



**User Manual for the  
*HE693DNT250***

# **DeviceNet Master (Scanner)**

**Fourth Edition  
June 23, 2000**

**MAN0054-04**



## PREFACE

**This manual explains how to use the Horner APG's DeviceNet Master (Scanner). .**

Copyright (C) 2000 Horner APG, LLC., 640 North Sherman Drive, Indianapolis, Indiana 46201. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior agreement and written permission of Horner APG, LLC.

All software described in this document or media is also copyrighted material subject to the terms and conditions of the Horner Software License Agreement.

Information in this document is subject to change without notice and does not represent a commitment on the part of Horner APG, LLC.

DeviceNet is a trademark of Open DeviceNet Vendors Association (ODVA).

Logicmaster, LM90, and SNP are trademarks of GE Fanuc

Windows and Windows-95 are trademarks of Microsoft Corporation

***For user manual updates, contact Horner APG, Technical Support Division, at (317) 916-4274 or visit our website at [www.heapg.com](http://www.heapg.com).***

## LIMITED WARRANTY AND LIMITATION OF LIABILITY

Horner APG, LLC. ("HE") warrants to the original purchaser that the DeviceNet Master (Scanner) manufactured by HE is free from defects in material and workmanship under normal use and service. The obligation of HE under this warranty shall be limited to the repair or exchange of any part or parts which may prove defective under normal use and service within two (2) years from the date of manufacture or eighteen (18) months from the date of installation by the original purchaser whichever occurs first, such defect to be disclosed to the satisfaction of HE after examination by HE of the allegedly defective part or parts. THIS WARRANTY IS EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES EXPRESSED OR IMPLIED INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE AND OF ALL OTHER OBLIGATIONS OR LIABILITIES AND HE NEITHER ASSUMES, NOR AUTHORIZES ANY OTHER PERSON TO ASSUME FOR HE, ANY OTHER LIABILITY IN CONNECTION WITH THE SALE OF THIS DEVICENET MASTER (SCANNER). THIS WARRANTY SHALL NOT APPLY TO THIS DEVICENET MASTER (SCANNER) OR ANY PART THEREOF WHICH HAS BEEN SUBJECT TO ACCIDENT, NEGLIGENCE, ALTERATION, ABUSE, OR MISUSE. HE MAKES NO WARRANTY WHATSOEVER IN RESPECT TO ACCESSORIES OR PARTS NOT SUPPLIED BY HE. THE TERM "ORIGINAL PURCHASER", AS USED IN THIS WARRANTY, SHALL BE DEEMED TO MEAN THAT PERSON FOR WHOM THE DEVICENET MASTER (SCANNER) IS ORIGINALLY INSTALLED. THIS WARRANTY SHALL APPLY ONLY WITHIN THE BOUNDARIES OF THE CONTINENTAL UNITED STATES.

In no event, whether as a result of breach of contract, warranty, tort (including negligence) or otherwise, shall HE or its suppliers be liable of any special, consequential, incidental or penal damages including, but not limited to, loss of profit or revenues, loss of use of the products or any associated equipment, damage to associated equipment, cost of capital, cost of substitute products, facilities, services or replacement power, down time costs, or claims of original purchaser's customers for such damages.

**To obtain warranty service, return the product to your distributor with a description of the problem, proof of purchase, post paid, insured and in a suitable package.**

## ABOUT PROGRAMMING EXAMPLES

Any example programs and program segments in this manual or provided on accompanying diskettes are included solely for illustrative purposes. Due to the many variables and requirements associated with any particular installation, Horner APG cannot assume responsibility or liability for actual use based on the examples and diagrams. It is the sole responsibility of the system designer utilizing the to appropriately design the end system, to appropriately integrate the DeviceNet Master (Scanner) and to make safety provisions for the end equipment as is usual and customary in industrial applications as defined in any codes or standards which apply. It assumed that the system designer is familiar with PLC programming and configuration.

**Note: The programming examples shown in this manual are for illustrative purposes only. Proper machine operation is the sole responsibility of the system integrator.**

## Revisions to this Manual

This version (MAN0054-04) of the DeviceNet™ User Manual contains the following revisions, additions, and deletions:

1. Revised Section 1.1 and 1.2 to include UCMM protocol and all four Message Body Format under UCMM.
2. Moved Chapter 3: "Configuration" to Chapter 6. Replaced Chapter 3 with "DeviceNet Node, DNT250 & PLC Relationships."
3. Replaced Chapter 4: "DeviceNet to PLC I/O Mapping" with a new Chapter 4: "DNT250 Status and Command Data Assemblies."
4. Replaced Chapter 5: "HE693DNT250 Network Operation" with a new Chapter 5: "Explicit Messaging Using the DNT250."
5. Added new configuration software to Chapter 6: Configuration.
6. Deleted Sections 7.3 and 7.4. Moved Table 7.1: General Error Codes to Appendix C.
7. Replaced Appendix A with "DNT250 Network Operation."
8. Replaced Appendix B with "Packet Timing."

**TABLE OF CONTENTS**

PREFACE .....	3
LIMITED WARRANTY AND LIMITATION OF LIABILITY .....	4
ABOUT PROGRAMMING EXAMPLES .....	4
TABLE OF CONTENTS .....	6
CHAPTER 1: INTRODUCTION .....	7
1.1 Overview .....	7
1.2 HE693DNT250 Features .....	7
1.3 Technical Specifications .....	8
CHAPTER 2: INSTALLATION .....	9
2.1 Connectors .....	9
2.1.1 RS-485 Connector .....	9
2.1.2 DeviceNet I/O Connector .....	9
2.2 LED Indicators .....	10
CHAPTER 3: DEVICENET NODE, DNT250 & PLC RELATIONSHIPS .....	13
3.1 General .....	13
3.2 Module to PLC Register Mapping .....	13
3.3 Node to PLC Register Mapping .....	13
3.4 Message Packets and Fragmentation .....	13
3.5 Physical Limitations .....	13
3.6 Data Assemblies .....	14
3.7 DNT250 Configuration Introduction .....	15
CHAPTER 4: DNT250 STATUS AND COMMAND DATA ASSEMBLIES .....	17
4.1 Register Requirements and Definitions of Data Assemblies .....	17
CHAPTER 5: EXPLICIT MESSAGING USING THE DNT250 .....	21
5.1 General .....	21
5.2 Explicit Messages and the DNT250 .....	21
5.3 Building Explicit Messages .....	22
5.4 How to Interpret Explicit Response Messages .....	24
5.5 Explicit Message Errors .....	24
CHAPTER 6: CONFIGURATION .....	25
6.1 General .....	25
6.2 DNT250 Module Configuration .....	25
6.3 PLC Rack Configuration .....	26
6.4 DNT250 Configurator .....	27
6.4.1 General .....	27
6.4.2 Configuration Procedures .....	28
CHAPTER 7: DEVELOPING USER SOFTWARE .....	37
7.1 Software Support for the Polled Connection .....	37
7.2 Software Support for the Explicit Connection .....	37
APPENDIX A: DNT250 NETWORK OPERATION .....	39
1 Sequence of Events .....	39
2 Establishing a Connection .....	39
APPENDIX B: PACKET TIMING .....	41
1 Message Packet Timing .....	41
2 Network Scan Time versus PLC Scan Time .....	41
APPENDIX C: DEVICENET GENERAL ERROR CODES .....	43

## CHAPTER 1: INTRODUCTION

### 1.1 Overview

The HE693DNT250 DeviceNet Scanner (DNT250) is an intelligent communications interface module that provides DeviceNet Scanner functionality to a GE Fanuc Series 90-30 PLC. The DNT250 allows the connection of up to 63 DeviceNet slave nodes. Depending on baud rate and the cable type used, the DeviceNet nodes may be located up to 1,500 feet from the PLC.

The DNT250 Polled data is mapped directly into the PLC's %I, %Q, %AI and %AQ registers. The ladder programmer can treat the DNT250 and its attached DeviceNet nodes as a large I/O module. Polled data is transferred between the DNT250 and the PLC using Backplane I/O which is fast and efficient.

The DNT250 provides two additional constructs known as Ladder Code Initiated Explicit Messaging (LIEM) and the Unconnected Message Manager (UCMM).

The LIEM allows the ladder code executing within the PLC to initiate dialogues with DeviceNet nodes through the use of explicit messaging. This enables the PLC to access data beyond that normally available using the Polled Connection. For LIEM, data is transferred between the DNT250 and the PLC using a technique known as Backplane Mail. Backplane Mail requires considerably more processing time than Polled messaging. LIEM should therefore only be used to access infrequently required data such as device configuration and tuning parameters.

The DeviceNet specification provides two methods of establishing communications between scanners and nodes. The first is known as "Group 2 Only". This was the only method provided by older DNT250s. The second method makes use of the "Unconnected Message Manager" (UCMM). The new DNT250 allows both of these methods to be used simultaneously. Any single slave device on a DeviceNet network can be either a Group 2 Only node or a UCMM node, never both. A DeviceNet network can be made up of a mixture of "Group 2 Only" and UCMM" slave devices.

The "Group 2 Only" method is by far the simplest. Because the UCMM is a far more intense protocol, very few manufactures supported it in the early days of DeviceNet. Times have changed. Now a large percentage of DeviceNet manufactures support the UCMM. As a result, the DNT250 has been upgrade to support both the UCMM and Group 2 Only protocols.

### 1.2 HE693DNT250 Features

The DNT250 allows a GE Fanuc Series 90-30 PLC to supervise a DeviceNet network. From the viewpoint of the ladder program running in the PLC, the DNT250 appears as a single, very high density I/O module.

The DNT250 supports the following DeviceNet features:

Baud Rates: 125K, 250K, and 500K,

UCMM protocol,

Group 2 Only protocol,

The Polled Connection,

The LIEM (explicit messaging),

Fragmentation on both polled and explicit connections,

All four Message Body Formats under UCMM.

When using Polled Messaging, data is read from the PLC's %Q and %AQ registers by the DNT250, formatted into DeviceNet packets and sent to the DeviceNet nodes. Likewise, data produced by the DeviceNet nodes is received by the DNT250, converted into PLC register notation and passed to the PLC's %I and %AI registers. This happens automatically without the need for block move, "COMREQ", or other ladder program instructions.

The ladder code running in the PLC can build Explicit Messages to access any network data that is known to be available through the node's Explicit Connection. See chapter 4 for more information on LIEM or explicit messaging.

### 1.3 Technical Specifications

The following HE693DNT250 DeviceNet Interface Module specifications are subject to change without notice.

<b>Table 1.1 - H3693DNT250 Specifications</b>			
DeviceNet Network Specifications			
<b>Parameter</b>	<b>Minimum</b>	<b>Maximum</b>	<b>Units</b>
DeviceNet Power Voltage	11	25	V
DeviceNet Power Load		65	mA
DeviceNet Signal Baud Rate	125	500	KHz
DeviceNet Signal Driver Fanout	0	63	Devices
PLC Power Load Specifications			
<b>Parameter</b>	<b>Minimum</b>	<b>Maximum</b>	<b>Units</b>
+5Vdc (LOGIC)	0	175	mA
+24Vdc (RELAY)	0	0	mA
+24Vdc (ISOLATED)	0	0	mA
Environmental Specifications			
<b>Parameter</b>	<b>Minimum</b>	<b>Maximum</b>	<b>Units</b>
Operating Temperature	0	+60	Deg C
Storage Temperature	-40	+85	Deg C
Humidity (non-condensing)	5	95	% RH



## CHAPTER 2: INSTALLATION

### 2.1 Connectors

#### 2.1.1 RS-485 Connector

The RS485 port allows the DNT250 to be connected to an RS-232 serial port of a PC. The port is physically and electrically identical to that used on the GE Fanuc Series 90-30. It allows the same cable and same PC serial port to be used to program both the PLC and the DNT250. The DNT250 does not accept SNP or Logicmaster commands. The DNT250 must be configured using the DN250CFG.EXE program.

**IMPORTANT:** Do not attempt to "talk to" the DNT250 using Logicmaster or SNP Protocol. Although no physical damage will result, the information in the DNT250 could be corrupted such that the DNT250 would appear "dead." A factory update could possibly be required to return the unit to normal operating conditions

To connect the DNT250 to a PC, the user must have an SNP adapter and a straight through 9 pin serial cable. The RS-485 port consists of a 15-pin female D-connector with the following pin descriptions:

Pin	Signal	Description
1	FGND	Frame Ground
2	DCD-	Data Carrier Detect -
3	DCD+	Data Carrier Detect +
4	-HHP	Hand Held Programmer Detect
5	+5V	RS485-RS232 Adapter Power
6	RTS-	Request to Send -
7	GND	Signal Ground
8	CTS+	Clear to Send +
9	RT	Receiver Termination
10	RXD-	Received Data -
11	RXD+	Received Data +
12	TXD-	Transmit Data -
13	TXD+	Transmit Data +
14	RTS+	Request to Send +
15	CTS-	Clear to Send -

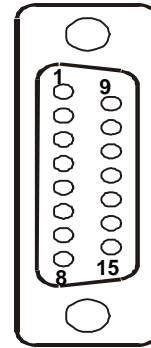


Figure 2.1 – Configuration Port (As Viewed from Front of Module)

#### 2.1.2 DeviceNet I/O Connector

The DeviceNet I/O connector consists of a 5-pin removable screw terminal with the following terminal descriptions:

Pin	Signal	Description
1	V-	Power -
2	CAN_L	Signal -
3	Drain	Shield
4	CAN_H	Signal +
5	V+	Power +

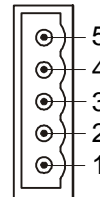


Figure 2.2 - DeviceNet I/O Connector (As Viewed from Front of Module)

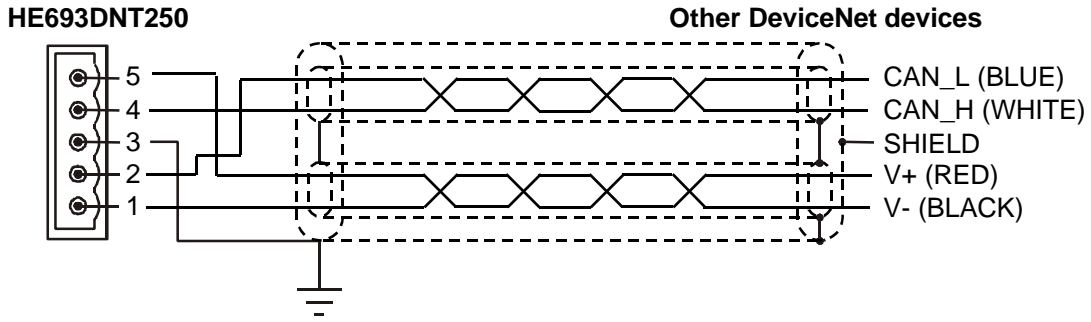


Figure 2.3 - DeviceNet Wiring

A user-supplied 121Ω ¼W 1% resistor is needed for termination at EACH END of the network cabling. Refer to the DeviceNet cabling specifications for proper location of the terminating resistor.

Table 2.3 - DeviceNet Cable Length vs Network Baud Rate		
DeviceNet Baud Rate	Maximum Cable Distance	
	Thick Cable	Thin Cable
125KHz	1640 Feet	320 Feet
250KHz	820 Feet	320 Feet
500kHz	320 Feet	320 Feet
DeviceNet Baud Rate	Maximum Drop Distance	
	Per Drop	Cumulative
125KHz	20 Feet	500 Feet
250KHz	20 Feet	250 Feet
500kHz	20 Feet	120 Feet

2.2 LED Indicators

The HE693DNT250 provides two bi-color (Red/Green) diagnostic LEDs on its front panel.

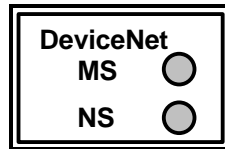


Figure 2.4 - LEDs

<b>Table 2.4 - Status Indicator States and Meaning</b>		
<b>Module Status LED (MS)</b>		
<b>For this state</b>	<b>Module Status LED is</b>	<b>Condition</b>
No Power	Off	There is no power applied to the module.
Device Operational	Green	The module is operating normally.
Device in Standby	Flashing Green	The module has no configuration.
Minor Fault	Flashing Red	Recoverable Fault
Unrecoverable Fault	Red	The module has an unrecoverable fault.
Device Self Testing	Flashing Red-Green	The module is in Self Test.
<b>Network Status LED (NS)</b>		
<b>For this state</b>	<b>Network Status LED is</b>	<b>Condition</b>
Not powered or not On-line	Off	The module may not be powered. Module is not on-line. Module has not completed Dup_MAC_ID test.
Module On-line and connected. Network OK	Green	The module is on-line and has connections in the established state.
Module On-line but not connected	Flashing Green	Module has passed the Dup_MAC_ID test and is on-line, but has no established connections to other nodes.
Connection Time-Out	Flashing Red	One or more Connections have Timed-Out.
Critical Link Failure	Red	Failed communication device.
Device Self Test	Flashing Red-Green	Device is in self test
Device in Standby	Flashing Red	The module has no configuration

NOTES

## CHAPTER 3: DEVICENET NODE, DNT250 & PLC RELATIONSHIPS

### 3.1 General

Unless otherwise noted, all numeric radices are in decimal. In a few cases the notation 0xhh is used to indicate Hexadecimal radix.

### 3.2 Module to PLC Register Mapping

The %I, %Q, %AI and %AQ register references within the PLC Rack configuration allow the starting location of each of these groups of registers to be assigned to specific locations within the PLC register space.

With respect to the examples in this manual, the PLC Rack register references are set to one. This simplifies our examples by assuming that the register reference assignments start one. That is, the first %I, %Q, %AI and %AQ registers are assigned to PLC register %I1, %Q1, %AI1 and %AQ1. Although these references can be assigned anywhere within the PLC register space, we have made these assignments for simplicities sake.

### 3.3 Node to PLC Register Mapping

The DNT250 DeviceNet Scanner also provides mapping between the PLC's data registers and each individual DeviceNet node. For each configured node, the DNT250 provides mapping between the PLC's %Q, %AQ, %I and %AI registers and the DeviceNet node's data assemblies. As a result, each nodes polled data is immediately available to the PLC ladder code by accessing the appropriate PLC register. Note that the Node to PLC register assignments must be contained within the Module register definitions defined in Module to PLC Register Mapping defined above.

### 3.4 Message Packets and Fragmentation

A DeviceNet message packet can contain from 0 to 8 bytes of data. If more than 8 bytes of data are required, the message must be broken up into two or more packets or fragments. This is called fragmentation. The DNT250 handles this automatically, without any intervention by the user or the ladder program running in the PLC.

Fragmentation works differently for Polled messages versus Explicit messages. In the case of Polled messages, each message fragment is sent as fast as possible. In the case of large Explicit messages, the sender will send only the first packet. The sender then waits for an ACK message from the receiver. Upon receiving the ACK, the sender then transmits the next packet then waits for another ACK. This process repeats until all message fragments have been sent. This ACK process requires a considerable amount of time when compared to Polled fragmentation. As a result, LIEM or explicit messaging should be limited to data that needs to be accessed infrequently such as tuning or configuration parameters.

### 3.5 Physical Limitations

There are physical limits on the amount of data that can be transferred between the DNT250 and the PLC. This limit is 254 bytes of input data and 254 bytes of output data. The input limit is computed as the sum of all %I registers, in terms of bytes, plus the number of %AI registers times 2. The output limit is computed as the sum of all %Q registers, in terms of bytes, plus the number of %AQ registers times 2. Because of this size restriction, the number of data bytes that can appear on the DeviceNet network is also limited accordingly. This is a physical limitation of the Series 9030 PLC backplane.

There are also limits, depending on the PLC model, on the number of %I, %Q, %AI and %AQ registers supported. Consult the specific PLC model's documentation for more information.

The smallest data item that DeviceNet can handle is one byte. If an application requires only a single %Q output or command bit, a group of 8 %Qs must be allocated. The remaining 7 will be unused (or wasted). The same is true for bit type input data.

### 3.6 Data Assemblies

For a Polled Connection, the data bytes within the DeviceNet messages must be sent or received in a predetermined order. This order is specified by "Data Assemblies". Data Assemblies are published by the manufacturer of each specific DeviceNet node. To properly map the DeviceNet data to or from the PLC, the data assemblies for each DeviceNet node must be understood. Consult data sheets from each DeviceNet node manufacture for specific details.

From the point of view of the PLC, all input is via %I and %AI registers and all output data is via %Q and %AQ registers. From the point of view of DeviceNet nodes, all I/O is via streams of data bytes, known as data assemblies, that it sends or receives. In some DeviceNet nodes, the data assembly formats are fixed, in others they are configurable.

The DNT250 accepts output data from the PLC in the form of %Q and/or %AQ registers. It then translates this output data into DeviceNet polled messages, and sends them to the addressed DeviceNet node. The DNT250 then receives DeviceNet polled response messages from the addressed DeviceNet node, translates these DeviceNet messages and passes the data on to the PLC where they are stored in the appropriate %I and/or %AI registers.

For the DNT250, it is expected that any bit data will be transmitted first, followed by any word data. For example, if eight (8) %Q registers and 3 %AQ registers are to be sent, the following Data Assembly would result:

BYTE	FUNCTION
0	Bit data %Q1 to %Q8
1	%AQ1 LSB
2	%AQ1 MSB
3	%AQ2 LSB
4	%AQ2 MSB
5	%AQ3 LSB
6	%AQ3 MSB

### 3.7 DNT250 Configuration Introduction

Configuration of the DNT250 is performed using a PC running the configuration program DNCFG.EXE. There are a number of steps required in order to properly configure the DNT250 DeviceNet scanner module. These steps include:

- Specify the DNT250's MACID
- Specify the network baud rate,
- Specify if explicit messaging is to be supported
- Specify the module offsets for each of the register types (%Q, %AQ, %I and %AI).
- Define the Scan List,  
And for each node included in the Scan List:
  - Specify the %I Reference Address
  - Specify the %I Size (in Bits)
  - Specify the %AI Reference Address
  - Specify the %AI Size (in Words)
  - Specify the %Q Reference Address
  - Specify the %Q Size (in Bits)
  - Specify the %AQ Reference Address
  - Specify the %AQ Size (in Words)
  - Specify the Expected Packet Rate for the Polled Connection

Each of these items are discussed below:

Specify the DNT250's MACID.

This is the network address that the DNT250 will respond to. All nodes on the network, including the DNT250 must be configured with a unique MACID.

Specify the network baud rate.

This is the data bit rate at which all nodes on the network will operate. All nodes on the network must be configured to operate with the same baud rate.

Specify if explicit messaging is to be supported.

If the application requires LIEM or explicit messaging to be performed with any network node, this check box must be marked. This will determine which Data Assemblies are selected for the DNT250.

Specify the module reference addresses for each of the register types (%Q, %AQ, %I and %AI).

When the PLC Rack configuration is performed the module register addresses (or Base Addresses) will be required. The module reference addresses specify the start of each block of registers that are assigned to the DNT250 for each register type.

Scan List.

Before the DNT250 can communicate with any DeviceNet nodes, it must be loaded with a Scan List. The Scan List specifies which DeviceNet MACIDs to query. Only those nodes specified in the Scan List are considered to exist on the network.

For each node included in the Scan List:

The DNT250 also needs additional information about each node in the Scan List. This information includes a Reference Address and Size for each of the PLC register types (%Q, %AQ, %I and %AI). These items tell the DNT250 where the specific node data is mapped within the PLCs registers and the size of that data. We also need to know the Expected Packet Rate for the node. The Expected Packet Rate is in terms of milliseconds. This value is sent to the node where it is used as an inactivity timeout. Internal to the node, the value is multiplied by four and is used to measure the time between accesses by the DeviceNet master. If the node does not receive any messages from the master within this period of time, the node will enter the timed-out state, effectively going off-line.

NOTES



## CHAPTER 4: DNT250 STATUS AND COMMAND DATA ASSEMBLIES

### 4.1 Register Requirements and Definitions of Data Assemblies

The DNT250 requires several PLC registers to be assigned to itself. The DNT250 actually provides two different sets of data assemblies. The specific set that is selected is determined by the Explicit Support bit in the DNT250 module configuration. If Explicit Support is selected, the DNT250 needs more registers in order to handle the Explicit Messaging. See Appendix B. The following tables define the DNT250 register requirements and definitions of each of these data assemblies:

Explicit Support?	No	Yes
Number of %Q's	8	8
Number of %AQ's	0	4
Number of %I's	8	16
Number of %AI's	4	4

	<b>Explicit Support = no</b>	<b>Explicit Support = yes</b>
%Q1	Stop DeviceNet Scanning	Stop DeviceNet Scanning
%Q2	Not Defined – Reserved	Send Explicit Message
%Q3	Not Defined – Reserved	Not Defined – Reserved
%Q4	Not Defined – Reserved	Not Defined – Reserved
%Q5	Not Defined – Reserved	Not Defined – Reserved
%Q6	Not Defined – Reserved	Not Defined – Reserved
%Q7	Not Defined – Reserved	Not Defined – Reserved
%Q8	Not Defined – Reserved	Not Defined – Reserved
%AQ1	Not Used (available)	Transmit Buffer %R Number
%AQ2	Not Used (available)	Transmit Buffer Size (in bytes)
%AQ3	Not Used (available)	Receive Buffer %R Number
%AQ4	Not Used (available)	Receive Buffer Size (in bytes)

Table 4.3 - DNT250 Input Data Assemblies		
	Explicit Support = no	Explicit Support = yes
%I1	Someone is Off Line	Someone is Off Line
%I2	DeviceNet scanning is stopped	DeviceNet scanning is stopped
%I3	Not Defined – Reserved	Explicit Transaction Complete
%I4	Not Defined – Reserved	Explicit Error – Buffer Allocation Error
%I5	Not Defined – Reserved	Explicit Error – Can't get %R buffer
%I6	Not Defined – Reserved	Explicit Error – Invalid MACID
%I7	Not Defined – Reserved	Explicit Error – Node not configured
%I8	Not Defined – Reserved	Explicit Error – Node not on-line
%I9	Not Used (available)	Explicit Error – Can't make explicit connection
%I10	Not Used (available)	Explicit Error – Operation timed out
%I11	Not Used (available)	Explicit Error – other DNT Error
%I12	Not Used (available)	Explicit Error – Receive buffer overrun
%I13	Not Used (available)	Explicit Error – Can't put %R buffer
%I14	Not Used (available)	UCMM Exp Enable Error
%I15	Not Used (available)	Not Defined – Reserved
%I16	Not Used (available)	Not Defined - Reserved
%AI1	On Line status, nodes 0-15	On Line status, nodes 0-15
%AI2	On Line status, nodes 16-31	On Line status, nodes 16-31
%AI3	On Line status, nodes 32-47	On Line status, nodes 32-47
%AI4	On Line status, nodes 48-64	On Line status, nodes 48-64

#### %Q1 Stop DeviceNet Scanning

Setting this bit causes the DNT250 to close or release all connections between itself and all DeviceNet nodes. No further communications will occur between the DNT250 and the DeviceNet nodes until the Stop DeviceNet Scanning bit is cleared. While the scanning is stopped, the DeviceNet Scanning is Stopped bit %I2 bit will be set.

#### %Q2 Send Explicit Message

When this command is issued the DNT250 will attempt a LIEM sequence. The specifics of the LIEM are discussed in chapter 4. %I3 through %I13 are used to report the completion status of this command.

#### %Q3-8 Not Defined - Reserved

These bits are reserved and can not be used for any purpose.

#### %AQ1 Transmit Buffer First %R Number

For LIEM sequences, this register must be loaded with the number of the first %R register of the explicit message transmit buffer. See chapter 4 for more details.

#### %AQ2 Transmit Buffer Size (in Bytes)

For LIEM sequences, this register must be loaded with the number of bytes of data to be sent to the target node. See chapter 4 for more details.

#### %AQ3 Receive Buffer First %R Number

For LIEM sequences, this register must be loaded with the number of the first %R register of the explicit message receive buffer. See chapter 4 for more details.

**%AQ4 Receive Buffer Allocation Size (in Bytes)**

For LIEM sequences, this register must be loaded with the size in bytes of the receive buffer. This value can be as large as the programmer likes, but an error will result if the node sends a message that is larger than will fit in the receive buffer. See chapter 4 for more details.

**%I1 Someone is Off Line**

This status bit indicated that one or more configured nodes are not on line.

**%I2 DeviceNet Scanning is Stopped**

This status bit indicates that the DeviceNet scanning is stopped as a result of the bit %Q1 Stop DeviceNet Scanning having been set by the PLC.

**%I3 Explicit Transaction Complete**

This status bit indicates that the requested LIEM sequence is complete. Once this bit is set, the PLC must check bits %I4 through %I14 which indicate if any errors have occurred.

**%I4 Explicit Error – Buffer Allocation Error**

If set, this LIEM status bit indicates that the values in %R1 and %R2 or %R3 and %R4 are in error. The transmit buffer and/or the receive buffer is/are outside the available %R registers of the specific PLC CPU model.

**%I5 Explicit Error – Can't get %R Buffer**

If set, this LIEM status bit indicates that the DNT250 was unable to read the transmit buffer from the PLC.

**%I6 Explicit Error – Invalid MACID**

If set, this LIEM status bit indicates that the MACID contained in the transmit buffer is too large.

**%I7 Explicit Error – Node Not Configured**

If set, this LIEM status bit indicates that the MACID contained in the transmit buffer is not configured.

**%I8 Explicit Error – Node Not On Line**

If set, this LIEM status bit indicates that the MACID contained in the transmit buffer is not On Line.

**%I9 – Explicit Error**

Can't make explicit connection. If set, this bit indicates that the explicit connection can not be made.

**%I10 Explicit Error – Operation Timed Out**

If set, this LIEM status bit indicates that the LIEM sequence timed out. Most likely the target node did not send a response in a timely manner.

**%I11 Explicit Error – Other DeviceNet Error**

If set, this LIEM status bit indicates that some other miscellaneous DeviceNet error occurred.

**%I12 Explicit Error – Receive Buffer Overrun**

If set, this LIEM status bit indicates that the response message received from the target node was too large to fit in the receive buffer indicated by %R3 and %R4.

**%I13 Explicit Error – Can't put %R Buffer**

If set, this LIEM status bit indicates that the DNT250 was unable to write the receive buffer to the PLC.

**%I14 Explicit Error – UCMM Exp Enable Error**

If set, this LIEM status bit indicates that an attempt was made to perform an LIEM to a UCMM type node without the UCMM Exp Enable bit set.

**%I15 Not Defined – Reserved****%I16 Not Defined – Reserved**

These bits are reserved and can not be used for any purpose.

**%A11 On Line Status of Nodes 00 - 15****%A12 On Line Status of Nodes 16 - 31****%A13 On Line Status of Nodes 32 - 47****%A14 On Line Status of Nodes 48 - 63**

These four %AI registers collectively contain 64 individual status bits. One bit is assigned to each of the 64 possible DeviceNet node addresses. Status bits for nodes 0 through 15 are mapped into the first %AI register. Status bits for nodes 16 through 31 are mapped into the second %AI register. Status bits for nodes 32 through 47 are mapped into the third %AI register. Status bits for nodes 48 through 63 are mapped into the fourth %AI register. The status bit for node 0 is placed in the least significant bit position of the first %AI register, with consecutively higher addressed nodes status bits being placed in the next higher bit positions. See the table below.

Each individual status bit is set TRUE if the corresponding DeviceNet node is On Line. On-Line is defined as the node's Polled connection being in the Established state and the node is communicating Polled data.

Table 4.4 – HE693DNT250 Unit Status Bits																
%AI[offset+0]																
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACID	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
%AI[offset+1]																
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACID	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
%AI[offset+2]																
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACID	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
%AI[offset+3]																
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACID	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48

**Note:** Many DeviceNet nodes also provide additional diagnostic data within their individual data assemblies.

## CHAPTER 5: EXPLICIT MESSAGING USING THE DNT250

### 5.1 General

The DNT250 DeviceNet Scanner, supports both POLLED and EXPLICIT connections.

Explicit Messaging requires a great deal of overhead in both the DNT250 and the PLC. Explicit Messaging requires access to the PLC's memory using the BackPlane Mail features of the Series 90-30 system. BackPlane Mail requests are handled only once per PLC logic scan. Large amounts of data will require multiple PLC scans for all of the data to be transferred between the PLC and the DNT250. If the PLC is also doing a significant amount of processing, requesting an Explicit Message can impact the DeviceNet Poll Sequence by several tens of milliseconds. As a result, Explicit Messages should be reserved for access to infrequently needed data, such as configuration or tuning parameters only.

The LIEM feature requires that the Explicit Support check box be selected when the DNT250 module is configured. The act of doing this selects the proper data assemblies for the DNT250.

The sequence below presents a general description of the process that must be executed to service an Explicit request.

1. The ladder code builds an explicit request message in a group of %R registers.
2. The ladder code plugs the start of and length of the transmit and receive buffers into four %AQ registers. The transmit and receive buffers must both be located in the %R register space of the PLC.
3. The ladder code then sets the "Send Explicit Message" command bit.
4. At the top of the Scan List, the DNT250 checks the "Send Explicit Message" command bit.
5. If the "Send Explicit Message" command bit is set, processing of the explicit request begins.
6. The four %AQ registers are examined and checked for validity.
7. The DNT250 requests the PLC to send the transmit buffer via Backplane mail.
8. The DNT250 then checks the MACID contained in the transmit buffer. Several checks take place, MACID out-of-range, referenced node not configured and addressed node not On Line.
9. Then the message is formatted and sent to the addressed node.
10. When the response message is received, a check is made to see if the allocated receive buffer is large enough to accept the message.
11. Copy the message to the receive buffer via Backplane mail.
12. Set the Explicit Transaction Complete bit.
13. If there were any error detected in any of the previous steps, the process is aborted and the appropriate error status bit along with the Explicit Transaction Complete bit are set.

See chapter 3 for descriptions of all of the command, status and error bits.

### 5.2 Explicit Messages and the DNT250

Early versions of the DNT250 did not support the UCMM but some did support explicit messaging for Group 2 Only type nodes.

The Group 2 Only nodes allow only one explicit messaging format. The UCMM supports three additional formats for a total of four formats.

For the least amount impact to those users of early DNT250s who are upgrading, the new DNT250 supports all of these different explicit message formats. First, the new DNT250 supports the legacy format of the early DNT250s. The new DNT250 also supports the UCMM formats as well.

With respect to the UCMM type nodes, as the DNT250 establishes a connection with the UCMM node, they negotiate which of the four explicit formats are to be used for all future explicit communications with that specific node. The DNT250 automatically converts format type as needed to properly communicate with the various UCMM nodes.

To select between the legacy and UCMM formats, there is a PLC Rack configuration parameter that must be set up. This parameter, "UCMM Explicit Enable" when cleared to FALSE tells the DNT250 to use the legacy explicit messaging mode to communicate with "Group 2 Only" type nodes only. The DNT250 can not do explicit messaging to UCMM type nodes in this mode. When the parameter, "UCMM Explicit Enable" is set to TRUE, the DNT250 can communicate via explicit messaging to both UCMM and "Group 2 Only" nodes.

### 5.3 Building Explicit Messages

As mentioned previously, the DNT250 provides two formats for explicit messages. The format used depends on the PLC Rack Parameter "UCMM Explicit Enable". When the "UCMM Explicit Enable" parameter is clear (FALSE), the legacy format is used.

In the following example, let's assume that the "UCMM Explicit Enable" parameter is clear or FALSE. We want to read the polled consumption size from the node at MACID 3, we also want to locate the transmit buffer at %R101, the receive buffer at %R51 and we will also allocate 20 bytes to the receive buffer.

BYTE NUMBER	REGISTER NUMBER	DESCRIPTION	TRANSMIT BUFFER DATA
0	R101 LSB	MACID (Node Address)	03
1	R101 MSB	Service Code (Get Attribute Single)	14
2	R102 LSB	Class ID (Connection Class)	05
3	R102 MSB	Instance ID (Polled Connection)	02
4	R103 LSB	Attribute # (Consumption Size)	07

REGISTER	DESCRIPTION	VALUE
%AQ1	Start of Transmit Buffer	101
%AQ2	Number of Bytes to Transmit	5
%AQ3	Start of Receive Buffer	51
%AQ4	Receive Buffer Allocation Size	20

Now, we will repeat the same example except lets assume that the "UCMM Explicit Enable" parameter is set (TRUE):

BYTE NUMBER	REGISTER NUMBER	DESCRIPTION	TRANSMIT BUFFER DATA
0	R101 LSB	MACID (Node Address)	03
1	R101 MSB	Service Code (Get Attribute Single)	14
2	R102 LSB	Class ID (Connection Class)	05
3	R102 MSB		00
4	R103 LSB	Instance ID (Polled Connection)	02
5	R103 MSB		00
6	R103 LSB	Attribute # (Consumption Size)	07

REGISTER	DESCRIPTION	VALUE
%AQ1	Start of Transmit Buffer	101
%AQ2	Number of Bytes to Transmit	7
%AQ3	Start of Receive Buffer	51
%AQ4	Receive Buffer Allocation Size	20

In the following example, let's assume that the "UCMM Explicit Enable" parameter is clear or FALSE. We want to write the polled expected packet rate to the node at MACID 3, we also want to locate the transmit buffer at %R101, the receive buffer at %R51, we will also allocate 20 bytes to the receive buffer.

BYTE NUMBER	REGISTER NUMBER	DESCRIPTION	TRANSMIT BUFFER DATA
0	R101 LSB	MACID (Node Address)	03
1	R101 MSB	Service Code (Set Attribute Single)	16
2	R102 LSB	Class ID (Connection Class)	05
3	R102 MSB	Instance ID (Polled Connection)	02
4	R103 LSB	Attribute (Expected Packet Rate)	07
5 & 6	R104	Data Value	1000

REGISTER	DESCRIPTION	VALUE
%AQ1	Start of Transmit Buffer	101
%AQ2	Number of Bytes to Transmit	7
%AQ3	Start of Receive Buffer	51
%AQ4	Receive Buffer Allocation Size	20

Now, we will repeat the same example except lets assume that the "UCMM Explicit Enable" parameter is set (TRUE):

BYTE NUMBER	REGISTER NUMBER	DESCRIPTION	TRANSMIT BUFFER DATA
0	R101 LSB	MACID (Node Address)	03
1	R101 MSB	Service Code (Set Attribute Single)	16
2	R102 LSB	Class ID (Connection Class)	05
3	R102 MSB		00
4	R103 LSB	Instance ID (Polled Connection)	02
5	R103 MSB		00
6	R104 LSB	Attribute (Expected Packet Rate)	07
7	R104 MSB	Data Value Low Byte	E8(Hex)
8	R105 LSB	Data Value High Byte	03

REGISTER	DESCRIPTION	VALUE
%AQ1	Start of Transmit Buffer	101
%AQ2	Number of Bytes to Transmit	9
%AQ3	Start of Receive Buffer	51
%AQ4	Receive Buffer Allocation Size	20

## 5.4 How to Interpret Explicit Response Messages

The normal, expected response from an Explicit Message is the Acknowledge Message:

BYTE NUMBER	DESCRIPTION	EXAMPLE
0	Number of	06
1	bytes received	00
2	MACID	01
3	*Service Code	0x90(Hex)
4	Optional Data	Xx
-	Optional Data	Xx
-	Optional Data	Xx
n	Optional Data	Xx
* Previously sent Service Code (0x10) + 0x80		

Note: The Most Significant Bit in the Service Code byte is used as a Response Bit indicating that the command was properly received. [Service Code 0x10 + Response Bit 0x80 = 0x90]. If any extra data needs to be returned, that data will be placed into subsequent bytes.

## 5.5 Explicit Message Errors

In the case that an Explicit Message requests a function that can not be performed by the addressed node, an Explicit Error Message will be returned in the Receive Buffer

An Explicit Error Message takes the following form:

Byte Number	Description	Example (values in HEX)	Word Offset	Value (from example)
0	Number of bytes received	06	%R1	0006
1		00		
2	MACID	01	%R2	0x9401
3	Service Code	94		
4	General Error Code	xx	%R3	xxxx
5	Additional Error Code	xx		

Like a "normal" response, the Service Code byte has the Most Significant Bit set as the Response Bit. Therefore, the actual Error Code is 0x14; with the Response Bit set the received code is 0x94. Error Code is 0x14 (0x94) has a special meaning. This indicates that the addressed node has detected an error and the two bytes following indicate the specifics to that error. The next byte indicates the General Error Code. See appendix C for a list of and definitions of General Error Codes. Some General Error Codes support an Additional Error Code if appropriate to indicate additional information.



## CHAPTER 6: CONFIGURATION

### 6.1 General

Before the DNT250 can be used, it must first be properly configured. This consists of two steps: Module Configuration and PLC Rack Configuration.

IMPORTANT: The module configuration must occur prior to configuring the PLC rack.

### 6.2 DNT250 Module Configuration

Included with the DNT250 module is a diskette containing the configuration utility, DNCFG.EXE. DNCFG.EXE is a 32 bit Windows application. It can be installed and run on most Windows 95, Windows 98 or Windows NT based PC. DNCFG.EXE will not run under DOS or Windows 3.1. The PC must be connected to the DNT250 via the module's integral RS-485 serial port.

Module operational parameters (i.e. MACID, baud rate, Polled Production size, Polled Consumption size and Expected Packet Rate) are configured on behalf of each node using DNCFG.EXE. This program also allows entire network configurations to be uploaded or downloaded to or from the DNT250, saved to or loaded from a disk, or sent to a printer for hard copy.

Exact details of using the DNCFG.EXE configuration utility can be found in chapter XXX. Before going there we need to discuss a number of concepts that must be understood before module configuration can be successfully completed:

1. Does the application require the use of LIEM (explicit messaging)? This will determine which data assembly must be used for the DNT250 itself.
2. If yes, is LIEM required to address any UCMM type nodes? This will determine if the "UCMM Exp Enable" bit needs to be set in the rack configuration.
3. At what baud rate does the network need to run? This is needed for the DNT250 module configuration as well as the configuration of each individual DeviceNet node. All nodes including the DNT250 must be configured for the same baud rate.
4. Is Data Assembly information available for each DeviceNet node in the network? This can be in the form of printed information in the manufactures manual or in the form of a EDS file provided by the manufacture. Note that some EDS files do not provide the nodes' polled consumption and production sizes. This information will also be required when the PLC Ladder Code is written.
5. Assign MACIDs (node addresses) to each DeviceNet node. The MACIDs assigned to each node is totally arbitrary. The only thing that might have an effect here is the fact that the DNT250 actually scans the nodes in order by MACID. Gaps can be left in the MACID assignments to allow for future nodes to be added, if desired. DeviceNet provides for nodes with lower MACIDs to have a higher network priority. But because of the manner in which the DNT250 scans the network, this priority advantage does not exist. The MACID for the DNT250 can be assigned any unused address. Every node on the network must be assigned a unique MACID.
6. All bit type input data must be assigned to a contiguous block of PLC %I registers and all word type input data must be assigned to a contiguous block of PLC %AI registers.
7. Furthermore, all bit type output data is required to be assigned to a contiguous block of PLC %Q registers and all word type output data must be assigned to a contiguous block of PLC %AQ registers.

### 6.3 PLC Rack Configuration

Configuration of the GE Series 90-30 Rack can be accomplished via several methods. In this discussion we will use Logicmaster 90 configuration software. In any case, module configuration (using DNCFG.EXE) must be completed before the rack configuration should be attempted.

There are four parameters that need to be understood before proceeding with the PLC Rack configuration. These parameters are:

1. Always 1  
This parameter must be set to 1.
2. Block Move  
This parameter selects one of two possible methods of moving data to and from the PLC's registers.
  - When set to 0, data is transferred between the PLC and the DNT250 either one byte or one word at a time. With this method, there exists the possibility to have a data consistence problem. It is possible that half of a PLC register could be updated on one PLC scan and the balance of the same register could be updated on the next PLC scan. This setting is only provided for compatibility with older versions of the DNT250.
  - When set to 1, all data is transferred between the PLC and the DNT250 as one contiguous uninterrupted block. It is not possible to have any kind of data consistence problem using this mode. This is the recommended setting for this parameter.
3. UCMM Explicit Enable  
This parameter allows selection between the old and new methods of explicit messaging. Note that the "Explicit Support" check box in the Module configuration must be checked before this parameter will have any effect.
  - When set to 0, the legacy mode of explicit messaging is selected. In this mode, explicit messaging can only be carried on with Group 2 Only types of nodes. It is not possible to conduct explicit messaging with UCMM type nodes in this mode. This mode is provided only for compatibility with older versions of the DNT250.
  - When set to 1, explicit messaging can be carried on with both UCMM and Group 2 Only types on nodes. This mode is suggested for all new system designs.
4. Wait Time  
This parameter allows the user to specify the length of time the DNT250 will wait for a response from a node. Using a zero for this parameter value denotes the use of a default value of 50 milliseconds. In most applications this value should be acceptable. In the case that there is a node on the DeviceNet that is sluggish in its response, a larger value can be used. Acceptable values include the series, 10, 20, 30, 40, ... 250 milliseconds. The timing resolution within the DNT250 is 10 milliseconds. Therefore, there is no advantage to use values that are not divisible by 10. Use caution with smaller numbers, if too small a value is used, nodes may be logged off line during LIEM explicit messaging.

The following sequence may be used to configure the PLC rack for a DNT250 module via Logicmaster 90 configuration software:

1. Start up the Logicmaster 90 configuration software.
2. At the Logicmaster 90 software screen select <F2> Configuration Package.
3. Create a program folder at the prompt (to later store data into) and press <ENTER>.
4. At the menu screen select <F1> I/O Configuration.
5. Use arrow keys to select the slot where the DNT250 will reside.
6. If a scanner configuration already exists in the slot you select, delete the old module by using the <shift F4> key.
7. Press <F8> (Other) to select module.
8. Select Foreign Module by pressing the <F3> key.
9. Using the information you copied or printed from the DTNCFG.EXE program, fill in values for the %I reference, %I size, %Q reference, %Q size, %AI reference, %AI size, %AQ reference and %AQ size parameters.
10. Set the parameter byte 1 on right side of screen to '1' and press <ENTER>
11. Set the parameter byte 2, Block Move, on right side of screen to the user selected value and press <ENTER>
12. Set the parameter byte 3, UCMM Explicit Enable, on right side of screen to the user selected value and press <ENTER>
13. Set the parameter byte 4, Wait Time, on right side of screen to the user selected value and press <ENTER>
14. Parameter bytes 5 through 16 should be set to zero.
15. Exit to main menu by pressing <ESCAPE> key *twice*.
16. Download the configuration to the PLC rack by pressing the <F9> key.
17. Press the <F2> key to save the configuration to the PLC. Press the <ENTER> key to begin store and overwrite.
18. Power down the PLC rack.
19. With the rack power off, plug the DNT250 module into the PLC slot selected during Step 5.
20. Power up the PLC rack.

If the module ever needs replaced or reconfigured, you must first delete any information appearing at the selected module's position. If you do not, the new configuration may not be properly recognized.

## 6.4 DNT250 Configurator

### 6.4.1 General

The diskette distributed with the HE693DNT250 module contains the HE693DNT250 DeviceNet Configuration Utility (DNCFG.EXE) setup program, setup.exe.

1. Insert the diskette into the A: (or B:) drive.
2. As always, it is a good practice to make a back-up copy of the installation diskette.
3. Execute the setup program, setup.exe, on the diskette and follow the instructions for installation.
4. The DNCFG.EXE utility program will be installed in the installation directory selected during setup.

DNCFG.EXE can work off line, with no module attached. Normally, DNCFG.EXE is attached to a HE693DNT250 module. In order to configure a HE693DNT250 module, the user must first connect it to the PC:

1. With the rack power off, plug the HE693DNT250 module into an open I/O slot of a GE Fanuc Series 90-30 rack.
2. Plug the SNP adapter (P/N HE693SNP232) into the 15-pin female RS485 port on the HE693DNT250 module.
3. Plug one end of the 9-pin female D to 9-pin female "straight-through" cable (P/N HE693CBLSNP) to the 9-pin end of the SNP Adapter. Connect the free end of the cable to the desired COM port on the PC. If the PC is equipped with 25-pin COM ports, use a standard 25-pin female to 9-pin male adapter.
4. Power up the PLC rack.

#### 6.4.2 Configuration Procedures

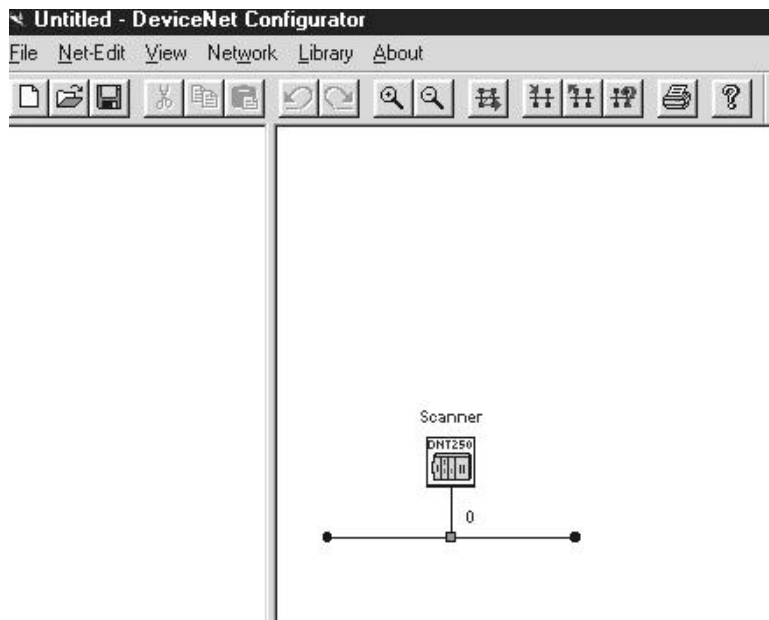
*The focus of this example configuration is to show how to configure a new network using the DNCFG.EXE Configuration Utility and creating a Node Template Library. This example shows one approach of configuring the network, but there are numerous approaches that can be used. After a network has been created, many of the same procedures can be used to edit the existing network.*

1. Identify all devices (nodes) that are going to be a part of the network. Although not mandatory, it is highly recommended that the user first create a template library for all the nodes on the network. This is helpful if the user intends to use the nodes in other configurations / applications, because it will decrease re-work. *If a template library is not desired, be sure to refer to Step 1 and 2; Refer to Step 7 for a summary of procedures.*

It is also recommended that the user obtain EDS files for each device on the network and place them into a directory. EDS files contain information that is specific to a device, and the EDS file can be obtained from the device's vendor. If an EDS file cannot be obtained, the user can contact the vendor for the information requested on the General Setup Screen, which is discussed later in the configuration procedure.

2. Copy the DNCFG.EXE Configuration Utility from the disk that is sent with the module. Place the DNCFG.EXE Configuration Utility into a file. Double-click the DNCFG.EXE Configuration Utility icon in the installation directory.

**Note:** The title of the screen indicates that the screen is untitled. It remains as such until the configuration is saved with a unique name.



**Figure 6.1 – Untitled Default Configuration Screen**

The screen has two window panes. The left pane is the **Product Template View**, which is used to store node configuration data in a template library. The user creates the template library as discussed later in the procedure. Once a template library is created, these templates can be dragged to the right pane on the screen. The nodes can then be configured.

The right pane is the **Network View** where the user can build a visual representation of a network depicting the DNT250 Master and all slave nodes (devices). The nodes are configured in this view.

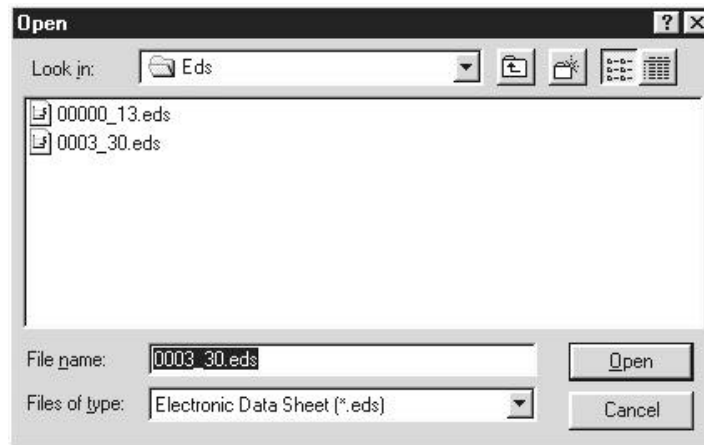
3. Creating a Node Template Library with *or* without using EDS Files.

**Note:** Creating a template for a node is not to be confused with configuring a node.

### Creating a Template for a Node using an EDS File

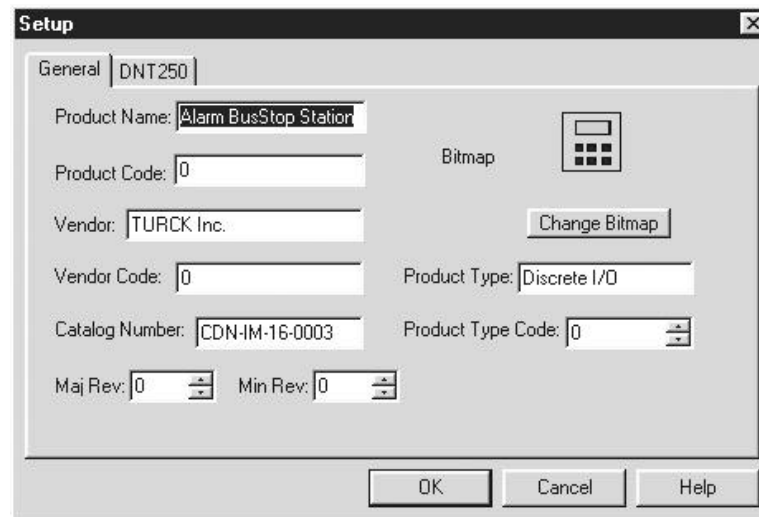
Note: Before importing an EDS file, it is recommended to save any changes already done to the network and library. A faulty EDS file can possibly cause malfunction in the program when the file is being "read" by the program. A faulty EDS file refers to one that does not conform to the DeviceNet standard.

To import EDS files, refer to Figure 6.1 and press **Library, Import from EDS File**. The following screen appears. Select the desired EDS file and press **Open**.



**Figure 6.2 – EDS Files**

The following screen appears.



**Figure 6.3 – Setup Screen in a EDS File (General Tab Selected)**

**General Tab:** This screen is not edited by the user, but it is recommended that the **Catalog Number**, **Maj. Rev.** and **Min. Rev.** numbers be updated by the user as needed. On the right side of the screen, a **Bitmap icon** can be double-clicked by using the left mouse button. Upon doing so, a bitmap file associated with the node can be selected (if a bitmap has been created) . The bitmap can be changed by pressing **Change Bitmap**.

Now, select the **DNT250 Tab**.

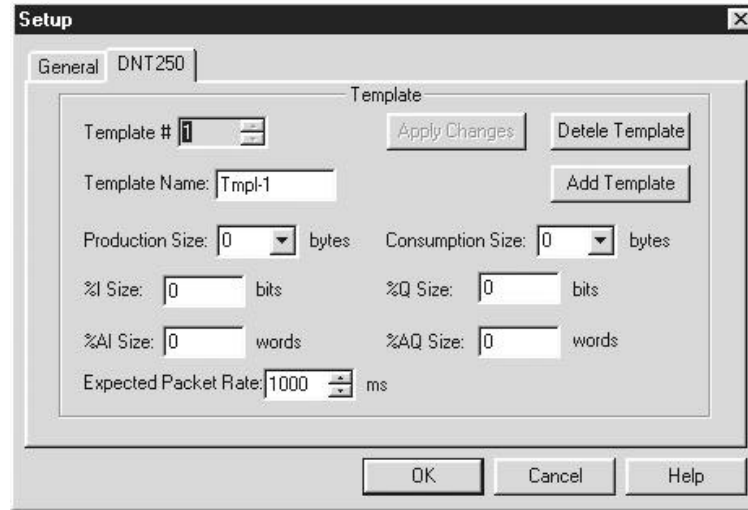


Figure 6.4 – Setup Screen in a EDS File (General Tab Selected)

**DNT250 Tab:** Each node can have one or *more* templates associated with it, and the user can select a desired template using the up-down arrows located next to the **Template #**. The user can enter an identifying name in the **Template Name** block.

The **Expected Packet Rate** is the time when the master establishes communication with a node. It sets the timer in the node, and the Expected Packet Rate is used as a basis as a time-out. If the master does not perform a scan within the **Expected Packet Rate x 4**, the slave node goes off-line until the master establishes communication with the node on the next scan.

**Note:** If EDS files have been used, it is recommended not to modify the **Production Size** or the **Consumption Size**. However, the register sizes can be edited as needed.

To delete the template, press **Delete Template**. In some instances, a node might need more than one template. In that case, press **Add Template** and repeat the template process. To check each template, the user can use the up-down arrows to select the desire template.

Press **OK**. Be sure to save the configuration with a unique name.

### **Creating a Node Template without EDS Files**

Right-click anywhere on the screen (see Figure 6.1) and select **Add New Node**. The following generic screen appears.

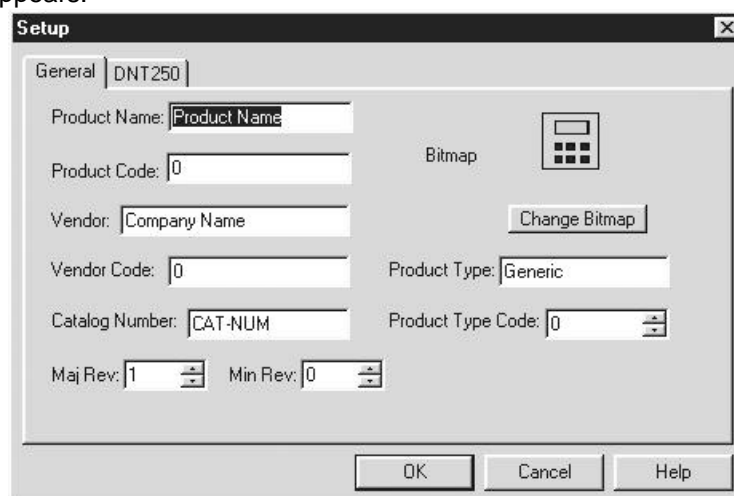


Figure 6.5 – Setup Screen in a EDS File (DNT250 Tab Selected)

**General Tab:** Although the user does not need to configure this screen, it is extremely useful to do so. To obtain information about the device including product and vendor codes, it may be necessary to contact the vendor. On the right side of the screen, a **Bitmap icon** can be double-clicked by using the left mouse button. Upon doing so, a bitmap file associated with the node can be selected (if a bitmap has been created) . The bitmap can be changed by pressing **Change Bitmap**.

Now, select the **DNT250 Tab:** Each node can have one or more templates associated with it, and the user can select a desired template using the up-down arrows located next to the **Template #**. The user can enter an identifying name in the **Template Name** block.

The **Expected Packet Rate** is the time when the master establishes communication with a node. It sets the timer in the node, and the Expected Packet Rate is used as a basis as a time-out. If the master does not perform a scan within the **Expected Packet Rate x 4**, the slave node goes off-line until the master establishes communication with the node on the next scan.

Specify the %I size, %AI size, %Q size, and %AQ size. Production and Consumption sizes change automatically to reflect the changes.

To delete the template, press **Delete Template**. In some instances, a node might need more than one template. In that case, press **Add Template** and repeat the process. To check each template, the user can use the up-down arrows.

Press **OK**.

4. Upon pressing OK and saving the template with a unique name (such as Project X), the following information appears in the Node Template Library on the left pane of the screen.

To create a template for the next node, either select **Library, Insert New Item** or **Import from EDS File** and repeat steps 3 and 4. The user can also right-click on the left pane and select **Insert New Item** or **Import from EDS File**.

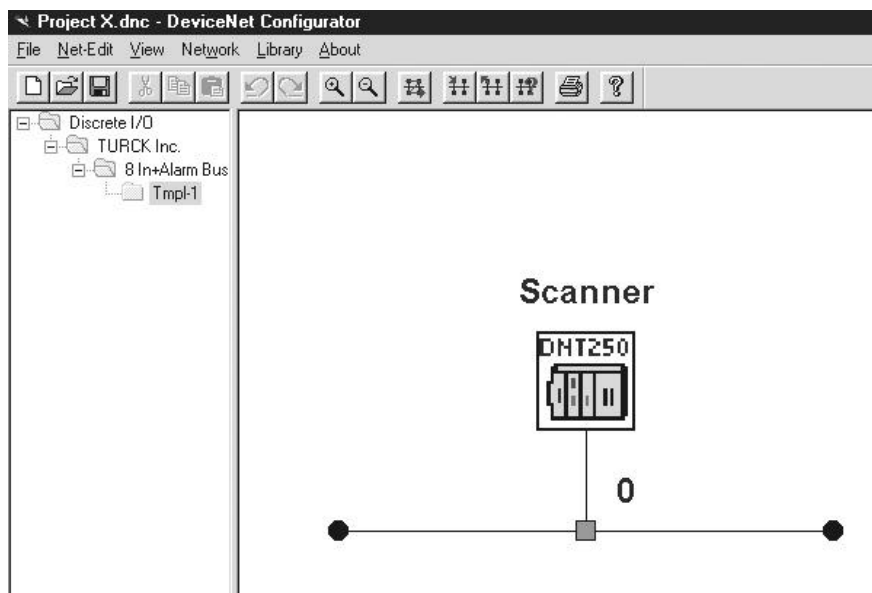


Figure 6.6 – Node Information Displayed on Left Pane

5. After the complete node template library has been created, it is now appropriate to configure the **Network Properties** screen (Figure 6.7). Afterwards, the nodes can be placed into the visual display of the network on the right side of the screen, and then each node can be individually configured.



**Network Properties**

Scanner:  DNT250  DNT350

Baud Rate: 125 KB

Explicit Support:

Series 90-30 CPU

Model: CPU311

%I Ref Addr: 1    %I Size: 8

%AI Ref Addr: 1    %AI Size: 4

%Q Ref Addr: 1    %Q Size: 8

%AQ Ref Addr: 1    %AQ Size: 0

Total input bytes [%I + %AI]: 9

Total output bytes [%Q + %AQ]: 1

Note: 1. The sum of all inputs, %I and %AI registers, can not exceed 254 bytes  
2. The sum of all outputs, %Q and %AQ registers, can not exceed 254 bytes

Cancel    OK

**Figure 6.7 – Node Information Displayed on Left Pane**

Select **Network, Properties** or double-click anywhere on the screen. A screen similar to Figure 6.7 appears. The user needs to select the scanner type, baudrate, and CPU. The user needs to set the starting **Reference Addresses** of the registers. Note that the size of the registers are based upon the CPU model selected. The size of the registers are not edited by the user, and they are automatically set by the software. It is important to pay attention to the total **Input Bytes** and **Output Bytes**, and the notes shown on the screen (Figure 6.7) regarding byte limits.

6. After configuring the Network Properties screen, it is appropriate to *place* the nodes (one at a time) into the visual display of the network on the right side of the screen and then **configure** the nodes individually.

Left-click on Tmpl-1 and drag and drop it to the right side of the screen (Figure 6.8). The node is automatically placed into the appropriate position. If other node templates are contained in the Node Template Library, repeat the drag and drop process.

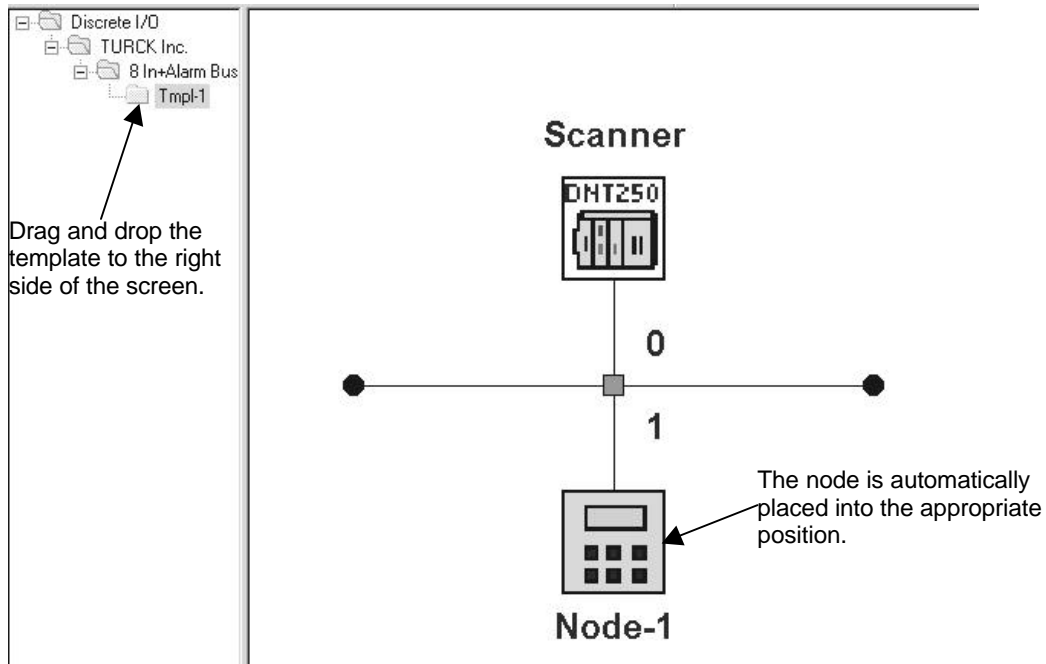


Figure 6.8 – Node Information Displayed on Left and Right Pane

The node (one master DNT250 or one of 63 possible slaves) can now be configured. Double-click Node 1, and the following screen appears in Figure 6.9.

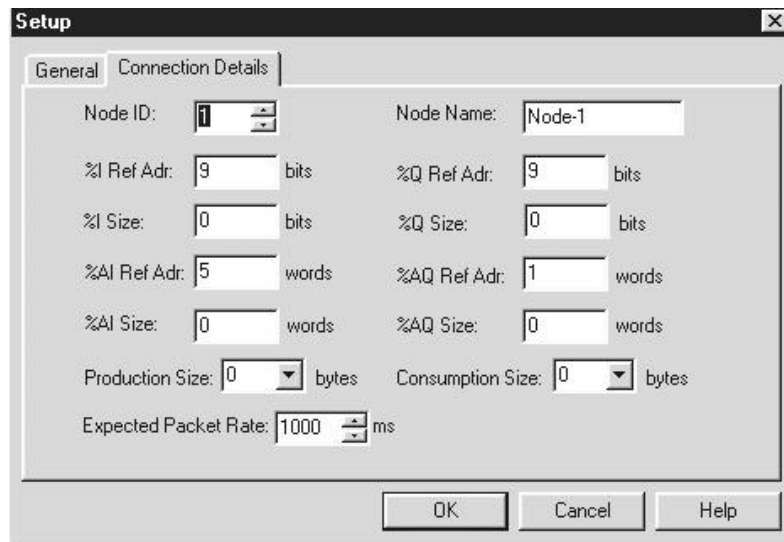


Figure 6.9 – Connection Details Tab (Node 1)

Enter the name and node number. The software program sets the register sizes automatically, but the user can change the starting reference addresses. After doing so, the user can choose to press **Network, Auto Remap** to remap the node in the network.

**Caution:** Before using the **Auto Remap** feature, be sure to save the configuration beforehand. Auto Remap is primarily intended for use in new configurations. Auto Remap reassigns the reference addresses for all nodes so that no gaps are left in the map. Auto Remap action can not be undone and needs to be used with caution.

The **General Tab** [not shown] contains information about the node as shown in Figure 6.3.

7. If a Node Template Library is not desired, right-click on the right pane and select **Network Properties** (if this has not been configured yet). Refer to Step 5 to configure the Network Properties screen. Then right-click on the right pane again and select **Add New Node**. Left-click on the new node and perform the procedures described in Step 3 (to use EDS Files) and Step 6 to configure the node.

8. After the network is built, using the mouse, press **Network, Select Com Port**. Select an appropriate communication port.

The configuration can now be **downloaded** from the PC to the DNT250 by selecting **Network, Download**. Configuration is now completed.

If a copy of the current configuration is needed, the user can select **Network, Upload** the configuration from the DNT250 to the PC.

To verify the current configuration, press the **Network, Verify**.

**Note:** While trying to establish communications during **upload, download or verification**, an error message may appear. If this happens, try again. If the error persists after two attempts,

- a. Check the module-PC connection
- b. Make sure that the Com Port is not in use by another application.
- c. Power cycle the PLC.

To document and save information about the node configuration, select **Network, Project Summary**.

To see a summary of I/O devices, select **Network, I/O Summary**.

NOTES

## CHAPTER 7: DEVELOPING USER SOFTWARE

### 7.1 Software Support for the Polled Connection

When using the Polled Connection, all inputs from and outputs to the DNT250 are handled by normal ladder programming techniques to set appropriate values and read any resulting values. Inputs from the DNT250 (and the DeviceNet nodes) are placed in %I or %AI registers; outputs are placed into %Q or %AQ registers. Which registers are used is a function of the configuration of the DNT250 and of the PLC Rack Configuration.

Data is passed via the DeviceNet Polled Connection automatically. Data is read from or written to the network devices according to the registers programmed using DNCFG.EXE and the LM90 Configurator Program. No extra ladder programming is required to send or receive data using Polled Messages.

### 7.2 Software Support for the Explicit Connection

The Explicit Connection requires additional ladder programming support.

The Explicit Connection must first be specifically enabled by marking the EXPLICIT SUPPORT check box in the DNCFG.EXE configuration program.

During PLC operation, the ladder program must check both the SEND EXPLICIT MESSAGE and EXPLICIT TRANSMISSION COMPLETE bits. Both bits must be clear. The program must not modify the %AQ registers nor the %R register buffers pointed to by the %AQ registers if either bit is set.

**NOTE:** If the Explicit Support check box is not marked, the EXPLICIT TRANSMISSION COMPLETE bit is permanently set to '1'.

Finding both bits clear, the ladder program may then proceed to build the transmit buffer in %R Registers. The ladder program must also modify the %AQ registers to point to the %R transmit and receive buffers.

%AQ1	Offset of Base %R Register for the Transmit Buffer
%AQ2	Size of the Transmit Buffer, <u>in bytes</u>
%AQ3	Offset of Base %R Register for the Receive Buffer
%AQ4	Size of the Receive Buffer, <u>in bytes</u>

The offset of the %R register holding the first word of the Explicit Transmit Buffer is then placed into the %AQ1. The SIZE of the buffer IN BYTES is placed in %AQ2.

The PLC ladder program must also define a Explicit Receive Buffer. The offset of the %R register to hold the first word of the received message is placed in %AQ3. The allocated size of the receive buffer IN BYTES is place in %AQ4. The PLC ladder code must be sure to allocate enough buffer space (%R registers) to handle the largest expected incoming message.

The ladder program then sets the SEND EXPLICIT MESSAGE bit. The ladder program must then monitor the EXPLICIT TRANSMISSION COMPLETE bit. This bit will be set when the command is complete. Fragmentation if required is handled automatically by the DNT250.

After the EXPLICIT TRANSMISSION COMPLETE bit is set, the ladder program must check the EXPLICIT ERROR BITS for any possible errors. If any errors exist, they should be processed by the ladder code. If there are any errors are present, any data in the receive buffer should be considered invalid. If there are no Explicit Errors, the ladder program can then process any received message in the receive buffer.

Finally, the ladder program must "acknowledge" the completion of the transmission by clearing the SEND EXPLICIT MESSAGE COMMAND bit. The DNT250 will respond by clearing the EXPLICIT TRANSMISSION COMPLETE bit.

The HE693DNT250 is now ready for another Explicit Message transmission.

Below is a flow chart for the Send Explicit Message command.

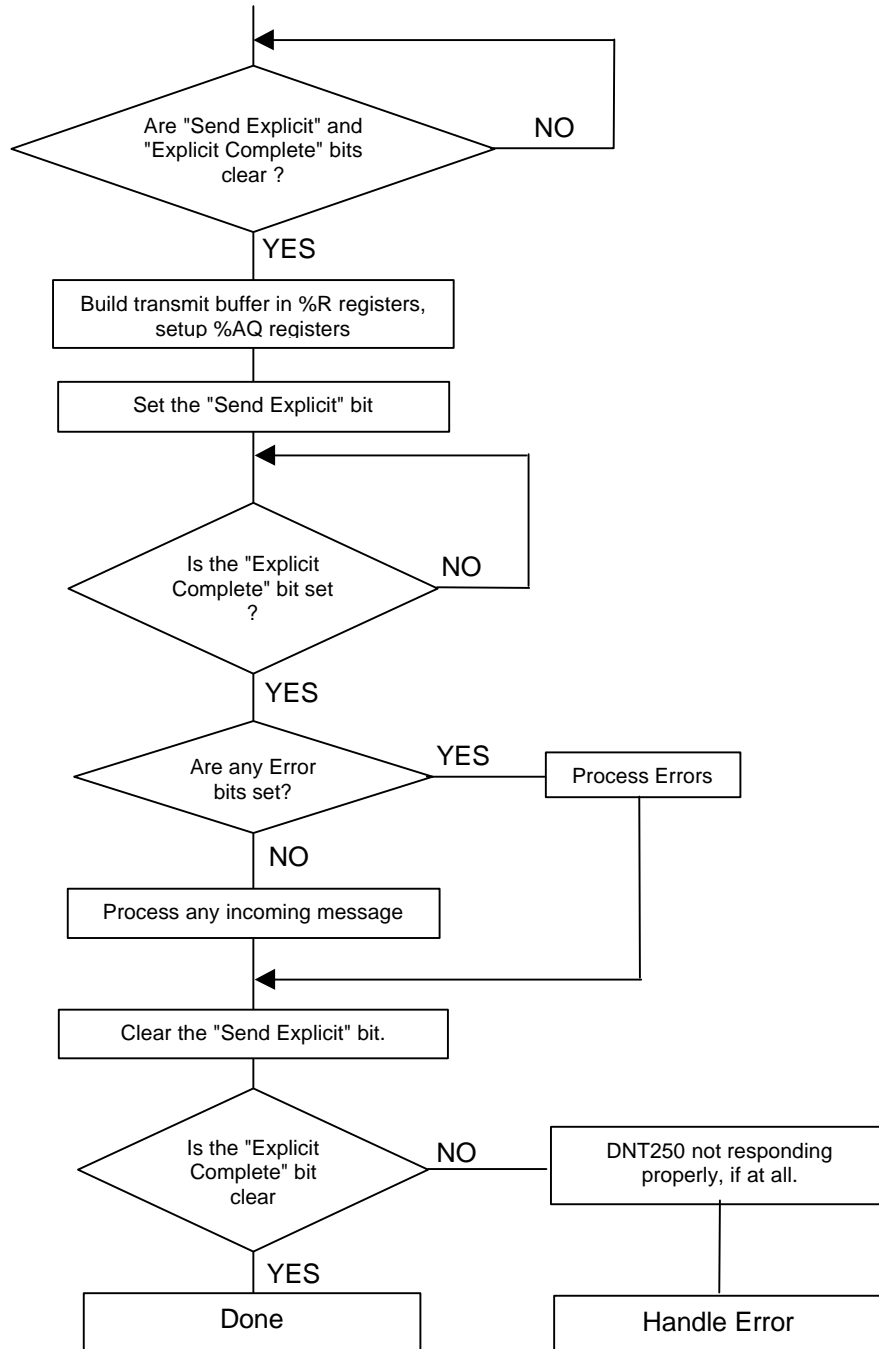


Figure 7.1 - Explicit Messaging Flow Chart

## APPENDIX A: DNT250 NETWORK OPERATION

### 1 Sequence of Events

The DNT250 follows a predetermined sequence of events as it attempts to communicate with the nodes connected to the network. Defined below is this sequence of events:

1. Starting at the top of the scan list (MACID 1), locate the first node that is configured.
2. If no nodes are configured, go to step 1.
3. If this node is On Line, go to step 7.
4. Attempt to establish a connection with the node. See Establishing a Connection below.
5. If a connection was successfully established, go to step 7.
6. If a connection was not successfully established, locate the next node that is configured and go to step 3.
7. Read the appropriate %Q and %AQ data from the PLC.
8. Build a Polled Command message sequence (fragmentation is required if the data size is greater than 8 bytes).
9. Sent the message(s) over the network to the node.
10. Setup a timeout timer.
11. Wait for the Polled response from the node.
12. If the timeout expires, take the node offline, locate the next node that is configured and go to step 3.
13. Send the response data to the appropriate PLC %I and %AI registers.
14. Locate the next node that is configured and go to step 3.

### 2 Establishing a Connection

There are several steps involved in the establishment of connections with DeviceNet nodes. The general steps involved include:

1. Attempt to establish an Explicit connection using the UCMM protocol.
2. If unsuccessful, attempt to establish an explicit connection using the Group 2 Only protocol.
3. If unsuccessful, give up on this node, return
4. Attempt to establish a Polled connection.
5. If unsuccessful, release the Explicit connection, give up on this node, and return.
6. Request the node's Polled Production size.
7. If unsuccessful, release the Explicit and Polled connections, give up on this node, and return.
8. Compare the nodes Polled Production size to the configuration file.
9. If different, release the Explicit and Polled connections, give up on this node, and return.
10. Request the node's Polled Consumption size.
11. If unsuccessful, release the Explicit and Polled connections, give up on this node, and return.
12. Compare the nodes Polled Consumption size to the configuration file.
13. If different, release the Explicit and Polled connections, give up on this node, and return.
14. Set the nodes Polled Expected Packet Rate.
15. If unsuccessful, release the Explicit and Polled connections, give up on this node, and return.
16. Set On Line status for the node and return.

In an effort to minimize adverse effects on nodes that are On Line, the DNT250 will only attempt to establish a connection with one Off Line node per complete pass through the Scan List.

Also, every few seconds the DNT250 will send an Explicit message containing a NOP to each On Line node. The purpose of this is to keep the nodes Explicit connection from timing out. Again, the DNT250 will only send one NOP message to a single node per complete pass through the Scan List.

If configured for LIEM (explicit messaging), the DNT250 will at the top of the scan list check to see if the ladder code is requesting that an explicit message be sent. If so, the explicit request is processed. This can occur only once per Polled scan.



## APPENDIX B: PACKET TIMING

### 1 Message Packet Timing

It is not easy to determine the exact transmission time of the DeviceNet Polled scan time. The table below provides actual timing of single message packets for various baud rates and message sizes. It must be understood that the table does not take into account the dead time between the message packets. There are a number of reasons for this dead time, but be assured that it does exist. The best way to get an accurate handle on the DeviceNet Polled scan time is to connect a good digital oscilloscope or logic analyzer to the DeviceNet cable while the system is running.

The amount of data to be transmitted determines the "packet size" and number of packets necessary. The baud rate determines the amount of time necessary to transmit one packet.

MESSAGE PACKET TRANSMISSION TIMES										
Number of Data Bytes in Message	0	1	2	3	4	5	6	7	8	
Number of BITS in message packet	47	55	63	71	79	87	95	103	111	
Transmission time @ 125K	376	440	504	568	632	696	760	824	888	μS
Transmission time @ 250K	188	220	252	284	316	348	380	412	444	μS
Transmission time @ 500K	94	110	126	142	158	174	190	206	222	μS

Given a message of 'X' bytes, it is first necessary to determine if fragmentation is involved, and if so how many packets will be sent:

If the size of the message ('X') is eight (8) bytes or less, use the time given in Table 8.

If the size ('X') is more than eight bytes, fragmentation is required:

Number of Full Packets = Message Size / 7

Size of additional packet = (remainder of (message size / 7) ) + 1

Time for full packets = Time for 8 bytes, taken from table.

Time for additional packet = Taken from table

Time for full message = Total for all Full Packets + Time for Additional Packet.

For example, suppose that a DeviceNet node requires a Poll Command of 27 bytes:

### 2 Network Scan Time versus PLC Scan Time

The user should not make any assumptions about the relationship between the PLC scan time and the DeviceNet Polled scan time. They are totally asynchronous to each other.

Any relationship between the PLC scan time and the DeviceNet Polled scan time can be determined only with a strong knowledge of the program running in the PLC and the DeviceNet configuration and operation or by conducting extensive testing.

NOTES

## APPENDIX C: DEVICENET GENERAL ERROR CODES

The following DeviceNet General Error Codes are defined within the DeviceNet specification and are supported by all DeviceNet nodes.

Table 1 – General Error Codes		
Error Code (in hex)	Name	Description of error
00 – 01	Reserved	
02	Resource Unavailable	Resource needed for the object to perform the requested service were not available
03 – 07	Reserved	
08	Service Not Supported	The requested service was not implemented or not defined for this Object Class/Instance
09	Invalid Attribute Value	Invalid attribute data detected
0A	Reserved	
0B	Already in requested mode	The object is already in the requested mode or state requested by the service
0C	Object State Conflict	The object can not perform the requested service in it's current mode or state
0D	Reserved	
0E	Attribute Not Settable	A request to modify a non-modifiable attribute was received
0F	Privilege Violation	Permission/privilege check failed
10	Device State Conflict	The device's current mode or state prohibits the requested service
11	Reply Data Too Large	The data to be transmitted is large than the allocated response buffer
12	Reserved	
13	Not Enough Data	The service did not supply enough data to perform the requested service
14	Attribute Not Supported	the attribute specified in the request is not supported
15	Too Much Data	The serve supplied more data than was expected
16	Object Does not Exist	The specified object does not exist in the device
17	Reserved	
18	No stored attribute data	The attribute data of the object was not stored prior to the requested service
19	store operation failure	The attribute data of this object was not saved by the object
1A – 1E		Reserved by DeviceNet
1F	Vendor Specific error	A vendor-defined error has occurred. See the Additional Byte for further information
20	invalid parameter	A parameter associated with the service was determined to be not valid
21 – CF	Reserved	
D0 – FF	Object Class and Service Errors	Vendor-specific Object and Class errors.

NOTES