

# **Process Manager Implementation Guidelines**

**PM12-500**

---



**Implementation  
Process Manager - 1**

***Process Manager  
Implementation Guidelines***

**PM12-500  
Release 500  
8/95**

---

# Copyright, Trademarks, and Notices

Printed in U.S.A. — © Copyright 1995 by Honeywell Inc.

Revision 01 – August 1, 1995

While this information is presented in good faith and believed to be accurate, Honeywell disclaims the implied warranties of merchantability and fitness for a particular purpose and makes no express warranties except as may be stated in its written agreement with and for its customer.

In no event is Honeywell liable to anyone for any indirect, special or consequential damages. The information and specifications in this document are subject to change without notice.

---

---

## About This Publication

This publication summarizes the Process Manager (PM) implementation process, guides you to procedures and references you need to implement PMs, describes PM operations considerations in implementing and using redundant NIMs, PMMs, and IOPs, and defines the UCN Node points and Node Specific points you must build for each PM. A similar publication contains comparable information for the Advanced Process Manager and can be found in the *Implementation/Advanced Process Manager -1* binder for that product.

This publication supports TDC 3000<sup>X</sup> software release 500.

Change bars are used to indicate paragraphs, tables, or illustrations containing changes that have been made to this manual effective with release 500. Pages revised only to correct minor typographical errors contain no change bars.



---

# Table of Contents

---

<b>1</b>	<b>INTRODUCTION</b>
1.1	Summary of Process Manager Implementation Tasks
1.2	References
1.2.1	References for Engineering Activities
1.2.2	References for Hardware Implementation
1.3	PM Implementation Dependencies
<b>2</b>	<b>PM OPERATIONAL CONSIDERATIONS</b>
2.1	The Process Manager Subsystem
2.2	PM Subsystem Redundancy
2.2.1	Redundant NIMs
2.2.2	Redundant PMMS
2.2.3	Redundant IOPs
2.3	Saving and Restoring of PM Information
2.3.1	Maintenance of Consistent NIM and PM Databases
2.3.2	IOP Database Security
<b>3</b>	<b>BUILDING POINTS</b>
3.1	Building UCN and Node-Specific Points
3.2	Data Point Building
3.3	Reconfiguration
3.3.1	To Add An IOP
3.3.2	To Delete An IOP
3.3.3	Changing the Control Point Mix or Scan Rate
<b>4</b>	<b>NIM LOADING AND PERFORMANCE</b>
4.1	Estimating NIM Loading
4.2	Assessment of NIM Loading
4.3	Use of a "Remote" NIM to Share Processing Load
4.3.1	Implementation of Two Logical Process Networks
4.3.2	Operational Considerations for Two Logical Process Networks
4.3.3	Functional Relationships on Two Logical Process Networks
<b>5</b>	<b>SERIAL DEVICE INTERFACE IMPLEMENTATION</b>
5.1	General Implementation Information
5.2	Manual/Auto Station Implementation
5.2.1	Input Functions from the Manual/Auto Station
5.2.2	Output Values and Control Functions Sent to the Manual/Auto Station
5.2.3	Additional Logic Implementation
5.2.4	Parameter Listings
5.3	Weigh Scale Implementation
5.3.1	Methods of Control
5.3.2	Control Schematic Example

---

# Table of Contents

---

<b>6</b>	<b>PEER-TO-PEER COMMUNICATION AND PERFORMANCE</b>
6.1	System Constraints on Peer-to-Peer Communication
6.1.1	Configuration Limits
6.1.2	Parameter Throughput Limit
6.1.3	Transaction Throughput Limit
6.2	Local and Remote Data
6.2.1	Transaction Load from Peer Traffic
6.2.2	Push/Pull Examples
6.3	Transaction Load from NIM Traffic
6.3.1	Typical NIM, Transaction Load = 8
6.3.2	Large NIM, Transaction Load = 21
6.4	Estimating Transaction Throughput
6.5	Designing To Handle Heavy Loads
6.6	Helpful Hints
6.6.1	Adding Connections to an Existing Configuration
6.6.2	Adding Another PM to an Existing UCN Configuration
6.6.3	Where Overruns Can Occur
6.6.4	Effect of Alarm Inhibit
6.6.5	Using Mostly One Transport Mechanism and a Little Bit of Another
6.6.6	Nonredundant Versus Redundant PMs
6.7	Peer-To-Peer Performance Monitoring
6.7.1	Peer-To-Peer Statistical Parameters
6.8	Troubleshooting PM Communications Overload
6.8.1	Troubleshooting Guidelines

## INDEX

## INTRODUCTION

### Section 1

*This introduction summarizes the Process Manager implementation tasks, lists publications that you will reference to implement Process Managers, and describes implementation dependencies.*

#### 1.1 SUMMARY OF PROCESS MANAGER IMPLEMENTATION TASKS

While most of the information in this publication relates to Process Manager (PM) functions, PM data points, and PM operating considerations, other implementation activities must also be completed to make the PM functional. The Engineering Personality activities listed below may be affected by the implementation of a PM or must be used to implement a PM. Note, however, that most of these activities are necessary to configure an LCN system even when a PM is not present.

See subsection 1.2 for references to instructions for each of these activities.

Activities named in THIS TYPEFACE are activated by targets on the Engineering Personality Main Menu.

- UNIT NAMES—The process units defined for each PM data point are established in this activity.
- AREA NAMES—The area name and descriptor for any units with PM points that are assigned to an area are established in this activity.
- LCN NODES—Identifies and defines the nodes on the Local Control Network. In the case of Network Interface Modules, this activity defines the node numbers for the NIMs and the process network number for the Universal Control Network.
- VOLUME CONFIGURATION—The NIM checkpoint volume, &8np, and the CL/PM sequences and LM ladder logic volume, &9np, are established in this activity. Volume &8np must have adequate storage space to accommodate the PM checkpoint data plus space to accommodate all other devices on all of the Universal Control Networks (UCNs) in this system. Volume &9np must have adequate space to accommodate all LM ladder logic programs and all CL/PM sequences.
- APPLICATION MODULE—Any AM points that are members of a control strategy that includes PM points are built in this activity. Connections to the PM points are defined in `tagname.parameter` form.
- NETWORK INTERFACE MODULE—The UCN point which defines where the UCN resides, and the node-specific points that define the nodes on that UCN, including the NIM and the PM, are built in this activity. Also, the PM data points are built in this activity.
- PICTURE EDITOR, FREE FORMAT LOGS, BUTTON CONFIGURATION—Any of pictures, logs, and buttons built by these activities can access PM points, once the points are built and loaded.
- HM HISTORY GROUPS—PM data point values for which continuous history is to be collected are defined in this activity, by assigning them to specific HM history groups.

- AREA DATA BASE—This activity defines how and where data for data points, including PM data points, are used and displayed in a given process area. The area database is the database loaded into a Universal Station, so that the database defines the process area monitored and controlled through the US.
- Control Language (CL)—User-written CL/PM programs are entered and compiled in this activity.

## 1.2 REFERENCES

### 1.2.1 References for Engineering Activities

- UNIT NAMES and AREA NAMES
  - Network Form Instructions* in the *Implementation/Startup and Reconfiguration - 1* binder
  - Network Data Entry* in the *Implementation/Startup and Reconfiguration - 1* binder
- VOLUME CONFIGURATION—Section 7 of the *Engineer's Reference Manual* in the *Implementation/Startup & Reconfiguration - 2* binder
- APPLICATION MODULE
  - Application Module Control Functions* in the *Implementation/AM - 1* binder
  - Application Module Algorithm Engineering Data* in the *Implementation/AM - 1* binder
  - Application Module Parameter Reference Dictionary* in the *Implementation/AM - 2* binder
  - Data Entity Builder Manual* in the *Implementation/Engineering Operations - 1* binder
- NETWORK INTERFACE MODULE
  - Process Manager Control Functions and Algorithms* in the *Implementation/Process Manager - 1* binder
  - Process Manager Parameter Reference Dictionary* in the *Implementation/Process Manager - 2* binder
  - Data Entity Builder Manual* in the *Implementation/Engineering Operations - 1* binder
- PICTURE EDITOR, FREE FORMAT LOGS, BUTTON CONFIGURATION
  - Instructions for these activities are in the *Implementation/Engineering Operations - 1* and *2* binders.

- HM HISTORY GROUPS

*HM History Group Form Instructions in the Implementation/Engineering Operations - 1 binder*

*Data Entity Builder Manual in the Implementation/Engineering Operations - 1 binder*

- AREA DATA BASE

*Area Form Instructions in the Implementation/Engineering Operations - 1 binder*

*Data Entity Builder Manual in the Implementation/Engineering Operations - 1 binder*

- Control Language (CL)

*CL/AM publications in the Implementation/Application Module - 2 binder*

*CL/PM publications in the Implementation/Process Manager - 2 binder*

*Data Entity Builder Manual in the Implementation/Engineering Operations - 1 binder*

## 1.2.2 References for Hardware Implementation

### 1.2.2.1 Site Planning

*LCN Planning manual in the System Site Planning binder*

*UCN Planning in the System Site Planning binder*

*UCN Installation in the Universal Control Network binder*

*Process Manager/Advanced Process Manager Planning in the System Site Planning binder*

### 1.2.2.2 Installation and Checkout

*LCN System Installation manual in the LCN Installation binder*

*LCN System Checkout manual in the LCN Installation binder*

*Process Manager/Advanced Process Manager Installation in the Implementation/PM/APM - 1 binder*

*Process Manager/Advanced Process Manager Checkout in the Implementation/PM/APM - 1 binder*

### 1.2.2.3 Service

*Five/Ten-Slot Module Service in the LCN Service - 2 binder (Network Interface Module)*

*Process Manager/Advanced Process Manager Service in the PM/APM/HPM Service - 1 binder*

### 1.3 PM IMPLEMENTATION DEPENDENCIES

Figure 1-1 shows which Process Manager implementation tasks depend on information entered in other tasks. This figure does not necessarily dictate the order in which the tasks must be completed, but it does show all of dependencies that must be satisfied before the PM can be fully operational.

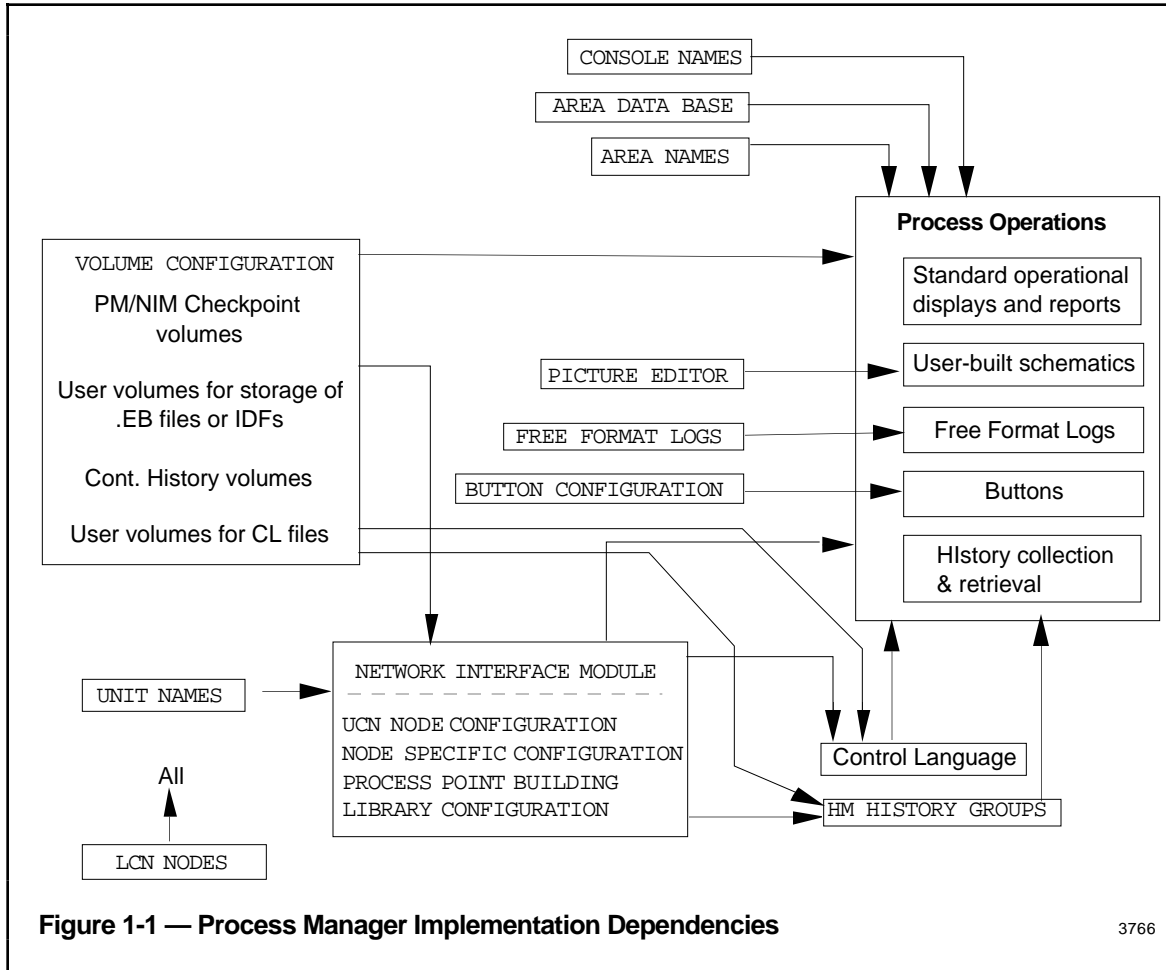


Figure 1-1 — Process Manager Implementation Dependencies

3766

## PM OPERATIONAL CONSIDERATIONS Section 2

*This section describes operational characteristics of the Process Manager that you should consider as you implement your PM(s).*

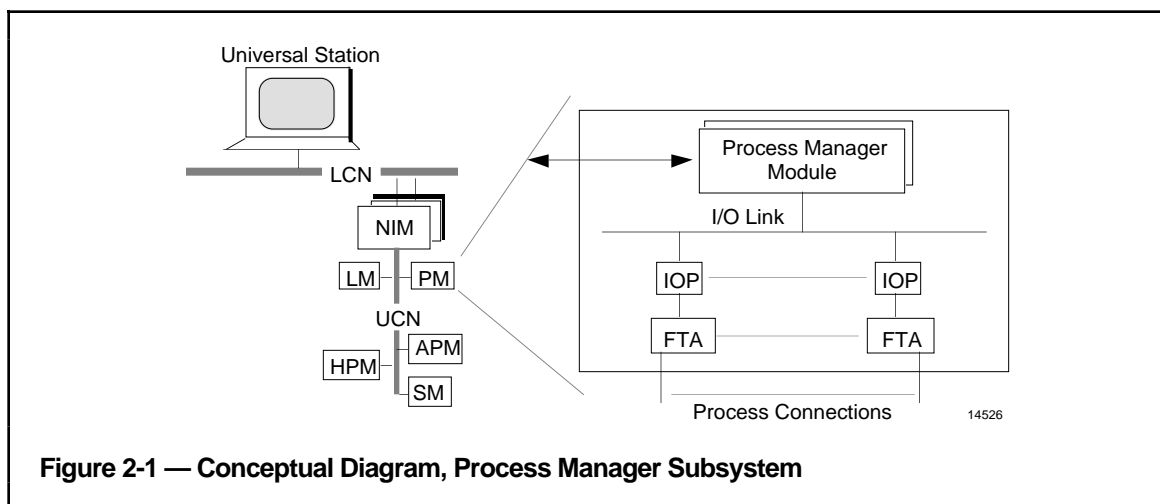
### 2.1 THE PROCESS MANAGER SUBSYSTEM

The Process Manager performs data acquisition and control functions, including regulatory, logic, and sequential control. The PM offers a powerful complement of prebuilt data acquisition and control algorithms, and it executes user-prepared sequence programs written in the PM version of Control Language, CL/PM.

Figure 2-1 is a conceptual diagram of a Process Manager subsystem.

The transition from the LCN to the Universal Control Network (UCN) is made by a Network Interface Module (NIM). Usually, the NIM consists of a pair of redundant NIMs.

The Process Manager consists of a Process Manager Module (PMM), I/O Processors (IOPs), and Field Termination Assemblies (FTAs) to which the process field wiring is connected. For additional reliability, PMMs can be configured as redundant pairs, as can High Level Analog Input and Output IOPs, Smart Transmitter Interface IOPs, and Digital Input/Output IOPs.



**Figure 2-1 — Conceptual Diagram, Process Manager Subsystem**

### 2.2 PM SUBSYSTEM REDUNDANCY

#### 2.2.1 Redundant NIMs

NIMs are intended to operate as redundant pairs, where one NIM is the primary and on-line, while its partner backs up the primary and maintains a copy of the primary's database. Should the primary NIM fail or be taken out-of-service, its partner takes over, becoming the new primary.

### 2.2.1.1 LCN and UCN Addresses for NIMs

LCN addresses for NIMs are established through the LCN NODES configuration activity on the Engineering Personality Main Menu and through address pinning on the LCN interface in the node electronics. To facilitate operation and the handling of unusual situations, such as unexpected fail overs or the separation of LCN segments, we recommend that you assign sequential addresses (odd/even) to each NIM pair.

UCN addresses for NIMs are established through the UCN NODE CONFIGURATION and NODE SPECIFIC CONFIGURATION activities on the NIM Build Type Select Menu, and through address pinning on the UCN interface in the node electronics. NIMs should be assigned to the lowest UCN addresses, starting with address one. Each NIM pair uses one odd and one even address. We recommend that for possible expansion, UCN addresses one through six be reserved for NIMs, even if though fewer than three NIM pairs are actually in use.

## 2.2.2 Redundant PMMS

PMMS can be configured as redundant pairs. One of the PMMs operates as the primary and its partner operates as its backup. Should the primary PMM fail or be taken out-of-service, its partner takes over, becoming the new primary.

### 2.2.2.1 UCN Addresses for PMMs

Each of the partner PMMs uses one UCN address (odd) and its partner (if present) uses the next address (even). We recommend that for possible expansion, UCN addresses one through six be reserved for NIMs, even though fewer than three NIM pairs are actually in use; therefore, the first PMM could start at UCN address seven and its partner PMM would use address eight.

If the UCN has a mix of PMs and other UCN products, the first PMM in each PM uses an odd UCN address and its partner PMM (if there is one) uses the next even address. Each of the other redundant UCN products, also use consecutive (odd/even) UCN addresses and each nonredundant UCN product uses one UCN address.

### 2.2.3 Redundant IOPs

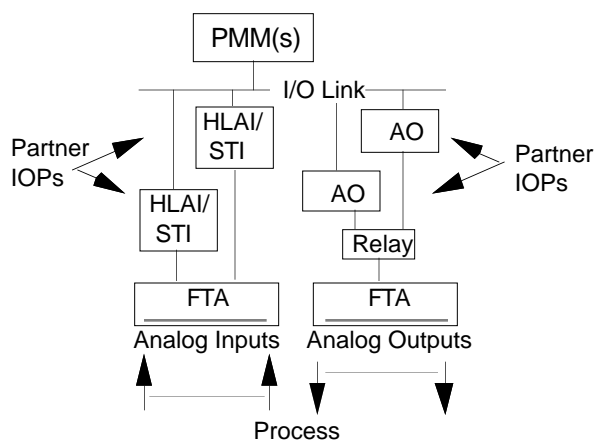
The following IOPs are available as redundant partners.

- Analog Output (AO)
- High Level Analog Input (HLAI)
- Smart Transmitter Interface (STI)
- Digital Input (DI)
- Digital Output (DO)

Either one of the partner IOPs operates as the primary and the other backs up the primary. Each of the partners connects to the process through a single Field Termination Assembly (FTA).

Both the primary and backup IOPs receive all data from the UCN or from the process simultaneously and, should the primary IOP fail, the backup IOP takes over automatically, becoming the new primary IOP. Such a failover is transparent to the remainder of the system, except for Universal Station displays that show IOP status information. A failover is completed in 100 milliseconds or less.

At the PM Detail Status display, you can request that the primary and backup roles for the partner IOPs be exchanged. To do this, select the backup IOP, then select RUN STATES, SWAP PRIMARY, and press ENTER.



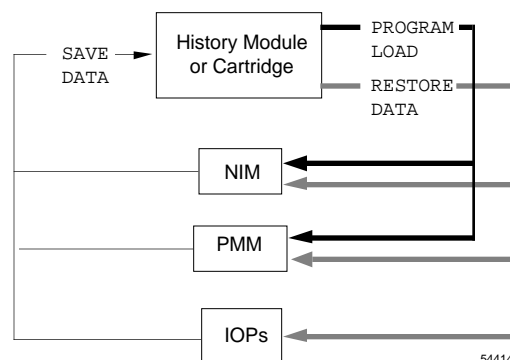
3771

Synchronization of the database in the partner IOPs is verified as the backup IOP checks that the the primary IOP received and responded to each data item, and by periodic comparisons of the databases in both IOPs.

In the sketch above, both HLA/STI IOPs and both STI IOPs monitor the inputs from the FTA. For AO IOPs, a relay is used to direct the analog output from the primary IOP to the FTA. A diagnostic routine verifies that the interconnections between the partner IOPs are functional and that the analog-output relay makes any changeover within a predetermined time interval.

## 2.3 SAVING AND RESTORING OF PM INFORMATION

Three of the five LOAD, SAVE, and RESTORE targets at the bottom of the UCN Status display save and restore NIM and Process Manager information as indicated in this diagram.



The functions of each of the three commands is as follows:

- **PROGRAM LOAD**—Loads the NIM and PMM software personality image from the &Inp volume on an HM, or from a cartridge or floppy, to the NIM and the selected PMM(s) in the selected Process Manager(s).
- **RESTORE DATA**—Restores point data stored in the &8np checkpoint volume on an HM, or from a cartridge or floppy, to the NIM and the PMM(s) in the selected Process Manager(s). For more information on checkpointing, refer to Section 21 of the *Engineer's Reference Manual in the Implementation/Startup and Reconfiguration - 2* binder.
- **SAVE DATA**—Saves point data in the NIM and the PMM(s) in the selected Process Manager(s) into the &8np checkpoint volume on an HM, or onto a cartridge or floppy. This target requests a “demand” checkpoint. Automatic checkpointing may also save this data at the established automatic checkpoint interval for this system. For more information on checkpointing, refer to Section 21 of the *Engineer's Reference Manual in the Implementation/Startup & Reconfiguration - 2* binder.

### 2.3.1 Maintenance of Consistent NIM and PM Databases

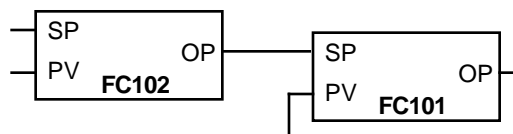
It is possible to create inconsistent NIM and PM databases through misuse of the point loading, checkpointing, and deleting functions. These inconsistencies can cause valid control connections between points to appear to operators as invalid.

Here are three scenarios that can cause inconsistent, mismatched data in the NIM and PM:

- (1) Save NIM and PM checkpoint data. (2) Use the Data Entity Builder to build and load points in the PM. (3) Before the new PM database is saved in the checkpoint files, shut down and reload the NIM, using the unmodified checkpoint as the data source (the PM data is not restored, so it still has data for the new points, but the NIM does not).

- (1) Use the Data Entity Builder to build and load points in the PM. (2) Save NIM and PM checkpoint data. (3) Use the DEB to delete a point (this removes it from the PM and NIM databases). (4) Restore the PM database, using the checkpoint data which still has the point that was deleted (the NIM database was not restored, so the PM has data for the point that was deleted but the NIM does not).
- (1) With the NIM's LOADSCOP parameter containing NIMAndPM, use the DEB to build and load PM points. (2) Change the NIM's UCN Node Configuration entity so that LOADSCOP contains NIMOnly, then delete a point (the delete affects the NIM only; the PM still has data for the point).

As an example of the confusion that can be caused by NIM/PM database mismatches, consider the PM points in this sketch. FC102 has a control-output connection to FC101. Any of the three scenarios described above could cause the NIM to lose the tag name for FC102. An operator at a Group or Detail display is not able to see any information about FC102. Even more confusing, FC101's setpoint (SP) may be changing and its output (OP) following, even though there is no apparent input. This is because the cascade is operating properly in the PM, but due to the mismatch in NIM and PM databases, the operator can't see FC102.



3773

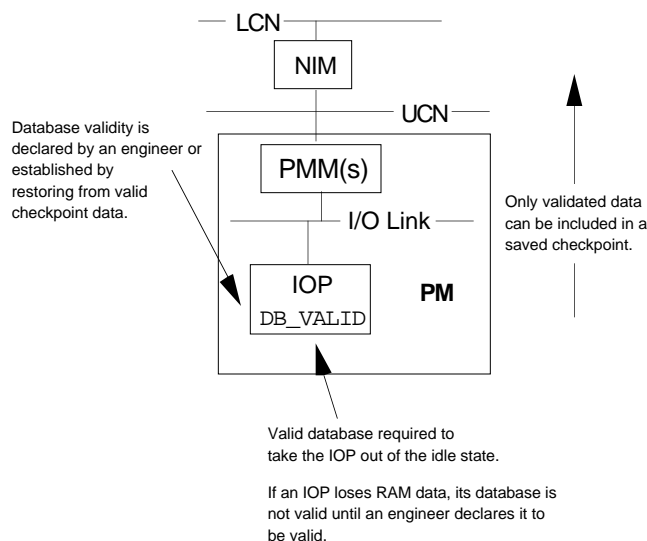
To avoid database mismatches, follow these recommendations:

- When you use the Data Entity Builder to load and to delete points, do so while UCN Node Configuration parameter LOADSCOP in the NIM contains NIMAndPM. If you need to change it to NIMOnly, be sure to change it back as soon as possible.
- Keep only one version of each of the checkpoints for each NIM/PM combination. If you have to have differing version, be aware that the NIM and PM checkpoints in different version may not match.
- Immediately after you load or delete points in a PM, use the SAVE DATA target on the PM Status display or PM Detail Status display to update the checkpoint.

## 2.3.2 IOP Database Security

An IOP cannot be changed from its idle (IDL) state to an operating state unless it has a valid database (normally, a nonidle IOP has OK status). Until the IOP is switched out of the idle state, it does not process data, and if it has an output value, that value does not change. An IOP accepts parameter value changes from Universal Stations and user-written programs, only when it is not idle. Therefore, the process is protected from incorrect output changes that might occur if the IOP were started without a validated database.

In an IOP that has lost its random-access memory (RAM) data, and no point data has yet been loaded into it, all parameter values in the IOP are the default values.



3774

These are two ways to validate an IOP's database:

- By using the `RESTORE MODULE` or `RESTORE ALL` Target on the PM Detail status display to restore a valid checkpoint in the IOP.
- By placing the key switch in the ENG position, selecting an IOP, and on the PM Detail status display, select `RUN STATES`, `VALIDATE IOP DB`, and `ENTER`.

After either of these two conditions is met, when the `STARTUP` target or the `VALIDATE IOP DB` target is selected on the PM Detail status display, IOPs with valid databases are shown as `DB_VALID`.

Therefore, an engineer can declare any IOP database as valid, but would do so only if he or she is sure that required point data has been loaded into the IOP from the DEB or by restoring the database from a valid checkpoint.

Data from an IOP with an invalid database cannot be included in a checkpoint, therefore, only valid data is included in the checkpoint data for a PM.

If power is removed from an IOP and restored shortly thereafter, the IOP may retain the database in its RAM, and in such a case, the database remains valid and restoration of data from a checkpoint or validating by an engineer is not necessary.

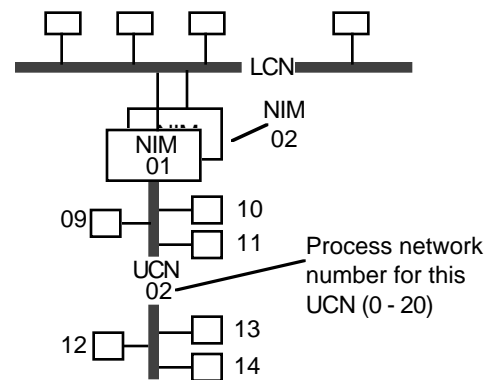
## BUILDING POINTS Section 3

This section defines the UCN node entities (points), node-specific entities (points), and process data points that must be built and loaded into the PMs.

### 3.1 BUILDING UCN AND NODE-SPECIFIC POINTS

A UCN point must be built for each node on a UCN, including each NIM, each PMM, and any redundant partners. Also, you must build a node-specific point for each PMM and each other node on the UCN. The UCN and LCN points are reserved entities. These entities must be built and loaded before data points can be loaded into the nodes on the UCN.

These entities are built with the Data Entity Builder. After you select NETWORK INTERFACE MODULE on the Engineering Main Menu, select the UCN NODE CONFIGURATION and NODE SPECIFIC CONFIGURATION picks to access the Parameter Entry Displays (PEDs) used to build them. For information about the values to be entered, refer to the *Process Manager Parameter Reference Dictionary* in the *Implementation/Process Manager - 2* binder.



3775

For the UCN example in sketch above, the following reserved entities would be built:

<u>UCN Node</u>	<u>UCN Point</u>	<u>Node-Specific Point</u>
NIM, UCN node no. 1	\$NM02N01	N/A
NIM, UCN node no. 2	\$NM02N02	N/A
PMM (or other), node no. 9	\$NM02N09	\$NM02B09
.	.	.
.	.	.
.	.	.
Node no. 14	\$NM02N14	\$NM02B14

In the example above, NIM 01 and NIM 02 are partners in a redundant node pair. The PMMs can also be paired as redundant partners (for more information, see Section 2). A UCN point and a node-specific point must be built for each PMM, including all redundant partners.

## 3.2 DATA POINT BUILDING

Data points are also built with the Data Entity Builder. After selecting NETWORK INTERFACE MODULE on the Engineering Main Menu, select PROCESS POINT BUILDING.

The resulting menu allows you to choose one of several types of points, for example Analog, Digital, Process Module, etc. Pulse Input points, and Smart Transmitter Interface points are subsets of Analog Input points.

If you choose Regulatory PV or Regulatory Control points, an additional menu on the second page of these displays allows you to choose further sub-types of those points such as Summer, Totalizer, PID, or Ramptoak points.

Process data point building consists of selecting options and entering parameters into the parameter entry screen displays and then loading the set of point specifications. Refer to the *PM Control Functions and Algorithms* manual for information about each point type. Refer to the *PM Parameter Reference Dictionary* for parameter details. The system startup guide contains brief examples of data point building.

Points can only be loaded when the designated slot is in the inactive state, or the IOP containing the designated slot is in the Idle state. Points are loaded by selecting appropriate targets on the Data Entity Builder's Command Menu or from the Parameter Entry Display by pressing a function key. The *Process Operations Manual* explains how to make points active or inactive. Deleting a point leaves that slot active with default parameters as a database. Individual slots can be set to inactive status through the Slot Summary display.

You can either store a copy of the parameter entry data to an Intermediate Data File (IDF) or to an Exception Build file. The IDF store is easier, but the Exception Build file is more adaptable for use in another system, or in later release software, and the parameters can be edited using the Text File Editor. Refer to the *Data Entity Builder Manual* for the procedures. Once the points are loaded and the database is set valid, the entire database can be checkpointed (saved) in a format that is easily reloaded in mass.

While building points, you may need to specify I/O connections to other points. The other points can be built as described above or they can be denoted as untagged (hardware reference) points. An untagged point is built by specifying a syntax that refers to the IOP type, IOP card number, slot number and parameter, all preceded by an exclamation point; for example, !AO12S03.OP. The complete format is explained in the *PM Control Functions and Algorithms* manual for applicable points. Untagged points are easy to specify and do not deduct from the maximum NIM point count. The disadvantage is that they are relatively invisible in the system and operators or maintenance technicians may not be aware of their presence. Therefore, you should keep careful records of the IOPs and slots where untagged points are implemented or use the Find Names utility to list them. Like tagged points, the slot must be inactive or the IOP idled to load them. You can check the slot status on the Slot Summary display.

### 3.3 RECONFIGURATION

There are several occasions when you may wish to reconfigure the PM database:

- To add an IOP
- To delete an IOP
- To change from one IOP type to another
- To add control points
- To delete control points
- To change the scan rate

If you wish to replace an existing IOP with the same type (for example, a later revision), no reconfiguration is necessary. The IOP may be replaced with power on and the process running.

Note that changing from one IOP type to another amounts to deleting one IOP and adding another. The various procedures are discussed in the following paragraphs.

Before starting you should —

- call up the UCN Status Display and fix any Soft Fail or Part Fail errors
- checkpoint the PM (save the data) to a cartridge and to the HM so that you can return to the original configuration, if necessary.

You will find it helpful to have the *Command Processor Operation* manual (for Find Names functions) and the *Data Entity Builder Manual* (to reconstitute/load points/delete points) available. Section 7 in the *Process Manager Service* manual provides information on inserting and removing the physical IOP board.

### 3.3.1 To Add An IOP

The IOP card can be inserted with power on and the process running.

Then, reconstitute the Node Specific configuration file, NMnnBxx (where nn = the Network number and xx = the node number on the UCN). The procedure follows.

To reconstitute the Node Specific configuration file—

- Select NETWORK INTERFACE MODULE from the Engineering Menu
- Select NODE SPECIFIC CONFIGURATION from the NIM BUILD TYPE Menu
- Enter the desired Network Number and Node Number into the NODE SPECIFIC CONFIGURATION display. If the word UNENTERED appears at the top, press ENTER until it disappears (repeat if necessary).
- Hold the CTRL key down and press 7.

The Parameter Entry Display should now contain a copy of the values that were loaded into the system. The following steps change the values and reload the PM.

- Page forward to the IO MODULE CONFIGURATION section and make the necessary changes (select IOP types and enter file/card numbers).
- To load the new configuration, hold the CTRL key down and press ~ (F12)
- Hold the CTRL key down and press 9 (F9) to return to the NIM BUILD menu.

#### NOTE

If you added an SOE card, you must also reconstitute the NIM entity under UCN NODE CONFIGURATION and set the TIMESYNC parameter to ENABLED. Then, reload the NIM entity as above.

- Select the Process Point Building target for the Point Build Menu. Choose a point type for the new IOP and configure points as needed.
- You can load each point by holding the CTRL key down and pressing ~ (F12) or you can write each point to an IDF and then execute a LOAD MULTIPLE.
- Manually checkpoint to the HM (twice) and to the backup media (twice).

### 3.3.2 To Delete An IOP

For the IOP being removed—

- Make sure that no points exist for those slots. If you do not have adequate records, you can reconstitute each slot for that IOP and make a list. Note that this will not locate hardware reference points (such as !DO03S05.SO). If you think hardware reference points were built for this IOP, try to find them using the Find Names function (refer to the Command Processor manual).
- Use Find Names to determine where each point is used. Delete those point references in CL programs and schematics. They will have to be recompiled. Point references in the Area Data Bases must be deleted. You must reinstall the Area Data Base, but if new points will be added to either the area or schematics or CL programs, wait until that is done.
- Set each point in the IOP module to INACTIVE or
- Set the IOP module to IDLE.
- Delete all points in the IOP.
- Reconstitute the Node Specific configuration file, NMnnBxx (see 3.3.1 for a procedure) and change the IOP module to NONE. Or, if it is being replaced by another IOP, this is the time to configure the new IOP type.
- To load the new configuration, hold the CTRL key down and press ~ (F12).
- Hold the CTRL key down and press 9 (F9) to return to the NIM BUILD menu.

#### NOTE

If you deleted an SOE card, you must also reconstitute the NIM entity under UCN NODE CONFIGURATION and set the TIMESYNC parameter to DISABLED. Then, reload the NIM entity as above.

- Manually checkpoint to the HM (twice) and backup media (twice).

The IOP card can now be removed with power on and the process running.

### 3.3.3 Changing the Control Point Mix or Scan Rate

The point mix and the scan rate are configured as a part of the PM Box Data Point. This is viewed at the Universal Station, on the Node Specific Configuration display. These are the choices for Regulatory PV and Regulatory Control points, and the flag, logic, numeric, digital composite points, etc., preceding the SCANRATE parameter on the configuration display. SCANRATE refers to the Regulatory/Logic point scan cycle (REG1LOG1, etc.).

#### To change the point mix and/or the scan rate—

- Set the PM to IDLE. If you can access the UCN Status display from another US, this step can wait until you are ready to load the new configuration.
- If you are deleting control points, use the Find Names function to determine where those points are used (refer to the *Command Processor* manual). Delete the point references in CL programs and schematics.
- Create a Selection List of all PM control points (but not IOP resident points). A selection list is just a list of points in a file. For the procedure, refer to Section 7 in the *Data Entity Builder Manual*. The UCN Status Display's Point Summary List, your IDF files, and good records can help you to build the Selection List.
- Using the Selection List created above, either Reconstitute Multiple to an IDF file, or Print System Entities to an exception build file. Section 7 in the *Data Entity Builder Manual* describes these procedures. In brief, select the Command Menu. Then\*—

#### To Reconstitute to an IDF file—

Select Reconstitute Multiple  
 Enter a Reference Pathname (for example NET>n timer>)  
 Enter a pathname for the IDF file (for example CTLPNTS.DB)  
 Enter the pathname name for your Selection List (for example SLIST.EL)  
 Press Enter. Wait for the operation to complete.

#### To Print System Entities to an Exception Build file—

Select Print Entities  
 Select Print System Entities  
 Enter a Reference Pathname (for example NET>n timer>)  
 Enter the pathname name for your Selection List (for example SLIST.EL)  
 Enter a Destination pathname for the exception build file (for example EXFILE.EB)  
 Press Enter. Wait for the operation to complete.

The IDF file (.DB ) or the Exception Build file (.EB) will be used later to reload the control points.

---

\*For the examples shown, when you only specify a file name, the Reference Path Name is prefixed to form a complete pathname. If any files are on removable media, specify the full pathname.

- Reconstitute the Node Specific Configuration file (see 3.3.1 for a procedure) and enter the necessary changes.
- To load the new configuration, hold the CTRL key down and press F12 (the PM must be in IDLE).

**NOTE**

If you increase the Scan Rate, more Processing Units are required (Refer to Section 3 in the *PM Control Functions and Algorithms* manual). If you exceed the Processing Units limit, the configuration changes are not loaded.

To reload the control points from your IDF or Exception Build file, go to the Command Menu. Then—

- To reload from an IDF file—
  - Select Load Multiple
  - Enter a Reference Pathname
  - Select With Overwrite
  - Enter a pathname name for the IDF file (for example CTLPNTS.DB)
  - Press Enter. Wait for the operation to complete.
- To reload from an Exception Build file—
  - Select Exception Build
  - Enter a reference pathname
  - Select With Overwrite
  - Select Load Entities
  - Enter a pathname for the exception build Source file (for example EXFILE.EB)
  - Enter a pathname for the IDF file for example BADPNTS.DB). This is for points that don't load (if any), but a file name is required.
  - Press Enter. Wait for the operation to complete.
- Restart the PM.
- Manually checkpoint to the HM (twice) and to the backup media (twice).

At this point, you will probably want to go build control points as allowed by the new configuration.



## NIM LOADING AND PERFORMANCE Section 4

*This section describes methods for estimating Network Interface Module loading and assessing the results of your estimates. Also described is the implementation of a second NIM (or NIM pair) on a UCN, to share the processing load.*

### 4.1 ESTIMATING NIM LOADING

Tables 4-1 and 4-2 provide examples of NIM loading estimates for the NIM in a performance cluster (for more information about a performance cluster, refer to the *Engineer's Reference Manual*). The cluster contains one PM, three Universal Stations, Application Module, and one History Module. The HM contains PM checkpoints and continuous history.

The Universal Stations are used to view standard displays and two custom schematics. Each custom schematic contains 250 parameters, 50 of which are on fast update. Refer to Table 4-3 for added schematic Load Factors. Refer to the Set Collection command in the *Picture Editor Reference Manual* for more information on fast update.

Use Table 4-1 for systems with a 68020 NIM and Table 4-2 for systems with a 68040 NIM. 68020 and 68040 refer to the NIMs microprocessor type. If the NIM contains a K4LCN board, it is a 68040 NIM. If not, it is a 68020 NIM.

A NIM load estimate is calculated by multiplying the value you entered in the Number column by the factor in the Load Factor column, entering the results in the Induced Load column, and adding the values in the Induced Load column. Comments following the tables explain more about the considerations. In the first example, the total induced load is 625, which is 62.5% of the maximum load allowed for a NIM.

You should make a loading estimate for each NIM in your system. If you have several NIMs, consider using a spread sheet on a personal computer to do your calculations. The following comments along with Tables 4-1 and 4-2 help to further explain the process:

#### **General**—

- Indicate the number of active LCN nodes, that is, those that are principally accessing information from the UCN resident devices. For the History Module, this means continuous history. Then apply the load factors as explained below.
- **Redundant Nodes**—Count redundant node pairs (NIMs, AMs, APMMs, HPMMs, PMMs, LMs, and SMs) as one node.

Table 4-1 — 68020 NIM Loading Estimator

Load Sources	Units to Be Entered in Number Column	Number	Load Factor	Induced Load
<b>US Induced Load</b>				
Universal and U <sup>X</sup> S Stations	Number principally accessing this NIM	3	15	45
Fast Schematic Displays	Number principally accessing this NIM	2	85	170
<b>UCN Induced Loads</b>				
PMs, APMs, HPMs, LMs and/or SMs on the UCN	Number of PMs, APMs, HPMs, LMs, and SMs on the UCN	10	10	100
<b>History Module Induced Load</b>				
Continuous History:				
68020 HM (2400 Points per minute)	Number of HMs principally accessing this NIM	1	30	30
68040 HM (3000 Points per minute)	Number of HMs principally accessing this NIM	0	40	
Checkpoints	Number of HMs checkpointing this NIM	1	70	70
<b>AM and A<sup>X</sup>Ms Induced Loads (90 Points/second)</b>				
AMs with 68020 microprocessor	Number principally accessing this NIM	1	150	150
AMs with 68040 microprocessor	Number principally accessing this NIM	0	200	
<b>CG Induced Loads</b>				
Computer Gateways (100 parameters/second)	Number principally accessing this NIM	1	60	60
<b>Total Induced Load:</b>				<b>625</b>
<b>Maximum Allowable Load:</b>				<b>1000</b>
<b>% of Maximum Allowable Load:</b>				<b>62.5</b>

Table 4-2 — 68040 NIM Loading Estimator

Load Sources	Units to Be Entered in Number Column	Number	Load Factor	Induced Load
<b>US Induced Load</b>				
Universal and U <sup>X</sup> S Stations	Number principally accessing this NIM	3	6	18
Fast Schematic Displays	Number principally accessing this NIM	2	35	70
<b>UCN Induced Loads</b>				
PMs, APMs, HPMs, LMs and/or SMs on the UCN	Number of PMs, APMs, HPMs, LMs, and SMs on the UCN	10	4	40
<b>History Module Induced Load</b>				
Continuous History:				
68020 HM (2400 Points per minute)	Number of HMs principally accessing this NIM	0	12	0
68040 HM (3000 Points per minute)	Number of HMs principally accessing this NIM	1	16	16
Checkpoints	Number of HMs checkpointing this NIM	1	28	28
<b>AM and A<sup>X</sup>Ms Induced Loads (90 Pts./second)</b>				
AMs with 68020 microprocessor	Number principally accessing this NIM	0	60	0
AMs with 68040 microprocessor	Number principally accessing this NIM	1	80	80
<b>CG Induced Loads</b>				
Computer Gateways (100 parameters/second)	Number principally accessing this NIM	1	24	24
<b>Total Induced Load:</b>				<b>276</b>
<b>Maximum Allowable Load:</b>				<b>1000</b>
<b>% of Maximum Allowable Load:</b>				<b>27.6</b>

**Schematics—**

- Assume that the schematics are principally accessing this NIM.
- For every Universal Station there is a base load as shown in Table 4-1 or 4-2. Add to this the load factor from Table 4-3 —

**Table 4-3— Schematic Load Factor**

Total Parameters in Schematic	Number of Parameters on Fast Update	Added Load Factor 68020	Added Load Factor 68040
100	0	0	0
100	50	65	26
150	0	5	2
150	50	70	28
200	0	15	6
200	50	75	30
250	0	20	12
250	50	85	34

**HMs—**

- The checkpoint load factor is in addition to the History Module factor. The 68020 HM Continuous History load factor is based on an HM with a 68020 processor collecting history for 2400 points per minute. The 68040 HM Continuous History load is based on an HM with a 68040 processor collecting history for 3000 points per minute.

**AMs and AXMs—**

- The AM load factor is based on a fully-loaded AM accessing data from this NIM. A fully loaded 68020 AM processes 90 points per second. A fully loaded 68040 AM processes 120 points per second.

**CGs—**

- The CG load assumes that it is requesting 100 parameters per second from the NIM.

**4.2 ASSESSMENT OF NIM LOADING**

The following are the NIM loading categories:

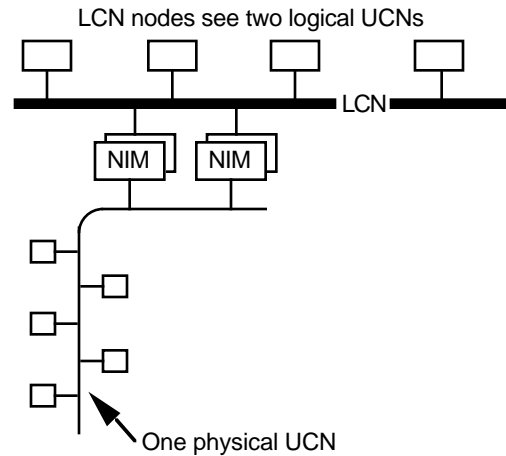
- A NIM whose total induced load is 750 (75%) or less can be expected to perform as specified under all actual system-use conditions.
- If the total induced load is between 750 and 1000 (75% to 100%), the NIM load is marginally acceptable, and display of information from this NIM and its reporting of events may occasionally be sluggish, especially during a process upset or a peak load such as multiple point loading.
- If the total induced load is above 1000 (100%), the NIM should be considered overloaded. If a failover to the backup NIM or some other system upset occurs, the view to the process may be temporarily lost.

### 4.3 USE OF A “REMOTE” NIM TO SHARE PROCESSING LOAD

An additional NIM (redundant NIM pair) can be added to the UCN and the LCN to share the processing load with another NIM. To use such a NIM you must adhere to certain rules for point assignments and operational practices.

#### 4.3.1 Implementation of Two Logical Process Networks

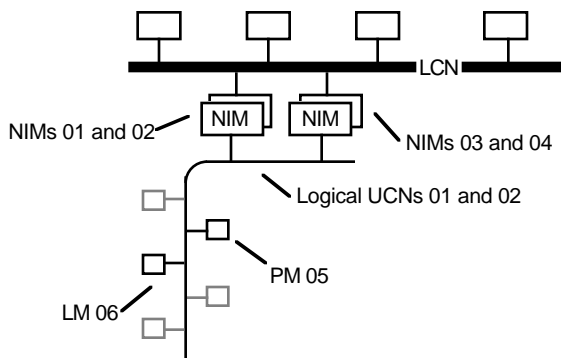
From the LCN viewpoint, the two NIMs (redundant NIM pairs) are on separate process networks, even though they are connected to the same physical UCN. The first NIM (configured as ThisNIM) is assigned to process network n (n is in a range from 1 to 20; each UCN and each Data Hiway is one process network) and the second NIM (configured as RemotNIM) is assigned to process network n+1. The assignment of the network numbers is arbitrary, but consistent, logical assignment simplifies operating practices.



3776

The NIMs to be configured as ThisNIM and RemotNIM and their process networks must be first defined in the NCF through the Engineering Personality's LCN NODES configuration activity. Then, all of the UCN nodes, including the NIMs, are defined on both process networks by building UCN entities (NIM points) and Node Specific entities (box points). In the UCN Node entities, about half of the nodes on each process network are configured with NODEASSN = ThisNIM and the remainder with NODEASSN = RemotNIM. Each node assigned as ThisNIM on process network n is assigned as RemotNIM on n+1, and each node assigned as ThisNIM on process network n+1 is assigned as RemoteNIM on network n.

For example, for the UCN node numbers in this sketch, you would build the following UCN node and node-specific entities:



3777

Node	UCN	UCN Entity Name	NODEASSN	Node Specific Ent. Name
NIM 01	01	\$NM01N01	ThisNIM	N/A
NIM 02	01	\$NM01N02	ThisNIM	N/A
NIM 03	01	\$NM01N03	RemotNIM	N/A
NIM 04	01	\$NM01N04	RemotNIM	N/A
PM 05	01	\$NM01N05	ThisNIM	\$NM01B05
LM 06	01	\$NM01N06	RemotNIM	\$NM01B06
NIM 01	02	\$NM02N01	RemotNIM	N/A
NIM 02	02	\$NM02N02	RemotNIM	N/A
NIM 03	02	\$NM02N03	ThisNIM	N/A
NIM 04	02	\$NM02N04	ThisNIM	N/A
PM 05	02	\$NM02N05	RemotNIM	\$NM02B05
LM 06	02	\$NM02N06	ThisNIM	\$NM02B06

As you build process points to reside on the two logical UCNs, assign approximately equal numbers of points to each UCN (parameter NTWKNUM), but take care to assign points that use peer-to-peer communication to the same UCN (peer-to-peer communication is through connections from points in one UCN node to points in other nodes on the same UCN).

### 4.3.2 Operational Considerations for Two Logical Process Networks

Use of the SAVE DATA target to checkpoint data from the UCN nodes and the restoration of checkpoint data to the nodes can be accomplished only from the UCN Status display for the process network the nodes are assigned to (NODEASSN = ThisNIM). If you try to do this from the wrong display, a “node assignment” error message appears. If some of the points in a UCN node are assigned to process network n and others are assigned to process network n+1, you will have to use SAVE DATA twice, once from each UCN Status display.

For automatic checkpointing to save all data, it must be enabled through the UCN Status displays for both process networks.

Alarming, message transfers, and event-initiated processing are handled by the NIMs and no special operational considerations are required.

### 4.3.3 Functional Relationships on Two Logical Process Networks

Successful implementation and use of two NIMs and logical process networks that share processing loads is more likely if you understand the relationships described here.

It is through UCN node configuration that the relationships of the two NIMs (or two pairs of redundant NIMs) and the two logical process networks are established. It is through node specific configuration that nodes on the UCN are assigned to one network or the other.

The relationships of PMs to the NIMs is defined in the UCN node point parameter NODEASSN, which contains either ThisNIM or RemotNIM. Two UCN node entities are configured for each UCN node, one on each process network. The PM's points are processed by the NIM on the process network for which the PM NODEASSN value is ThisNIM.

UCN nodes configured with NODEASSN = RemotNIM appear on the UCN Status display for the process network associated with the NIM that is not processing their points, even though they don't logically belong to that network. The boxes for UCN nodes that logically belong to a process network are green and the boxes for nodes that logically belong to the other process network are yellow.

Because process points are assigned to a process network as the points are built, a PM can contain points that belong to one network and other points that belong to the other network. A point's database resides partly in the PM and partly in the NIM that has the same process network assignment as the point.

Consider this checkpointing example for process networks 1 and 2: when the data for a PM is checkpointed, all point data for the PM is saved in the checkpoint directory for process network 1, but the NIM-resident data for the points assigned to network 2 is not saved until another checkpoint operation saves the data for network 2 in the directory for that network.



## SERIAL DEVICE INTERFACE IMPLEMENTATION

### Section 5

*This section describes methods to connect various devices to the Process Manager through a Serial Device Interface.*

#### 5.1 GENERAL IMPLEMENTATION INFORMATION

The Serial Device Interface (SDI) allows various devices to be easily connected to the PM. Field Termination Assemblies (FTA) connected between the device(s) and the SDI I/O Processor (IOP) contain firmware to map the device data so that it mimics data from a Smart Transmitter. Each type of device uses a corresponding type of FTA designed to process the I/O and return any status or control signals required by that type of device. The SDI IOP itself must be configured as a Smart Transmitter Interface module (STIM). Note that up to two SDI FTAs can be connected to each SDI IOP. The first FTA uses the first 8 STI points; the second uses STI points 9–16. Each FTA can support one device type.

This section discusses the unique implementation of each Serial Device Interface option. Section two of the *Process Manager Control Functions and Algorithms* manual describes the characteristics, constraints, and operating considerations for SDI options. It also provides charts showing how the critical parameters relate between the PM and the serial device.

#### 5.2 MANUAL/AUTO STATION IMPLEMENTATION

Physical installation of the Manual/Auto Station is described in the *Process Manager Installation* manual and in the *Manual/Auto Station Installation/Operation Manual* (see References).

Only the following configurations are supported per FTA to avoid a soft failure diagnostic timeout.

**Table 5-1 — Supported M/A Configurations**

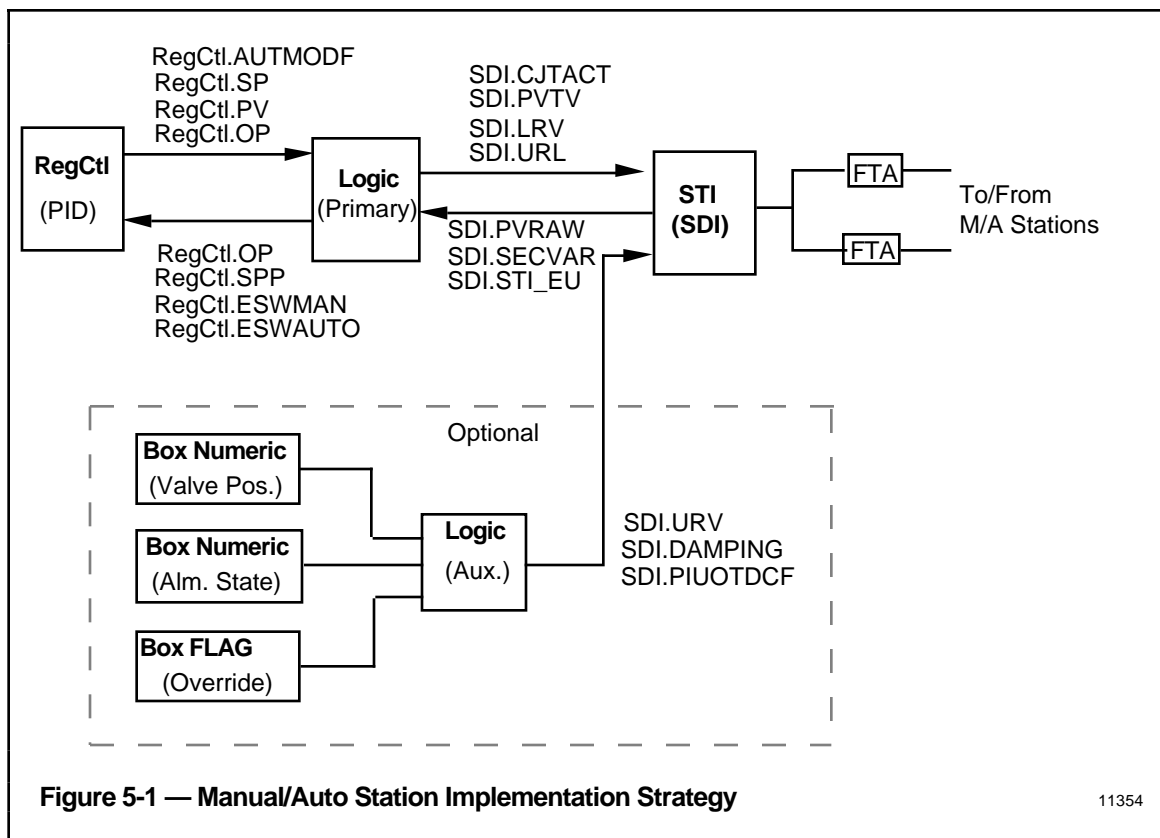
Full Configuration	Plus Basic Configuration	Number of Active M/A Stations Per FTA
4	0	4
3	1	4
2	2	4
1	3	4

PM implementation typically requires that you build a PID Regulatory Control point, configure the SDI IOP as an STI module (during Node Specific Configuration), and build at least one and preferably two logic points.

The second logic point is optional and is used to write the following information back to the Manual/Auto Station:

- valve position
- the alarm state
- the override flag

Figure 5-1 illustrates the signal flow necessary to fully implement each M/A Station. In this drawing and those that follow, RegCtl refers to the Regulatory control point; SDI refers to the SDI IOP configured as an STI point, and Logic refers to a logic point.



## 5.2.1 Input Functions from the Manual/Auto Station

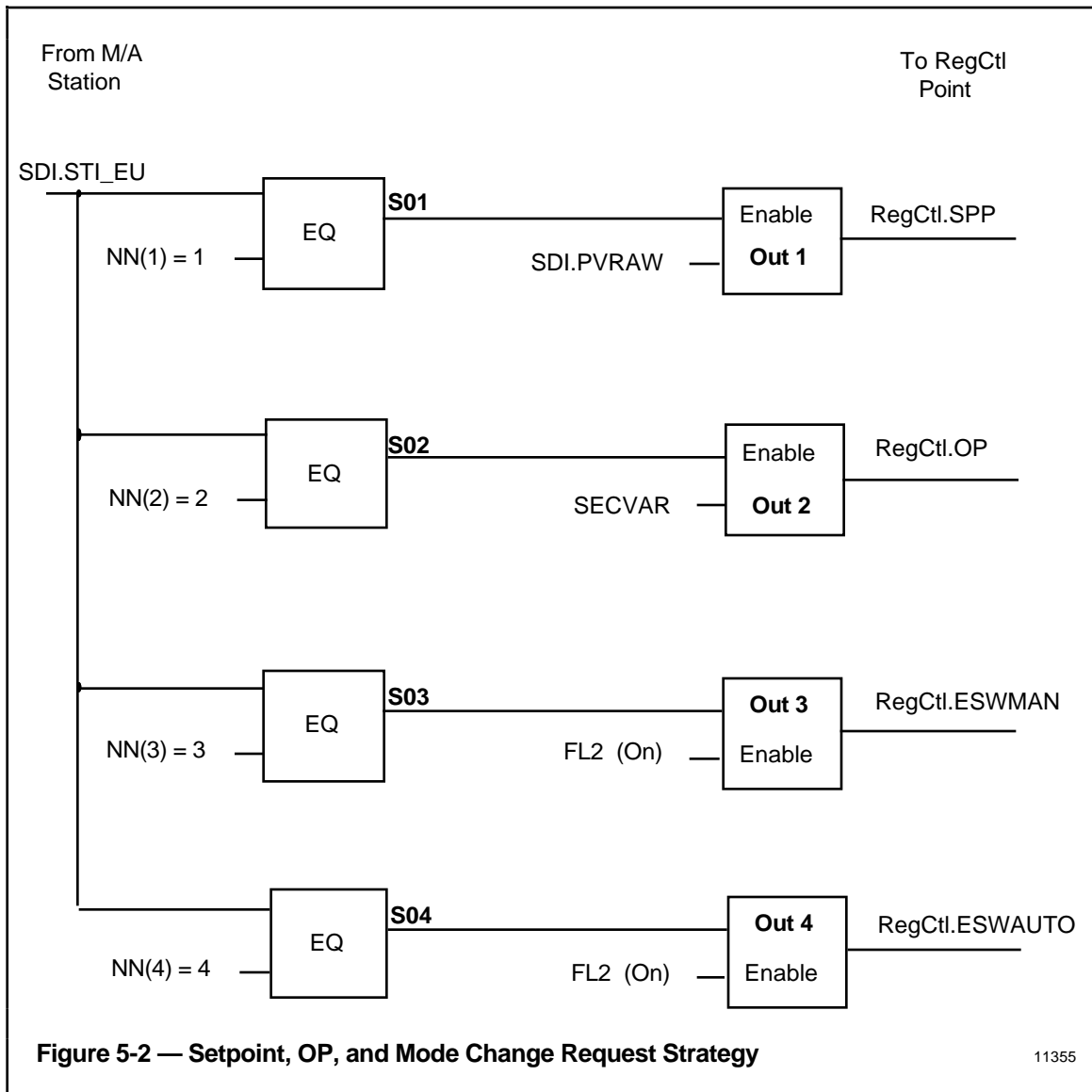
Signals sent from the Manual/Auto Station include mode change requests, also Setpoint and OP change requests along with the Setpoint and OP values. The configuration needed to handle these signals is discussed in the following paragraphs.

### 5.2.1.1 Setpoint, OP, and Mode Change Requests

The Manual/Auto Station request for an SP, OP, or mode change is determined by a change in the value of parameter STI\_EU. Figure 5-2 shows how the primary logic point in the PM is configured with a series of “Equals” blocks to detect values of 1, 2, 3, or 4. The value of STI\_EU is normally 0, but temporarily changes to 1 through 4 depending on the request (refer to the *Process Manager Control Functions and Algorithms* manual).

A Setpoint change request causes SO1 to enable the PVRAW value through logic block Output 1. PVRAW is the desired setpoint from the Manual/Auto Station. When the STI\_EU parameter equals two, an OP change request enables SECVAR through logic block Output 2. SECVAR contains the requested OP value from the Manual/Auto Station.

When there is a Manual/or auto mode change request from the Manual/Auto Station, STI\_EU enables SO3 or SO4, respectively, to turn on the appropriate external switching flag to the regulatory control point.

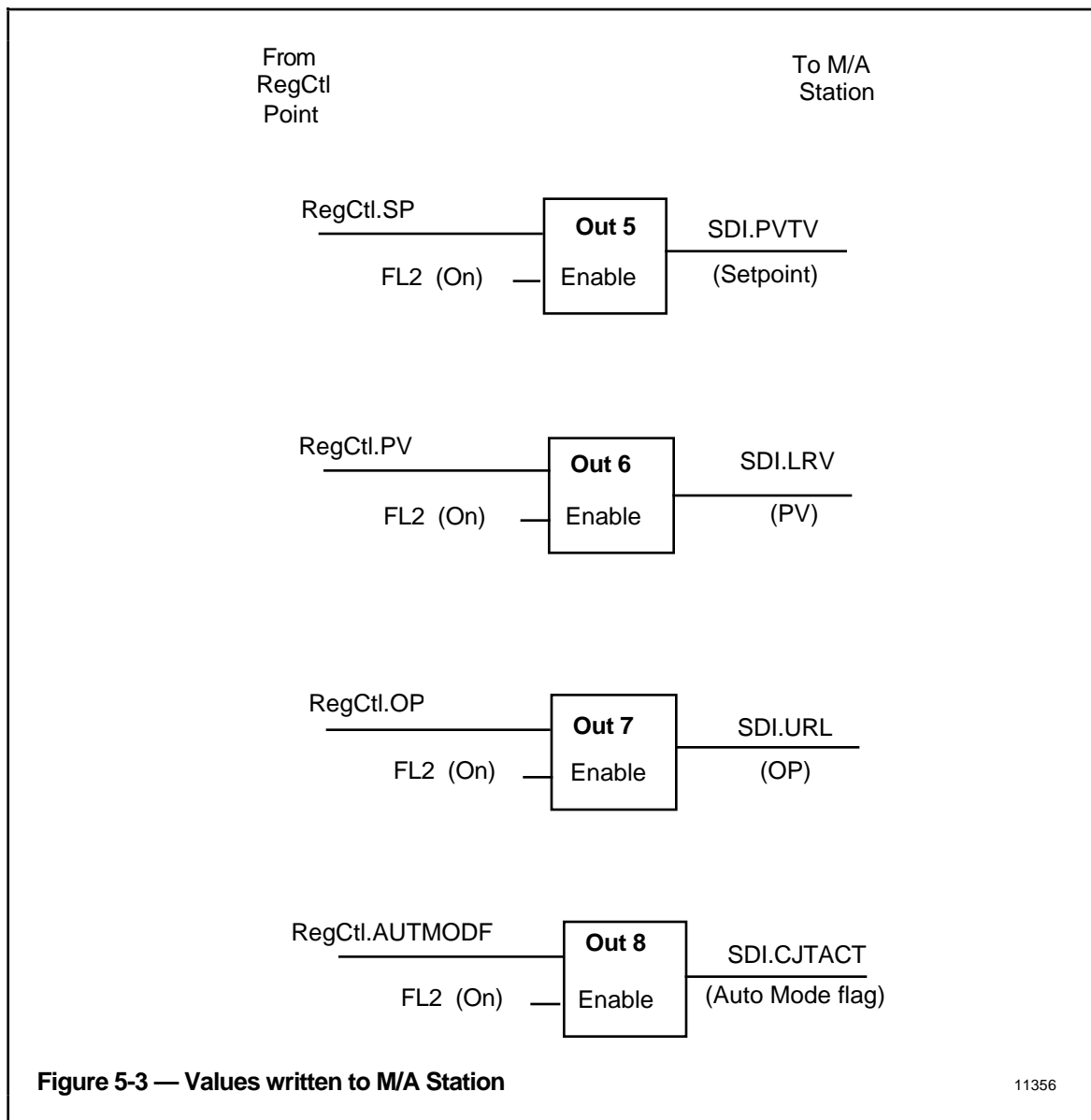


## 5.2.2 Output Values and Control Functions Sent to the Manual/Auto Station

Signals sent back to the Manual/Auto Station for display purposes include the mode, Setpoint, and OP. An auxiliary logic point can be used to send the Override Flag, alarms, and the valve position.

### 5.2.2.1 Fundamental Signals Sent to the Manual/Auto Station

Figure 5-3 illustrates how the Setpoint, PV, OP, and mode flag from the regulatory control point are written back to the Manual/Auto Station using logic gates.

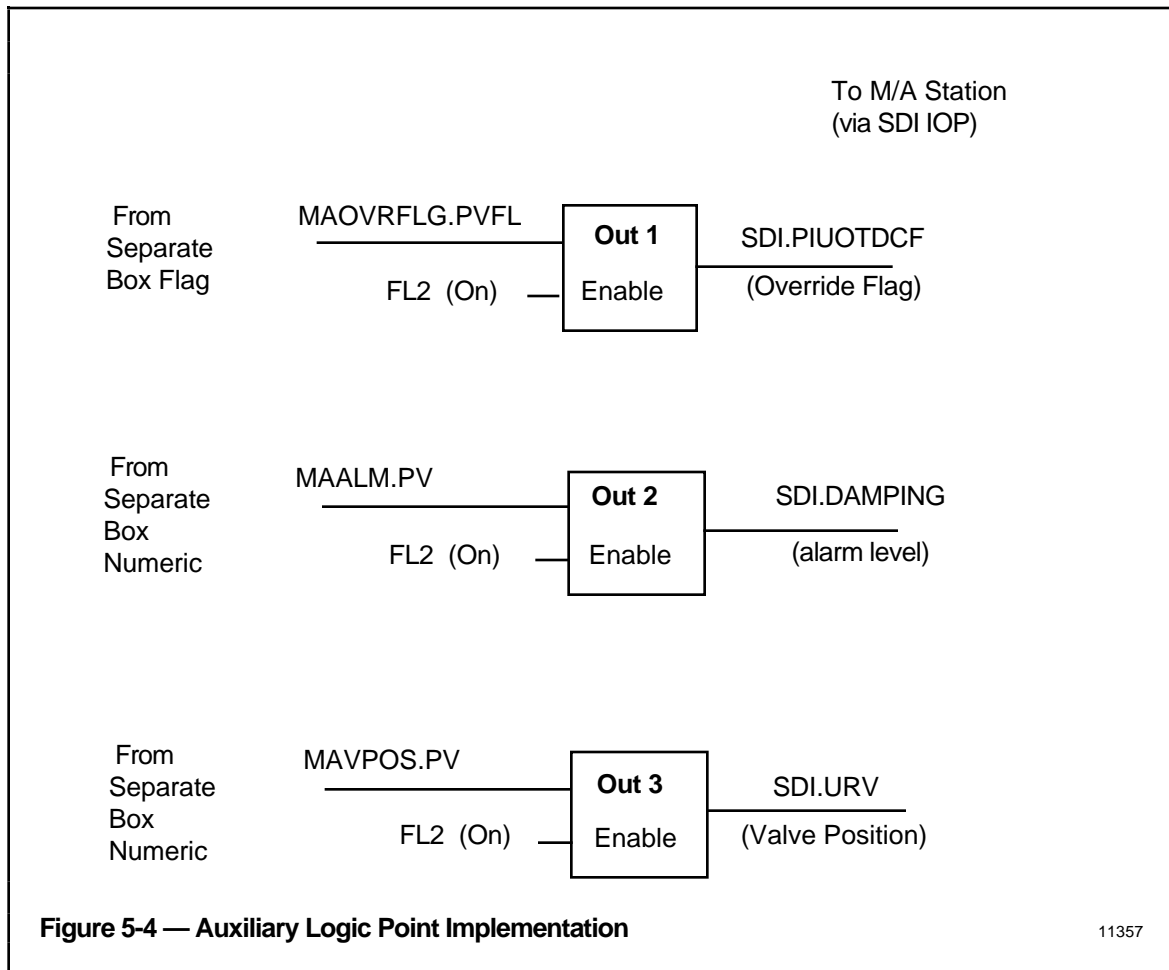


## 5.2.3 Additional Logic Implementation

The logic functions described in this section are optional but provide useful functions.

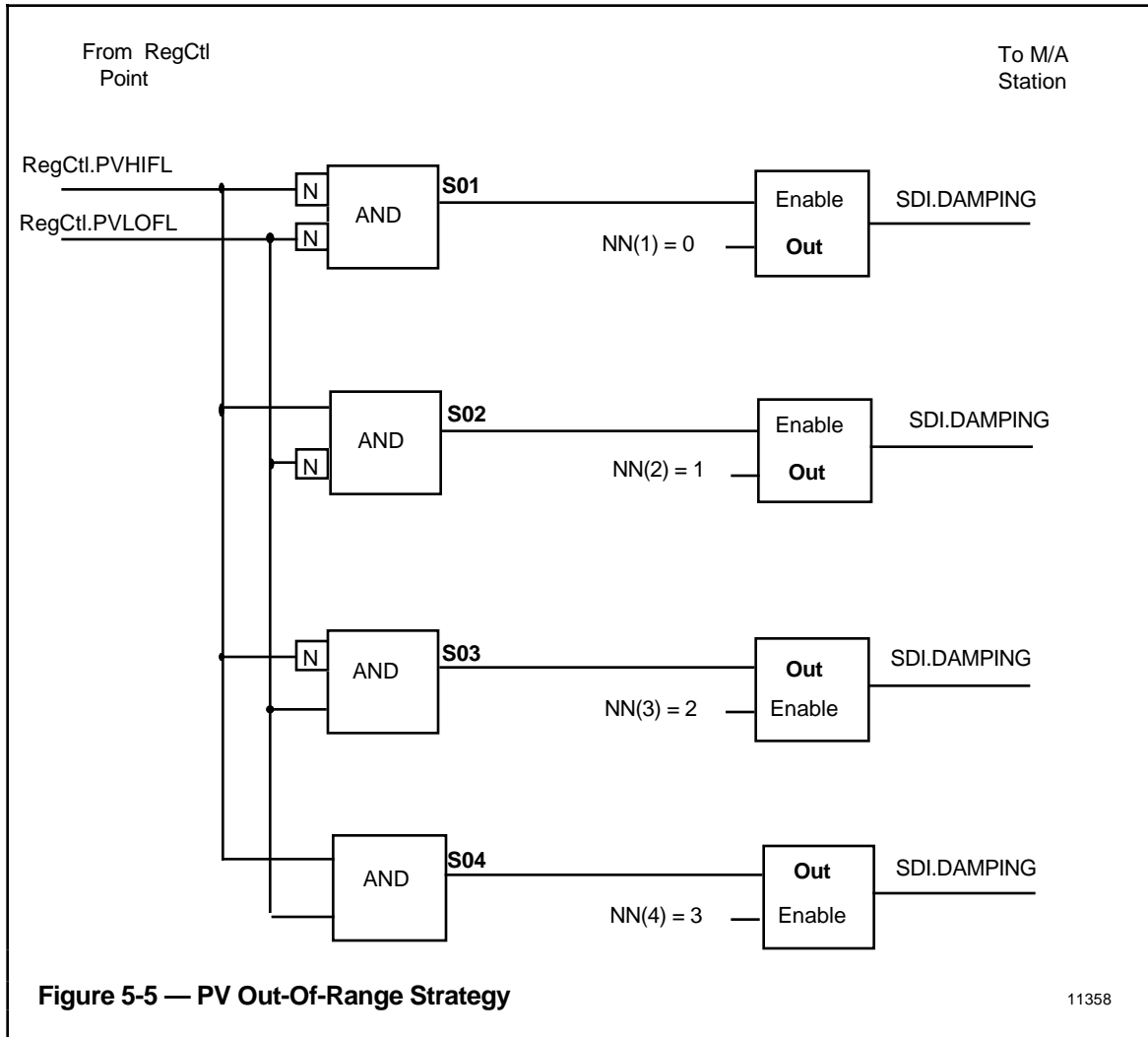
### 5.2.3.1 Auxiliary Logic Point Implementation

A second logic point, referred to as the Auxiliary Logic Point, is optional and can be used to feed back the override flag, an alarm status, and valve position to the Manual/Auto Station. Figure 5-4 illustrates how these signals can be returned to the Manual/Auto Station.



## 5.2.3.2 Alternative Alarming Strategy

Figure 5-5 illustrates an alarming scheme whereby the PV High and Low Alarm Flags can be used to feed back PV out-of-range alarms to the Manual/Auto Station. PVHIFL on alone sets alarm 1, PVLOFL on alone sets alarm 2, and alarm 3 is set when both are on (this is an alternative to using logic gate Out 2 as shown in Figure 5-4).



## 5.2.4 Parameter Listings

The following listings are intended to help you implement the strategies described for the Manual/Auto Station. Use your own point names, unit number, network, node and slot numbers. Use default values for parameters not included. The terminology SDI refers to the SDI IOP configured as a STI point and RegCtl refers to the associated PID Regulatory Control Point. Use your own point names in place of “SDI” and “RegCtl.”

**Table 5-2 — Primary PM Logic Point**

Parameter	Entry	Parameter	Entry
LOGMIX	12_8_8	LOSRC(6)	L5
LISRC(1)	SDI.PVRAW	LODSTN(6)	SDI.LRV
LISRC(2)	SDI.SECVAR	LOENBL(6)	FL2
LISRC(3)	SDI.STI_EU	LOSRC(7)	L6
LISRC(4)	RegCtl.SP	LODSTN(7)	SDI.URL
LISRC(5)	RegCtl.PV	LOENBL(7)	FL2
LISRC(6)	RegCtl.OP	LOSRC(8)	L7
LISRC(7)	RegCtl.AUTMODFL	LODSTN(8)	SDI.CJTACT
NN(1)	1.0	LOENBL(8)	FL2
NN(2)	2.0	LOGALGID(1)	EQ
NN(3)	3.0	R1(1)	L3
NN(4)	4.0	R2(1)	NN1
LOSRC(1)	L1	DEADBAND(1)	0.5
LODSTN(1)	RegCtl.SPP	LOGALGID(2)	EQ
LOENBL(1)	SO1	R1(2)	L3
LOSRC(2)	L2	R2(2)	NN2
LODSTN(2)	RegCtl.OP	DEADBAND(2)	0.5
LOENBL(2)	SO2	LOGALGID(3)	EQ
LOSRC(3)	SO3	R1(3)	L3
LODSTN(3)	RegCtl.ESWMAN	R2(3)	NN3
LOENBL(3)	FL2	DEADBAND(3)	0.5
LOSRC(4)	SO4	LOGALGID(4)	EQ
LODSTN(4)	RegCtl.ESWAUTO	R1(4)	L3
LOENBL(4)	FL2	R2(4)	NN4
LOSRC(5)	L4	DEADBAND(4)	0.5
LODSTN(5)	SDI.PVTV		
LOENBL(5)	FL2		

**Table 5-3 — Auxiliary PM Logic Point**

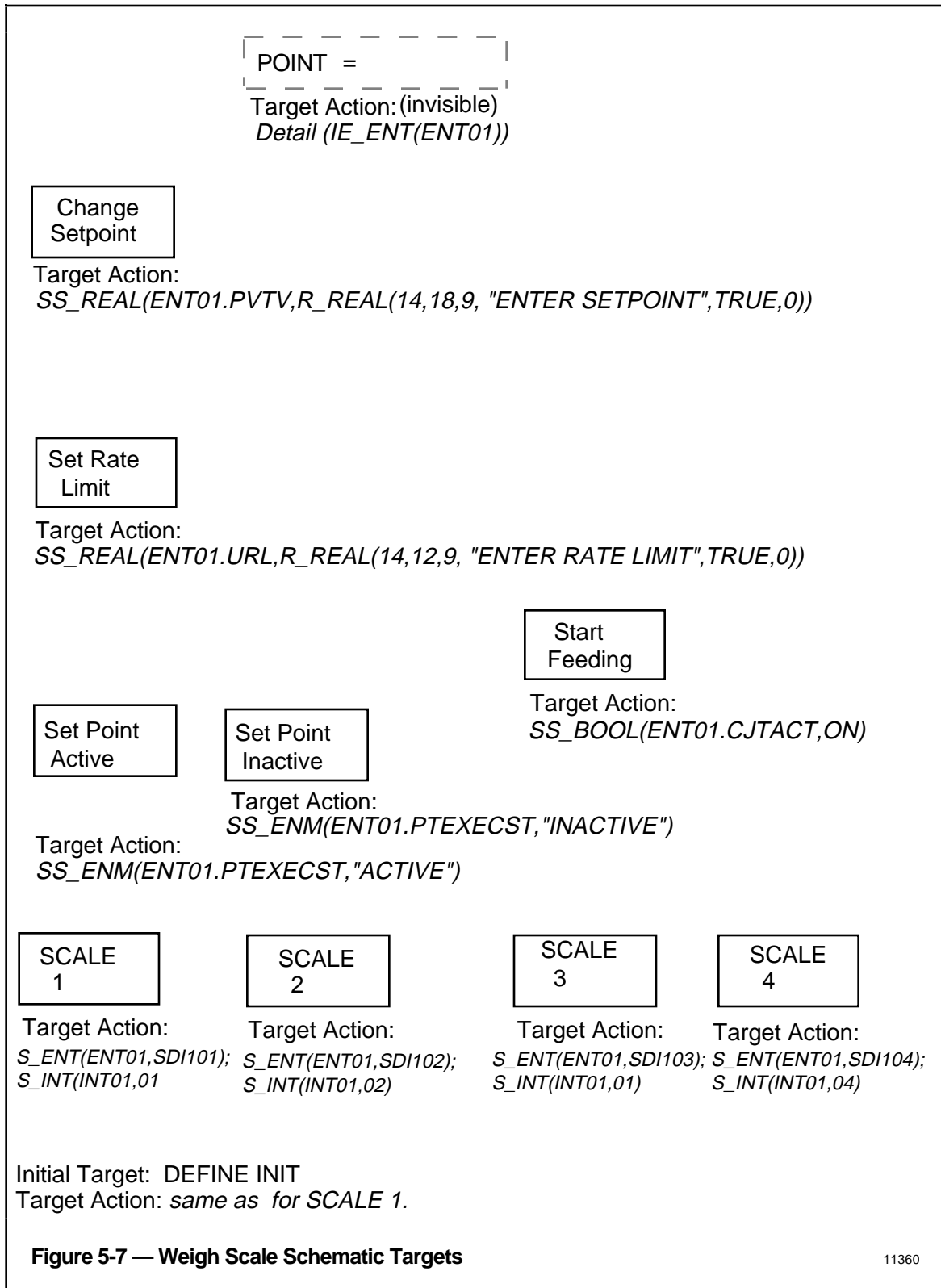
Parameter	Entry
LOGMIX	12_8_8
LISRC(1)	MAOVRFLG.PVFL
LISRC(2)	MAALM.PV
LISRC(3)	MAVPOS.PV
LOSRC(1)	L1
LODSTN(1)	SDI.PIUOTDCF
LOENBL(1)	FL2
LOSRC(2)	L2
LODSTN(2)	SDI.DAMPING
LOENBL(2)	FL2
LOSRC(3)	L3
LODSTN(3)	SDI.URV
LOENBL(3)	FL2

**Table 5-4 — PM STIM (SDI) Point**

Parameter	Entry
NODETYP	PM
PNTFORM	FULL
PNTMODTY	STIM
SENSRTYP	SPT_DP
PVCHAR	LINEAR
STI_EU	INH2O
DECONF	PV_SV_DB
DAMPING	0.0
INPDIR	DIRECT
PVEUHI	100.0
PVEULO	0.0
PVEXEUHI	100.0
PVEXEULO(	0.0

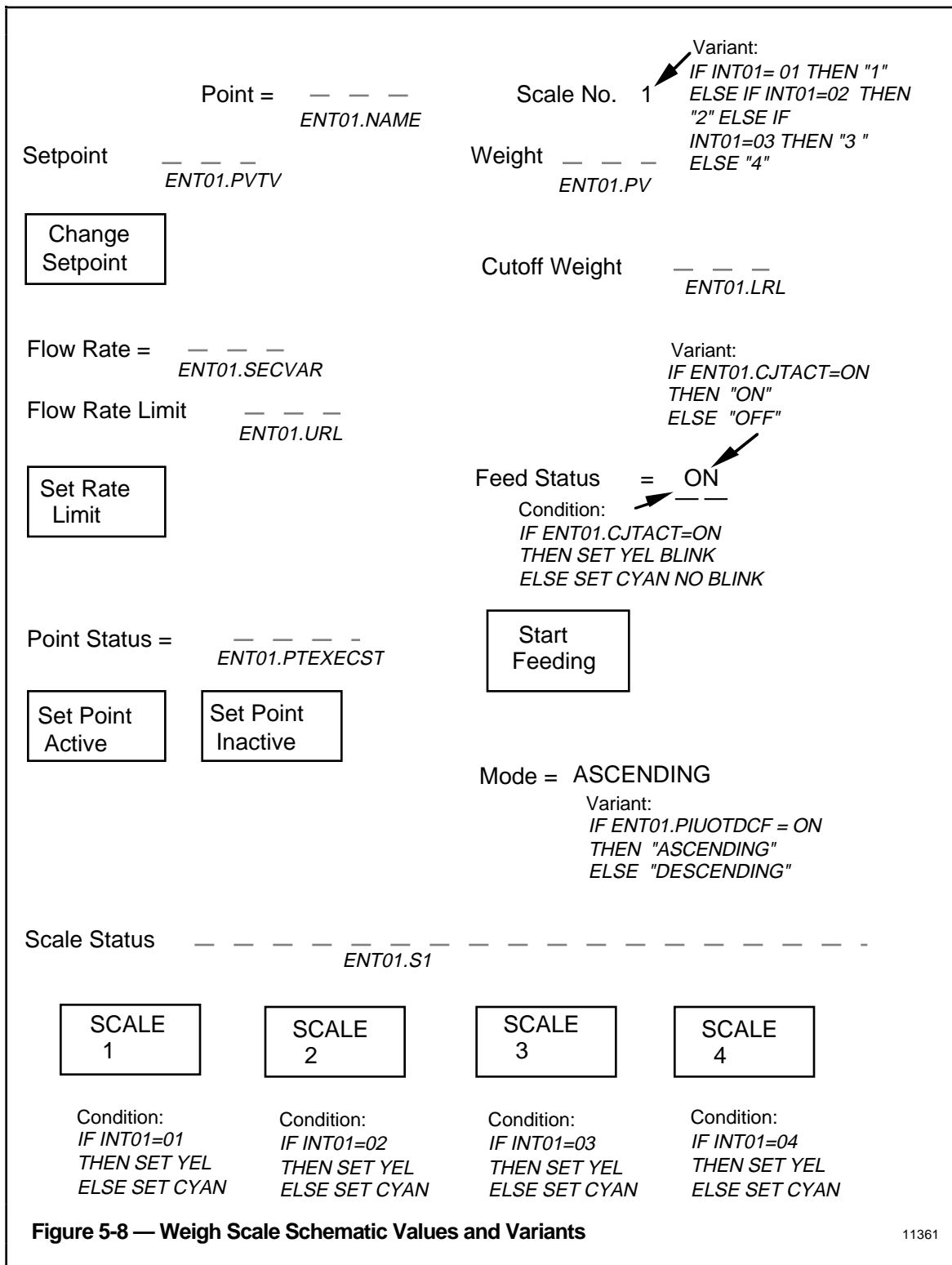


Figure 5-7 shows the targets. Action sequences are shown in italics.



11360

Figure 5-8 shows the values, variants, and conditional behavior.



### 5.3.2.1 Control Schematic Description

To understand the control, you should refer to the table of Weigh Scale/STI parameters in the *Process Manager Control Functions and Algorithms* manual. For help in understanding Picture Editor functions, refer to the *Picture Editor Reference Manual*.

The following comments help to explain the example schematic—

- Upon invoking the schematic, an initial target selects Scale 1 as described below. This guarantees that the DDB variables contain acceptable starting values.
- Selecting one of the four scale targets loads the DDB Entity ENT01 with the proper point reference. It also loads the DDB Integer INT01 with the scale number 1, 2, 3, or 4. Conditional behavior changes the selected target from white to yellow.
- At the top of the schematic, a value displays the contents of ENT01 to show the point name. A variant tests INT01 and displays the selected scale number. An invisible target over the word “Point =” is provided to call up the detail display if needed.
- The Change Setpoint and Set Rate Limit targets open ports for an operator to type in desired values.
- The Start Feeding target stores ON for parameter CJTACT to turn the feed on. A value displays the text “ON” after the words “Feed Status.” Below the word “ON,” a text line turns yellow and blinks when a conditional behavior test finds CJTACT = ON. When the setpoint is reached and feed is stopped at the scale, CJTACT returns to Off. “OFF” is displayed for the Feed Status and the line returns to steady-state cyan.
- A variant tests parameter PIUOTDCF and displays ascending or descending mode. If you need to change the mode, you can do so at the Detail Display.

### 5.3.2.2 Point Configuration

A typical STI point configured for use with the schematic is shown in Table 5-5. Default values are acceptable in most cases. Some output values are set to 0 and some control parameters are initially configured to OFF for safety. Ascending weight is assumed; if descending is desired, set PIUOTDCF to OFF.

Use your own point name, unit number, network, node, and slot numbers. Use default values for parameters not included. The last four entries contain required values.

Four points, (SDI101 through SDI104) were built for use in this example to show how four scales could be controlled. Configuration of the points was the same except for the name and slot number. Because each scale is connected to a separate FTA, scales 3 and 4 would require a second SDI IOP and its two associated IOPs.

**Table 5-5 — APM STIM (SDI) Point**

Parameter	Entry
NODETYP	APM
PNTFORM	FULL
PNTMODTY	STIM
SENSRTYP	STT
PVCHAR	LINEAR
STI_EU	MV
DECONF	
LRV	1
URL	0
DAMPING	0
PIUOTDCF	ON (Ascending)
CJTACT	OFF
INPDIR	DIRECT
PVEUHI	100.0 (Required)
PVEULO	0.0 (Required)
PVEXEUHI	100.0 (Required)
PVEXEULO	0.0 (Required)

### 5.3.2.3 Schematic Enhancements

You could add a drawing of a scale and insert parameters in appropriate places.

Some functions can be eliminated and others could be added. For example, if you only have one scale, eliminate the Point and Scale identifiers and the scale targets. Substitute the point ID in all expressions for ENT01. You may want to add a target to call the Detail Display for that point.

If you need to change from ascending to descending mode frequently, that functionality could be built into the schematic.

You can build the scale targets into another schematic to call a subpicture with scale control functions.



## PEER-TO-PEER COMMUNICATION AND PERFORMANCE

### Section 6

*This section discusses Peer-to-Peer Transaction Throughput constraint in terms that allow general estimates on system capability to be made before implementation of the overall control strategy has been carried out. A description of the performance statistics and methods that can be used to troubleshoot performance problems are included.*

#### 6.1 SYSTEM CONSTRAINTS ON PEER-TO-PEER COMMUNICATION

System constraints limit the quantity of data that can be transported between peer PMs. These constraints include limits on

- the number of parameters that can be passed using specific transport mechanisms (configuration limits).
- the total number of parameters that can be serviced in each second (Parameter Throughput).
- the number of transactions that can be processed each second (Transaction Throughput).

##### 6.1.1 Configuration Limits

Peer to peer configuration limits are discussed in Section 3 of the *Control Functions and Algorithms Manual*.

##### 6.1.2 Parameter Throughput Limit

The limit on Parameter Throughput in the PM has been estimated at 400 parameters per second. The sum of all parameter requests to any individual PM should not exceed this value.

While it is important that system configuration not exceed this limit, previous experience indicates that the Transaction Throughput limit discussed below is usually reached first.

##### 6.1.3 Transaction Throughput Limit

When parameters are transported over the UCN, they are sent in groups called messages. Two kinds of messages are involved: request messages and response messages.

- Request messages are those sent by the node that wants the data to the node that owns the data. They list all the parameters to be transported.
- Response messages are sent by the node that owns the data to the node that requested the data. They contain the actual parameter values.

All parameter transport requires a complete cycle of message request and message response. Such a cycle is called a transaction. Note that transactions occur when peer nodes communicate and also when NIMs and PMs communicate in response to LCN initiated requests.

The rate at which transactions occur is called the Transaction Throughput. Like Parameter Throughput, it is one component of the overall load on the PM. In most applications, Transaction Throughput is a more important component of load than Parameter Throughput.

Lab measurements have shown that in most configurations, the PM can handle 50 transactions per second.

In general, when a peer-to-peer configuration is designed, it should target a Transaction Throughput at or below 50 for steady-state and peak-load conditions alike. However, if this limit is exceeded by small amounts at infrequent intervals, the effect is small. UCN overruns result and some of the peer transactions will slow down.

Large and frequent overrun counts or UCN overrun soft failures are signs that Transaction Throughput is too heavy.

**Cases where 50 transactions per second is too much**—Certain configurations can have communications overruns at Transaction Throughput loads below 50.

**CAUTION**

Application engineers should be particularly cautious if a significant number of peer-to-peer connections make reference to IOP resident data.

The PM peer-to-peer communication capability is primarily intended to support transport of Control Processor resident data. If large amounts of IOP owned data must be transported, they should first be fetched into the Control Processor by a local connection. The peer-to-peer connections can then reference the data from its Control Processor location.

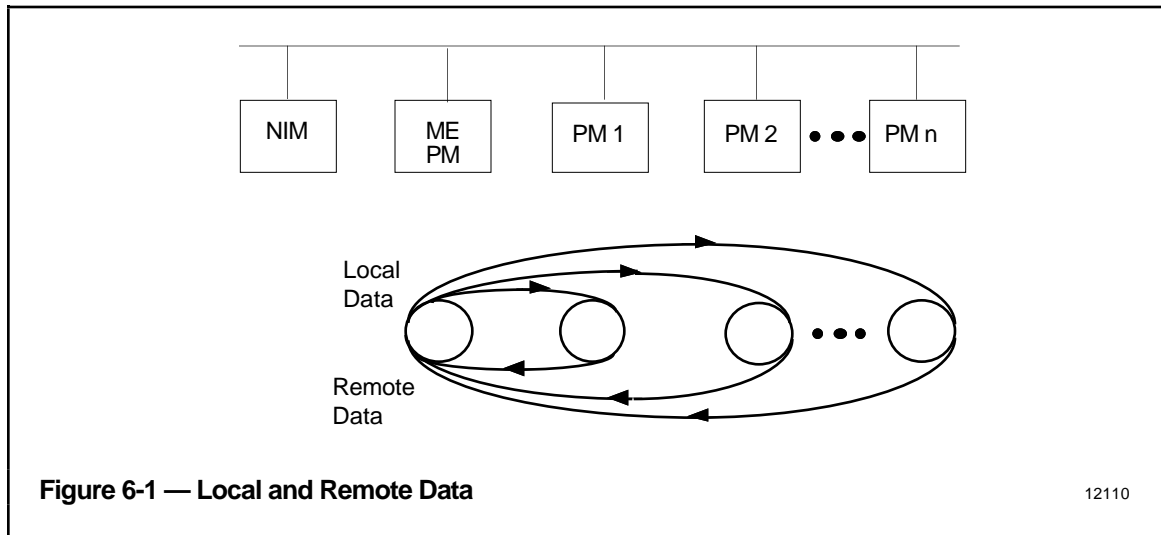
## 6.2 LOCAL AND REMOTE DATA

In the following discussion, Transaction Throughput values are always quoted in units of transactions per second.

Imagine a UCN with  $P + 1$  PM nodes operating as peers and one NIM node connected to the LCN (see Figure 6-1).

For a PM called ME, data transactions can be divided into two classes:

- those that transport remote data not owned by ME into ME
- those that transport local data owned by ME into other nodes.



Remote data can be transported by pulls (reads) from ME or by pushes (writes) from the P other nodes. Similarly, Local data can be transported by pushes from ME or by pulls from the P other nodes. The distinction between remote and local has to do with who owns that data that is to be transported, not with whether pushes or pulls are used to effect the transport.

For nodes that are exclusively data sinks, the peer-to-peer load is caused only by remote data. For nodes that are exclusively data sources, the situation is reversed.

### 6.2.1 Transaction Load from Peer Traffic

Transaction load on ME, generated by the P peer PM nodes, depends strongly on two factors:

- the number of other nodes communicating (P), and
- the specific mechanisms used to transport data between ME and the peer nodes.

**Node Count Effects**—The dependence on P is linear.

**Configuration Effects**—The dependence on configuration is harder to characterize. Relevant factors include the following:

- **ME is a data source, data sink, or both.** If peer transactions transport only local data, ME is a pure source. If peer transactions transport only remote data, ME is a pure sink. Using ME as both a source and a sink, typically doubles the transaction load.
- **Data is transported by pulls, pushes, or both.** Data transport to and from ME is accomplished primarily by pull or push mechanisms, or by a combination of the two. It is advantageous to use exclusively one or the other.
- **Data is transported by continuous points, sequence points, or both.** Data transport to and from ME is accomplished primarily by continuous point mechanisms (connections configured in Regulatory PV, Regulatory Control, or Logic points), by sequence point mechanisms (Read and Write statements in Process Module Data points), or by a combination of the two. It is advantageous to use exclusively one or the other.

### 6.2.2 Push/Pull Examples

The following examples illustrate the range of variation possible due to configuration options:

#### Example 1: Very Inefficient Transaction Usage

Both pushes and pulls are used to accomplish data transport. Both continuous and sequence point mechanisms are used.

Remote Transaction Load =  $10 \times P$

Local Transaction Load =  $10 \times P$

#### Example 2: Moderately Efficient Transaction Usage

Transport mechanisms are restricted to pushes. Both continuous and sequence mechanisms are used.

Remote Transaction Load =  $6 \times P$

Local Transaction Load =  $6 \times P$

#### Example 3: Moderately Efficient Transaction Usage

Transport mechanisms are restricted to pulls. Both continuous and sequence mechanisms are used.

Remote Transaction Load =  $6 \times P$

Local Transaction Load =  $6 \times P$

#### Example 4: Moderately Efficient Transaction Usage

Transport mechanisms are restricted to continuous mechanisms. Both pushes and pulls are used.

Remote Transaction Load =  $6 \times P$

Local Transaction Load =  $6 \times P$

#### Example 5: Moderately Efficient Transaction Usage

Transport mechanisms are restricted to sequence mechanisms. Both pushes and pulls are used.

Remote Transaction Load =  $8 \times P$

Local Transaction Load =  $8 \times P$

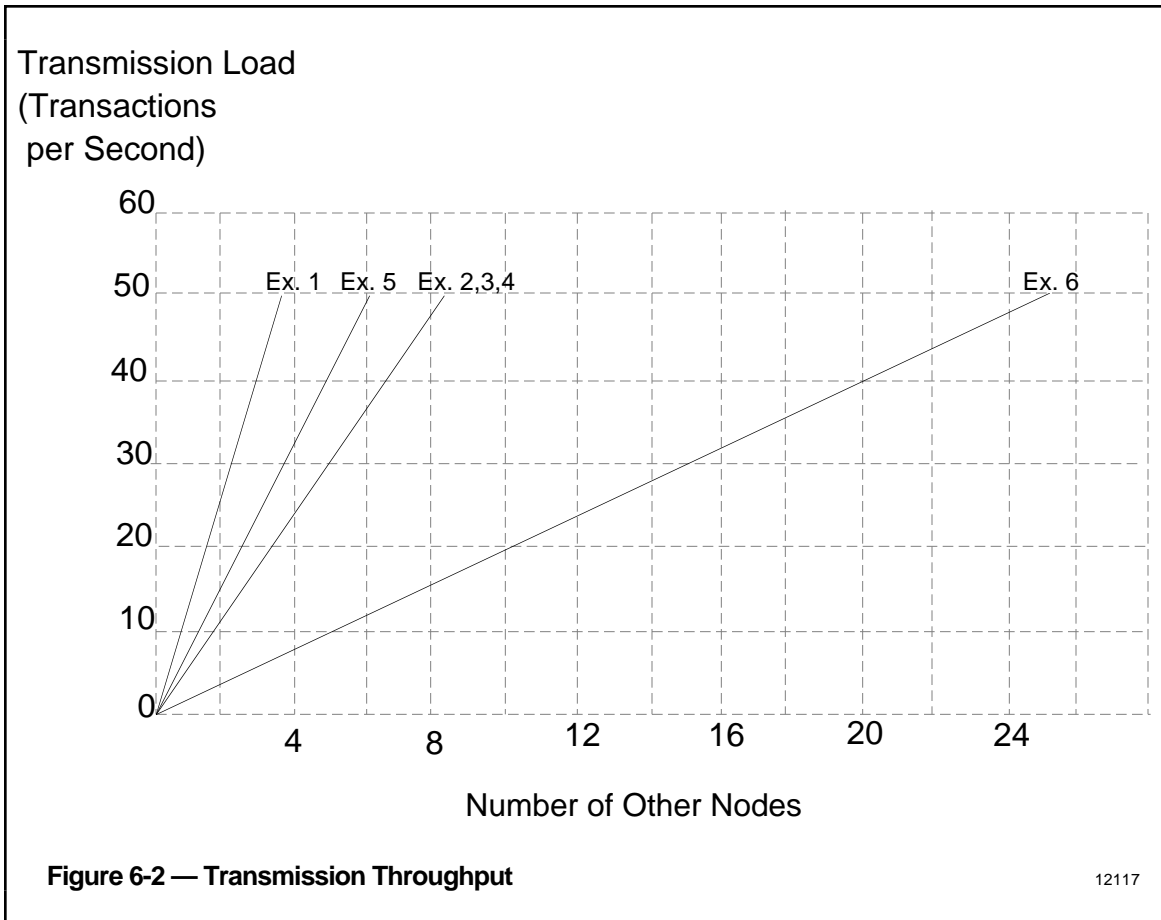
#### Example 6: Efficient Transaction Usage

Only pulling is used for data transport. Only continuous mechanisms are used.

Remote Transaction Load =  $2 \times P$

Local Transaction Load =  $2 \times P$

Figure 6-2 shows the dependence of transaction load on node count for each of the examples listed above. Each plot applies to transport of either remote or local data.



Note that for certain types of applications, even the approach used in example 6 may impose too much load. This is further described in subsection 6.5

### 6.3 TRANSACTION LOAD FROM NIM TRAFFIC

Transaction load imposed by the NIM upon ME can vary greatly depending on static factors such as LCN configuration. It also depends on dynamic factors such as:

- how many active displays draw data from ME
- how many AM or CM strategies pull and push to ME
- whether history data is being collected
- how many events are being generated by ME.

To roughly characterize these variations, two categories of loads can be considered:

- typical
- large

### 6.3.1 Typical NIM, Transaction Load = 8

This might be generated by an LCN with the following nodes and activities:

- one 68020 AM
- one CM50 generating requests at its maximum rate
- one HM generating requests at its typical rate
- one group displays on fast update
- one group displays on normal update
- one schematic display of about 100 parameters
- one detail display

### 6.3.2 Large NIM, Transaction Load = 21

This might be generated by an LCN with the following nodes and activities:

- two 68020 AMs
- one CM50 generating requests at its maximum rate
- one HM generating requests at its maximum rate
- two group displays on fast update
- two group displays on normal update
- three schematic displays of about 100 parameters each
- two detail displays

## 6.4 ESTIMATING TRANSACTION THROUGHPUT

Use the following procedure to estimate Transaction Throughput:

1. Identify how many PMs will be connected to the UCN.
2. Consider each PM one at a time thinking of it as the ME node.
3. Identify how many peer nodes will be communicating with ME (P).
4. Estimate whether ME is primarily a data source, primarily a data sink, or both.
5. Consider the kind of data to be transported and whether all or just a few transport mechanisms must be used.
6. Use the graph in Figure 6-2 to gauge the Transaction Throughput used to transport remote data.
7. Use the graphs in Figure 6-2 to gauge the Transaction Throughput used to transport local data.
8. Gauge the transaction load that will be posed by the NIM. Estimate it as either large or typical.
9. Compute an estimate of total transaction load by summing the load from peer local data, peer remote data, and the NIM. This number will serve as an upper limit on the actual transaction load that the configuration will generate.
10. Check how close the estimated total transaction load comes to the limit of 50 transactions per second. Note that configurations exceeding this limit may still run symptom free, particularly if the number of parameters transported in each transaction is low.
11. Repeat this procedure for each PM that uses peer to peer communication.

## 6.5 DESIGNING TO HANDLE HEAVY LOADS

Although the PM supports a variety of peer-to-peer transport mechanisms, applications that are expected to impose a heavy UCN communication load are best served by focusing on one of the two sequence mechanisms (Sequence Reads or Sequence Writes) and excluding others.

The reason for this is that the ability to program how and when peer-to-peer requests are made gives the application engineer the highest level of control over communications loading.

There are three load minimization strategies that can be used only with Sequence Reads and Sequence Writes.

- **Continuous Data That Can Tolerate a Low Rate of Update**—Some data may need to be transported over the UCN for regulatory or other kinds of continuous control activities. However, the update rate may not need to be as high as one or two Hz. If this is the case, Sequence Reads and Sequence Writes allow the control engineer to set the update rate as low as control requirements allow.
- **Status Data That Changes by Exception**—Various kinds of status data tend to change at infrequent intervals. To fetch or send this kind of data at a regular rate, however slow, could be wasteful. With Sequence Writes, this data can be sent to receivers only when it changes.
- **Data Associated with Common Equipment**—Batch applications frequently involve common equipment such as pipe headers which are used by multiple PMs. Status and process variable data associated with this equipment needs to be made available to each PM as needed. For many such applications, the common equipment may be used by only one PM at a time.

In applications like these, using a continuous push or pull mechanism to transport the data could be extremely wasteful as these mechanisms are active constantly, even when the recipient node has no need of the data. A more efficient approach is to use Sequence Writes.

With Sequence Writes, the PM that owns the equipment wiring can send the necessary data to the one node currently using the equipment.

In general, applications that need the most efficient peer-to-peer transport mechanism should use Sequence Reads or Sequence Writes.

## 6.6 HELPFUL HINTS

### 6.6.1 Adding Connections to an Existing Configuration

Adding more peer-to-peer connections may or may not have an effect on transaction load, depending on the particulars of the configuration. What matters is whether the new connection causes a new transport mechanism to be used or causes an operative transport mechanism to access an additional node.

For example, suppose that points RP0901, RC0901, and LG0901 each fetch three parameters from PM 11. If several more pulls from PM 11 were added into any Regulatory PV, Regulatory Control, or Logic points in PM 9, then no additional transaction load would result on either PM 9 or PM 11. However, if the pulls were directed to node 13 or if they were performed by a sequence running in PMDP0901, then the transaction load would increase at both the requesting and responding nodes.

### 6.6.2 Adding Another PM to an Existing UCN Configuration

When adding a PM to an existing configuration, do a careful analysis on the amount of peer data the node will source, the amount it will sink, and the node(s) which will be partner(s) in transactions.

Estimate transaction loads using the method described above for the new node and any nodes to which it may communicate. If estimated transaction loads severely exceed the limit of 50 transactions per second, then consider alternative system implementations.

### 6.6.3 Where Overruns Can Occur

In some heavily loaded peer-to-peer configurations, UCN overruns occur on the nodes(s) that have the highest transaction load. However, in other cases, overruns occur on nodes with transaction loads of less than 50. This is caused by peer access to a node that is overloaded and, therefore, cannot respond quickly to the request.

### 6.6.4 Effect of Alarm Inhibit

Since event reporting is one of the PM activities that causes transaction load, an intermittent or steady event load that is heavy can contribute to peer-to-peer communication problems. It is often thought that inhibiting alarms at the NIM should ameliorate this effect; this is incorrect. Event inhibit affects the transport of events on the LCN. Event traffic on the UCN remains unaffected.

### 6.6.5 Using Mostly One Transport Mechanism and a Little Bit of Another

If the purpose is to minimize transaction load, there is no advantage to making small or partial use of a particular transport mechanism. The reason for this is that if a new transport mechanism is used, even if for only a single parameter to a single other node, additional transactions are required.

Consider Example 2, described previously, that uses only the continuous pull mechanism to transport peer-to-peer data. If such a configuration were changed by adding a single Regulatory Control point running once per second, doing peer-to-peer pushes, then the local transaction load would go from  $2 \times P$  to  $2 \times P + 1$ . Conversely, adding more Regulatory Control or Logic point pulls, as long as the points were running on the same cycle and directed to the same node, would not affect the transaction load at all.

### 6.6.6 Nonredundant Versus Redundant PMs

A nonredundant PM in a peer-to-peer strategy generates somewhat less transaction load than a redundant PM. This is true even when the secondary is not in the backup state. However, the specification of 50 transactions per second does apply to the redundant case.

Nonredundant nodes have a slightly higher capability. The difference arises from the fact that redundancy imposes additional communication burden in the form of database backup operations in the Control Processor and status scans between the primary and secondary communications processors.

## 6.7 PEER-TO-PEER PERFORMANCE MONITORING

### 6.7.1 Peer-to-Peer Statistical Parameters

The statistics described in the following paragraphs are used to analyze peer-to-peer communications efficiency and to determine if a problem exists or is imminent. They are particularly useful for isolating problems to a node or a few nodes. Most of these statistics are presented on the Toolkit displays (refer to section 10 in the *Engineer's Reference Manual*).

In Release 410 (and later) software, the following types of statistics are provided:

- Percentage of free time available to the Central Processing Unit (CPU Free)
- Transaction Trip Times,
- Request/Response Transaction Throughput,
- Requests/Response Parameter Throughput,
- Node Access Rate,
- Event Throughput Rates, and
- Miscellaneous.

In general, three dimensions of statistics are reported: maximum, minimum, and average.

- Maximum values are collected for all statistics,
- Minimum values are only collected for the CPU Free statistics, and
- Average values are computed as a running average over the previous minute of samples.

Each parameter is briefly described in terms of the type of statistic it supplies. Additional information can be found in the *PM Parameter Reference Dictionary*.

### 6.7.1.1 CPU Free Statistics

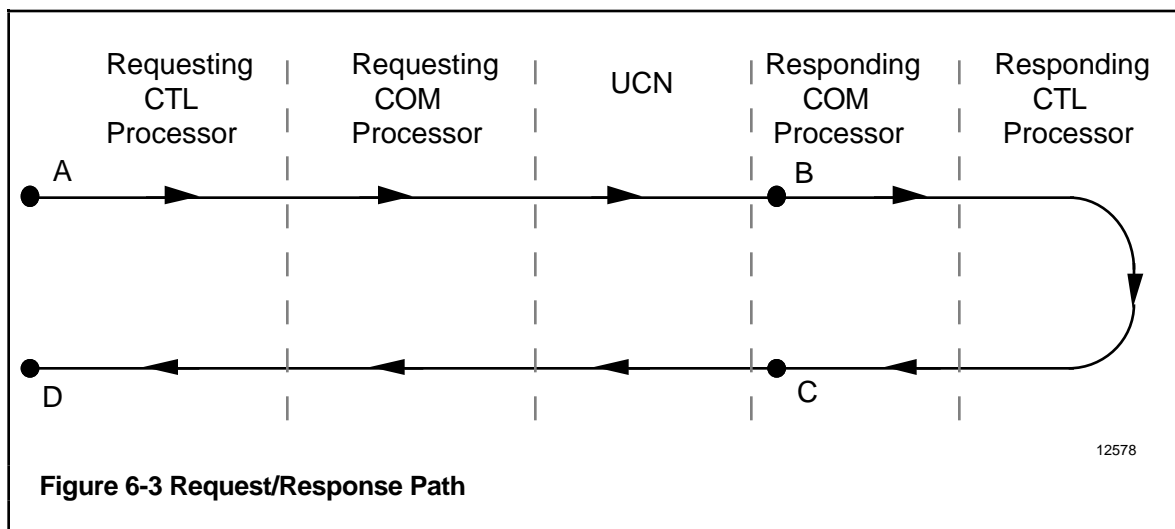
These statistics indicate the percentage of time in each second that the specified CPU is idle. Statistics are available for two CPUs: the Communication (COM) Processor and the Control (CTL) Processor.

The parameters are:

Parameter	Statistic Measured
<b>COMCFAVG</b>	Average CPU Free Time for the Communications Processor
<b>COMCFMAX</b>	Maximum CPU Free Time for the Communications Processor
<b>COMCFMIN</b>	Minimum CPU Free Time for the Communications Processor
<b>CTLCFAVG</b>	Average CPU Free Time for the Control Processor
<b>CTLCFMAX</b>	Maximum CPU Free Time for the Control Processor
<b>CTLCFMIN</b>	Minimum CPU Free Time for the Control Processor

UCN trip time statistics measure how many milliseconds it takes for a transaction to traverse part or all of its circuit.

Peer-to-peer request messages typically contain requests directed toward more than one responding node. Figure 6-3 shows the simplest case of a single responding node. When multiple responders are involved, request time values are based on the slowest response path.



**Request trip time**—Request trip time statistics provide the time required to go from A to D. They are measured in the requesting Control (CTL) Processor.

**Response trip times**—Response trip times statistics provide the time required to go from B to C. They are measured in the responding Communication (COM) Processor.

Trip time statistics are made available for fetch and store transactions. Store trip times would typically be a little larger than fetch trip times because more processing is generally required for parameter store operations.

Statistics are also provided that give a weighted sum of response fetch and store trip times.

The Transaction Trip Time parameters are shown in the following table:

<b>Parameter</b>	<b>Statistic Measured</b>
<b>FRQUTAVG</b>	Average Fetch Request Trip Time
<b>FRQUTMAX</b>	Maximum Fetch Request Trip Time
<b>FRSPTAVG</b>	Average Fetch Response Trip Time
<b>FRSPTMAX</b>	Maximum Fetch Response Trip Time
<b>SRQUTAVG</b>	Average Store Request Trip Time
<b>SRQUTMAX</b>	Maximum Store Request Trip Time
<b>SRSPTAVG</b>	Average Store Response Trip Time
<b>SRSPTMAX</b>	Maximum Store Response Trip Time
<b>TRATAVG</b>	Average UCN Response Trip Time (includes both fetches and Stores)
<b>TRATMAX</b>	Maximum UCN Response Trip Time (includes both fetches and stores)

### 6.7.1.3 Parameter and Transaction Throughput Statistics

These parameters specify the number of transaction and parameter accesses serviced by the node under examination. They include both fetches and stores. Transactions to the secondary APM are not included in the accumulated counts.

The parameters are:

Parameter	Statistic Measured
<b>NPARAVG</b>	The average number of parameters serviced per second (both requests and responses) to all other nodes.
<b>NPARMAX</b>	The maximum number of parameters serviced per second (both requests and responses) to all other nodes.
<b>NPRQUAVG(n)</b>	The average number of parameter requests serviced per second (issued to node n) when n = 1, 3, 5, ... 63 (n = 0 returns the average total number of parameter requests to all other nodes).
<b>NPRQUMAX(n)</b>	The maximum number of parameter requests serviced per second (issued to node n) when n = 1, 3, 5, ... 63 (n = 0 returns the maximum total number of parameter requests to all other nodes).
<b>NPRSPAVG(n)</b>	The average number of parameter responses serviced per second (issued to node n) when n = 1, 3, 5, ... 63 (n = 0 returns the average total number of parameter responses to all other nodes).
<b>NPRSPMAX(n)</b>	The maximum number of parameter responses serviced per second (issued to node n) when n = 1, 3, 5, ... 63 (n = 0 returns the maximum total number of parameter responses to all other nodes).
<b>NTRAAVG</b>	The average number of transactions per second (both requests and responses) to all other nodes.
<b>NTRAMAX</b>	The maximum number of transactions per second (both requests and responses) to all other nodes.
<b>NTRQUAVG(n)</b>	The average number of transaction requests per second (issued to node n) when n = 1, 3, 5, ... 63 (n = 0 returns the average total number of transaction requests to all other nodes).
<b>NTRQUMAX(n)</b>	The maximum number of transaction requests per second (issued to node n) when n = 1, 3, 5, ... 63 (n = 0 measures requests issued to all other nodes).
<b>NTRSPAVG(n)</b>	The average number of transaction responses per second (issued to node n) when n = 1, 3, 5, ... 63 (n = 0 returns the average total number of transaction responses to all other nodes).
<b>NTRSPMAX(n)</b>	The maximum number of transaction responses per second (issued to node n) when n = 1, 3, 5, ... 63 (n = 0 returns the maximum total number of transaction responses to all other nodes).

#### 6.7.1.4 Node Access Rate Statistics

These parameters specify the number of UCN nodes that the node under examination communicates with within a one second period of time. They include both fetches and stores.

Parameter	Statistic Measured
<b>NORQUAVG</b>	The average number of nodes to which requests are made each second.
<b>NORQUMAX</b>	The maximum number of nodes to which requests are made each second.
<b>NORSPAVG</b>	The average number of nodes to which responses are made each second.
<b>NORSPMAX</b>	The maximum number of nodes to which responses are made each second.
<b>NOTRAAVG</b>	The average number of nodes accessed each second (includes both requests and responses).
<b>NOTRAMAX</b>	The maximum number of nodes accessed each second (includes both requests and responses).

#### 6.7.1.5 Event Throughput Statistics

These parameters specify the number of events generated by the PM per second. The parameters are:

Parameter	Statistic Measured
<b>NEVTAVG</b>	The average number of events per second (this statistic uses a two second average rather than the one minute average used on all other statistics).
<b>NEVTMAX</b>	The maximum number of events per second.

#### 6.7.1.6 Miscellaneous Parameters

Parameter	Meaning
<b>BNDRESET</b>	The maximum/minimum (bounds) Statistic Reset flag.
<b>NSCANITM*</b>	Specifies the number of scan items in the UCN Scan Table
<b>SUMSLTSZ*</b>	Specifies the total amount of memory allocated for CL programs.

**Resetting Statistic Values**—Maximum, minimum, and average statistics can be reset at any time. This allows the user to establish a known start time at which statistics gathering begins. Resetting the statistics has the following effects:

- Maximum values are set to zero and then grow to larger values as new instantaneous values are computed.
- Minimum values are set to the maximum range limit and then shrink to smaller values as new instantaneous values are computed.
- Average values are set to the current instantaneous value.

\*These are viewed on the PM Diagnostic Display-Control Configuration page.

Statistic Reset typically occurs as the result of a target selection at the UCNSUMM, NODESTA1 or NODESTA2 displays. The effect of selecting the reset target at these three displays differs as indicated below.

- When reset is commanded from UCNSUMM all statistics in all UCN nodes are initialized as described previously.
- When reset is commanded from NODESTA1 all statistics in the selected node are initialized.
- When reset is commanded from NODESTA2, all statistics in the selected node are initialized.

If desired, Statistics Reset can also be controlled from CL programs. To do so, the parameter BNDRESET is written to ON at each node where statistics are being monitored. On a read, BNDRESET always returns off.

## 6.8 TROUBLESHOOTING PM COMMUNICATIONS OVERLOAD

### 6.8.1 Troubleshooting Guidelines

While it is best for TDC 3000<sup>X</sup> applications to be designed so that communication overload does not occur this is not always done. When overload problems occur, refer to the tips on the following pages and answer the questions below.

- What are the signs of PM overload?
- How do I know that the physical components of UCN communications are operating correctly?
- How can I tell that the overload is in the PM and not the NIM?
- How do I know which PMs are reporting symptoms?
- How do I know which PMs are the ones in overload?
- What are the possible causes of overload?
- What if the symptoms occur under peak load but not steady-state conditions?

### 6.8.1.1 Cases where PM Overload Is Indicated

Several different symptoms can indicate overload of PM communications capabilities:

- Intermittent black-outs or question marks appearing where UCN data is shown on US screens.
- Diagnostic time-out soft failures reported by a PM COM processor.
- UCN overrun soft failures reported by a PM.
- Moderate to high levels of UCN overrun counts.

Note that for the first symptom there are a variety of causes including the following:

- Poor physical communication integrity on the UCN. This might result from poor cable connections, poor cable layout, marginal modems or other communication equipment.
- NIM communications overload.

### 6.8.1.2 Distinguishing Physical Network Problems from PM Communications Overload

Poor network integrity can be identified through the use of the UCN communication statistics displays which include the following:

- **UCN COMM display**—This display is invoked by selecting the UCN COMM target on the UCN Status display. It shows UCN communication statistics for 16 nodes at a time. Refer to the *Universal Control Network Guidelines* document for explanation of how to interpret statistics.
- **UCN Statistics display**—This display is invoked by selecting the UCN STATS target on the PMM Diagnostic Display. It shows communications statistics for one UCN node at a time. Refer to the *Universal Control Network Guidelines* document for explanation of how to interpret statistics.
- **UCNEVENT and UCNCOMM Schematics**—These schematics show communication statistics for 16 UCN nodes at a time. Refer to the *Universal Control Network Guidelines* document for an explanation of how to interpret statistics.

If these displays indicate physical communication problems those problems should be corrected before doing further troubleshooting or making any other system changes.

### 6.8.1.3 Distinguishing NIM Communication Overload from PM Communications Overload

Symptoms at the Universal Stations can be caused by NIM overload as well as PM overload. To distinguish these two cases do the following:

- Use the on-line loading statistics available for the NIM to develop a thorough understanding of the NIM load. Crucial to this understanding will be the steady state value of the NIM CPU Free statistic (viewed on the Toolkit displays). If it is below 20% the NIM is a possible cause of US symptoms. NIM performance statistics can be observed from several different schematics. Schematic PERFMENU provides an overview of available LCN and UCN performance schematics. Schematics useful for studying NIM loading are CPUCHKR, PARCHKR, NODEPRF and NIMTRND.
- Make a similar study for all PMs referenced by the US displays showing symptoms. As with the NIM, Communications Processor CPU Free values below 20% warrant further investigation. PM performance schematics (UCNSUMM, NODESTA1 and NODSTA2) can be invoked from schematic PERFMENU as can NIM performance schematics. Further information about how to use UCNSUMM, NODESTA1 and NODSTA2 is presented below.
- Although not always present, when event journals show one or more occurrences of the message "job queue full", this is a definite indication of NIM overload.

### 6.8.1.4 Identifying PMs or AM That Are Reporting Symptoms

It is entirely straightforward to identify the node showing symptoms. As a result this is typically known by the time troubleshooting begins:

- If the symptom is a soft failure then the journal event includes the number of the reporting node.

One case that is slightly more subtle concerns UCN Overruns:

- If the overrun rate is too low to cause a soft failure, the symptom may be observed by selecting the CONTROL CONFIG target on the PMM Diagnostic Display. UCN Overrun statistics are shown along with Point Processing and IO Link Overrun statistics at the bottom of the screen. Either the CURRENT HOUR UCN ACCESS OVERRUNS or PREVIOUS HOUR UCN ACCESS OVERRUNS, or both will be non-zero if overruns are occurring. Overruns of about 50 per hour or less are not cause for great concern, but you should investigate counts greater than this.
- Information can also be obtained by looking at the MAXIMUM REQUEST TRIP TIME parameters on the NODESTA1 schematic. If either value is over 500 ms, then overruns are probably occurring from time-to-time. If either is over 750 ms, then overruns are definitely occurring at some rate.

Note that the node or nodes showing symptoms are not necessarily the same as the nodes causing the symptoms. For example if PM 5 is making requests to PM 9, and PM 9 is overloaded, PM 5 could show overruns even though its own Communications processor was lightly loaded.

### 6.8.1.5 Identifying Heavily Loaded PMs

Because overload at one node can cause symptoms at another node, a method is needed to track down the overloaded nodes:

- Call up the UCNSUMM schematic. This display was designed to give summary information for all nodes configured on the UCN. Four types of information are presented on this display:
 

Node Access Rate,	Parameter Throughput, and
Transaction Throughput,	Transaction Response Trip Times.
- For locating overloaded nodes, Response Trip times are the most useful. For this statistic, average values over 300 ms or maximum values over 500 ms may indicate problems. However, regardless of the absolute value shown, locate the node with the highest average Response Trip Time.
- For the node identified call up the NODESTA1 schematic and look at average and minimum COM CPU Free statistic. If average values are less than 20%, or minimum values are less than 10% then this node is probably causing symptoms.
- After a node with low COM CPU Free has been located, continue to use UCNSUMM and NODESTA1 schematics until all nodes with average COM CPU Free below 20% have been located.
- Transaction Throughput values shown on the UCNSUMM schematic are harder to interpret than Response Time values. There are configurations that run well with average Transaction Throughput greater than 50. Conversely, there are configurations which run poorly with Transaction Throughput under 30. Variations are caused by factors such as the magnitude of Parameter Throughput, the types of parameters being transferred, and the location of the data being transferred.

Note that requesting IO Link data from the UCN can be particularly costly to COM Processor performance. As a rule of thumb, an average Transaction Throughput value of 40 or greater can be considered high.

- The Parameter Throughput values shown on the UCNSUMM schematic are also hard to interpret. While 400 Is generally a fairly high value, systems can run flawlessly with values considerably higher.

### 6.8.1.6 Identifying the cause of Node overload:

**Event Traffic**—PM communications overload can sometimes be caused by high steady state event throughput.

- Call up the NODESTA1 schematic. Look at the average Event Throughput. Values greater than 10.0 indicate heavy activity that could be overloading the Communications Processor.
- If heavy event load is present, enable the event journals and examine them for chattering events.

**Parameter Transfers on the UCN**—After overloaded nodes have been identified the next step is to analyze system configuration to see why. This process may involve examination of schematic source code, point build IDF files, CL source code or other system configuration data generated by application engineers. It can be very labor intensive. Node performance statistics can assist somewhat in this process.

- Call up the UCNSUMM schematic. Look at the average Node Access Rate. Values greater than 7 may indicate that too many nodes are making requests to the node in question.
- Call up the NODESTA2 schematic. This display may be used to show which UCN nodes are placing a communication burden on the node in overload. Any node that is not communicating with the node selected at schematic invocation shows zeroes for all statistics. Nodes that are communicating show non-zero values.
- Look at the Transaction Throughput data on the NODESTA2 schematic. Investigate nodes that are placing a high Transaction Throughput burden on the overloaded node (for either request or response transactions).
- Look at the Parameter Throughput data on the NODESTA2 schematic. Investigate nodes that are placing a high Parameter Throughput burden on the overloaded node (for either request or response transactions).

Two other tools may be helpful to analyze the configuration of peer connections on a UCN. These are the Find Names and Documentation Tool utilities. Refer to the *Command Processor Manual* in the *Implementation/Engineering Operations - 1 binder* for information on these functions. Together they can be used to construct tables of all peer-to-peer connections that put a load on each node.

Constructing such tables involves two general steps:

- Use Find Names to generate lists of all entity-parameter references in all checkpoints and all PM/CL programs used in the UCN configuration.
- Use the Documentation Tool's filtering capabilities to isolate those entity-parameter references that indicate peer connections.

### 6.8.1.8 When Symptoms Seem to Occur Only under Peak Conditions

When a node experiences overload during peak conditions, it will generally be marginal under steady-state conditions. Thus, an examination of steady-state data will almost always identify nodes that are overloaded, and are therefore causing symptoms.

- In rare cases where analysis of data taken under normal operating conditions is not sufficient, trends can be used. Parameter values displayed in the UCNSUMM, NODESTA1 and NODESTA2 schematics have corresponding names indicated as well.
- For example, on UCNSUMM the characters "TRAT..." appear under the Transaction Response Trip Time column. The three dots indicate that "MAX" may be appended to get the name of the parameter displayed in the MAX column while "AVG" may be appended to get the name of the parameter displayed in the AVG column. Knowing this, take trends of \$NMuuBnn.TRATMAX and \$NMuuBnn.TRATAVG to see if any node shows strong deviations over a long period of time. The symbol "uu" here denotes the UCN number while "nn" denotes the node number of the PM in question.
- NODESTA1 follows the same convention for displaying parameter names as UCNSUMM. For example in the CPU Free Percentage area of the screen, COMCF..." indicates that three statistical values for COM Processor CPU Free may be trended by referring to \$NMuuBnn.COMCFMIN, \$NMuuBnn.COMCFMAX, or \$NMuuBnn.COMCFAVG.
- NODESTA2 follows the same convention with one additional detail. Note that under the Transaction Throughput Response column "NTRSP...(i)" appears. This indicates that \$NMuuBnn.NTRSPAVG(i) gives the average number of responses per second that the node under study is sending to node i, while \$NMuuBnn.NTRSPAVG(0) gives the total number of responses per second that the node under study is servicing for all other nodes.



---

# Index

---

Topic	Section Heading
Activities, Engineering Main Menu	1.1
Addresses	
NIMs	2.2.1.1
PMMs and LMs	2.2.1.2
Assessment of NIM Loading	4.2
Auto/Manual Station ( <i>see Manual/Auto Station</i> )	
Checkout of Hardware, References	1.2.2.2
Configuration, UCN and Node-Specific Points	3
Considerations, Operational	2
Consistent NIM and PM Databases, Maintenance of	2.3.1
Databases	
IOP Security	2.3.2
NIM and PM, Consistency of	2.3.1
Data Point Building	3.2
Dependencies, Implementation	1.3
Entities, UCN Points and Node-Specific Points	3
Estimating NIM Loading	4.1
Hardware Implementation, References	1.2.2
Implementation	
Dependencies	1.3
Hardware, References	1.2.2
Summary of Tasks	1.1
Two Logical Process Networks	4.3.1
Information, Saving and Restoring	2.3
Installation and Checkout, References	1.2.2.2
Introduction	1
IOPs	
Database Security	2.3.2
Redundant	2.2.3
LCN and UCN Addresses for NIMs	2.2.1.1
Loading	
NIM Performance	4
NIMs, Assessment of	4.2
NIMs, Estimates of	4.1
NIMs, Two NIMs to Share Load	4.3
Logical Processing Networks	4.3.1
Networks, Processing, Implementation of Two	
Logical Process Networks	4.3.1
Manual/Auto Station	5.2
NIMs	
Addresses	2.2.1.1
Consistency of NIM and PM Databases	2.3.1
Loading and Performance	4
Loading, Assessment of	4.2
Loading, Estimates of	4.1
Redundant	2.2.1
Sharing Processing Load	4.3
Node-Specific Points and UCN Points	3
Operational Considerations	2
Overruns, Peer-to-Peer	6.6.3

---

# Index

---

Topic	Section Heading
Peer-to-Peer Communication	6
System Constraints	6.1
Statistical Parameters	6.7.1
Throughput Limits	6.1.2, 6.1.3
Performance Monitoring	6.7
Procedure for Estimating Transaction Throughput	6.4
Troubleshooting Communications Overload	6.8
Performance, NIMs	4
PMMs	
Addresses	2.2.1.2
Redundant	2.2.2
Point Building	3
Process Manager Subsystem	2.1
Processing Load, Two NIMs to Share	4.3
Process Networks, Implementation of Two	
Logical Process Networks	4.3.1
Publications	1.2
Reconfiguration	3.3
Redundancy	2.2
IOPs	2.2.3
NIMs	2.2.1
PMMS	2.2.2
References	1.2
Engineering Activities	1.2.1
Hardware Implementation	1.2.2
Installation and Checkout	1.2.2.2
Service	1.2.2.3
Site Planning	1.2.2.1
Remote NIM to Share Processing Load	4.3
Saving and Restoring of PM Information	2.3
Security, IOP Database	2.3.2
Serial Device Interface (SDI) Implementation	5
Service References	1.2.2.3
Sharing Processing Load, Two NIMs	4.3
Site Planning, References	1.2.2.1
Statistical Parameters, Peer-to-Peer	6.7.1
Summary of Process Manager Implementation Tasks	1.1
Task Dependencies	1.3
Toledo Weigh Scale ( <i>see Weigh Scale Implementation</i> )	
Two Processing Networks	4.3.1
UCN Addresses	
NIMs	2.2.1.1
PMMs and LMs	2.2.1.2
UCN Points and Node-Specific Points	3
Use of a "Remote" NIM to Share Processing Load	4.3
Weigh Scale Implementation	5.3

# READER COMMENTS

Honeywell IAC Automation College welcomes your comments and suggestions to improve future editions of this and other publications.

You can communicate your thoughts to us by fax, mail, or toll-free telephone call. We would like to acknowledge your comments; please include your complete name and address

**BY FAX:** Use this form; and fax to us at (602) 313-4108

**BY TELEPHONE:** In the U.S.A. use our toll-free number 1\*800-822-7673 (available in the 48 contiguous states except Arizona; in Arizona dial 1-602-313-5558).

**BY MAIL:** Use this form; detach, fold, tape closed, and mail to us.

Title of Publication: **PM Implementation Guidelines** Issue Date: **8/95**

Publication Number: **PM12-500**

Writer: **B. Damours**

**COMMENTS:** \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**RECOMMENDATIONS:** \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

NAME \_\_\_\_\_ DATE \_\_\_\_\_  
TITLE \_\_\_\_\_  
COMPANY \_\_\_\_\_  
ADDRESS \_\_\_\_\_  
CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP \_\_\_\_\_  
TELEPHONE \_\_\_\_\_ FAX \_\_\_\_\_

(If returning by mail, please tape closed; Postal regulations prohibit use of staples.)

Communications concerning technical publications should be directed to:

Automation College  
Industrial Automation and Control  
Honeywell Inc.  
2820 West Kelton Lane  
Phoenix, Arizona 85023-3028

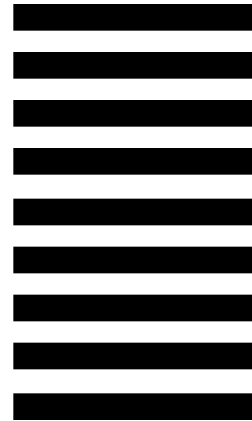
FOLD

FOLD

From: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE USA



Cut Along Line

**BUSINESS REPLY MAIL**  
FIRST CLASS      PERMIT NO. 4332      PHOENIX, ARIZONA

POSTAGE WILL BE PAID BY ....

**Honeywell**

Industrial Automation and Control  
2820 West Kelton Lane  
Phoenix, Arizona 85023-3028

Attention: Manager, Quality

FOLD

FOLD

Additional Comments:



**Honeywell**

---

**Industrial Automation and Control**  
Honeywell Inc.  
16404 North Black Canyon Highway  
Phoenix, Arizona 85023-3033

*Helping You Control Your World*