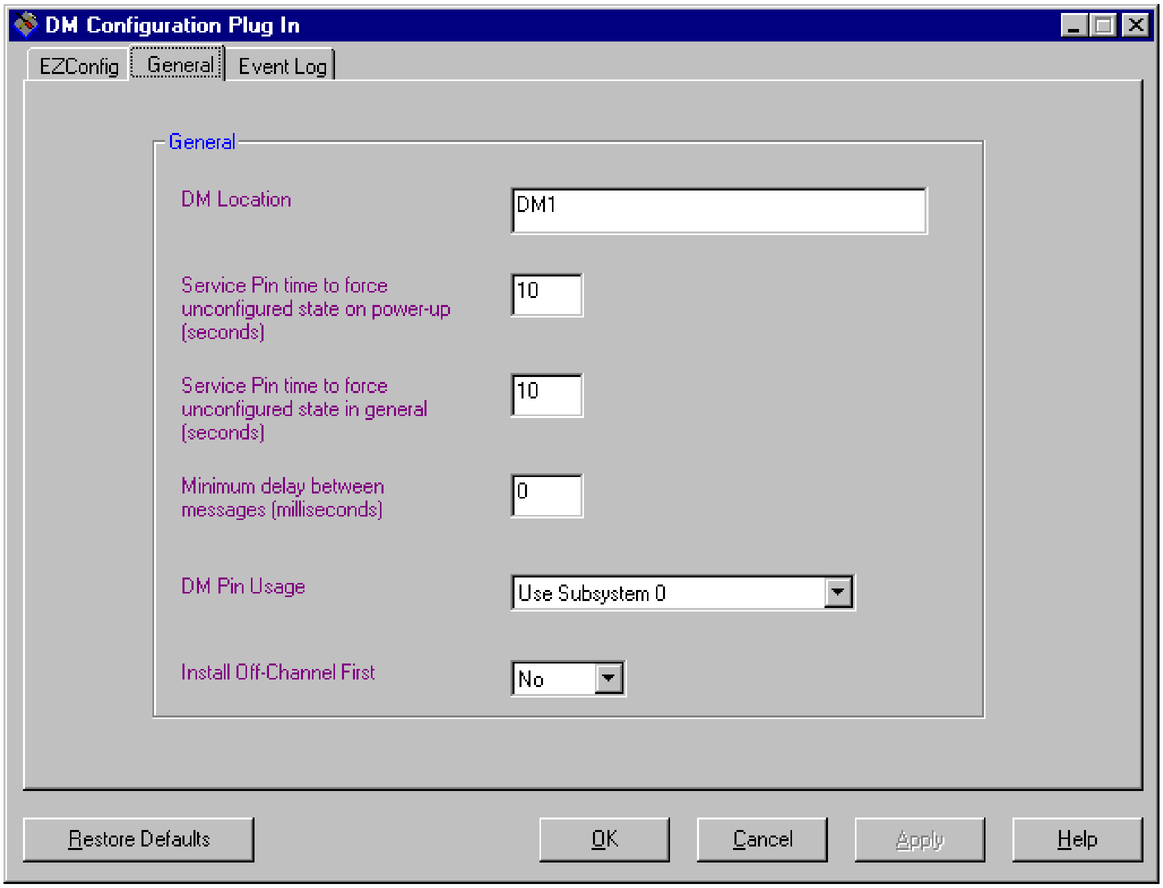
The restore defaults button may be used at any time to restore all configuration property values to their factory defaults**.**

Note:

The *Restore Defaults* button will restore all values on all tabs of the plug-in.

General

The `General` tab appears in figure 5.15.



DM Configuration Plug In

General

DM Location: DM1

Service Pin time to force unconfigured state on power-up (seconds): 10

Service Pin time to force unconfigured state in general (seconds): 10

Minimum delay between messages (milliseconds): 0

DM Pin Usage: Use Subsystem 0

Install Off-Channel First: No

Figure 5.15 - DM Configuration Plug In, General Tab

This tab allows following properties to be set:

DM Location

DM Location is the standard LONMARK location configuration property and is unrelated to the device or Device Manager location strings described above. See the *LONMARK Application Layer Interoperability Guidelines* for details about this configuration property.

Service pin time to force unconfigured state on

The amount of time in seconds that the service pin must be pressed and held to cause the Device *power-up*

Manager to go unconfigured. This feature provides a means of placing the Device Manager in a known state. To cause the Device Manager to go unconfigured, power off the Device Manager, press and hold the service pin, and then power on the Device Manager while still holding the service pin. After the number of seconds defined by this configuration property, the Device Manager will enter the unconfigured state. Setting this value to 0 disables this feature.

Service pin time to force unconfigured state in general

Similar to the functionality described above except that power cycling the Device Manager is not required force the unconfigured state. Simply press and hold the Device Manager service pin for the number of seconds defined by this configuration property to cause the Device Manager to enter the unconfigured state. Setting this value to 0 disables this feature.

Minimum delay between messages

The delay in milliseconds inserted between all messages the Device Manager sends on the network. This feature is useful in situations where it is necessary to limit the bandwidth used by the Device Manager by inserting a delay between ***each*** message.

The cumulative effect of this parameter is significant. For example, assume that it requires 20 to 100 messages to discover and commission a single device. If the minimum delay between messages is set to 100ms, and it requires 50 messages to commission a device, 5 seconds will be added to the commissioning time of each device. In a network of 30 devices, this delay will add at least two minutes of additional overhead. Generally this parameter should be set to zero to allow the Device Manager to operate as quickly as possible.

DM Pin Usage

DM Pin Usage determines how the Device Manager's subsystem selection IO pins are used. Setting this property to `Use subsystem 0` tells the Device Manager that it will be loaded with a managed device list that contains a single subsystem. The Device Manager will always apply this subsystem to the network. Setting this property to `Use IO Pins as DB number` tells the Device Manager that it will be loaded with a managed device list that contains several subsystems and that the Device Manager should read its 4-bit subsystem selection input to determine which subsystem to apply to the network.

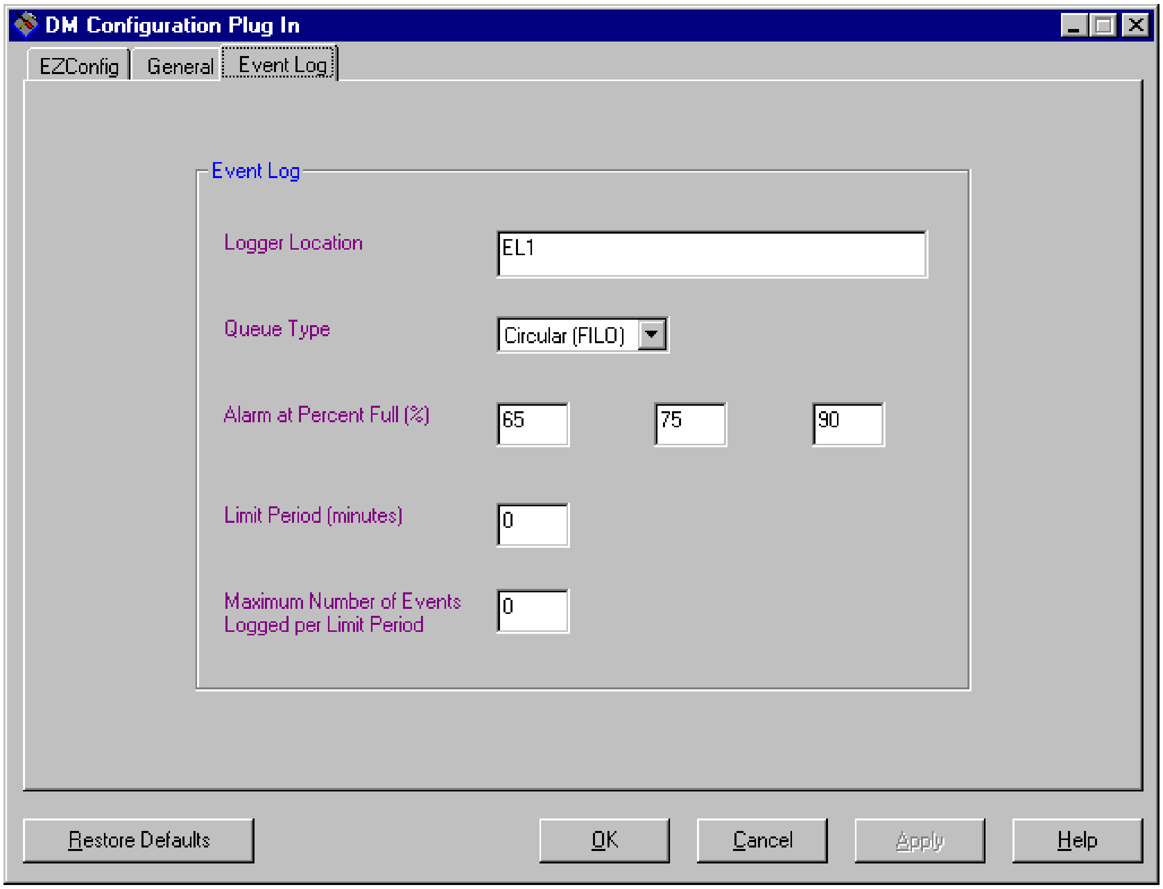
See Chapter 4 for more information on the 4-bit subsystem selection input feature.

Install Off-Channel First

This configuration property is only applicable in systems where the Device Manager is responsible for managing a router. In such systems, on power-up, the Device Manager will first search (try to discover) for a router. No devices will be commissioned until the router is found and commissioned. Once the router is commissioned this parameter governs whether the Device Manager installs the local channel first or the far channel first.

Event Log

The `Event Log` tab appears in figure 5.16:



Event Log

Logger Location: EL1

Queue Type: Circular (FILO)

Alarm at Percent Full (%): 65 75 90

Limit Period (minutes): 0

Maximum Number of Events Logged per Limit Period: 0

Figure 5.16 - DM Configuration Plug In, Event Log Tab

This tab allows the following properties to be set:

Logger Location

Logger Location is the standard LONMARK location configuration properties for the logger object embedded in the Device Manager, and is unrelated to the device or Device Manager location strings as described above. See the *LONMARK Application Layer Interoperability Guidelines* for details about this configuration property.

Queue Type

Determines whether the Device Manager event log is set up as a circular or linear queue.

A circular queue overwrites the oldest data point once the available memory for the queue is reached, maintaining only the most recent events or data points.

A linear queue does not overwrite old events or data points, and instead writes data until the queue is full. Once full, no more data points are logged, thereby retaining the old data-points.

Alarm at Percent Full

The datalogger object can announce to the network that its internal log is becoming full by using the `DL_Level_Full` network variable. Setting this configuration property causes the Device Manager to send `DL_Level_Full` when the event log reaches the specified percentage full marks. One `DL_Level_Full` message is sent as the event log fills past each mark. No messages are sent if the event log is defined to be a circular queue.

For example, if this property is set to values of 65, 75, and 90, as the log fills three `DL_Level_Full` network variable updates will occur: one when the log is 65% full; one when the log is 75% full; and one when the log is 90% full.

Limit Period & Maximum number of events logged per limit period

These configuration properties limit the number of events logged within a given period. In some systems, failure conditions can cause the same event (or data point) to be logged repeatedly over a short period of time, filling the log with a single event or data point that is registered many times over a short period. Correctly setting these configuration properties minimizes this effect. By establishing a sliding window of time where only the defined number of events may be logged.

`Maximum number of events logged per limit period` is an integer that defines the maximum number of events that can be logged in the time period specified by `Limit Period`.

`Limit Period` is an integer representing the length in minutes of the logging limit.

6

Device Manager Commands, Reports, and Events

This chapter describes the various internal functions of the Device Manager that are exposed through the Device Manager's command, report and event interfaces.

Device Manager Functional Blocks

The Device Manager contains three functional blocks that can be dropped onto a LonMaker for Windows Integration tool network drawing: the Node object, the Device Manager, and the Event Log. The network variables on these functional blocks are connected and configured as described below.

Node Object (LM Object Index 0)

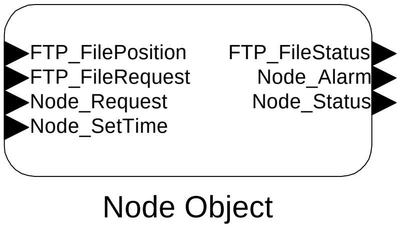


Figure 6.1 – Node Object Functional Block

The Device Manager contains a Node Object functional block (see Figure 6.1) with the network variables described in table 6.1.

Table 6.1 – Node Object Network Variables

NV Name	SNVT ID	Comments
`FTP_FileRequest`	73	`FTP_FileRequest`, along with `FTP_FileStatus` and `FTP_FilePosition` are used to implement interoperable file transfer. Refer to the *LONMARK Interoperable File Transfer Guidelines* for details.
`Node_Alarm`	88	Standard LONMARK alarm used to report under-range and over-range conditions.
`FTP_FilePosition`	90	`FTP_FilePosition`, along with `FTP_FileStatus` and `FTP_FileRequest` are used to implement interoperable file transfer. Refer to the *LONMARK Interoperable File Transfer Guidelines* for details.

`Node_SetTime`	84	Event log timestamps are derived from this reference. Set this network variable using an external device (such as the LonPoint SCH-10 Scheduler Module) to provide a time reference. This network variable should be updated from the external source once every few minutes to ensure synchronized timestamps between devices in the network. If this network variable is not initialized, the Device Manager will use an internal software clock to timestamp events. This internal clock is not synchronized to local time, however it provides the reader of the event log with a general indication of how much time has elapsed between events. If the Device Manager is reset, the clock will also be reset.
`Node_Status`	93	Standard LONMARK `Node_Status` network variable used to manage objects (disable, enable, etc). See the *LONMARK Application Layer Interoperability Guidelines* for complete details.
`FTP_FileStatus`	74	`FTP_FileStatus`, along with `FTP_FileRequest` and `FTP_FilePosition` is used to implement interoperable file transfer. Refer to the *LONMARK Interoperable File Transfer Guidelines* for details.
`Node_Request`	92	Standard LONMARK `Node_Request` network variable used to manage objects (disable, enable, etc). See the *LONMARK Application Layer Interoperability Guidelines* for complete details.

Device Manager (LM Object Index 14)

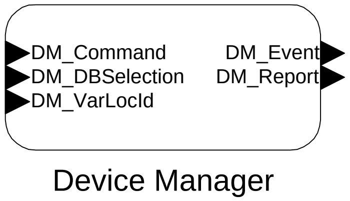


Figure 6.2 – Device Manager Functional Block

The core logic for managing the Device Manager network is implemented in the Device Manager functional block (see figure 6.2). The network variables on this functional block are described in table 6.2.

Table 6.2 – Device Manager Functional Block Network Variables

NV Name	SNVT ID	Comments
DM_VarLocId	8	RESERVED
DM_DBSelection	8	RESERVED
DM_Command	0	Input network variable that allows an external device to interact with the Device Manager. For example, sending the Device Manager a DM_QUERY_DEVICE_PROPS requests the Device Manager to produce a report on device properties.
DM_Report	0	Output network variable that allows the Device Manager to report status information to an external device.
DM_Event	0	Output network variable that the Device Manager can update to tell an external device of some event. See the *Event Log* section, later in this chapter, for more information.

DM_Command

This input network variable can be used by an external device to issue commands; such as report requests. This network variable is a union of several structs, each of which contains a command type as its first field. See dmobj.h in \LONWORKS\DM\include for more details.

Supported commands include:

DMCMD_QUERY_DM_PROPS

The DMCMD_QUERY_DM_PROPS command is used to cause the Device Manager to issue a DmProps report using the DM_Report output network variable. This command uses the DmQueryDmProps struct defined in dmobj.h.

DMCMD_QUERY_DEVICE_PROPS

The DMCMD_QUERY_DEVICE_PROPS command is used to cause the Device Manager to issue a DmDeviceProps report using the DM_Report output network variable. This command uses the DmQueryDeviceProps struct defined in dmobj.h.

DM_Report

This output network variable provides several reports in response to DM_commands. This network variable is a union of several structs, each element of which contains a report type as its first field. See dmobj.h in \LONWORKS\DM\include for more details.

Supported reports are:

DMREP_DM_PROPS

The DMREP_DM_PROPS report is issued by the Device Manager in response to DMCMD_QUERY_DM_PROPS command received on the DM_Command input network variable. This report uses the DmProps struct defined in dmobj.h.

DMREP_DEVICE_PROPS

The DMREP_DEVICE_PROPS report is issued by the Device Manager in response to DMCMD_QUERY_DEVICE_PROPS command received on the DM_Command input network variable. This report uses the DmDeviceProps struct defined in `dmobj.h`.

DMREP_DEVICE_DISCOVERED

The DMREP_DEVICE_DISCOVERED report is issued by the Device Manager each time a new device is found. If the auto-commission bit of the corresponding entry in the Managed Device List is set to zero (FALSE), commissioning does not occur. If the auto-commission bit is set to 1 (TRUE), the Device Manager updates this output network variable when a device is found and immediately commissions the newly discovered device.

A device is considered to have been *discovered* after the Device Manager receives the device's Neuron ID (either via the discovery process or via service pin message) and successfully reads its Program ID, Location String, and state variable. At this point, a DMREP_DEVICE_DISCOVERED report is generated and a DMEVNT_FOUND_xxx event is recorded in the Device Manager internal event log.

This report uses the DmDeviceDiscovered struct defined in `dmobj.h`.

DMREP_FATAL_ERROR

The DMREP_FATAL_ERROR report is issued by the Device Manager when the Device Manager encounters a fatal error and can no longer function. The internal event log may be examined to determine the cause of the failure. This report uses the DmFatalErrorReport struct defined in `dmobj.h`.

DM_Event

Whenever an event occurs, the Device Manager firmware writes an event code (`DmEventType`) to its internal event log, and optionally sends the event code on this network variable (see *Send Async* under *Event Log*, later in this chapter). Event codes are defined by the `DmEventType` enumeration (see *Event List*, below). The event log contains entries of the type `DmEventLogEntry`. See `dmobj.h` in `\LONWORKS\DM\include` for more details.

Event Log (LM Object Index 15)

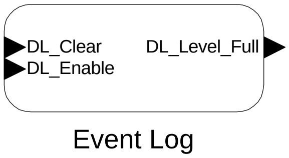


Figure 6.3 – Event Log Functional Block

The Device Manager contains an Event Log functional block (see figure 6.3) that captures network events which can be useful for network troubleshooting. The network variables on this functional block are described in table 6.3.

Table 6.3 – Event Log Functional Block Network Variables

NV Name	SNVT ID	Comments
DL_Clear_1	95	Setting this network variable to TRUE clears the event log. FALSE has no effect. Once clear, event logging resumes as events occur.
DL_Enable_1	95	Enables event tracking if set to TRUE, disables event tracking if set to FALSE
DL_Level_Full_1	21	Emitted as the event log fills depending on *Alarm at Percent Full* configuration property. An external device can also poll this network variable

Each event contains the following fields:

Event Num (1 byte) The enumerated value associated with this event as defined in dmobj.h.

Severity (1 byte) The user defined severity of the event. This value is kept only in the event list and is intended for use as documentation date, i.e., the value has no effect on the event. The default is 0 for all events. This value may be changed using the *Event List Editor*, described later in this chapter.

Send Async (1 byte) A TRUE or FALSE value which determines whether the event will be sent on the DM_Event network variable when the event is generated. If set to FALSE, generated events will be logged in the Device Manager event log (see *Event Log* later in this chapter) but not sent on the DM_Event network variable. If set to TRUE, generated events will be logged in the event log and sent on the Device Manager network variable. The default is FALSE for all events except DMEVNT_DEVICE_COMMISSIONED, for which the default is TRUE. This value may be changed using the *Event List Editor* described later in this chapter.

Parameter (2 bytes) The parameter returned by this event, if this event returns a parameter.

Text (30 bytes) The text associated with this event. This value may be changed using the *Event List Editor* described later in this chapter.

See *File Transfer* later in this chapter for details on how to read the event log.

Event List

The following events can be generated by the Device Manager:

DMEVNT_RESET

Event Number: 3

Parameter: None

Default Text: DM-20 reset

The DMEVNT_RESET event is issued when the Device Manager experiences a reset. This can occur due to a power cycle, power surge, transient, physical reset, or logical reset. The Device Manager Neuron Chip resets and the database selection algorithm is executed.

DMEVNT_REBOOT

Event Number: 4

Parameter: None

Default Text: DM-20 reboot

The DMEVNT_REBOOT event is issued when the Device Manager experiences a reboot. This can occur due to a physical reboot or logical reboot. The Device Manager Neuron chip reboots, the Device Manager self-installs, and the database selection algorithm is executed, but database information is retained.

DMEVNT_DATABASE_SELECTED

Event Number: 5

Parameter: Database ID

Default Text: Database selected

The DMEVNT_DATABASE_SELECTED event is issued when the database selection algorithm executes successfully. The Device Manger will then create its Managed Device List based on the selected database.

DMEVNT_INVALID_DATABASE_ID

Event Number: 6

Parameter: Invalid Database ID

Default Text: Invalid database ID

The DMEVNT_INVALID_DATABASE_ID event is issued when the database selection algorithm executes, but the database ID exceeds the number of subsystems. The Device Manger cannot proceed with creating the Managed Device List.

DMEVNT_FOUND_UNCNFG

Event Number: 8

Parameter: Device index of the found device

Default Text: Unconfigured device found

The DMEVNT_FOUND_UNCNFG event is issued when the Device Manager discovers an unconfigured device. If the device can be uniquely associated with a device in the database using its Program ID and location string, it is integrated into the system.

DMEVNT_FOUND_CNFG

Event Number: 9

Parameter: Device index of the found device

Default Text: Configured device found

The DMEVNT_FOUND_CNFG event is issued when the Device Manager discovers an configured device. If the device can be uniquely associated with a device in the database using its Program ID and location string, it is integrated into the system.

DMEVNT_MISSING

Event Number: 10

Parameter: Device index of the missing device

Default Text: Device missing

The DMEVNT_MISSING event is issued when a device in the system does not respond to a Device Manager ping. The device may be malfunctioning or out of communication. The Device Manager will no longer consider the device functioning, i.e., if another device is found which matches its Program ID and location ID, it will be integrated into the system.

DMEVNT_DEVICE_CONFIG_ERROR

Event Number: 11

Parameter: Device index of the device that exhibits the behavior

Default Text: Device config error

The DMEVNT_DEVICE_CONFIG_ERROR event is issued when a device in the system has the wrong state (should be configured-online), wrong subnet/node address, or wrong location string. The device may have been changed by another LNS tool. The device will be recommissioned by the Device Manager.

DMEVNT_DEVICE_COMMISSIONED

Event Number: 12

Parameter: Device index of the commissioned device.

Default Text: Device commissioned

The DMEVNT_DEVICE_COMMISSIONED event is issued when a device in the system is successfully commissioned by the Device Manager. When this event is generated, the device has been integrated into the network.

NOTE: The default Send Async value for this event is TRUE.

DMEVNT_DEVICE_COMMISSION_FAILED

Event Number: 13

Parameter: Device index of the device that failed to commission.

Default Text: Device commissioned

The DMEVNT_DEVICE_COMMISSION_FAILED event is issued when a device is discovered, but cannot be successfully commissioned. This could happen if the device loses power or goes applicationless during the commissioning process. The device is not integrated into the network, but if it remains attached and functional, the Device Manager will discover it and attempt to commission it again.

DMEVNT_FATAL_ERROR

Event Number: 16

Parameter: None

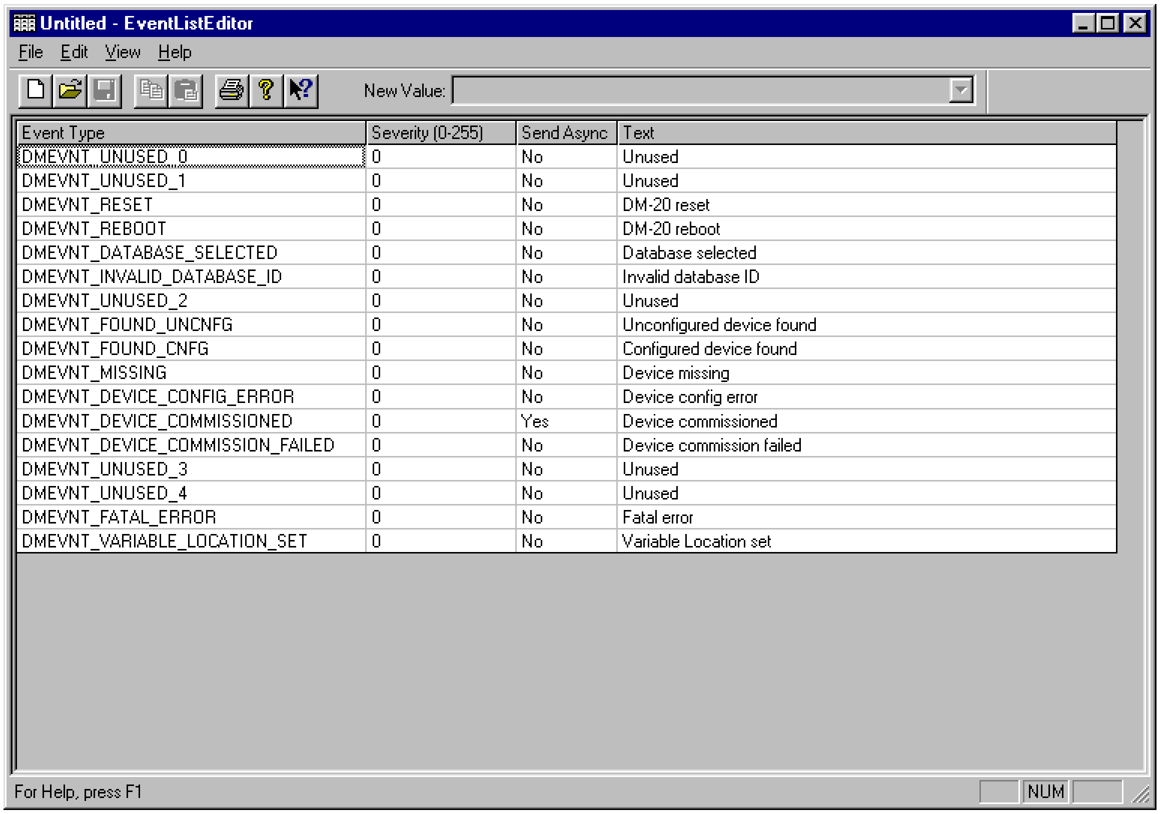
Default Text: Fatal error

The DMEVNT_FATAL_ERROR event is issued when the database algorithm executes successfully, but the database is corrupted or fails to open. The Device Manager cannot proceed with generating the Managed Device List until this problem is resolved.

Using the Event List Editor

The Device Manager can report several events of interest to monitoring applications. Each event has a text string, an event code, a severity flag, and a flag indicating whether the Device Manager should send the event as it occurs. The severity of events, and whether events should be sent asynchronously is application dependent. Each user must decide what makes sense for each system and application. By default all events have the same severity and only the DEVICE_COMMISSIONED event is sent asynchronously. However, severity, whether to send asynchronously, and the text string associated with each event can be changed. The *Event List Editor* utility is used to make these changes.

Start the Event List Editor by selecting PROGRAMS | Echelon Device Manager | Device Manager Event List Editor from the Windows start menu. The Device Manager Event List Editor follows standard Windows conventions to edit, print, save, and retrieve files.



Event Type	Severity (0-255)	Send Async	Text
DMEVNT_UNUSED_0	0	No	Unused
DMEVNT_UNUSED_1	0	No	Unused
DMEVNT_RESET	0	No	DM-20 reset
DMEVNT_REBOOT	0	No	DM-20 reboot
DMEVNT_DATABASE_SELECTED	0	No	Database selected
DMEVNT_INVALID_DATABASE_ID	0	No	Invalid database ID
DMEVNT_UNUSED_2	0	No	Unused
DMEVNT_FOUND_UNCNFG	0	No	Unconfigured device found
DMEVNT_FOUND_CNFG	0	No	Configured device found
DMEVNT_MISSING	0	No	Device missing
DMEVNT_DEVICE_CONFIG_ERROR	0	No	Device config error
DMEVNT_DEVICE_COMMISSIONED	0	Yes	Device commissioned
DMEVNT_DEVICE_COMMISSION_FAILED	0	No	Device commission failed
DMEVNT_UNUSED_3	0	No	Unused
DMEVNT_UNUSED_4	0	No	Unused
DMEVNT_FATAL_ERROR	0	No	Fatal error
DMEVNT_VARIABLE_LOCATION_SET	0	No	Variable Location set

Figure 6.4 – Event List Editor

File Transfer

The following files can be transferred using the LonTalk® File Transfer Protocol (FTP). See the LonTalk File Transfer engineering bulletin (part number 005-0025-01) and the *LONMARK Application Layer Interoperability Guidelines* for information and instructions.

ftpDM_EventList

The event list is uploaded from the Device Manager to the user via the LonTalk FTP. `DmEventDescriptor` defines the record structure. (See `dmobj.h` in `\LONWORKS\DM\include` for details.) Typically, the user first uploads this event list, then correlates the items in the list to the events in the event log to produce readable displays. The attributes of the events can also be changed (for example, enable the `SendAsync` attribute or change the text) and download it to the Device Manager. The Event List Editor, described above, allows this file to be updated.

FILE TYPE	8020
FILE INDEX	4

ftpDLEventLog	The event log. The event code (DmEventType) field is used to correlate entries in this log with the text strings stored in the Device Manager via ftpDM_EventList. Each event requires 10 bytes of storage. The event log can be downloaded using the LonTalk FTP. The record structure is `DmEventLogEntry.` See `dmobj.h` in `\LONWORKS\DM\include` for more details.

FILE TYPE	8010
FILE INDEX	3

Managed Device List	Each entry in the *Managed Device List* is of the type `ManagedDeviceEntry`.

FILE TYPE	8023
FILE INDEX	6

CAUTION: DO NOT Use UCPTDriveExternalHealthLED

The UCPTDriveExternalHealthLED configuration property exists in both the DM-20 and DM-21 devices, and is intended for Echelon's internal testing purposes.

The Device Manager plug-in will not allow the value of this configuration property to be changed because setting the value of this configuration property to TRUE and simultaneously providing +5VDC on the subsystem selection pin can reduce the life of the Neuron chip.

DO NOT USE ANOTHER TOOL TO CHANGE THE VALUE OF THIS CONFIGURATION PROPERTY.

7

Device Manager Advanced Topics

This chapter describes the various advanced topics that may be of interest for special installations.

Database Synchronization

Several Device Managers may be used together to manage a large system (see *Using Subsystems* in Chapter 1). There were two ways in which this can be accomplished:

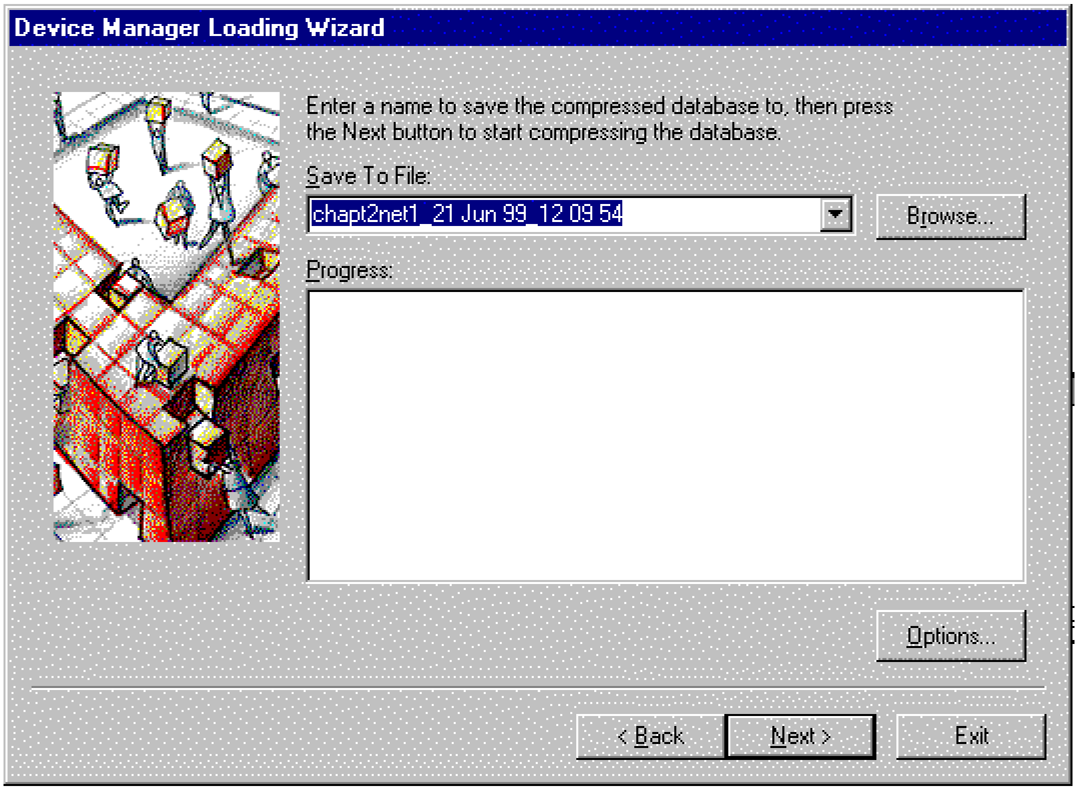
1. All subsystems can be loaded into every Device Manager, and each Device Manager can be instructed at runtime as to which subsystem to apply based on its 4-bit subsystem selection input;

2. Each Device Manager is loaded with a single subsystem, and is responsible for managing only that portion of the network associated with its subsystem.

When all subsystems fit into the Device Manager's memory, the first option provides much more flexibility by permitting Device Managers to be programmed in the factory and shipped to the field since any Device Manager can manage any part of the system based on the 4 bit subsystem selection input.

The first option raises the problem of database synchronization. Imagine an installation where five Device Managers are needed to manage the network. The five Device Managers are programmed and installed in year 1. In year 4 it is necessary to replace a Device Manager. Over the three-year period enhancements have been made to the original database. If the enhanced database was loaded into a replacement Device Manager and then installed in the four year old network, there is a possibility that the enhanced Device Manager would be incompatible with the original database.

To easily detect such an incompatibility, assign a database version number when the managed device list is loaded into the Device Manager. This is accomplished on the window of the Device Manager Loading Wizard shown in figure 7.1 (see step 11 of Example 1 in Chapter 5).



Device Manager Loading Wizard

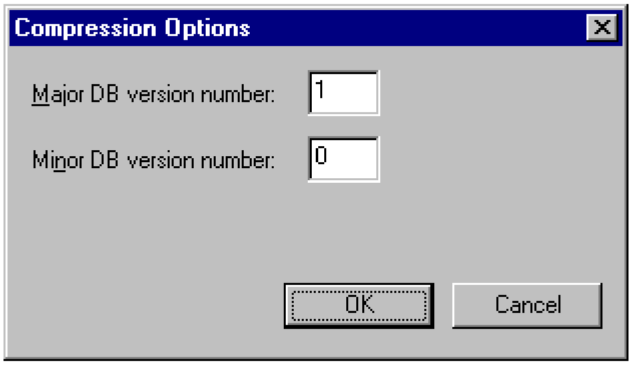
Enter a name to save the compressed database to, then press the Next button to start compressing the database.

Save To File: chapt2net1_21 Jun 99_12 09 54

Progress:

Figure 7.1 – Select Database Name

Before clicking Next and compressing the database set the database version by clicking the *Options* button to open the dialog shown in figure 7.2.



Compression Options

Major DB version number: 1

Minor DB version number: 0

Figure 7.2 – Database Version Number

Select a database version that makes sense for your application. Each version can be numbered from 0 to 65535. Once a database is assigned and all Device Managers are loaded with the same database it is a simple matter for a device in the system to monitor and compare the database versions of all the Device Managers in the system. If a discrepancy in the database version is detected among the Device Managers, the device can flag an operator to indicate the potential error condition.

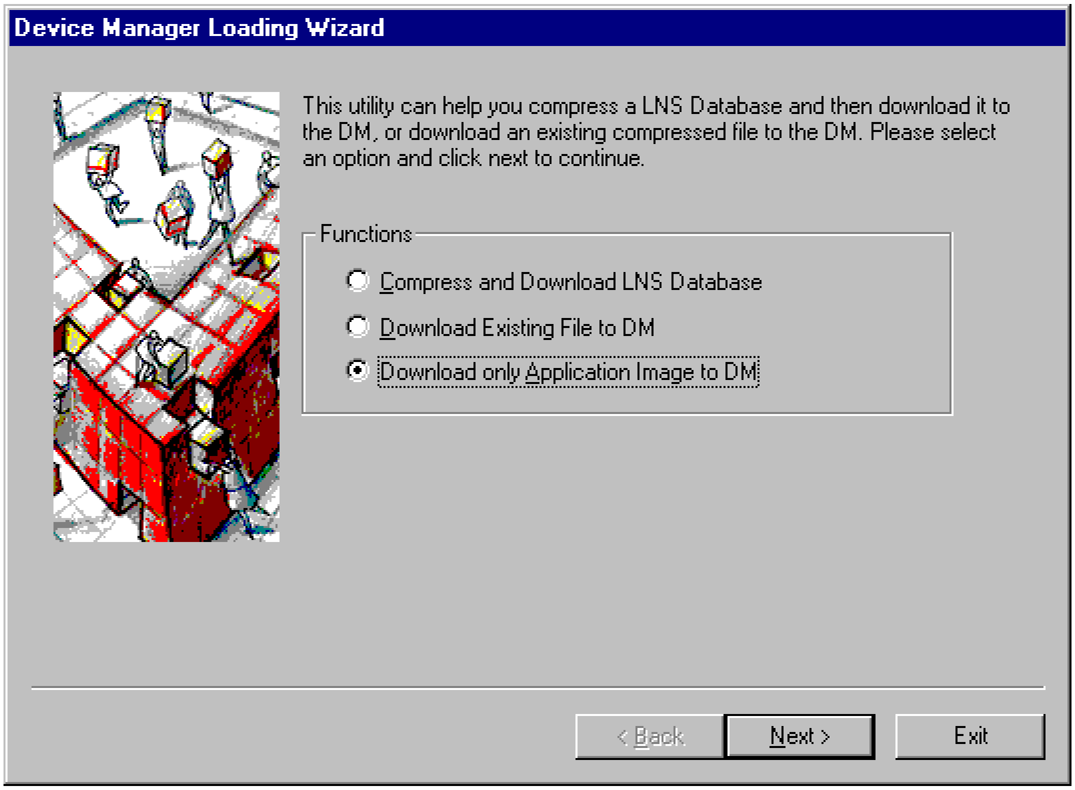
The simplest means for a device to monitor Device Manager databases is for the device to periodically issue a DMCMD_QUERY_DM_PROPS command, which will cause the Device Managers to respond with DMREP_DM_PROPS reports.

The DMREP_DM_PROPS report contains a field that includes the database number.

Updating DM Firmware

Newer versions of the Device Manager firmware can be loaded into a Device Manager by using the loading wizard. To load a new firmware version into device manger, follow these steps:

1. Start the Device Manager Compression Wizard and select `Download only Application Image to DM`. The window shown in figure 7.3 will appear.



This utility can help you compress a LNS Database and then download it to the DM, or download an existing compressed file to the DM. Please select an option and click next to continue.

Functions

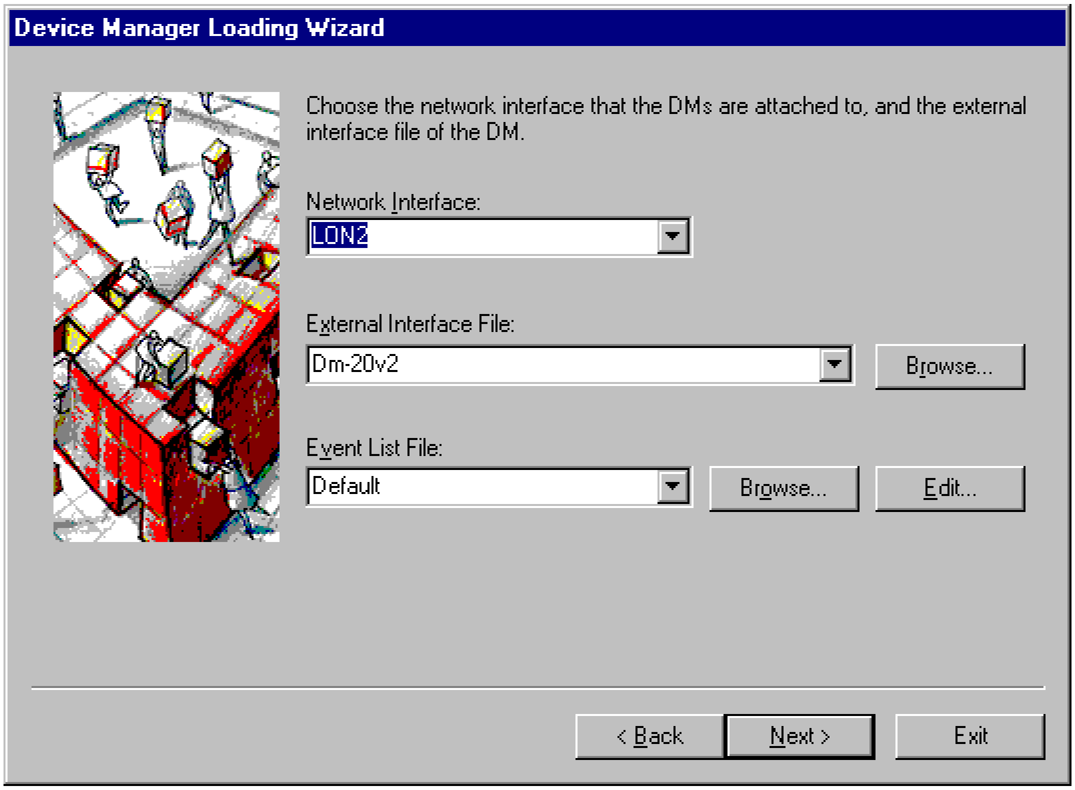
○ Compress and Download LNS Database

○ Download Existing File to DM

● Download only Application Image to DM

Figure 7.3 – Downloading a New Application Image

2. Click Next. The window shown in figure 7.4 will appear.



Device Manager Loading Wizard

Choose the network interface that the DMs are attached to, and the external interface file of the DM.

Network Interface: LON2

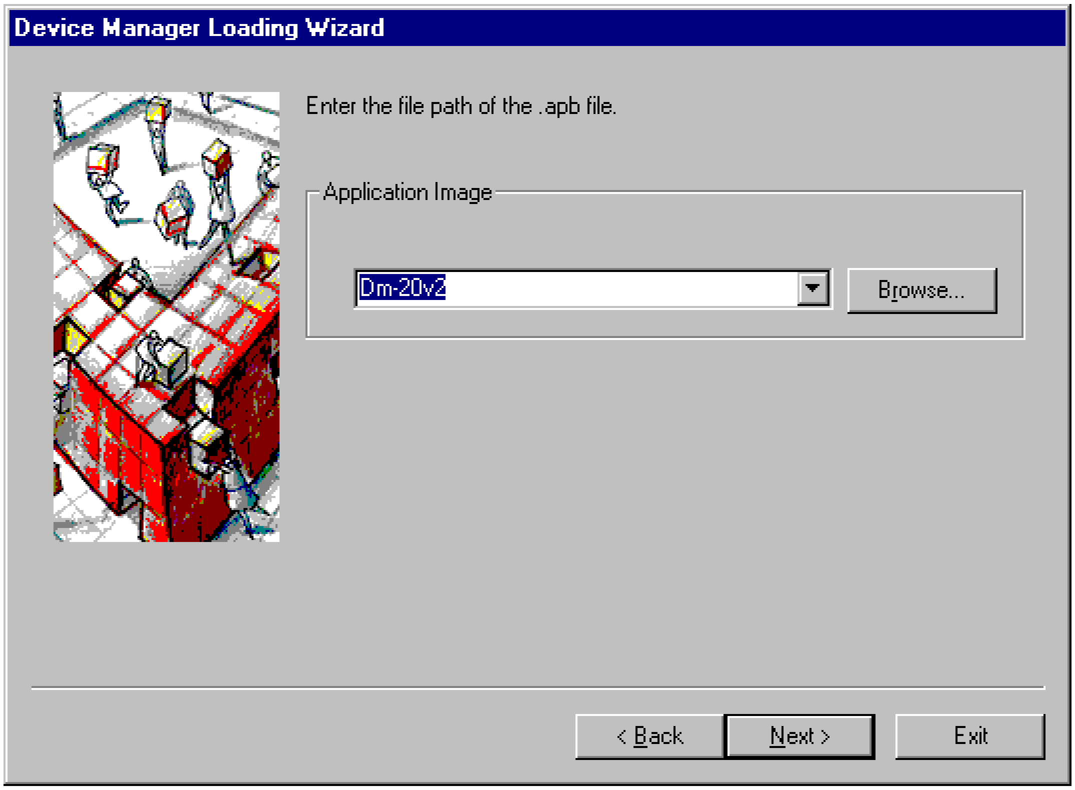
External Interface File: Dm-20v2

Event List File: Default

Figure 7.4 – Select Network Interface and XIF

Select the network interface, XIF file, and event list for this download.

3. Click Next. The window shown in figure 7.5 will appear.



Device Manager Loading Wizard

Enter the file path of the .apb file.

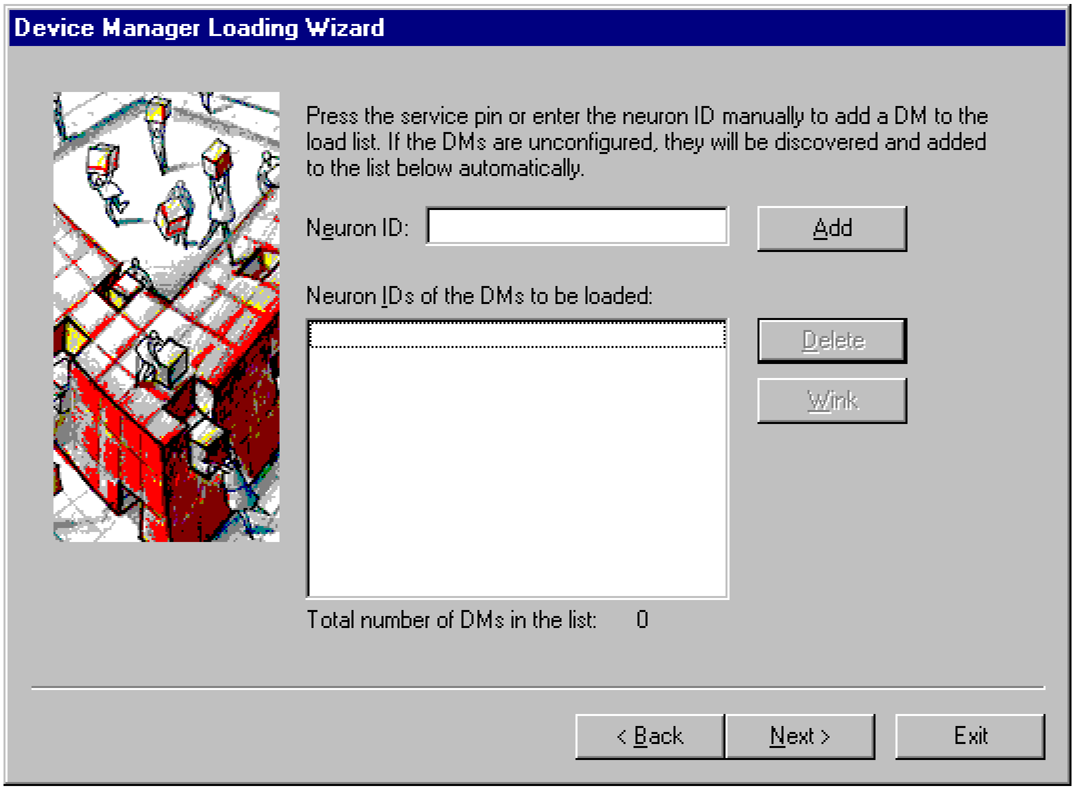
Application Image

Dm-20v2

Figure 7.5 – Select Application Image

Select the firmware image to load (.APB file). The APB file selected must match the XIF file specified in the previous step.

4. Click Next. The window shown in figure 7.6 will appear.



Device Manager Loading Wizard

Press the service pin or enter the neuron ID manually to add a DM to the load list. If the DMs are unconfigured, they will be discovered and added to the list below automatically.

Neuron ID:

Neuron IDs of the DMs to be loaded:

Total number of DMs in the list: 0

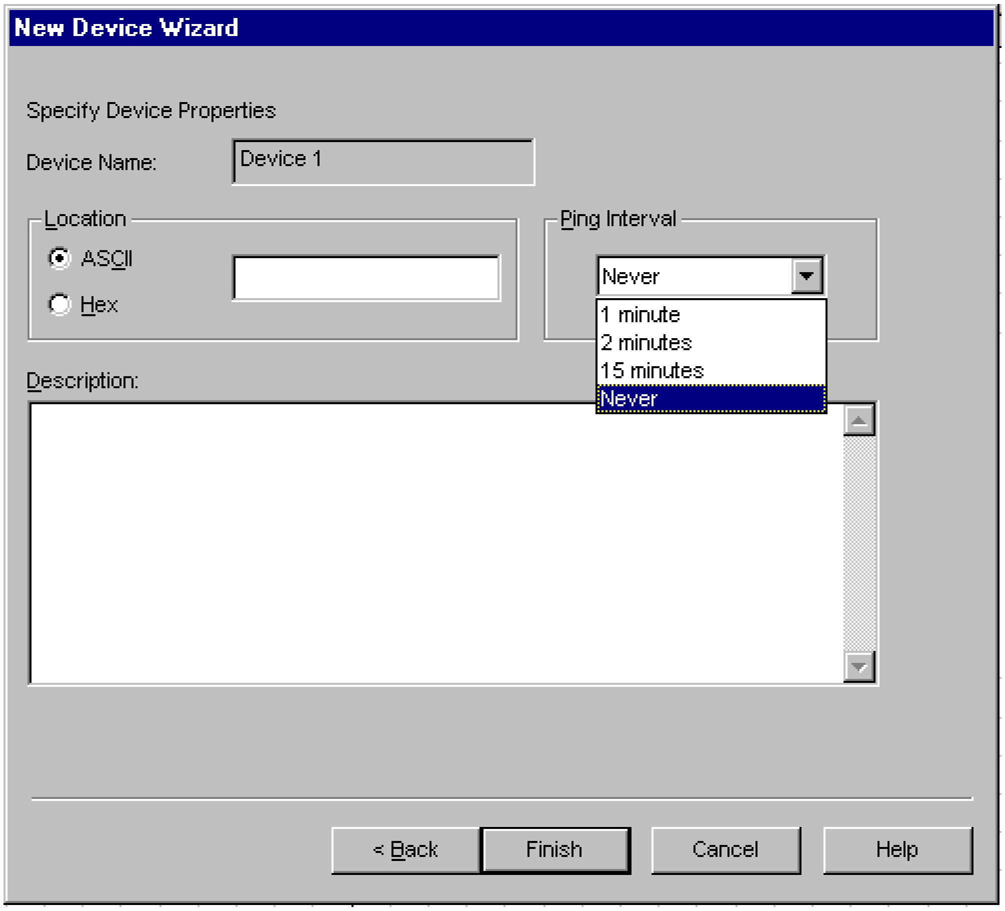
Figure 7.6 – Select Device Manager Modules

Select the Device Managers to be loaded by pressing service pins. The Compression Wizard will cache the Neuron ID numbers of the Device Managers and load them in batch mode.

5. Click Next. The Compression Wizard will display a progress window and load each Device Manager.

Temporary Devices

It may be desirable to have devices temporarily integrated into a network supervised by a device manager. Portable test equipment, for example, might need to be bound into the network for service, but generally would be absent when the network is running under normal conditions. In such cases it is usually better if the device is not pinged by the Device Manager. Such a temporary device can be identified by setting its ping interval on the screen of the LonMaker for Windows tool device wizard shown in figure 7.7.



New Device Wizard

Specify Device Properties

Device Name: Device 1

Location

● ASCII

○ Hex

Ping Interval: Never

Description:

Figure 7.7 – Select Ping Interval

Selecting `Never` or `15 minutes` mark the device as being a permanent device. Selecting `1 minute` or `2 minutes` identifies temporary devices. All permanent devices must respond to pinging before the Device Manager will indicate via its status LED that all devices have been found.

Unsupported Features

The Device Manager does not support the following features:

NSI Devices

The Device Manager cannot manage NSI devices. Use a network integration tool such as the LonMaker for Windows Integration Tool to manage NSI devices.

Dynamic Network Variables and Rate Estimates

The Device Manager can add network variables to a device that supports them (such as a host-based MIP device), provided the device is not an NSI and the dynamic network variables were added to the compressed database. The Device Manager cannot set self-documentation text or rate estimates for dynamic network variables.

Network Variable Name

The Device Manager cannot set a network variable's name in a host device.

Network Variable self-documentation text

The Device Manager cannot associate a text string with each network variable declaration as documentation.

Remote

Neither the Device Manager Compression Wizard, the Device Manager Event List Editor, or the Device Manager plug-in can run remotely.

Authentication

The Device Manager does not support authentication.

Appendix A

Troubleshooting

This appendix describes the process of troubleshooting the Device Managers and software.

Troubleshooting the DM-20/21 Hardware

Power LED is extinguished

- The power supply providing power to the module may have lost primary power. Check the power supply output for the presence of power.
- The power wiring may not be correctly connected to the module. Use a volt meter to check for power on Base Plate power terminals.
- The input voltage may be too low due to an excessive load on the power supply providing power to the module, excessively long power cabling, or shorted power wiring. Use a volt meter to check for the correct voltage level on the base plate power terminals. Correct cabling length if too long. Repair faulty power cable.
- The Device Manager firmware version may be a down-revision prior to 2.0. An error in the pre-2.0 firmware prevented the power LED from illuminating.

The Service LED is flashing

The module is unconfigured. This is the normal state of a new module. The module is commissioned when a database is downloaded into it by the Device Manager Loading Wizard.

The Service LED is illuminated

The module is applicationless and has lost its application software program. The module must be reloaded by the Device Manager Loading Wizard. See Chapter 8 for more information.

No communication between the Device Manager and the network

- The network wiring may not be correctly connected to module.
- Network cabling may be shorted, open, or otherwise damaged. Use a continuity meter to check cabling.
- One or more routers on the network may be not commissioned, not powered, or otherwise not functioning correctly.
- A router may be installed with the incorrect router channel type.

Erratic or improper communication between the Device Manager and the network

- The network wiring may not be securely connected to Base Plate terminals. The optimum tightening torque for a Base Plate screw terminal is 4 lbs. in. (0.5Nm) maximum.
- The Device Manager module may not be properly seated in the Base Plate. Ensure that the module is pressed firmly into the Base Plate and secured with suitable fixing screws.
- There may be improper or missing network termination. A free topology TP/FT-10 channel requires one Model 44100 Terminator located anywhere on the channel. A bus topology TP/FT-10 channel requires two Model 44101 terminators, one at each end of the bus.

- There may be excessive network cabling. See Chapter 3 for a discussion of the maximum cable distances permissible on the twisted pair channels. If the channel cabling is too long, install one or more LPR Routers in series with the network cabling. Refer to the transceiver user's guides for more detailed cabling information.
- There may be improper network cabling. See Chapter 3 for a discussion of the types of cabling that are approved for use on the TP/FT-10 channel. In some cases it is possible to correct network communications by installing one or more LPR Routers in series with the network cabling. In other cases the cabling must be replaced with an approved cable.
- There may be an excessive number of modules on a channel. See Chapter 3 for a discussion of the number of devices that can be installed on a channel. If too many modules are on the channel, install one or more LPR Routers in series with the network cabling.
- Modules may be connected to the wrong channel of an LPR Router. Care must be taken to ensure that correct types of devices are connected to the two channels of a router. LonPoint modules should only be connected to a TP/FT-10 channel.

Troubleshooting the Device Manager Loading Wizard

"The DM named xx cannot be compressed because another DM exists in the same subsystem hierarchy…" is displayed during compression.

There may be another Device Manager in a sub-subsystem. When a subsystem is compressed, all the sub-subsystems of that subsystem are also compressed, as illustrated in Figure A-1. In fact, compressing subsystem_1 is equivalent to compressing subsystem_combine. This may conceal Device Managers that were inadvertently associated with subsystems embedded within a subsystem.

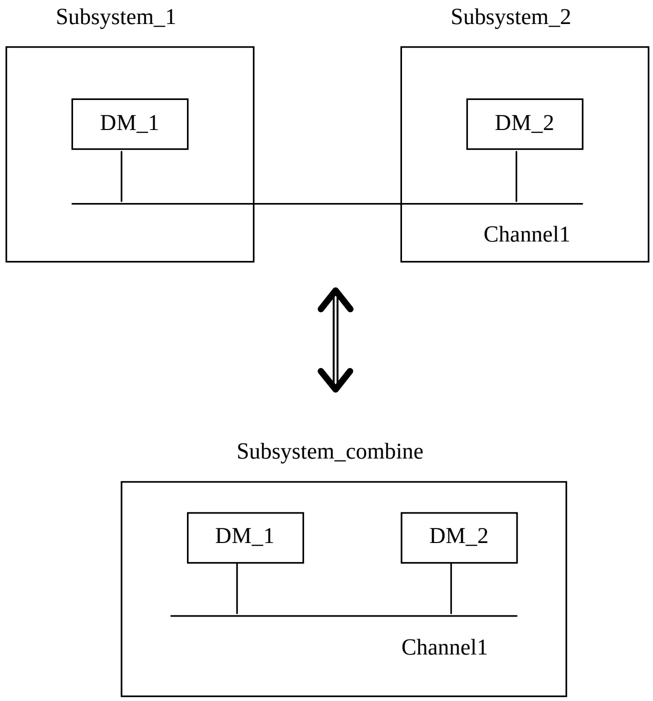


Figure A-1 - Illustrating recursive compression

"This channel contains more than one DM-20/21..." is displayed during compression

- A Device Manager may be sharing the same channel in another subsystem. Two Device Managers cannot share the same channel, even if they are used in peer subsystems. The reason that this is disallowed is because a race condition may result if two Device Managers attempt to commission the same device.
- Another Device Manager may be in a sub-subsystem sharing the same channel.

During loading, this error is given: "Cannot commission device because..."

- The Device Manager may not be properly connected to the network. Commission may fail if the connection is not stable.
- There may be a short circuit in the network.
- There may be too much traffic in the network, causing the network buffers to overload. Only the Device Manager to be loaded should be attached to the network. It is also recommended that Device Managers by deconfigured before running the Loading Wizard.
- The Device Manager may be applicationless. If a Device Manager is applicationless, an application image must be loaded before commissioning can occur.

Some error concerning the ~~~td database is displayed

The temporary database may be corrupted. The Loading Wizard uses a temporary database for loading. If it is corrupted loading cannot be accomplished.

To solve this problem, run the Loading Wizard with the –d flag, and then press the Delete Temporary DB button. This will delete the corrupted database and a new one will be created.

Troubleshooting the DM-20/21 Installation

The Device Manager status LED should only flash once per interval if it is operating correctly and has finished commissioning all the devices in the managed device list.

The Device Manager status LED is flashing twice per interval (Discovery incomplete)

- All of the devices in the database may not be physically installed on the network. The Device Manager cannot commission all of the devices if they are not physically installed.
- The location string may not be correctly set on one or more physical devices. The Device Manager cannot find non-unique devices unless their location strings are set correctly.
- Some device may not have been deconfigured before the Device Manager was installed into the network. If a device is not deconfigured, and its domain ID is different from that of the Device Manager, then the Device Manager cannot discover the device. An alternative is approach is to press the service pin on all the devices after the Device Manager is installed. The Device Manager will discover the device if it receives the device's service pin.

The DM status LED is flashing three times per interval (Opening database)

The Device Manager firmware image may not correspond to the external interface file used when the Device Manager was defined in the LonMaker network database. If this is the case, then the Device Manager will be unable to commission itself. If the Device Manager contains old firmware, it should be re-loaded using the Device Manager Loading Wizard. If the external interface file used to define the Device Manager is old, perform the following steps to upgrade the database:

1. Open the network database using the LonMaker for Windows Integration Tool.
2. For all the Device Managers in the database, repeat the steps 3-6.
3. Right-click the Device Manager shape and select *Replace* from the shortcut menu.
4. In the device replace wizard, select the Load .XIF and browse to the new .XIF file location (in the LONWORKS directory under `Import\Echelon\DM\Version2\dm-20v2.xif`).
5. An alert may appear warning that the new template is not compatible with the old template and that the functional blocks will have to be removed. Click Yes.
6. Drop and connect new Device Manager functional block shapes as described in the *LonMaker for Windows Integration Tool User's Guide.*

7. Configure the new Device Manager shapes using the Device Manger plug-in as described in Chapter 6.

The Device Manager status LED is flashing four times (No subsystem selected)

- The network input pins may not have the correct value. If the network input pin has a value larger than the largest subsystem index in the Device Manager, then the Device Manager cannot select a subsystem.
- The Device Manager may be attempting to select Subsystem 0, but the Device Manager 4 bit subsystem selection input configuration property may be improperly set (see Chapter 6).

The Device Manager takes a long time to select a database

The value of the configuration property MaxSendTime in the Device Manager LONMARK object may not be 0. This value should be set to 0 in most cases. If the value is set too high (>50 milliseconds) it will take a long time for the Device Manager to commission itself. The value should be set to greater than zero only if the Device Manager is sending messages too quickly and generating too much network traffic. The highest practical setting is 50 milliseconds.

Glossary

This glossary defines LONWORKS terms that may be encountered when using the Device Manager.

Configuration Property

Configuration properties are properties within a device that are set during network design and installation to specify device set points and limits. For example, the Device Manager pings devices on its network at a specified interval where the interval is determined by setting a configuration property within the Device Manager.

Configuration properties can be set using the LonMaker browser or a plug-in application specially created for the device whose property is to be to set. Many manufactures ship plug-ins with their devices.

Configuration properties can be associated with a device, a functional block, or a network variable.

Device Manager Loading Wizard

A software application that takes a set of subsystems from an LNS database as input and provides a Managed Device List as output. This utility is also used to load the Managed Device List into a Device Manager.

Device Manager Plug-in

A software module that extends the core functionality of the LonMaker tool. This plug-in extends the user interface of the LonMaker tool to easily allow you to set various device properties. For example, the Device Manager plug-in sets how often the Device Manager pings devices on its network, how often the Device Manager tries to discover new devices, and how many times a device can fail to respond to a "ping" before the Device Manager flags that device as missing.

The advantage of using a plug-in over using the LonMaker browser is that plug-ins can range check values that are input by the user to prevent improper configurations. Plug-ins generally provide a superior user interface to the LonMaker browser

External Interface

Every device has an *external interface* that defines the LONMARK objects (functional blocks), network variables, and message tags on the device. The LonMaker tool can extract an external interface directly from a device or it can import an external interface file provided by the device manufacturer. Systems managed with a Device Manager typically use external interface files provided by the device manufacturer, because the network is usually designed offline.

LNS

The standard network operating system for LONWORKS control networks. LNS provides the essential directory, installation, management, monitoring, and control services for LONWORKS networks.

LNS Database

A collection of information that defines the network images of all devices in a network. An *LNS database* contains information about all device, routers, channels, default configuration properties, timing parameters, network variable and message tag connections, polling intervals, bound connections, and subsystems. The LonMaker tool combines the information from multiple pages of a Visio drawing into a single LNS database.

Location String A six byte area of the Neuron Chip's internal EEPROM. The installer can initialize this string using a network tool, or it can be initialized by the application running in the Neuron Chip.

It is critical that each device managed by a Device Manager is uniquely identifiable by reading a combination of device program IDs and Location Strings. The Device Manager uses these two resources to identify which entry in the *Managed Device List* corresponds to the devices it finds.

LonMaker for Windows Integration Tool The LonMaker for Windows Integration Tool is a software product supplied by Echelon. Based on two well-established standards, LNS and Visio™ Technical, the LonMaker for Windows Integration Tool software package combines Visio's drawing capabilities with the networking capabilities of LNS. A LONWORKS network is created by dropping devices onto a page and dragging connections between those devices.

LONWORKS Device A device that communicates on a LONWORKS network and uses the LonTalk protocol as defined by Echelon.

Managed Device List The Managed Device List is a subset of the information contained in an LNS database. This subset is loaded into the Device Manager module using the Device Manager Loading Wizard. The Device Manager uses the information in this list to manage its devices.

Network Variable LONWORKS devices communicate through the use of *network variables.* Each Neuron Chip-based device can have up to 62 network variables. A host processor based device that communicates to a Neuron Chip via Echelon's Microprocessor Interface Program (MIP) or Network Services Interface firmware (NSI) may have up to 4096 network variables.

Network Tools such as the LonMaker for Windows Integration Tool can be used to connect the output network variable on one device to an input network variable on another device.

Every network variable represents a path through which data can flow into or out of a device via the network. All network variables are defined as either input or output, which determines whether the network variable handles data going into, or out of, the device.

Program ID An area in the internal EEPROM of the Neuron Chip used to store a manufacturer ID, device class, device subclass, and model number for a device. The Device Manager uses the program ID to identify the devices it discovers. Devices with the same program ID must have identical external interfaces.

See the *Neuron C Programmer's Guide*, *LONMARK 3.0 Application Layer Guidelines*, and *Neuron Databook* for more information about the program ID.

Subsystem Devices, routers, and functional blocks are contained within *subsystems*. With the LonMaker tool, each page in a Visio drawing generally refers to a subsystem. Subsystems are useful for grouping related devices. Subsystems may also be placed within other subsystems, allowing the creation of a subsystem hierarchy for large networks.

An LNS database can contain many subsystems. For example, a network might consist of HVAC, lighting, and security subsystems. These can be further divided into subsystems for each floor, and each floor can be divided into subsystems for each room.