

**GFK-0466**

**New In Stock!**

**~~GE Fanuc Manuals~~**

<http://www.pdfsupply.com/automation/ge-fanuc-manuals/programming-software/GFK-0466>

**programming-software**

**1-919-535-3180**

Logicmaster 90

[www.pdfsupply.com](http://www.pdfsupply.com)

**Email:** [sales@pdfsupply.com](mailto:sales@pdfsupply.com)



# *GE Fanuc Automation*

---

*Programmable Control Products*

## **Logicmaster™ 90 Series 90™ -30/20/Micro Programming Software**

**User's Manual**

*GFK0466L*

*September 1998*

# Warnings, Cautions, and Notes as Used in this Publication

## Warning

**Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.**

**In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.**

## Caution

**Caution notices are used where equipment might be damaged if care is not taken.**

## Note

Notes merely call attention to information that is especially significant to understanding and operating the equipment.

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware or software, nor to provide for every possible contingency in connection with installation, operation, or maintenance. Features may be described herein which are not present in all hardware and software systems. GE Fanuc Automation assumes no obligation of notice to holders of this document with respect to changes subsequently made.

GE Fanuc Automation makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. No warranties of merchantability or fitness for purpose shall apply.

The following are trademarks of GE Fanuc Automation North America, Inc.

Alam Master	CIMSTAR	Helpmate	PROMACRO	Series Six
CIMPLICITY	GENet	Logicmaster	Series One	Series 90
CIMPLICITY90-ADS	Genius	Modelmaster	Series Three	VuMaster
CIMPLICITYPowerTRAC	Genius PowerTRAC	ProLoop	Series Five	Workmaster
VersaMax				

This manual describes the features that are used to create ladder logic user programs for Series 90™-30 PLCs, Series 90™-20 PLCs, and Series 90™ Micro PLCs. These features are available with Release 9.02 of the Logicmaster™ 90-30/20/Micro programming software.

## Revisions to This Manual

Release 9.02 of Logicmaster and the Release 9 CPUs include several enhancements. Refer to the following sections for information about these new features.

- There are new 350 and 360 series CPUs. One of the new features most of these new models offer is communications through embedded serial communications ports. Refer to page 10-18 for information about configuring the ports.
- The 364 model CPU includes built-in TCP/IP Ethernet communications capabilities. Refer to pages 10-21 and following for information about this feature. Also, refer to the *TCP/IP Ethernet Communications for Series 90™ PLCs* manual (GFK-1541) for detailed information about this new CPU.
- There is a new function added to the instruction set: the Sequential Event Recorder (SER). There is a brief discussion of this function on page 3-13 of this manual and a detailed discussion in Chapter 4 of the *Series 90™-30/20/Micro Programmable Controllers Reference Manual* (GFK-0467K or later).

In addition, in our effort to improve the quality of Logicmaster documentation, there are clarifications and corrections in several places within this manual.

## Content of This Manual

The information in this book is arranged as chapters that correspond to the main features of the Logicmaster 90-30/20/Micro software. This manual contains the following chapters:

**Chapter 1. Introduction:** provides an overview of the features available with the Logicmaster 90-30/20/Micro programming software. After you read chapter 1, refer to the chapters that describe the functions you want to use.

**Chapter 2. Operation:** explains what you need to know to install and start up the software. It also explains the format of the software screens.

**Chapter 3. Program Editing:** describes program entry and edit features.

**Chapter 4. Reference Tables:** describes how to display tables of reference values, change formats in reference tables, force references, and override bit-oriented references.

**Chapter 5. PLC Control and Status:** describes how to control and modify the operation of a connected PLC. These features include how to change the PLC operating state,

display and access PLC privilege levels, display and clear PLC and I/O faults, display PLC and program memory usage, display configured reference sizes, and modify sweep parameters.

**Chapter 6. Programmer Setup:** explains how to set up the programmer for communication with the PLC, and how to select the programmer operating mode.

**Chapter 7. Program Folders:** describes how to create, select, rename, modify, copy, or delete program folders.

**Chapter 8. Program Utilities:** describes how to use program utility functions to transfer programs and tables between the programmer and the PLC, to compare programs and data in the programmer with programs and data in the PLC, and to clear PLC memory. This chapter also describes how to transfer a program, register table, and configuration between the PLC and EEPROM (or Flash), and to compare a program, register table, or configuration in the PLC with a program, register table, or configuration in EEPROM.

**Chapter 9. Print Functions:** describes how to use the print functions to enter printer parameters, create files containing information to be printed, and print copies of programs, reference tables, and display screens.

**Chapter 10. I/O Configuration:** describes how to use the configuration software to configure I/O modules.

**Chapter 11. CPU Configuration:** describes how to use the configuration software to set the operating characteristics of the CPU.

This manual contains the following appendices:

**Appendix A. Programming Lesson:** provides a sample programming lesson with simple instructions for creating a program folder, creating a program, entering a variable declaration, adding ladder logic to the program, printing the program, and exiting the programmer.

**Appendix B. Configuration Lesson:** provides a sample configuration lesson with simple instructions for creating a program folder and configuring various modules.

**Appendix C. Programmer Environment Setup:** describes how to modify the setup parameters.

**Appendix D. User Command Menu:** describes how to maintain a file of DOS-executable commands outside of the Logicmaster software packages.

**Appendix E. Instruction Mnemonics:** lists mnemonics that can be typed to display program instructions while searching through or editing a program.

**Appendix F. Key Functions:** lists the special keyboard assignments used for the Logicmaster 90 software.

**Appendix G. Files Created with Logicmaster 90-30/20/Micro Software:** describes the files created by Logicmaster 90-30/20/Micro software that are associated with the program or folder.

**Appendix H. Common User Errors:** describes common problems which may occur and the corrective action to take.

**Appendix I. Variable Declaration Table Import/Export Using Comma Separated Variable (CSV) Format:** describes the format and requirements you need to consider when using the importing or exporting feature for Variable Declarations.

## Related Publications

- Series 90™-30/20/Micro Programmable Controllers Reference Manual* (GFK-0467).
- Logicmaster™ 90 Series 90-30 and 90-20 Important Product Information* (GFK-0468).
- Series 90™-30 Programmable Controller Installation Manual* (GFK-0356).
- Series 90™-20 Programmable Controller Installation Manual* (GFK-0551).
- Series 90™-30 I/O Module Specifications Manual* (GFK-0898).
- Series 90™ Programmable Coprocessor Module and Support Software User's Manual* (GFK-0255).
- Series 90™ PCM Development Software (PCOP) User's Manual* (GFK-0487).
- CIMPLICITY™ 90-ADS Alphanumeric Display System User's Manual* (GFK-0499).
- CIMPLICITY™ 90-ADS Alphanumeric Display System Reference Manual* (GFK-0641).
- Alphanumeric Display Coprocessor Module Data Sheet* (GFK-0521).
- Series 90™-30 High Speed Counter User's Manual* (GFK-0293).
- Series 90™-30 and 90-20 PLC Hand-Held Programmer User's Manual* (GFK-0402).
- Power Mate APM for Series 90™-30 PLC - Standard Mode User's Manual* (GFK-0840).
- Power Mate APM for Series 90™-30 PLC - Follower Mode User's Manual* (GFK-0781).
- Series 90™-30 Genius® Communications Module User's Manual* (GFK-0412).
- Series 90™-30 Enhanced Genius® Communications Module User's Manual* (GFK-0695).
- Series 90™ PLC Serial Communications User's Manual* (GFK-0582).
- Series 90™ Programmable Controller RS-422/RS-485 to RS-232 Converter Data Sheet* (GFK-0550).
- Series 90™-30 Genius® Bus Controller User's Manual* (GFK-1034).
- Series 90™-70 FIP Bus Controller User's Manual* (GFK-1038).
- Series 90™-30 FIP Remote I/O Scanner User's Manual* (GFK-1037).
- Field Control™ Distributed I/O and Control System Genius® Bus Interface Unit User's Manual* (GFK-0825).
- Series 90™ Micro Programmable Logic Controller User's Manual* (GFK-1065).

## *We Welcome Your Comments and Suggestions*

At GE Fanuc Automation, we strive to produce quality technical documentation. After you have used this manual, please take a few moments to complete and return the Reader's Comment Card located on the next page.

*David Bruton  
Sr. Technical Writer*



<b>Chapter 1</b>	<b>Introduction .....</b>	<b>1-1</b>
	<b>Section 1: Product Overview .....</b>	<b>1-1</b>
	What You Will Need .....	1-2
	The MS-DOS Version Needed .....	1-3
	Help Screens .....	1-3
	Key Functions .....	1-3
	Before You Begin .....	1-3
	<b>Section 2: Configuration Software .....</b>	<b>1-4</b>
	CPU Configuration .....	1-4
	PLC Time and Date .....	1-4
	PLC Memory Allocation .....	1-4
	Fault Display and Clearing .....	1-4
	System Response to Faults .....	1-4
	I/O Configuration .....	1-5
	<b>Section 3: Programming Software .....</b>	<b>1-6</b>
	Creating or Editing a Program .....	1-6
	Displaying Tables of Reference Values .....	1-7
	Start/Stop PLC Execution .....	1-7
	Fault Display and Clearing .....	1-7
	PLC Sweep Time Display and Change .....	1-7
	PLC and Program Memory Information .....	1-8
	Selecting the Programmer Operating Mode .....	1-8
	System Security .....	1-8
	Setting Up PLC Communications Using a WSI Board .....	1-9
	Setting Up PLC Serial Communications for Standard Serial COM Ports .....	1-9
	Program Folders .....	1-9
	Transferring Programs .....	1-10
	Printing Programs and Configuration .....	1-10
<b>Chapter 2</b>	<b>Operation .....</b>	<b>2-1</b>
	<b>Section 1: Hardware Setup (WSI Version) .....</b>	<b>2-2</b>
	Installing the Work Station Interface Board .....	2-2
	Grounding .....	2-2
	Cabling .....	2-2

<b>Section 2: Hardware Setup (Standard Serial Communications Version)</b>	<b>2-3</b>
<b>Section 3: Software Installation</b>	<b>2-4</b>
AUTOEXEC.BAT and CONFIG.SYS Files	2-4
Installing Logicmaster Under Windowsr 95	2-6
Installing Logicmaster Under Windows NTt	2-6
Running Other Software with the CONFIG.SYS File for Logicmaster 90-30/20/Micro Software	2-7
Installation Instructions	2-8
Programmer Setup	2-12
Using a Modem	2-13
<b>Section 4: Startup/Exit</b>	<b>2-16</b>
Starting the Programming or Configuration Software	2-17
Exiting (Quitting) the Programming or Configuration Software	2-18
Programming Software Main Menu	2-19
Configuration Software Main Menu	2-21
<b>Section 5: Keyboard Functions</b>	<b>2-24</b>
Keyboards Supported	2-24
Key Functions	2-24
Keyboard Macros (Teach Mode)	2-24
Creating a Teach Sequence	2-25
Playing Back a Teach Sequence	2-25
Pausing the Playback of a Teach Sequence	2-25
Using a Teach File to Run Software from an MS-DOS Batch File	2-26
<b>Section 6: Screen Format</b>	<b>2-27</b>
Function Key Assignments	2-27
Message Line	2-28
Command Line	2-28
Status Information	2-28
PLC/Programmer Status: Definitions	2-29
Selecting the Programmer Operating Mode	2-30
<b>Chapter 3 Program Editing</b>	<b>3-1</b>
<b>Section 1: Ladder Logic Program Elements</b>	<b>3-2</b>
Data Zoom	3-16
Using the Data Zoom Feature	3-16
<b>Section 2: Program Format</b>	<b>3-18</b>
Creating or Editing Program Logic	3-19
Structure of a Ladder Logic Rung	3-19
Ladder Logic Language Rules	3-20

<b>Section 3: Program Entry</b> .....	<b>3-23</b>
Using Mnemonics .....	3-23
Inserting Logic Elements .....	3-24
Inserting Functions .....	3-27
Exiting Rung Entry .....	3-29
Using the Cursor to Select a Reference Table .....	3-29
<b>Section 4: Program Annotation</b> .....	<b>3-30</b>
Entering Nicknames and Reference Descriptions .....	3-31
<b>Section 5: Variable Declaration Table</b> .....	<b>3-33</b>
Displaying the Variable Declaration Table .....	3-34
Entering Variable Declarations .....	3-36
Copying a Variable Declaration .....	3-37
Editing Variable Declarations .....	3-39
Deleting Variable Declarations .....	3-39
Searching for Variable Declarations .....	3-39
Using Goto .....	3-39
Cut/Pasting Variable Declarations .....	3-39
Automatically Inserting References .....	3-40
Viewing the Identifier Table .....	3-40
Importing to and Exporting from the Variable Declarations Table .....	3-41
Comma Separated Variable (CSV) Format .....	3-41
Importing SNF Formatted Files .....	3-41
Exporting SNF (CSV) Formatted Files .....	3-43
<b>Section 6: Rung Comments</b> .....	<b>3-46</b>
Inserting a Rung Comment .....	3-46
Adding Text .....	3-47
User-Defined Footers in Listings .....	3-48
Creating Borders .....	3-50
Starting a New Page of Comments .....	3-51
Printing a Title .....	3-51
Creating Longer Comments .....	3-51

<b>Section 7: Changing the Display Mode</b> .....	<b>3-52</b>
<b>Section 8: Subroutine Blocks</b> .....	<b>3-57</b>
Adding Subroutine Block Declarations .....	3-58
Editing Subroutine Block Declarations .....	3-58
Deleting Subroutine Block Declarations .....	3-59
Searching for Subroutine Block Declarations .....	3-59
Using Goto .....	3-59
Zooming into Subroutine Block Logic .....	3-59
Locking/Unlocking Subroutines .....	3-60
Locking a Subroutine .....	3-61
Unlocking a Subroutine .....	3-63
Permanently Locking a Subroutine .....	3-64
Displaying the Lock Status of the Subroutine .....	3-64
Periodic Subroutines .....	3-65
Executing a Periodic Subroutine .....	3-65
Restrictions on Use of the Periodic Subroutine .....	3-66

<b>Section 9: Rung Edit</b> .....	<b>3-67</b>
Editing a Rung .....	3-69
Entering Insert or Edit Mode .....	3-69
Entering Instructions .....	3-69
Entering/Modifying Data Types .....	3-71
Moving the Cursor within a Rung .....	3-71
Entering a Reference Address .....	3-72
Entering Nicknames .....	3-72
Using Vertical and Horizontal Links .....	3-72
Using Continuation Coils and Contacts .....	3-73
Deleting an Element .....	3-74
Using Open Space Functions .....	3-74
Completing (Accepting) Rung Entry .....	3-75
Entering an Instruction Length .....	3-75
Viewing Variable Declarations .....	3-76
Deleting a Rung .....	3-76
Selecting Rungs .....	3-77
Cutting Selected Rungs .....	3-78
Pasting Previously Cut Rungs .....	3-79
Writing Selected Rungs to a File .....	3-80
Including Rungs from a File .....	3-81
Open Space Functions .....	3-82
Move Logic Right .....	3-83
Move Logic Down .....	3-84
Delete Column .....	3-85
Delete Row .....	3-86
Delete Instruction .....	3-87
Increment/Decrement Reference Address .....	3-88
Incrementing/Decrementing within a Rung .....	3-89
Auto-Next Highest Reference .....	3-90
<b>Section 10: Editor Options</b> .....	<b>3-92</b>
Coil Checking .....	3-92
Automatically Inserting References .....	3-96
<b>Section 11: Search Function</b> .....	<b>3-98</b>
Using Search and Replace .....	3-101
Implicit Search .....	3-103
Search by Reference Type .....	3-103
Quick Search for a Coil .....	3-103

	<b>Section 12: Online Editing/Monitoring</b> .....	<b>3-104</b>
	Inserting or Editing Rungs (Block Edit) .....	3-104
	Block Edit (“Bumpless”) Run Mode Store Function (6.61 or Later CPUs)	
	.....	3-104
	Substitutions .....	3-105
	Modifying Instructions .....	3-110
	Using the Modify Softkey .....	3-111
	Modifying a Reference Address or Constant .....	3-111
	Forcing and Overriding Discrete References .....	3-113
	Changing Register Values .....	3-114
<b>Chapter 4</b>	<b>Reference Tables</b> .....	<b>4-1</b>
	<b>Section 1: Displaying Reference Tables</b> .....	<b>4-2</b>
	Displaying a Reference Table .....	4-3
	Using the Cursor to Select a Reference Table .....	4-3
	Moving the Cursor in a Reference Table .....	4-3
	<b>Section 2: Changing Reference Table Values</b> .....	<b>4-4</b>
	Changing a Register Reference .....	4-4
	ASCII String Entry .....	4-5
	Changing the Reference Display Mode (ALT-N) .....	4-7
	Forcing a Discrete Reference .....	4-8
	Changing the Values of a Word of Discrete References .....	4-8
	<b>Section 3: Overrides</b> .....	<b>4-9</b>
	Using Overrides .....	4-10
	Removing Overrides .....	4-10
	<b>Section 4: Changing Display Formats</b> .....	<b>4-11</b>
	Discrete Reference Tables .....	4-11
	Register Reference Tables .....	4-12
	Display Formats .....	4-13
	Changing the Display Format .....	4-15
	Changing the Format of a Table .....	4-15
	Timer/Counter Format .....	4-17
	Returning to Default Values .....	4-18
	System Reference Table .....	4-19
	<b>Section 5: Mixed Reference Tables</b> .....	<b>4-20</b>
	Editing the Title .....	4-21
	Defining a Mixed Table .....	4-22
	Deleting a Line .....	4-23
	Moving the Cursor .....	4-24
	Timer/Counter Format .....	4-24

<b>Chapter 5</b>	<b>PLC Control and Status</b> .....	<b>5-1</b>
	Run/Stop the PLC .....	5-2
	PLC Password Protection .....	5-3
	Changing the Privilege Level .....	5-4
	Creating, Changing, or Removing Passwords .....	5-5
	Enabling/Disabling Passwords .....	5-6
	OEM Protection .....	5-7
	PLC Fault Table .....	5-8
	Fault Table Entries .....	5-9
	Number of Faults in the PLC Fault Table .....	5-10
	Zooming into the PLC Fault Table .....	5-11
	Clearing the Fault Table .....	5-11
	I/O Fault Table .....	5-12
	Fault Table Entries .....	5-13
	Number of Faults in the I/O Fault Table .....	5-13
	Zooming into the I/O Fault Table .....	5-14
	Clearing the Fault Table .....	5-15
	PLC Memory Used .....	5-16
	Block Memory Usage .....	5-17
	Configured Reference Sizes .....	5-18
	PLC CPU Sweep Control .....	5-19
	Active Constant Sweep Mode Setting .....	5-20
<b>Chapter 6</b>	<b>Programmer Setup</b> .....	<b>6-1</b>
	<b>Section 1: Programmer Operating Mode</b> .....	<b>6-2</b>
	Mode Selection .....	6-2

<b>Section 2: Selecting SNP Connections</b> .....	<b>6-3</b>
<b>Section 3: PLC Communications Serial Port Setup</b> .....	<b>6-4</b>
Standard Serial COM Port .....	6-4
Requirements .....	6-5
Port Requirements .....	6-5
Configuring Memory for the Communications Driver .....	6-6
MS-DOS Memory Areas Defined .....	6-7
Communications Driver Load Order .....	6-8
Running Logicmaster 90 Software .....	6-9
Setting Up the Computer's Serial Port .....	6-10
Setting Up the Port .....	6-11
Displaying Port Settings .....	6-11
Saving the Port Setup .....	6-12
Displaying File Settings .....	6-12
Port Usage Conflicts .....	6-12
Memory Manager Specifications .....	6-13
Using High Memory Area on an 80386 (Workmaster II) Or Higher Computer .....	6-14
Using Upper Memory Block on an 80386 (Workmaster II) Or Higher Computer .....	6-15
Using Conventional Memory on an 80386 (Workmaster II) Or Higher Computer .....	6-16
Using Expanded Memory on an 80386 (Workmaster II) Or Higher Computer .....	6-16
Using Video RAM on an 80386 (Workmaster II) or Higher Computer ..	6-17
Tested Configurations .....	6-18
TCP/IP Communications .....	6-18
<b>Section 4: WSI Serial Port Setup</b> .....	<b>6-19</b>
Setting Up a Port .....	6-20
Displaying Port Settings .....	6-20
Saving the Port Setup .....	6-20
Displaying File Settings .....	6-20
<b>Section 5: View Modes Setup (ALT-N)</b> .....	<b>6-21</b>
Changing the View Mode .....	6-22



	<b>Section 6: Serial Printer Setup .....</b>	<b>6-23</b>
<b>Chapter 7</b>	<b>Program Folders .....</b>	<b>7-1</b>
	Drawer .....	7-1
	Program Folder Names .....	7-2
	TEMP Program Folder .....	7-2
	Using Program Folder Functions .....	7-2
	Selecting/Creating Program Folders .....	7-4
	Selecting a Previously Created Folder with a New 351 or 352 (or Later) Configuration .....	7-4
	Auto-Select Function .....	7-5
	Automatic Folder Selection .....	7-6
	Deleting Program Folders .....	7-7
	Backing Up Program Folders .....	7-8
	Restoring Program Folders .....	7-9
	Renaming Program Folders .....	7-11
	Clearing Program Folders .....	7-12
	Locking/Unlocking Program Folders .....	7-13
	Copying Program Folders .....	7-14
	Copying to Another Folder on the Hard Disk .....	7-14
	Copying to a Diskette .....	7-15
<b>Chapter 8</b>	<b>Program Utilities .....</b>	<b>8-1</b>
	Loading from PLC to Programmer .....	8-3
	Loading Reference Tables from the PLC to Save in the Computer Folder .....	8-5
	Storing to PLC from Programmer .....	8-5
	Run Mode Store Function .....	8-8
	Verifying a Program with the PLC .....	8-9
	Clearing PLC Memory .....	8-11
	Clear Reference Override Tables .....	8-12
	Special Considerations for Clearing Reference Overrides .....	8-12
	Read/Write/Verify EEPROM/Flash .....	8-13
<b>Chapter 9</b>	<b>Print Functions .....</b>	<b>9-1</b>
	Print Functions Menu .....	9-2
	<b>Section 1: Printer Parameters .....</b>	<b>9-3</b>
	Changing Setup Printer Parameters .....	9-3
	Printer Parameters .....	9-4
	<b>Section 2: Selecting a Screen Print Device .....</b>	<b>9-5</b>
	Sending Screen Prints to a Printer .....	9-5
	Sending Screens to a File .....	9-6

<b>Section 3: Print Program</b> .....	<b>9-7</b>
Print Program Parameters .....	9-8
Shortcuts for Printing Program Logic .....	9-9
Cross References .....	9-10
<b>Section 4: Print Reference Tables</b> .....	<b>9-14</b>
<b>Section 5: Print Coil References</b> .....	<b>9-16</b>
<b>Section 6: Print Configuration</b> .....	<b>9-18</b>
Accessing the Print Configuration Screen .....	9-19
Pagination Guidelines .....	9-20
<b>Section 7: Print Function Examples</b> .....	<b>9-21</b>
<b>Chapter 10 I/O Configuration</b> .....	<b>10-1</b>
<b>Section 1: Default Configuration</b> .....	<b>10-2</b>
<b>Section 2: Displaying the I/O Configuration Rack Screen</b> ....	<b>10-4</b>
Configuration Validation .....	10-6
Copying Configuration from Slot to Slot .....	10-6
Changing the Configuration of a Slot .....	10-7
Replacing the Module with Another of the Same Type .....	10-7
Replacing the Module with One of a Different Type .....	10-7
Deleting the Configuration of a Module .....	10-7
Moving a Module to Another Slot .....	10-8
Saving the Configuration to Disk .....	10-8
<b>Section 3: Configuring the CPU</b> .....	<b>10-9</b>
Selecting a Different CPU Module .....	10-13
Run/Stop Switch Configuration (350 and higher CPUs) .....	10-17
Memory Protection Configuration for 350 and Higher CPUs .....	10-18
Port Configuration for 351, 352, and 363 CPUs .....	10-18
TCP/IP Ethernet Configuration on 364 CPUs .....	10-21
Ethernet Parameters for the 364 CPU .....	10-21
User-Configurable Memory Limits for 351 and Higher CPUs .....	10-24
<b>Section 4: Selecting the Base Rack</b> .....	<b>10-26</b>
Base Rack Power Supply .....	10-29
Selecting an Expansion Rack .....	10-29
<b>Section 5: Configuring the Model 211 CPU</b> .....	<b>10-31</b>
I/O Base Selection for the CPU 211 .....	10-33

<b>Section 6: Configuring a Micro PLC</b> .....	<b>10-36</b>
Expansion Units for the Micro PLC .....	10-42
Micro PLC Memory Configuration Limits .....	10-43
Configuration Rules and Miscellaneous Information .....	10-44
<b>Section 7: Configuring 90-30 I/O Modules</b> .....	<b>10-45</b>
Configuring Generic I/O Modules .....	10-47
Configuring an I/O Link Interface Module .....	10-48
Configuring an I/O Link Master Module .....	10-49
<b>Section 8: Configuring an HSC or Embedded HSC</b> .....	<b>10-50</b>
Configuring the Embedded HSC for a CPU 211 .....	10-56
<b>Section 9: Configuring a PCM Module</b> .....	<b>10-57</b>
Configuring a PCM .....	10-58
Selecting the Configuration Mode .....	10-59
PCM CFG Mode .....	10-59
PROG PRT Mode .....	10-60
CCM ONLY Mode .....	10-61
PROG/CCM Mode .....	10-62
CCM/PROG Mode .....	10-62
BASIC Mode .....	10-62
BAS/CCM Mode .....	10-62
<b>Section 10: Configuring a TCP/IP Ethernet Module</b> .....	<b>10-63</b>
Configuring a TCP/IP Ethernet Module .....	10-63
<b>Section 11: Configuring a CMM Module</b> .....	<b>10-67</b>
Configuring a CMM .....	10-67
Selecting the Configuration Mode .....	10-68
CCM ONLY Mode .....	10-69
RTU ONLY Mode .....	10-70
CCM/RTU Mode .....	10-70
RTU/CCM Mode .....	10-70
SNP ONLY Mode .....	10-71
SNP/CCM Mode .....	10-72
CCM/SNP Mode .....	10-72
SNP/RTU Mode .....	10-72
RTU/SNP Mode .....	10-72
<b>Section 12: Configuring a Motion Product                   (Formerly Configuring an APM)</b> .....	<b>10-73</b>
Configuring a Single Axis Motion Mate APM .....	10-74

<b>Section 13: Configuring an ADC Module</b> .....	<b>10-82</b>
<b>Section 14: Configuring a GCM or Enhanced GCM</b> .....	<b>10-83</b>
Enhanced Genius Communications Module .....	10-85
<b>Section 15: Configuring a Genius Bus Controller</b> .....	<b>10-88</b>
<b>Section 16: Configuring a High Density Analog Output Module</b>	<b>10-90</b>
Other Configuration Considerations .....	10-92
<b>Section 17: Configuring an Analog Combo Module</b> .....	<b>10-94</b>
Other Configuration Considerations .....	10-95
Analog Combo Configuration Parameters .....	10-96
<b>Section 18: Configuring a Third-Party Module</b> .....	<b>10-99</b>
<b>Section 19: Configuration Reference View</b> .....	<b>10-101</b>
Displaying the Reference View Table .....	10-102
Moving the Cursor .....	10-103
Displaying the Detail Screen .....	10-103
Displaying the Rack Screen .....	10-104
Overlapping References .....	10-105
<b>Chapter 11 CPU Configuration</b> .....	<b>11-1</b>
Storing the CPU Configuration to the PLC .....	11-1
PLC Date and Time .....	11-2
Changing the PLC Date and Time .....	11-3
SNP ID .....	11-4
Changing the SNP ID Name .....	11-4
PLC Memory Limits .....	11-5
Configurable Memory .....	11-5
<b>Appendix A Programming Lesson</b> .....	<b>A-1</b>
Help Screens .....	A-1
Starting the Lesson .....	A-1
Exiting the Programmer .....	A-1
Creating a Program Folder .....	A-2
Creating a Program .....	A-4
Entering a Variable Declaration .....	A-5
Adding Ladder Logic to the Program .....	A-7
Creating a Subroutine Block .....	A-16
Printing the Program .....	A-23
Ending the Lesson .....	A-24

<b>Appendix B</b>	<b>Configuration Lesson for the Series 90-30 PLC</b> .....	<b>B-1</b>
	Help Screens .....	B-1
	Starting the Lesson .....	B-1
	Creating a Program Folder .....	B-2
	Displaying the Rack Configuration .....	B-4
	Configuring a CPU Module .....	B-5
	Configuring a 90-30 I/O Module .....	B-7
	Configuring a Genius Communications Module .....	B-10
	Ending the Lesson .....	B-11
<b>Appendix C</b>	<b>Programmer Environment Setup</b> .....	<b>C-1</b>
	Displaying the Setup Screen .....	C-1
	Selecting Terminal and Printer Options .....	C-3
	Specifying Palette Colors .....	C-4
	Completing the Disk Drive Setup .....	C-5
	PLC Communications Options .....	C-6
	Standard Serial COM Port Installed .....	C-6
	Standard Serial COM Port and Driver Memory Area .....	C-6
	PLC Communications Options— Both WSI and Serial COM Port Installed .....	C-7
	PLC Communications— WSI Version with Configurable Interrupt Request .....	C-8
	PLC Communications— WSI Version with Non-Configurable Interrupt Request .....	C-9
	Saving the Programmer Environment Setup .....	C-9
<b>Appendix D</b>	<b>Instruction Mnemonics</b> .....	<b>D-1</b>
<b>Appendix E</b>	<b>Key Functions</b> .....	<b>E-1</b>
<b>Appendix F</b>	<b>User Command Menu</b> .....	<b>F-1</b>
	Accessing the User Command Menu .....	F-1
	Creating a COMENU.DAT file .....	F-2
	Example Comenu Definition File .....	F-3
<b>Appendix G</b>	<b>Files Created with Logicmaster 90-30/20/Micro Software</b> .....	<b>G-1</b>
	Files in the Program Folder .....	G-2
	Files in the Logicmaster 90-30/20/Micro Home Directory .....	G-4
	Cross Reference Data Files .....	G-4

<b>Appendix H</b>	<b>Common User Errors</b> .....	<b>H-1</b>
	Error Message: "File System Error" .....	H-1
	Error Message: "Read/Write PLC Initialization Aborted" .....	H-1
	Error Message: "Constant Out of Range" .....	H-1
	Error Message: "Invalid CPU" .....	H-1
	Error Message: "No Communications" .....	H-2
	Cannot Install the Software .....	H-2
	Error Message: "Comm Driver Not Loaded" .....	H-2
	Printer Output is Garbled .....	H-2
	Error Message: "Port/File Access Denied" .....	H-3
	System Software Error ID: 0000 EX: 0000 .....	H-3
	Busy Message Displayed After Load .....	H-3
	Error Message: "Error Detected in WSI Board" .....	H-3
	Trouble Communicating with the Series 90-30 or Series 90-20 PLC	
	Error Message: "Error Detected in WSI Board Port"	
	Error Message: "Error Loading Code into WSI Board" .....	H-4
<b>Appendix I</b>	<b>Variable Declaration Table Import/Export Using Comma Separated Variable (CSV) Format</b> .....	<b>I-1</b>
	Comma Separated Variable (CSV) Format and SNF Format ....	I-1
	General SNF Format Rules .....	I-1
	Header Section Rules .....	I-2
	Field Names Section Rules .....	I-2
	Data Section Rules .....	I-3
	The Header Section .....	I-3
	The Field Names Section .....	I-4

# Chapter 1

## Introduction

---

---

### Section 1: Product Overview

Logicmaster 90-30/20/Micro programming software is part of a family of products used to configure and program the full line of Series 90-30, Series 90-20, and Micro programmable controllers.

Configuration is the process of assigning logical addresses, as well as other characteristics, to the hardware modules in the system. It may be done either before or after programming, using the configuration software; however, it is recommended that configuration be done first.

Programming consists of creating an application program for a PLC. Because Series 90-30 PLCs, Series 90-20 PLCs, and Micro PLCs have a common instruction set, all can be programmed using this software.

#### Note

In this manual, the WSI version of Logicmaster 90-30/20/Micro software refers to using a Work Station Interface Board in the programmer to provide serial communication between the programmer and the attached PLC. The standard serial communications version of Logicmaster 90-30/20/Micro software refers to using ports COM1, COM2, COM3, or COM4 instead of the Work Station Interface Board.

## What You Will Need

To run Logicmaster 90-30/20/Micro software, you will need:

- A computer with a hard disk:
  - A Workmaster® II industrial computer with a 101-key keyboard, or
  - A personal computer with an Intel 80386 or higher processor and a minimum of 2 Megabytes of memory, or
  - A Zenith™ Mastersport™ SL notebook computer.
- At least 4 Megabytes of **free** disk space.
- Both the Logicmaster 90-30/20/Micro Release 5 WSI and the Standard Serial COM Port versions require a **minimum** of 520 KB (532,480 bytes) of available DOS application memory in order to run. The Standard Serial COM Port version requires either a **minimum** of 564 KB (577,536 bytes) of available DOS application memory, or 520 KB (532,480 bytes) of available DOS application memory *and* 42 KB of available High Memory Area, Upper Memory Block or Expanded Memory for the COM port driver (see page 6-6 and following for details about the Standard Serial COM Port version and memory management).

Both versions require a minimum 1024 KB of Lotus/Intel/Microsoft Expanded Memory (LIM EMS 3.2 or higher) for optimum performance. If additional DOS application memory (also called “low memory”) is needed, system software error ID: 0000 EX: 0000 will occur. Remove any unneeded TSR (Terminate and Stay Resident) programs and any unnecessary device drivers from the AUTOEXEC.BAT and CONFIG.SYS files to make more memory available. For additional information, see chapter 2, section 3, of this manual.

<sup>™</sup> Zenith and Mastersport are trademarks of Zenith Data Systems Corporation.  
<sup>®</sup> MS-DOS is a registered trademark of Microsoft Corporation.



## The MS-DOS Version Needed

To run Logicmaster 90-30/20/Micro software, MS-DOS® Version 5.0 (or higher) must be installed on your computer.

Logicmaster 90-30/20/Micro software provides foreign keyboard support, depending on the configuration of MS-DOS residing on the host computer. Consult your MS-DOS User's Manual for information on configuring for your country.

## Help Screens

Logicmaster 90-30/20/Micro software includes detailed Help screens. These Help screens are loaded onto the hard disk of your programmer during the software installation procedure and are readily accessible. To access the Help screens, press **ALT-H** for help, **ALT-I** for instruction mnemonic help, or **ALT-K** for key help.

## Key Functions

Appendix E, "Key Functions," lists the keyboard functions that are active in the Logicmaster 90-30/20/Micro software environment. Appendix E also contains a perforated Help card which can be removed from this manual. This information may also be displayed on the programmer screen by pressing **ALT-K** to access key help.

## Before You Begin

You will find Logicmaster 90-30/20/Micro programming software easy to use and to understand. When you are ready to begin, turn to chapter 2. It will tell you:

- How to install the software in your computer.
- How to start up the software.
- How to use your computer's keyboard to perform special programming functions.
- How to enter data, move the cursor, and read the status information on your screen.

After starting up the software, try the short programming lesson in appendix A and the configuration lesson in appendix B.

When you are ready to use the programming software to create a program, monitor a system, or perform any of its other functions, you will find instructions in other chapters of this book.

## Section 2: Configuration Software

### CPU Configuration

Configuration software is used to display and modify the characteristics of the CPU (such as memory allocation, PLC ID, and time of day). Section 3 of chapter 10 explains how to complete the CPU configuration for your system. This will set up or change the system features described below.

#### PLC Time and Date

The PLC maintains the current time and date. These settings can be displayed and changed using the CPU configuration function in Model 331 or higher CPUs. A time-of-day clock is not available in CPU Models 321, 323, 311, 313, 211, nor in Micro CPUs Models 004 and lower.

#### PLC Memory Allocation

You can display the current memory allocated to both discrete references and register references.

#### Fault Display and Clearing

When the programmer is monitoring a PLC, it displays faults that are stored in the PLC fault table. All faults are identified by time, date, and location. Faults can be cleared from the fault table display.

#### Note

CPU Models 321, 323, 311, 313, 211, and Micro CPUs Models 004 and lower do not support a time-of-day clock. Entries for CPU date and time are displayed as 00-00 00:00:00 in the fault tables.

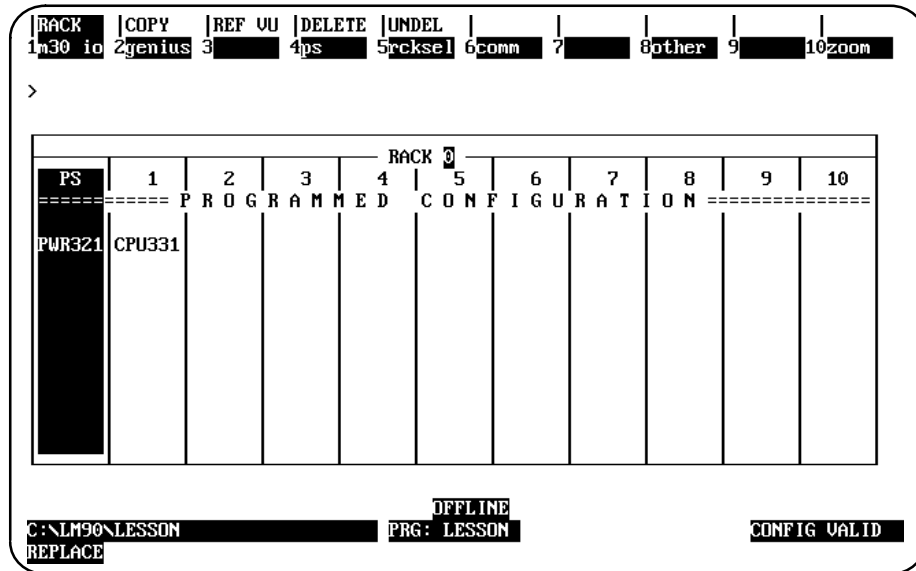
#### System Response to Faults

To assure safe operation of the control system, the PLC must be able to respond appropriately to certain types of faults. Fatal faults cause the CPU to set fault references and then go to **STOP** mode. Diagnostic faults cause the CPU to set fault references, but the PLC keeps operating.

## I/O Configuration

The I/O configuration function is used to describe the modules that are present in the PLC racks, to assign logical addresses, and select options for individual modules. These logical addresses are independent of physical location or function. Chapter 10 explains how to complete the I/O configuration for your system.

The I/O configuration rack screen represents the appearance of the Series 90-30 I/O rack. Use the Next and Previous page keys, or the Up and Down cursor keys, to display another rack. Then use the Left and Right cursor keys to move the cursor to the slot to be displayed or configured.



To complete the I/O configuration, you will:

1. Select the module present in each slot.
2. Assign each module a reference address. The configuration software automatically supplies the next highest reference address for each module; however, this address may be changed.
3. For some modules, you may also select options, such as the counter type for a High Speed Counter Module.

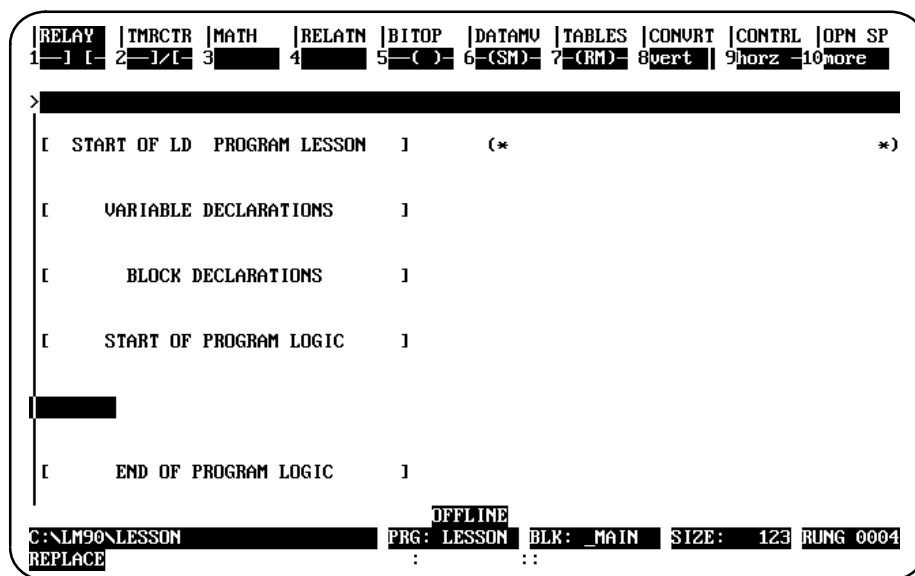
Editing features make it easy to copy, move, replace, delete, or undelete configurations.

## Section 3: Programming Software

### Creating or Editing a Program

Chapter, “Program Editing,” describes how to create and edit programs.

The basic elements of a program are shown when the programming screen is first selected:



Marker	Description
Variable Declarations	To access the variable declaration table, move the cursor to this marker and press Zoom (F10). Nicknames and reference descriptions can then be entered in the table.
Block Declarations	A program can include more than one block of logic. Additional blocks, known as subroutines, can be called from other blocks. When that is done, blocks must be declared before they are called.  The main block has a block declaration table. This table lists all blocks which are part of the complete program.  Blocks do not have block declaration tables. However, blocks can be called from the main block or from any block in the program.
Start/End of Program Logic	All logic is placed between these two markers. To enter logic, place the cursor on the [ END OF PROGRAM LOGIC ] marker and press Insert (F1).

Editing functions include rung insert/edit, select, cut, delete, paste, include, and write. In addition, search and goto functions are provided to position the cursor on a particular rung or element. An optional feature of the search function is the replacement of the search target with a user-specified element and/or reference address.

Chapter 3, section 4, “Program Annotation,” describes how annotation can be added to a program to make the program easier to read and understand. Three types of annotation (nicknames, reference descriptions, and rung comments) are supported by Logicmaster90-30/20/Microsoftware.

## Displaying Tables of Reference Values

Chapter 4, “Reference Tables,” explains how to use the reference tables feature to display the current values of program references. If the programmer is connected to a PLC and in **ONLINE** or **MONITOR** mode, the values shown in the table are from the PLC. In **OFFLINE** mode, they are from the current folder. There are separate tables for each type of program reference; for example, all discrete inputs (%I), all discrete outputs (%Q), and all registers (%R). In addition, there are 99 user-defined tables called mixed reference tables.

The format of individual items or an entire reference table can easily be changed to units that are suitable to your application. You can also return a standard reference table to its default format and fill the table locations with zero.

## Start/Stop PLC Execution

PLC program execution is started or stopped from the Run/Stop PLC screen, or by pressing ALT-R from any screen. For more information, refer to chapter 5, “PLC Control and Status.”

## Fault Display and Clearing

When the programmer computer is monitoring an operating PLC system, any faults that have occurred are displayed in one of two fault tables. PLC faults are listed in the PLC fault table. Faults from the I/O system are listed in the I/O fault table. All faults are identified by time, date, and location.

Additional information about each fault can be displayed by positioning the cursor on the fault in the fault table and pressing the Zoom (F10) softkey.

Faults can be cleared from the fault table displays. For information about the PLC and I/O fault tables, refer to chapter 5, “PLC Control and Status.”

## PLC Sweep Time Display and Change

Chapter 5, “PLC Control and Status,” explains how to use the programming software to:

- Display the current CPU sweep time. This is the amount of time required for one complete cycle of program execution, I/O scan, communications windows, and other functions.
- See whether or not **CONSTANT SWEEP TIME** mode is enabled, and read its setting.
- Display the time period of the watchdog timer. This timer is used to shut down the CPU if the sweep time is too long.
- Display the times currently allocated for the programmer window and the system communications window.

## PLC and Program Memory Information

Chapter 5, “PLC Control and Status,” also explains how to display information about:

- PLC memory size.
- PLC memory used and available.
- Program memory used for logic, data, declarations, and annotation.
- Memory used for configured program references.

## Selecting the Programmer Operating Mode

Both the programmer and the configuration software operate in three modes: **OFFLINE**, **MONITOR**, and **ONLINE**. In **OFFLINE** mode, no data transfer takes place between the computer and the PLC. Programs and configuration data may conveniently be developed in **OFFLINE** mode, with or without the computer connected to a PLC. In **MONITOR** mode, if communications have been established between the computer and the PLC, the computer can read data from the PLC but may not transfer data to it. With communications established in **ONLINE** mode, programs and other data can be transferred between the PLC and the computer.

If you are using a Workmaster or CIMSTAR I industrial computer, you may configure the Logicmaster 90-30/20/Micro software to use the keyswitch to select the operating mode. For those computers without a keyswitch, or if the keyswitch is not enabled, mode selection can be made by:

- Pressing the ALT and M keys simultaneously. Repeatedly pressing **ALT-M** switches the operating mode from **OFFLINE** to **MONITOR** to **ONLINE** and then back to **OFFLINE**.
- Going to the Programmer Setup screen (Shift-F7) and selecting an operating mode. For more information on using the Programmer Setup screen to select the operating mode, refer to chapter 6, “Programmer Setup.”

## System Security

Security for configuration and program functions consists of:

- A range of four privilege levels for the PLC, which may be protected using passwords.
- A software lock that can be applied to individual program folders.
- A selectable operating mode.
- A software lock that can be applied to individual subroutine blocks.

Passwords are a configurable feature of the Series 90-30, 90-20, and Micro PLC. Their purpose is to provide different levels of access privilege for the PLC when the programmer is in **ONLINE** mode. Passwords may not be set, or the password access (privilege) level may not be changed, when the programmer is in **OFFLINE** mode. The use of passwords is optional and may be set up using the status functions in the configurator. Passwords may be used to restrict changing I/O and PLC configuration data, changing programs, and clearing faults.

Chapter 3, section 8, “Subroutine Blocks,” describes how individual subroutine blocks may be locked and unlocked.

Chapter 5, “PLC Control and Status,” describes the programmer’s privilege levels and explains how to enter passwords. Chapter 5 also describes how passwords are specified.

Program folders can be protected to prevent accidental changes to program and configuration information. Each program folder can be either “locked” or “unlocked”. After a folder is locked, its contents cannot be changed or deleted. For more information on locking program folders, see chapter 7, “Program Folders.”

## Setting Up PLC Communications Using a WSI Board

For the WSI version of Logicmaster 90-30/20/Micro software, the serial port on the Work Station Interface (WSI) Board in the programmer provides serial communication between the programmer and the attached PLC. The COMSET serial port setup function is used to configure the WSI serial port, and to save or recall those configurations from disk files. Refer to chapter 6, “Programmer Setup,” for instructions on WSI serial port setup.

## Setting Up PLC Serial Communications for Standard Serial COM Ports

For the standard serial communications version of Logicmaster 90-30/20/Micro software, the COM1, COM2, COM3, or COM4 serial port may be set up to provide serial communications between the programmer and the attached PLC. The COMSET serial port setup is used to configure the COM1, COM2, COM3, or COM4 serial port, and to save or recall those configurations from disk files. Refer to chapter 6, “Programmer Setup,” for instructions on PLC communications serial port setup.

## Program Folders

Each program and the corresponding configuration is assigned to a subdirectory called a program folder. Both the configuration software and the programming software use a set of program utility functions to create and maintain program folders. Chapter 7, “Program Folders,” explains how to use the program folder functions to:

- Select another program folder.
- Create a new program folder.
- Delete an unneeded program folder.
- Rename a program folder.
- Make a backup copy of the current program folder.
- Restore a program folder with its backup copy.
- Clear the contents of the current program folder.
- Lock/unlock a program folder.
- Copy the contents of one program folder into the current program folder.

Application programs and related files can be stored on the same hard disk as the software. They can also be copied *using Logicmaster folder utilities* to floppy disks for portability or for independent storage.

## Transferring Programs

The program utility functions, described in chapter 8, “Program Utilities,” are used to:

- Transfer programs and configuration between the PLC and the programmer.
- Clear part or all of the program and/or configuration from PLC memory.
- Compare a program or configuration in the PLC with the current folder to determine if they are the same.
- Transfer a program, register (%R) table, and configuration between the PLC and EEPROM.
- Compare a program, register (%R) table, or configuration in the PLC with EEPROM.

## Printing Programs and Configuration

The programming software includes a complete set of print functions. You can select the contents and format of the printout. In addition, you can print cross references for the references used in the program.

You can also print copies of your system configuration by using the print function in the configuration software. Refer to chapter 9, section 5, “Print Configuration,” for information about printing configuration screens.



# Chapter 2

## Operation

---

---

This chapter explains what you will need to know to install and start up the Logicmaster 90-30/20/Micro software. It also explains keyboard use and content of the Logicmaster 90-30/20/Micro software screens.

Chapter 2 contains the following sections:

Section	Title	Description	Page
1	Hardware Setup (WSI Version)	Describes the two Work Station Interface Boards which are available. Section 1 also provides information on grounding and cabling.	2-2
2	Hardware Setup (Standard Serial Communications Version)	Describes the serial card which must be installed in your computer in order to use the standard serial communications version of Logicmaster 90-30/20/Micro software.	2-3
3	Software Installation	Explains how to install the Logicmaster 90-30/20/Micro programming and configuration software on your computer's hard disk.	2-4
4	Startup/Exit	Describes how to start up Logicmaster 90-30/20/Micro software.	2-16
5	Keyboard Functions	Describes keyboard functions in the Logicmaster 90-30/20/Micro software environment.	2-24
6	Screen Format	Shows the format of the display screen and describes the information that appears.	2-27

## Section 1: Hardware Setup (WSI Version)

### Installing the Work Station Interface Board

The Work Station Interface (WSI) Board provides a high-performance serial interface between a Series 90-30 or Series 90-20 PLC and the programmer for the WSI version of Logicmaster90-30/20/Microsoftware.

There are two Work Station Interface Boards available; one is for an AT-type computer, and the other is for a PS/2-type computer. The WSI Board is included as part of a package with Logicmaster 90-30/20/Micro programming software. When a Workmaster II computer is ordered as the programming device, the WSI Board is installed at the factory. For installation instructions on other programming devices, consult the computer manufacturer's instructions for option boards.

### Grounding

Be sure the computer has a ground connection in common with the CPU rack. This is usually done by connecting the programmer computer to the same power source (with the same ground reference point) as the rack.

#### Warning

**If the programmer is not connected as described above, damage to the Work Station Interface Board can occur. Erratic control operation may also result. If the programmer is online to an operating system, possible erratic operation may cause conditions which are hazardous to personnel and equipment.**

### Cabling

The communications cable that connects the computer to the PLC is a shielded twisted pair cable with a 3-pin D connector on each end. The maximum length for this cable is 50 feet. After installing the Work Station Interface Board in your computer, connect the computer to the PLC by first attaching the cable to the Work Station Interface Board and then to the port on the rack power supply.

---

## *Section 2: Hardware Setup*

### *(Standard Serial Communications Version)*

If you have the Logicmaster 90-30/20/Micro version of software which communicates with the PLC using the standard serial communication ports, a serial card with COM1, COM2, COM3, or COM4 must be installed in the computer in order to communicate with the PLC. Refer to GFK-0356, *Series 90-30 Programmable Controller User's Manual*, or GFK-0551, *Series 90-20 Programmable Controller User's Manual*, for instructions on establishing a serial connection between the Series 90-30, Series 90-20, or Micro PLC serial port and the serial port on the programming computer, without having a Work Station Interface Board installed in the computer.

## Section 3: Software Installation

To use Logicmaster 90-30/20/Micro software, it must be installed on the programmer computer's hard disk. MS-DOS Version 5.0 or higher must already exist on the hard disk.

The Logicmaster 90-30/20/Micro installation procedure creates three subdirectories on the hard disk:

Software Version	Subdirectories Created
WSI	\LM90 \LM90\P30 \LM90\C30
Standardserialcommunications	\LM90 \LM90\P30S \LM90\C30S

### AUTOEXEC.BAT and CONFIG.SYS Files

Before starting to install Logicmaster 90-30/20/Micro software, check the content of your hard disk root directory to see whether the files CONFIG.SYS and AUTOEXEC.BAT are present. Operation of Logicmaster 90-30/20/Micro software requires these files to be present. If they are not, they are installed automatically. If your hard disk already has these two files, you may be asked during installation whether the install process should modify them. If you prefer, you can edit your existing files for use with Logicmaster 90-30/20/Micro software.

The CONFIG.SYS (System Configuration) file is a short, readable file that describes the configuration of MS-DOS. Different software packages may use different system configurations. For all Logicmaster 90-30/20/Micro applications, the file must contain at least these two lines:

```
Files = 20
Buffers = 40
```

If you want to check the content of an existing CONFIG.SYS file, you can use the TYPE command. (Recommended buffer size is 40; minimum is 15.)

- The WSI version of Logicmaster 90-70 software requires a minimum of 520 KB (532,480) bytes of available MS-DOS application memory in order to run.

The Standard Serial COM Port version requires either a **minimum** of 564 KB (577,536 bytes) of available DOS application memory, or 520 KB (532,480 bytes) of available DOS application memory *and* 42 KB of available High Memory Area, Upper Memory Block or Expanded Memory for the COM port driver (see page 6-6 and following for details about the Standard Serial COM Port version and memory management).

Both versions require a minimum 1024 KBofLotus/Intel/MicrosoftExpanded Memory (LIM EMS 3.2 or higher) for optimum performance. If additional DOS application memory (also called "low memory" or "conventional memory") is needed, system software error ID: 0000 EX: 0000 will occur. Remove any unneeded

TSR (Terminate and Stay Resident) programs and any unnecessary device drivers from the AUTOEXEC.BAT and CONFIG.SYS files to make more memory available.

The following configurations will assist you in setting up your computer system to run Logicmaster 90-30/20/Micro software optimally on 386/486 computers with a minimum of 2 MBRAM.

LM90-30/20/Micro Software (Standard MS-DOS Version 5.0)	Configuration
CONFIG.SYS	DEVICE=C:\DOS\HIMEM.SYS DOS=HIGH,UMB DEVICE=C:\DOS\EMM386.EXE1024RAMX=CE00-CFFF FILES=20 BUFFERS=40

**Note**

The above EMM386.EXE switch "X=CE00-CFFF" is used to reserve this area of memory for the Series 90-30/20/Micro WSI card. If the computer does not have a WSI card, the "X= CE00- CFFF" is not required; however, the "EMM386.EXE 1024 RAM" specification enhances performance and may be necessary if you have over 537 MB of free hard disk space (see next paragraph below). If you have less than 537 MB of free hard disk space, the special precautions discussed below are unnecessary.

**If you have over 537 MB of free hard disk space and you have the statement DEVICE=C:\DOS\EMM386.EXENOEMS in your CONFIG.SYS, you will receive the following error: (00032) An unknown error occurred during install initialization.** To prevent this error from occurring, you must specify Expanded Memory in your CONFIG.SYS, e.g., DEVICE=C:\DOS\EMM386.EXE 1024 RAM. If you are using a commercial memory manager, such as QEMM™ or 386MAX™, this precaution may not be necessary. (See IPI GFK-0350 which comes with this Logicmaster release for examples of tested configurations using QEMM™ and 386MAX™.)

In addition, some folders may require additional memory. If additional memory is required, system software error ID: 0000 EX: 0000 will be displayed. Check the AUTOEXEC.BAT and CONFIG.SYS files to remove any device drivers and Terminate and Stay Resident (TSR) programs in order to free more RAM. Logicmaster 90-30/20/Micro software does not require the ANSI.SYS device driver.

The AUTOEXEC.BAT file must have (drive ID):\LM90 added to the existing path. (drive ID) is the letter which corresponds to the hard disk drive where Logicmaster 90-30/20/Micro software is installed.

If you need to run Version 4.01 or later of Logicmaster 90-30/20/Micro software from a disk that does not contain the software, add this to your AUTOEXEC.BAT file:

```
SET $PLCROOT=(drive ID):\LM90
```

™ 386MAX is a trademark of Qualitas, Inc.  
™ QEMM-386 and QRAM are trademarks of Quarterdeck Office Systems.

## Installing Logicmaster Under Windows® 95/98

You can run Logicmaster under Windows 95 or 98, but you cannot install it without rebooting into DOS mode. Although Windows 95 and Windows 98 support most standard DOS programs, the Logicmaster distribution diskettes contain a copy protection track that cannot be read using normal DOS file services. You must shut down the Windows 95/98 operating system and restart the computer in DOS mode before running the Logicmaster 90 Install program for both Logicmaster and SFC. Follow these steps for a successful installation.

- Click the Windows 95/98 Start button and select Shut Down.
- From the Shut Down dialog box, select the “Restart the computer in MS-DOS mode” option and press **Enter**.
- At the DOS prompt, insert the first Logicmaster distribution diskette and type **A:** (or **B:**), press **Enter**, then type **INSTALL**.
- Proceed with the standard installation.
- Configure your CONFIG.SYS and AUTOEXEC.BAT files as discussed previously.
- Restart your computer.

## Installing Logicmaster Under Windows NT™

You can run Logicmaster under Windows NT, but you cannot install it without going into DOS mode. To do so, restart your computer and select the option that takes you into standard DOS mode. Then follow the steps shown above for Windows 95/98 beginning with the third step.

If you do not have MS-DOS on your NT system, you can temporarily reboot from a floppy drive with an MS-DOS system disk if your NT system uses FAT-based file system.

### Note

Logicmaster 90-30TCP/IP Ethernet for DOS (IC641SWP313) cannot be run from a DOS window under Windows®, Windows 95/98, or Windows NT. For the other versions of Logicmaster, refer also to the IPI accompanying the software disks for additional information about installing Logicmaster.

There is a new version of Logicmaster (IC641SWP316) that was specifically designed as a DOS application that installs and works under Windows NT and Windows 95. This new version provides SNP or TCP/IP Ethernet communications to Series 90-70 PLCs and uses standard TCP/IP Windows drivers to provide maximum compatibility in a network environment. As a program designed for Windows NT and Windows 95, it consumes less of the system resources, easily handling the multitasking attributes of the Windows NT or Windows 95 operating system.

---

## Running Other Software with the CONFIG.SYS File for Logicmaster 90-30/20/Micro Software

Other types of software may require different entries in the CONFIG.SYS file. It is not always possible to combine the requirements for multiple software packages in one CONFIG.SYS file. In that case, you must maintain multiple versions of the CONFIG.SYS file with different file names, or if you are using MS-DOS version 6 or higher, you can create multiple boot options within one config.sys—see your DOS manual for information on this feature. Your MS-DOS manual contains other information about the CONFIG.SYS file that may be useful to you.

If you have loaded device drivers for special devices or a local area network, which terminate and stay resident, these programs may need to be removed before installing and running the Logicmaster 90-30/20/Micro software. The MS-DOS CHKDSK command may be used to determine available memory (bytes free).

Logicmaster 90-30/20/Micro software does not require the ANSI.SYS device driver; it may be removed in order to obtain more memory.

### Note

For MS-DOS version 6.0 or later, use the MEM /C /P command to display a list of all programs in memory and whether they are loaded in Conventional or Upper Memory. MS-DOS version 6 or later also includes the **MEMMAKER** command to optimize upper memory usage. Run it selecting Custom mode and select **YES** for options adding Upper Memory.

## Installation Instructions

1. Boot up the computer using MS-DOS.
2. Be sure CONFIG.SYS has files set to at least 20, i.e., FILES=20.
3. Remove the write-protect tab from the disk.
4. Insert the Logicmaster 90-30/20/Micro software disk into the computer's disk drive.
5. At the MS-DOS prompt, enter the designation of the disk drive followed by a colon. For example, if the disk is in drive A, type **A:** and press the **Enter** key.
6. Begin the installation procedure by typing **INSTALL** and pressing the **Enter** key. The following screen is displayed.

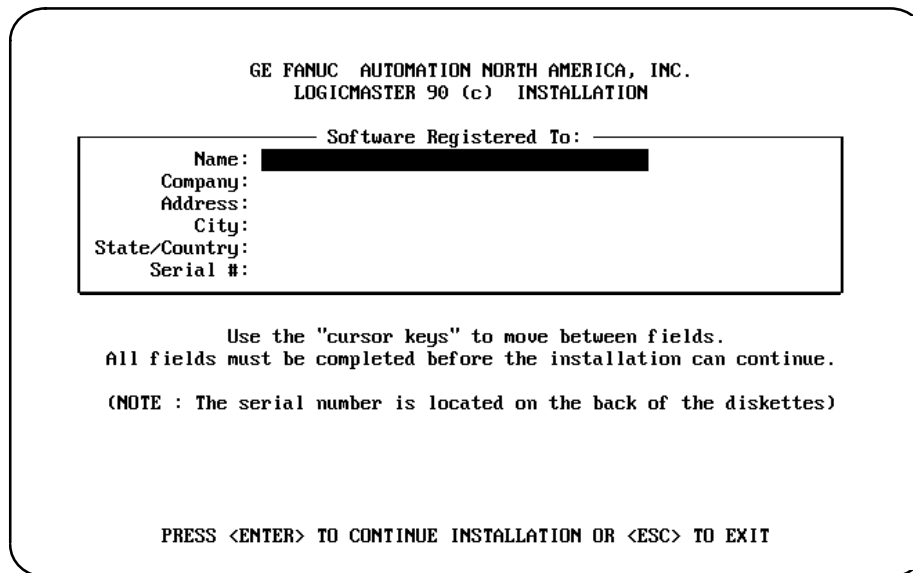
```
GE FANUC AUTOMATION NORTH AMERICA, INC.  
LOGICMASTER 90 (c) INSTALLATION  
  
The Logicmaster 90 installation process involves transferring files  
from one or more distribution diskettes to the hard disk on your  
computer. Please enter the destination drive (or use the default  
drive that is provided).  
  
DESTINATION DRIVE (Hard Disk) : C  
  
Press <ENTER> to accept selection or <ESC> to exit
```

### Note

If you are using Windows 95/98 or Windows NT as your operating system, you must restart your computer in (16-bit) DOS mode. Otherwise, an "ERROR: Disk Error, the target sector was not found" or "[+2553] Not Original Master Diskette" or a similar message will be displayed.



- 7. Specify the hard disk drive and press the **Enter** key, or just press the **Enter** key if the default is correct. The following screen is displayed if this is the first installation of this disk.



### Note

The *Serial Number* field must contain the serial number from your registration card or the back of the distribution disks.

- 8. Use the cursor keys to move between fields. Each field is validated as you move off the field. You must fill in all of the fields and press the **Enter** key in order for the installation process to continue. If any fields are empty or invalid when the **Enter** key is pressed, the first invalid field is highlighted and an error message is displayed. The system will prompt you to confirm that the registration information is correct as displayed.

If the registration data is not correct, press: **N** (No) and correct the registration information. When the registration data is correct, press: **Y** (Yes). The data is then encoded and written onto the master distribution disk. This screen is displayed until the registration data is successfully written to the master disk, or until the installation program is aborted by pressing the **Escape** key. You may print this screen by pressing the **Print Screen** key.

9. When the registration data is successfully written to the master disk, the installation process can begin. The following screen is displayed after the registration data has been entered, or on future Logicmaster installations of the same software package. (Registration is not required once the data is written to the master disk.) Read the licensing agreement.

```

                                GE FANUC AUTOMATION NORTH AMERICA, INC.
                                LOGICMASTER 90 (c) INSTALLATION

                                _____ Software Registered To: _____
                                Name: John Doe
                                Company: XYZ Manufacturing
                                Address: 121 Industrial Park Drive
                                City: Chicago                State/Country: IL
                                Serial #: 6123456789

                                COPYRIGHT (c) 1993 GE FANUC AUTOMATION NORTH AMERICA, INC.
                                Published in a limited, copyright sense and all rights,
                                including trade secret rights are reserved. Unauthorized
                                use of the information or program is strictly prohibited.

                                Installation of this software reaffirms acceptance of the
                                terms and conditions of the license agreement distributed
                                with this product.

                                PRESS <ENTER> TO CONTINUE INSTALLATION OR <ESC> TO EXIT

```

This information is displayed each time an installation is performed. Pressing the **Enter** key after reading this screen means you agree to comply with the stated terms. Press the **Escape** (**ESC**) key to terminate **INSTALL** and return to **MS-DOS**.

10. First, the Installation procedure creates the \LM90 directory in the root directory. INSTALL checks the files AUTOEXEC.BAT and CONFIG.SYS in the root directory of the hard disk. These files must contain certain commands to ensure that Logicmaster90-30/20/Microsoftwareexecutes properly. If neither file exists, they are both created. If either file already exists, INSTALL will ask if the files should be automatically modified.

```
Modifications to AUTOEXEC.BAT and CONFIG.SYS may be necessary for the
Logicmaster 90 software to execute properly.

AUTOEXEC.BAT - The path statement needs to be modified to
include a path to the Logicmaster 90 software.

CONFIG.SYS - The number of files and buffers may be adjusted
to match the Logicmaster 90 software requirements.

Should these changes be made automatically? (Y/N) 
```

If you want the AUTOEXEC.BAT and CONFIG.SYS files to be automatically modified, enter **Y** (Yes) or press the **Enter** key. If there were already versions of those files in the root directory, they will be renamed to AUTOEXEC.L90 and CONFIG.L90, respectively.

If you already have AUTOEXEC.BAT and/or CONFIG.SYS files and plan to edit them yourself, enter **N** (No). The following screen is displayed:

```
The following modifications to AUTOEXEC.BAT and CONFIG.SYS need to be
made for the Logicmaster 90 software to execute properly.

AUTOEXEC.BAT

Append the following subdirectory name to the existing path:
C:\LM90

CONFIG.SYS

Modify the maximum number of files and buffers to at least:
BUFFERS=15
FILES=20

Delete the following lines, if present:
DEVICE=GEXDISK.SYS
DEVICE=GEXM.SYS

CONFIRM : Should these changes be made automatically? (Y/N) 
```

11. If you change your mind and want INSTALL to automatically update the AUTOEXEC.BAT and CONFIG.SYS files, enter **Y** (Yes). Otherwise, enter **N** (No) or press the **Enter** key.

Installing Logicmaster 90 on C:

If this is a first time installation, all necessary subdirectories will be created and the files copied. If Logicmaster 90 already exists, it will be updated from the distribution diskette(s). This installation will take several minutes.

**WORKING**

12. INSTALL attempts to create three subdirectories under the root directory on the hard disk, and transfers the appropriate files to those subdirectories. If the subdirectories already exist, any files in them are deleted and the new files installed. If folders exist, they are not deleted.
13. INSTALL will then prompt you to insert any other disks. After all the files have been transferred, the final installation screen is displayed.
14. The computer must now be re-booted in order to complete the installation process. After removing the last disk used during the installation process, press **CTRL-ALT-Delete** to re-boot the computer. When the MS-DOS prompt is displayed, enter **LM90** to start up the software.

## Programmer Setup

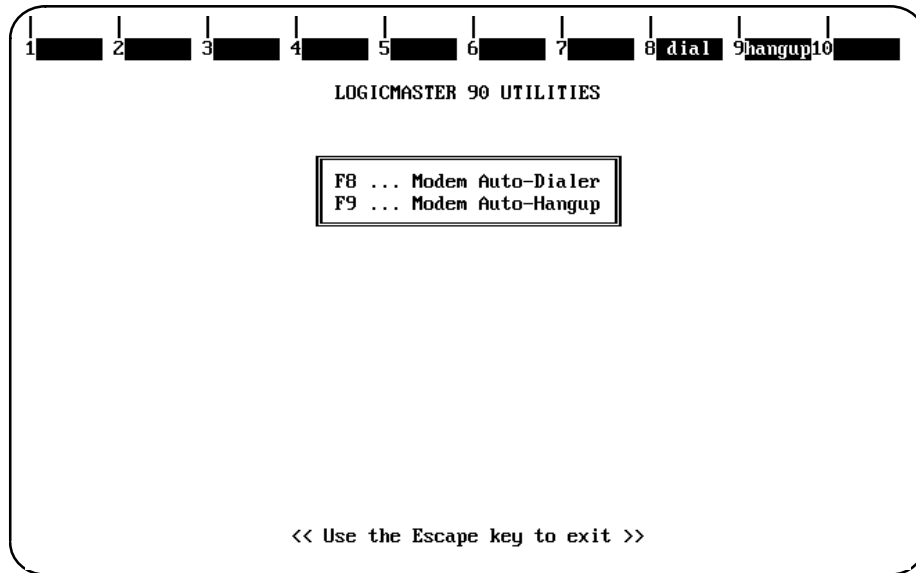
A default setup file is created during installation. Only in special circumstances will this file need to be changed. Refer to appendix C, "Programmer Environment Setup," for instructions on modifying the setup parameters.

## Using a Modem

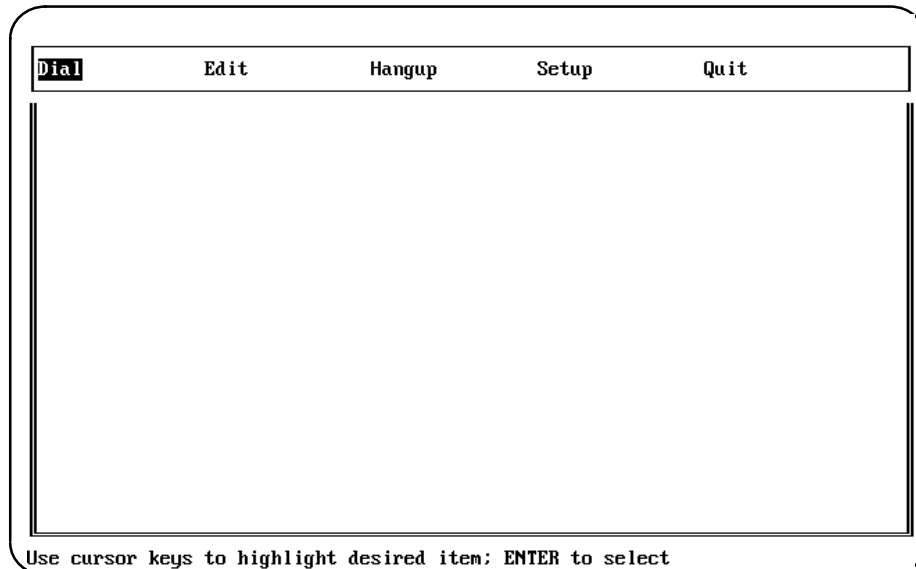
Logicmaster 90-30/20/Micro software has a modem auto dial feature which can be accessed from the main menu of Series 90 PLCs and functions by selecting Logicmaster 90 Utilities (F7).

### Note

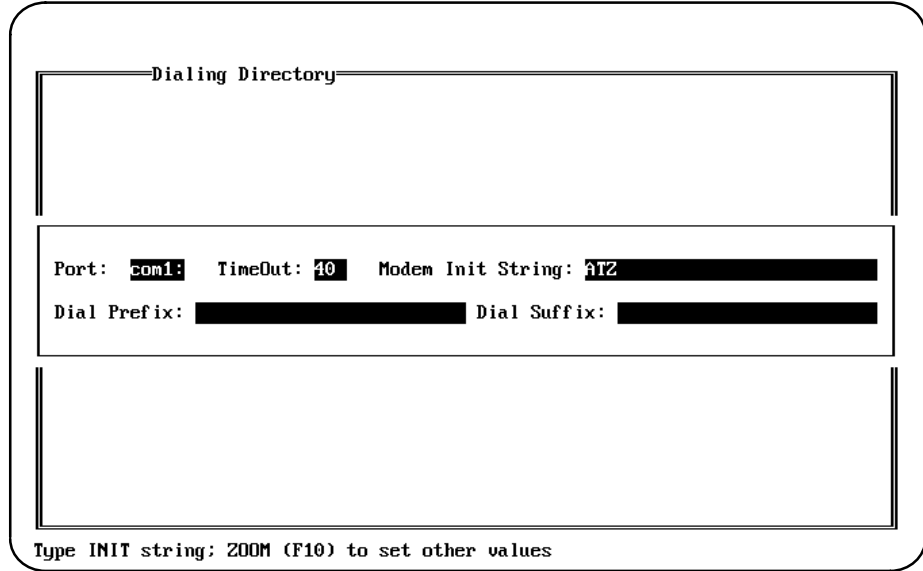
The Logicmaster 90-30/20/Micro modem auto dial feature only supports COM1 and COM2.



1. Press **F8** to select the modem auto dial feature.



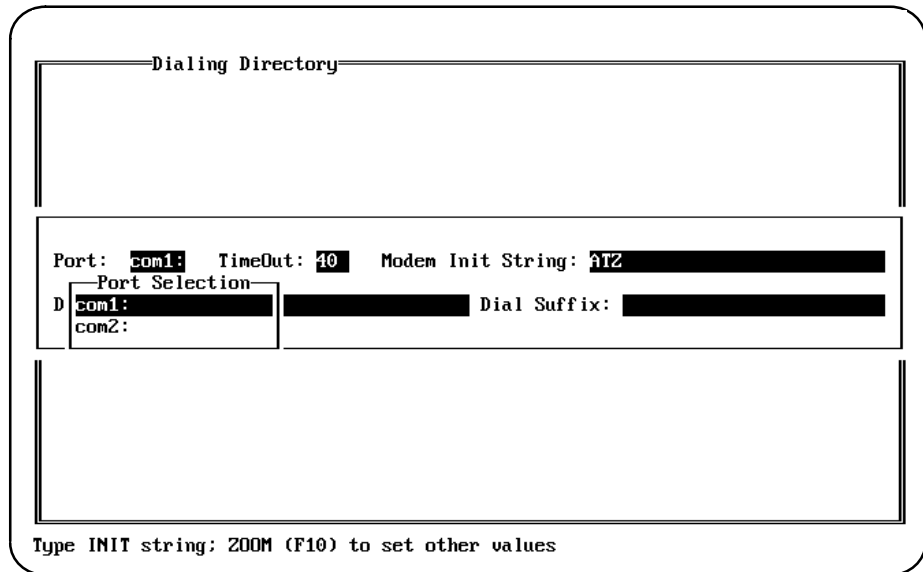
- 2. To set the modem parameters, move the cursor to *Setup* and press the **Enter** key.



- A. Complete the fields on the screen displayed, using the Enter key to move among the fields and the right/left cursor keys to move within each field. Press **Zoom** (F10) to display other values, as shown in the following screen.

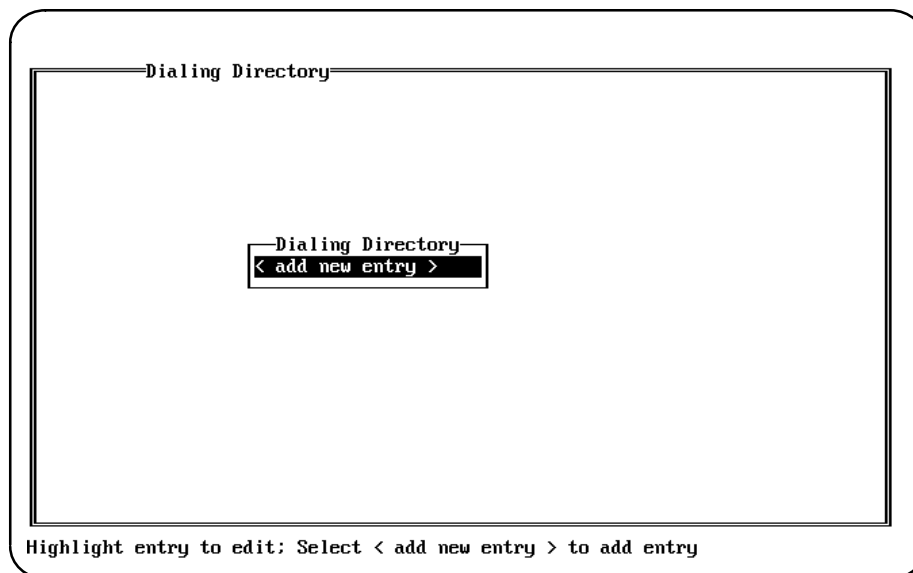
### Note

You can add any non-numerical characters that your modem accepts for the dial prefix or suffix, e.g., commas to add pauses (particularly useful when trying to connect to a number in a different country). See your modem user’s manual for acceptable non-numerical characters.



- B. Press the **Enter** key to select a parameter for each field. When all the fields are complete, press the **Escape** key.

3. To edit an entry or enter a new listing in the directory, move the cursor to *Edit* and press the **Enter** key.



- A. Highlight the entry you wish to edit, or select: *< add new entry >* and press the **Enter** key to add a new listing.
  - B. Complete the fields on the screen displayed, using the **Enter** key to move among the fields and the right/left cursor keys to move within each field. Press **Zoom (F10)** to display the choices for each parameter, cursor to one of the choices, and then press **F10** again or the **Enter** key to select the parameter. When all the fields are complete, press the **Escape** key.
4. To dial a number, press the **Enter** key with the cursor on *Dial*. Use the cursor keys to select an entry, and then press the **Enter** key to dial the number.
  5. Move the cursor to *Hangup* and press the **Enter** key to abort the call and hang up the modem.
  6. To exit the modem, cursor to *Quit* and press the **Enter** key, or press **ALT-F8**.

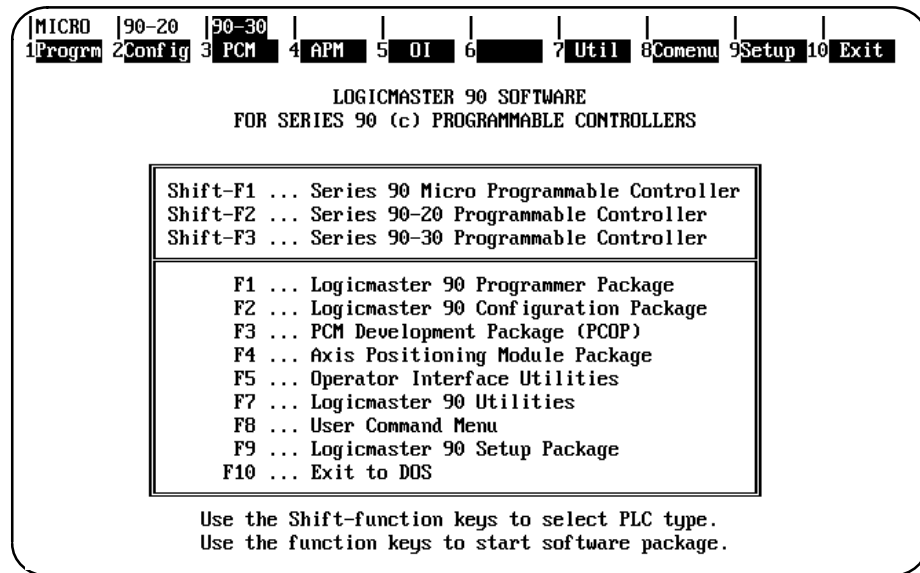
### Note

All computers with modems or add-in card or PCMCIA slot modems include communications software. You may choose to use one of these packages to dial and connect Logicmaster to a remote PLC, especially if you are using COM3 or COM4. If you use other communications software, set up the modem to stay connected (i.e., NOT to hang up the line) when exiting the software (so that you can start Logicmaster). Usually, this means forcing DTR high or telling the modem to ignore DTR. After exiting Logicmaster, you must re-enter the modem software package to send a command, such as the command to hang up the line.

## Section 4: Startup/Exit

To start up Logicmaster 90-30/20/Micro software:

1. At the MS-DOS prompt, type **LM90** and press the **Enter** key. The menu of Series 90 PLCs and functions is displayed.



### Note

Beginning with Release 3, a teach file can be used to enter the Logicmaster 90-30/20/Micro software. This entry teach file will contain the keystrokes previously taught while entering the programming or configurator software. Information on using a teach file to enter the Logicmaster 90-30/20/Micro software can be found in chapter 2, section 5, “Keyboard Functions.”

2. Use the function keys to select the programming, configuration, or PCM configuration software.
3. If only one version of Logicmaster 90-30/20/Micro software (either WSI or standard serial communications) is installed, it will be run automatically. If both versions are installed, the Logicmaster 90 Setup Package (F9) can be used to select the version to run. Refer to appendix C, “Programmer Environment Setup,” for instructions on modifying the programmer environment setup.



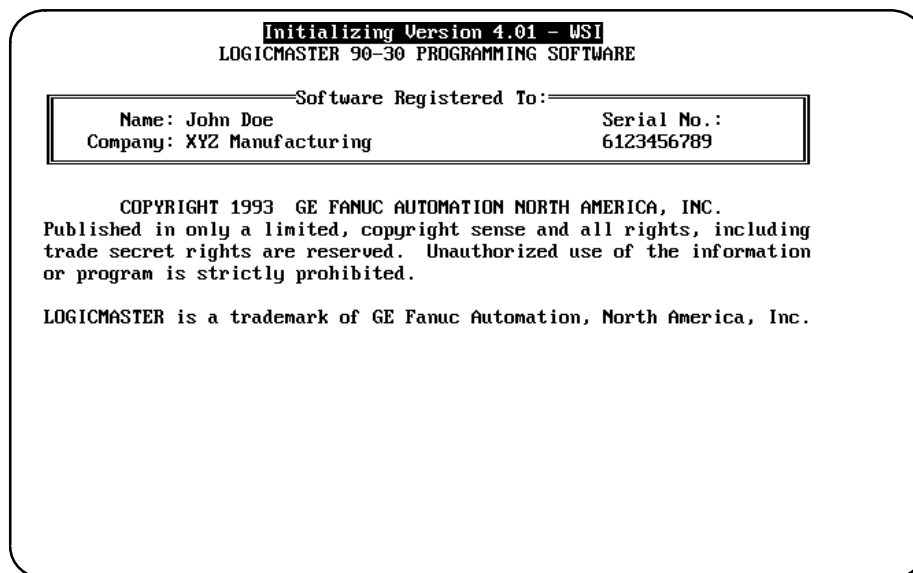
## Starting the Programming or Configuration Software

To load the programming software, select Program (F1) from the menu of Series 90 PLCs and functions, shown above. If you wish to load the configuration software, select Config (F2), or select PCM (F3) to load the PCM configuration software. (For information on the PCM development software package, refer to the *Series 90 PCM Development Software (PCOP) User's Manual*, GFK-0487.)

The Series 90-30, Series 90-20, and Micro PLCs use the same Logicmaster 90-30/20/Micro software package for programming. The default PLC type can be selected using the appropriate shift-function key sequence, as described below, but any of the three PLC types can be configured using the configuration package.

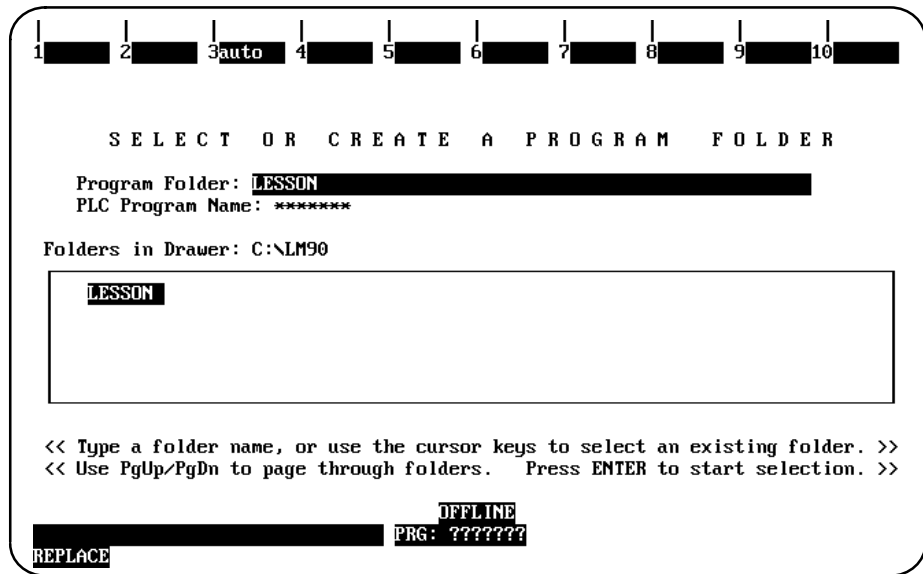
- If the configuration file does not already exist in the selected program folder and the Micro (Shift-F1) key is pressed before booting the Logicmaster software, the reference defaults and CPU will be that of a Micro CPU.
- If the configuration file does not already exist in the selected program folder and the 90-20 (Shift-F2) key is pressed before booting the Logicmaster software, the reference defaults and CPU will be that of a Model 211 CPU.
- If the configuration file does not already exist in the selected program folder and the 90-30 (Shift-F3) key is pressed before booting the Logicmaster software, the reference defaults and CPU will be that of a Model 331 CPU.

After selecting **Program (F1)** or **Config (F2)**, the following screen appears:



When the Logicmaster 90-30/20/Micro software powers up, it attempts to automatically select a program folder:

- If there is a program folder beneath the drawer last selected, whose name matches the program name in the attached PLC, that folder is selected.
- If the current directory is a program folder and its name matches the program name in the attached PLC, that folder is selected.
- If there is a program folder beneath the current directory, whose name matches the program name in the attached PLC, that folder is selected.
- If the current directory is a program folder, it is selected.
- Other wise, the initial select screen is displayed with the name of the last folder selected in the **Folder** field.



### Note

At serial baud rates of 1200 or less, the folder cannot be automatically selected during startup. Once in the programming software, however, you may use the auto function to automatically select a folder. (Refer to the information on “Automatic Folder Selection” in chapter 7, “Program Folders.”)

Enter a name of up to seven characters for the program folder. (The name is also used for the program.) After entering a name, press the **Enter** key. If you need more information about program folders, refer to chapter 7, “Program Folders.”

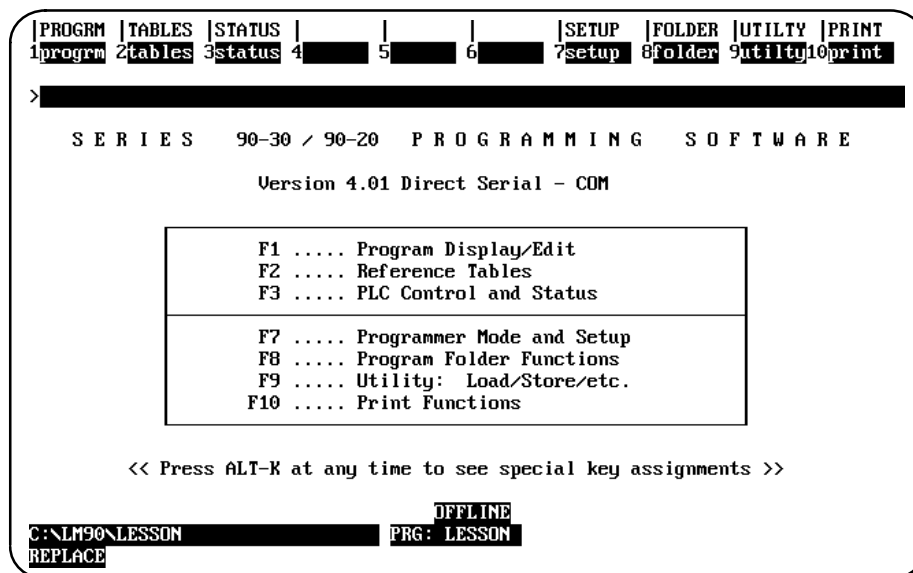
## Exiting (Quitting) the Programming or Configuration Software

To exit the software at any time, press **CTRL-Break** and confirm the prompt, or repeatedly press the **Escape** key at each prompt. You should always exit the software before turning the computer off; otherwise, some changes may be lost.

If you respond to the exit prompt (“Exit Logicmaster 90 Package? (Y/N)”) by entering a **1** instead of a **Y** (Yes) or **N** (No), a bookmark feature saves the current context. The next time you run the software, you will return to the same general location when you re-enter the software.

## Programming Software Main Menu

If you selected **Program (F1)** from the menu of Series 90 PLCs and functions, the Programming Software main menu shown below is displayed. This menu is used to access the primary functions of Logicmaster 90-30/20/Micro programming software.

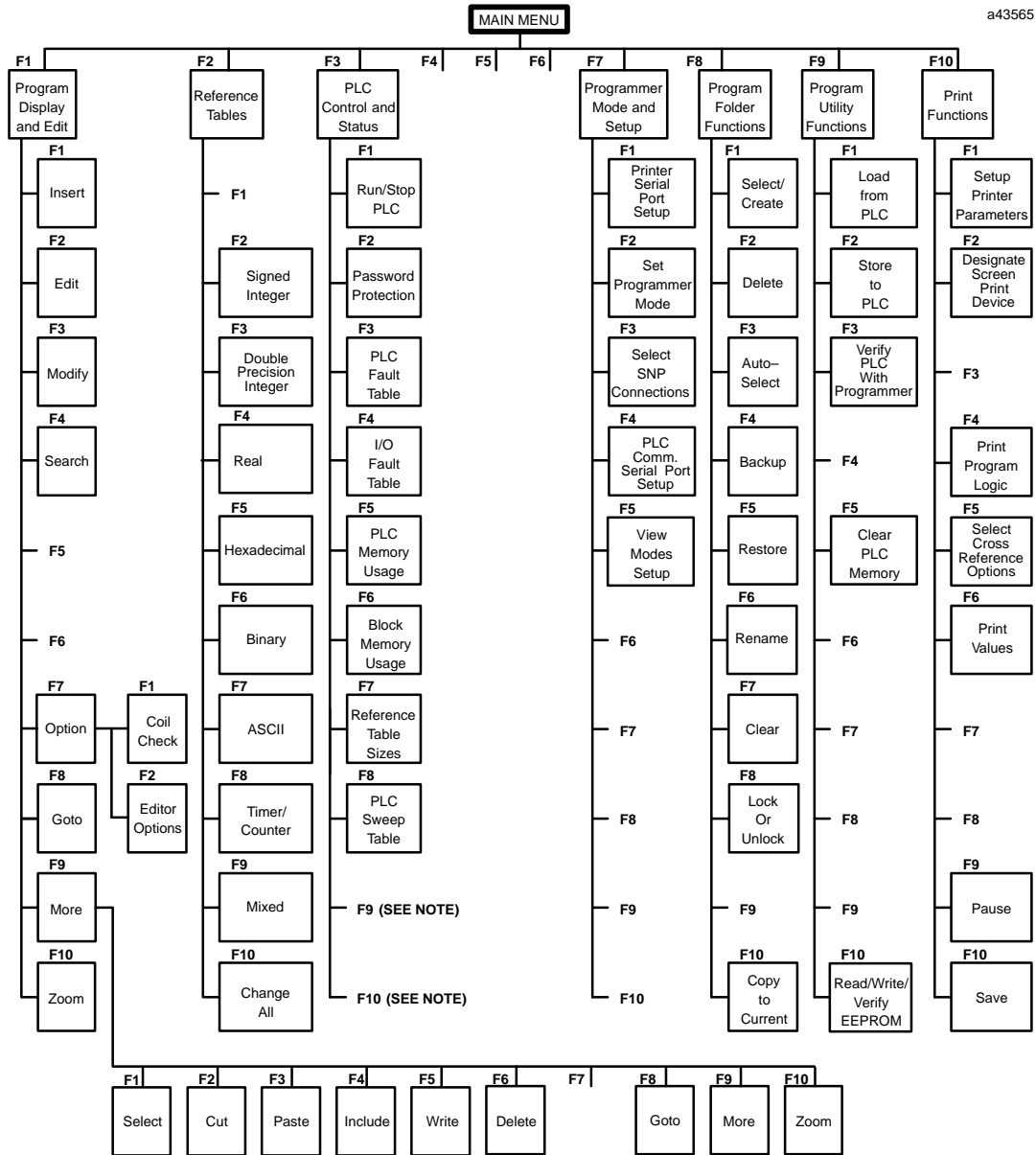


### Note

For the WSI version of Logicmaster 90-30/20/Micro software, **WSI** is displayed after the version number in the screen shown above. For the standard serial communications version of software, **COM** is displayed after the version number.

Function Key	Function	Description
F1	Program	Create or edit a program or monitor program logic. Chapter 3 describes the program display and explains how to create or edit a program.
F2	Tables	Display and change reference data. To select a particular reference table, enter the reference type on the command line before pressing <b>F2</b> . Chapter 4 describes monitoring data and explains how to change tables and variable data.
F3	Status	Select the status functions. These functions include displays of I/O faults and PLC faults. The memory configuration and the current PLC access level can also be viewed. Chapter 5 describes these functions.
F7	Setup	Display and change serial port setup and other programmer configuration. Chapter 6 describes these functions.
F8	Folder	Create, select, clear, rename, delete, lock, or back up a program folder. Chapter 7 explains how to use these functions.
F9	Utility	Load, store, or verify a program, or clear PLC memory. Chapter 8 explains how to use the utility functions.
F10	Print	Print a program folder. Chapter 9 describes the print functions.

The function keys remain active after selecting a programming function. You can go directly from one programming function to another without returning to the main menu by simultaneously pressing the **Shift** key and the desired function softkey. The bottom of the main menu screen contains three lines of status information. Chapter 2, section 6, "Screen Format," explains the content of the screen's status area.

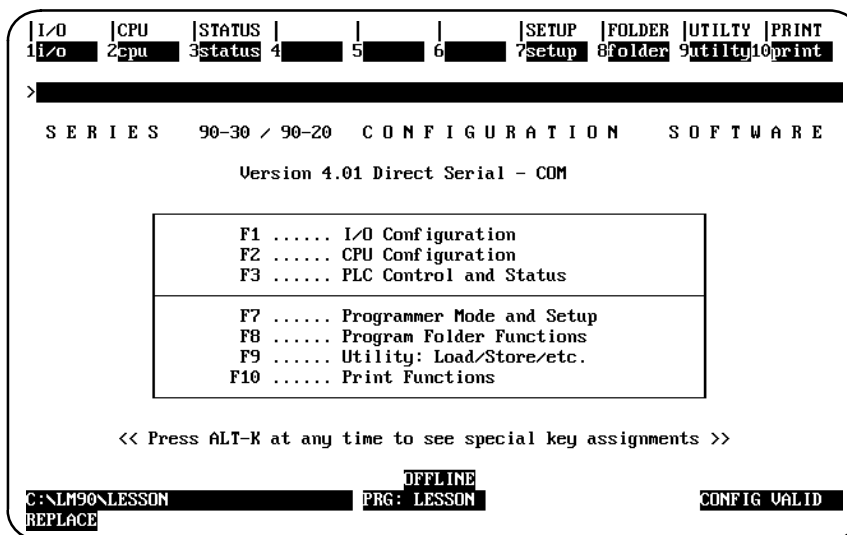


**NOTE:** Press Insert Rung (F1), Edit Rung (F2), or Modify Rung (F3) to display the programming instruction menus.  
 When Password Protection (F2) is pressed (PLC Control and Status functions), F9 will become the OEM softkey.  
 When PLC Fault Table (F3) or I/O Fault Table (F4) is pressed (PLC Control and Status functions), F9 will become the Clear softkey and F10 will become the ZOOM softkey.

Figure 2-1. Programming Software Menu Tree

## Configuration Software Main Menu

If you selected **Configuration (F2)** from the menu of Series 90 PLCs and functions, the Configuration Software main menu shown below is displayed. The menu lists the primary functions of the software. The first three menu entries are used to select PLC configuration functions. The remaining entries are used to select support functions.



### Note

For the WSI version of Logicmaster 90-30/20/Micro software, **WSI** is displayed after the version number in the screen shown above. For the standard serial communications version of software, **COM** is displayed after the version number.

Function Key	Function	Description
F1	I/O	I/O configuration is the process of describing to the software the content of a PLC system's rack(s). Chapter 10 tells how to complete I/O configuration.
F2	CPU	CPU configuration sets the operating characteristics of the CPU. Chapter 11 describes the CPU configuration.
F3	Status	The status function displays I/O faults and PLC faults. You will also use this function to display PLC information such as memory usage and to assign passwords. Chapter 5 describes these displays.
F7	Setup	The programmer setup function sets up the computer's serial ports, current operating mode, and PLC connection. Refer to chapter 6 for information on the setup function.
F8	Folder	Program folder functions are a group of file-handling utilities which you will use to create, delete, back up, and limit access to configuration and program files. Chapter 7 describes these functions.
F9	Utility	Program utilities are used to transfer information between the computer and the PLC. An additional program utility is used to clear memory in the PLC. Chapter 8 explains how to use the program utilities.
F10	Print	Print functions are used to generate configuration printouts and to set up a destination for screen prints. Chapter 9 describes these print functions.

The functions are selected using the function keys shown on the top line of the screen. You can go directly from one function to another without returning to the main menu by using the shift function keys.

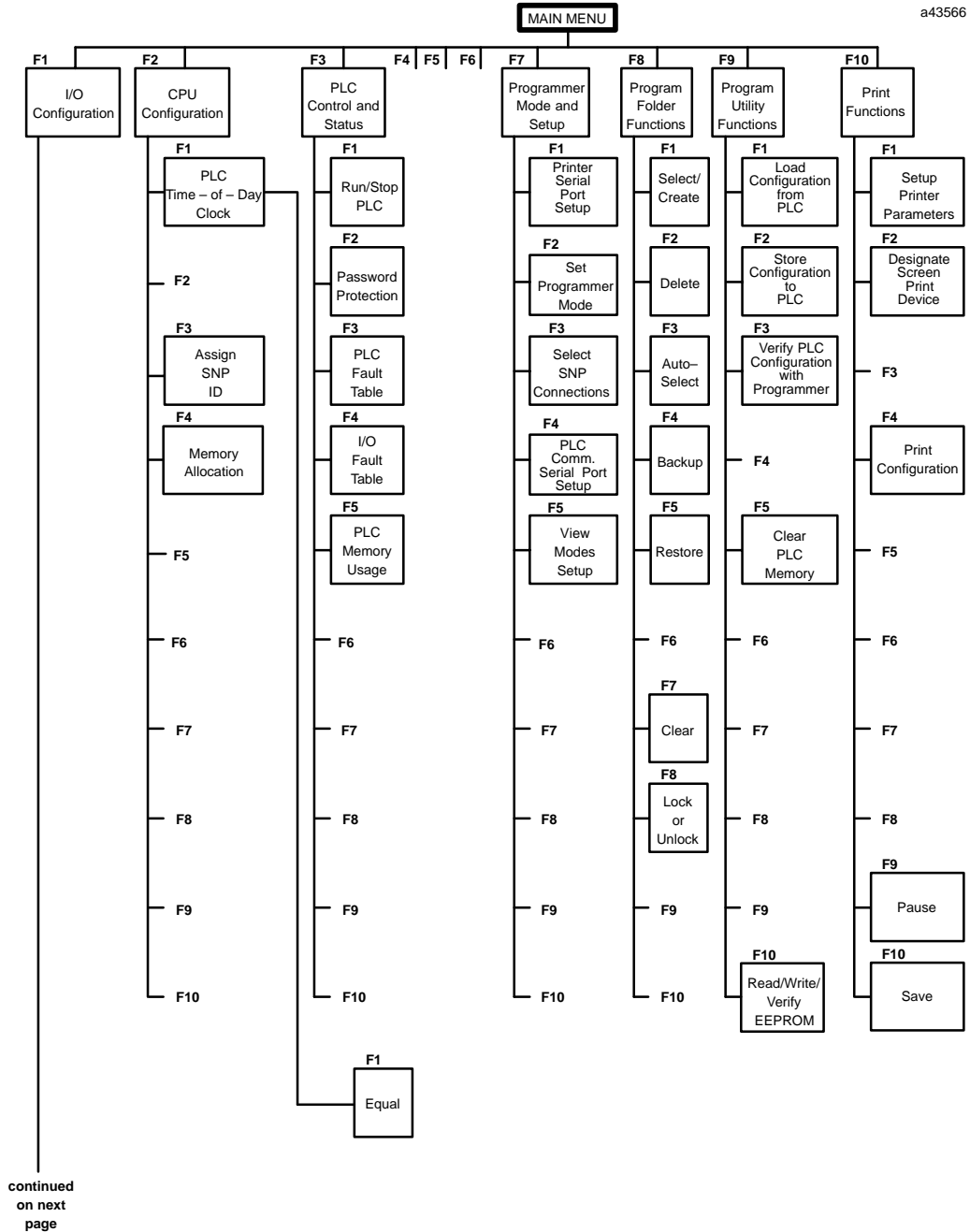


Figure 2-2. Configuration Software Menu Tree

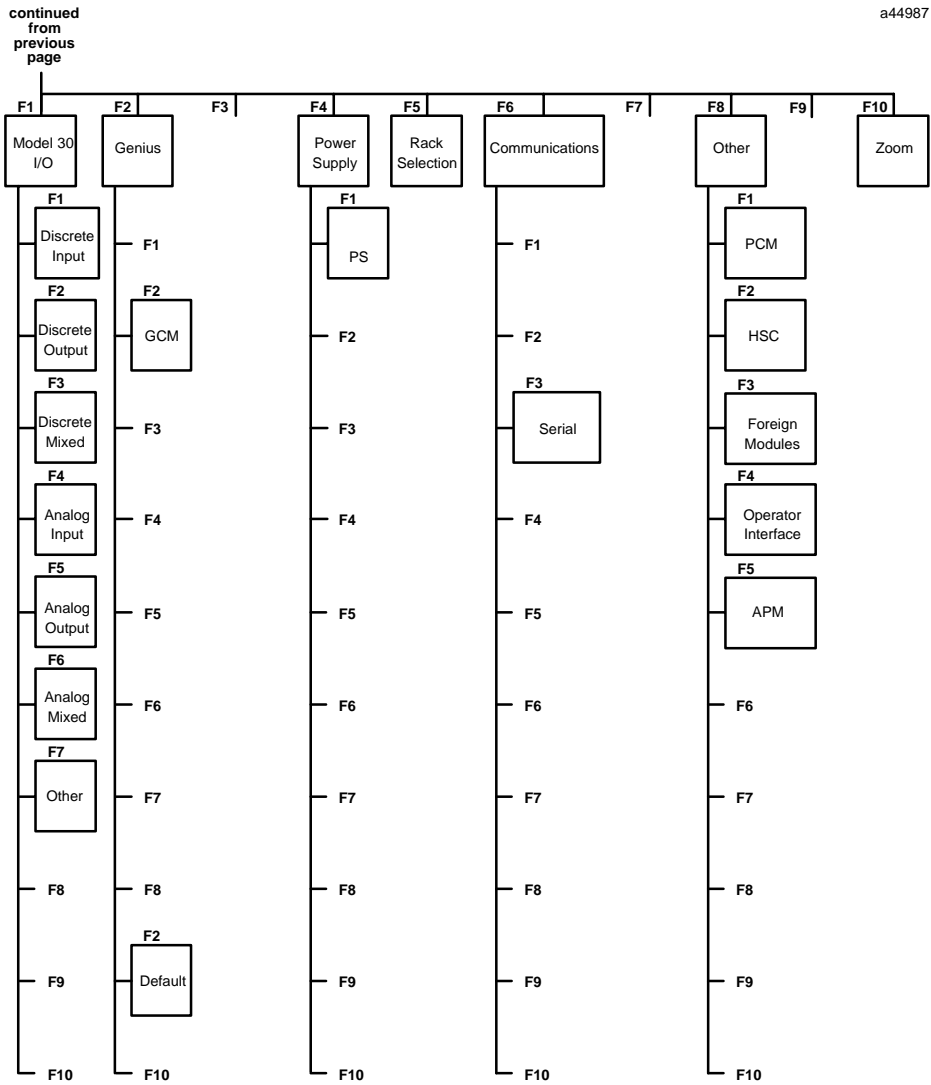


Figure 2-2. Configuration Software Menu Tree (cont'd)

## Section 5: Keyboard Functions

This section describes the keyboard functions that are active in the Logicmaster 90-30/20/Micro software environment. It also describes playback functions, which can be used to assign sequences of frequently-used keystrokes to a file for simple recall later on.

### Keyboards Supported

Logicmaster 90-30/20/Micro programming software can be used with the 83-key or 101-key keyboard for an IBM PC personal computer, or a Workmaster, Workmaster II, or CIMSTAR I industrial computer. Other types of keyboards for the IBM PC-XT, PC-AT, or IBM-compatible computer may work with Logicmaster 90-30/20/Micro software, but have not been tested. The 91-key keyboard for the Workmaster computer, developed for use with Logicmaster 6 programming software, would be difficult to use since Logicmaster 90-30/20/Micro software was designed to use with standard keyboards.

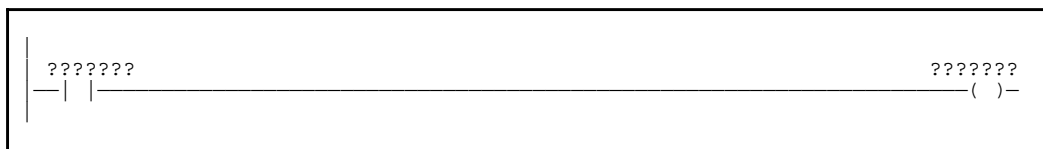
### Key Functions

Appendix E, “Key Functions,” lists the keyboard functions that are active in the Logicmaster 90-30/20/Micro software environment. Appendix E also contains a perforated Help card which can be removed from this manual. This information may also be displayed on the programmer screen by pressing **ALT-K** to access key help.

### Keyboard Macros (Teach Mode)

Logicmaster 90-30/20/Micro software can be instructed to record sequences of keystrokes and play them back with a single keystroke. These sequences of keystrokes are referred to as keystroke macros or teach sequences.

A combination of keystrokes might represent a series of frequently used functions or part of a program that you want to duplicate. An example of a simple keystroke macro would be the sequence of keystrokes needed to create the following line of logic:



The auto-next highest reference address function, described at the end of chapter 3, section 9, “Rung Edit,” provides another example of using a keystroke macro.

The keystrokes that make up a keystroke macro are stored in a file named KEYx.DEF, where x is a digit in the set 0 to 9 that you choose when you create the keystroke macro. This file is called a teach file; it is located in the current program folder.



## Creating a Teach Sequence

Follow the steps below to create a keystroke macro:

1. Select the starting point in Logicmaster 90-30/20/Micro software at which you wish your macro to begin. You must remember this starting point when you execute or play back the macro you are about to record. For example, if you are editing ladder diagram logic, you must ensure that a condition such as whether you were in **EDIT** mode when you started recording holds when you are playing back the macro.
2. It is often simplest to begin the macro from an easy-to-remember starting point such as the main menu, the Print menu, or the Edit Program commands. When you wish to begin recording keystrokes, press and hold the **ALT** key while pressing the **T** key. Then, select a playback file by pressing one of the key combinations between **ALT-0** and **ALT-9**. If you select a key combination that has been used previously in the current folder, the software will ask you if you want to overwrite it. Logicmaster 90-30/20/Micro software will now record all the keystrokes that you make.
3. When you have completed your sequence of keystrokes, press **ALT-Q** to exit **TEACH** mode. The software will stop recording keystrokes and create a teach file.

## Playing Back a Teach Sequence

Before replaying the keystroke macro, be sure that the Logicmaster 90-30/20/Micro software is in the state it was in when you started recording the macro. Then, press the **ALT-digit** combination you were prompted for when you started recording. The software will rapidly execute the keystrokes you recorded as though they were entered from the keyboard.

The only keyboard input recognized during playback is **ALT-A**, which causes the Logicmaster 90-30/20/Micro software to ask you if you want to stop the playback.

If an error condition is encountered during playback, an error message is displayed, playback is paused, and you will be prompted to continue or stop the key sequence.

## Pausing the Playback of a Teach Sequence

A pause may be entered in a keystroke macro. If you are recording a sequence of keystrokes and need to pause playback to examine something on the screen, press **ALT-Z** at the point where you would like playback to pause. This may be done as many times as you like while recording a macro. A message will be displayed telling you that a user prompt key (a pause) has been recorded.

When the Logicmaster 90-30/20/Micro software encounters the **ALT-Z** key combination during playback, it displays a prompt and waits until the **space bar** is pressed to continue.

## Using a Teach File to Run Software from an MS-DOS Batch File

Beginning with Release 3.02, Logicmaster 90-30/20/Micro software may be run using a teach file, which contains keystrokes previously taught while entering the Logicmaster 90-30/20/Micro programming or configuration software packages. This feature allows **LM90** to be invoked from an MS-DOS batch file, to play back keystrokes to perform desired function(s), and to return to the batch file to execute the next command. This can be helpful when executing multiple print requests.

The keystroke macro is stored as the text file <filename>.tch. The file name must identify either the programming or configuration software package. For example, prg9030.tch could be used as the file name for the programming software and cfg9030.tch could be used for the configurator software. The file extension must be **.tch**.

To begin recording keystroke macro, enter the executable path, at the MS-DOS prompt, which corresponds to the version of software installed on your computer:

Software Version	Executable Path
WSIprogrammer	c:\lm90\p30\prg9030
WSIconfigurator	c:\lm90\c30\cfg9030
Standardserialcommunicationsprogrammer	c:\lm90\p30s\prg9030
Standardserialcommunicationsconfigurator	c:\lm90\c30s\cfg9030

Enter the name of the teach file for this macro (e.g., prg9030.tch) and the letter **t** after the path. The **t** indicates **TEACH** mode. For example, to begin recording keystrokes to a teach file named prg9030.tch, enter:

```
C:\lm90\p30\prg9030 prg9030.tch t
```

To end the teach sequence, press **ALT-Q**, or **CTRL-Break** to end the teach sequence and exit the software.

To play back the keystroke macro, enter the path and teach file name, followed by the letter **p** for **PLAYBACK** mode. For example, to play back a macro for the programming software, enter:

```
C:\lm90\p30\prg9030 prg9030.tch p or c:\lm90\p30\prg9030 p
```

To initialize the communications driver for the standard COM port version of Logicmaster 90 software, the batch file must contain the line: **c:\lm90\ws130 i** before the command to create or play back the teach file. The line: **c:\lm90\ws130 r** must be added after the command to create or play back the teach file in order to remove the communications driver. For example:

```
c:\lm90\ws130 i
c:\lm90\p30s\prg9030 p
c:\lm90\ws130 r
```

## Section 6: Screen Format

This is a typical bit-oriented reference table display for Logicmaster 90-30/20/Micro software.

```

|PROGRM |TABLES |STATUS |      |      |      |SETUP |FOLDER |UTILITY |PRINT
1|       |2int  |3dint |4     |5hex  |6bin  |7ascii |8tmctr |9mixed |10hgall
>
                                INPUT STATUS
                                %I0001 DWELL_T
00001  00000000 00000000 00000000 00000000 00000000 00001000 00000000 00000000
00065  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00129  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00193  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00257  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00321  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00385  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00449  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

ID: A0001  RUN/OUT EN  22ms SCAN  ONLINE  L4 ACC: WRITE LOGIC  LOGIC EQUAL
C:\LM90\LESSON          PRG: LESSON
REPLACE                %I0001 : DWELL_T :: weld dwell timer

```

The main part of the screen shows menus, data, and other information related to the function you are currently using.

### Function Key Assignments

The top two lines of the screen show the functions that are currently available. The main functions displayed in the upper line are selected by pressing the **Shift** key with a function key, **F1** through **F10**.

The active function appears in a reverse-video block. In the following example, the active main function is reference tables.

```

|PROGRM |TABLES |STATUS |      |      |      |SETUP |FOLDER |UTILITY |PRINT
1|       |2int  |3dint |4     |5hex  |6bin  |7ascii |8tmctr |9mixed |10hgall

```

The lower line displays the secondary functions that can be selected with that main function key active. In the example above, they are the display formats that are available while using reference tables. These formats are selected by pressing a function key, **F1** through **F10**, without using the **Shift** key.

## Message Line

Directly under the function key area is the message line. Errors in command syntax, or those discovered while executing commands or selections, are displayed on the message line. Prompts for additional information required from the user are also displayed on the message line.

## Command Line

The command line, identified by the > prompt, displays typed entries, such as data to be used for the table being displayed. It is used to enter instructions, references, or comments.

## Status Information

The bottom three lines display information about the status of the programmer, the PLC, the program, and the keyboard. The information displayed will change depending on the programmer mode. For example:

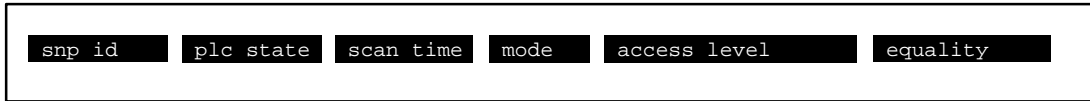
```

ID CONVEY  RUN/ENABLE  25msFIXED  ONLINE  L3 ACC: WRITE LOGIC  LOGIC NOT EQ
C:\ACME\CONVEYOR\CVLINE3  PRG: CVLINE3
REPLACE
  
```

Status Area	Description
Top line	Displays information about the attached PLC and the programmer.
Second line	Identifies the current program.
Third line	Shows the status of the keyboard; whether Caps Lock, Scroll Lock or Num Lock is active, and whether the keyboard is in <b>REPLACE</b> or <b>INSERT</b> mode. For some programming functions, the third status line displays additional information, described in later sections of this manual. Items appear in the status information area only when needed.

## PLC/Programmer Status: Definitions

The top line of the status area displays information about a PLC and about the programmer. Some of the information displayed reflects selections made using the PLC monitoring functions. For information about using these functions, refer to chapter 5, "PLC Control and Status."



Items that may appear on this line are explained in the following table. (Items will not appear on the status lines if they are not used for the currently active function.)

Field	Description
SNP ID	The Series Ninety Protocol (SNP) identifier assigned to the PLC.
PLC State	The current status of the CPU:  <b>RUN/OUTEN:</b> PLC running the logic program, outputs enabled. <b>STOP/IOSCAN:</b> PLC stopped, not executing the logic program, scanning/O. <b>STOP/NOIO:</b> PLC stopped, not executing the logic program, noI/Oscan. <b>STOP/FAULT:</b> PLC stopped due to a fault; check fault tables.
Scan Time	The CPU sweep time in milliseconds. This is followed by the type of the scan, which may be:  <b>SCAN:</b> Each scan executed as fast as possible. <b>FIXED:</b> Constant sweep timer enabled, scan fixed to set time limit. <b>OVER:</b> Constant sweep timer enabled, scan exceeds the set time limit.
Mode	The current mode of the programmer connected to the CPU:  <b>OFFLINE:</b> No communications with PLC, or no PLC attached. <b>ONLINE:</b> Actively communicating with the PLC. <b>MONITOR:</b> Same as ONLINE, but programmer cannot modify the contents of the PLC.
AccessLevel	The password access level of the PLC:  <b>LEVEL 4:</b> Change password, write logic/configuration. <b>LEVEL 3:</b> Write logic/configuration, PLC stopped. <b>LEVEL 2:</b> Write data, clear fault tables. <b>LEVEL 1:</b> Read PLC only.
Equality	This field compares the program in the PLC with the version in the folder:  <b>LOGICEQUAL:</b> Both program versions are the same. <b>LOGICNOTEQ:</b> Program versions may be different. <b>BLOCKEDIT:</b> The PLC program is the same as that in the current program folder, the current block is being edited online and may be different in the programmer. The block can be stored to the PLC by pressing <b>ALT-S</b> .

## Selecting the Programmer Operating Mode

Both the programmer and the configuration software operate in three modes: **OFFLINE**, **MONITOR**, and **ONLINE**. In **OFFLINE** mode, no data transfer takes place between the computer and the PLC. Programs and configuration data may conveniently be developed in **OFFLINE** mode, with or without the computer connected to a PLC. In **MONITOR** mode, if communications have been established between the computer and the PLC, the computer can read data from the PLC but may not transfer data to it. With communications established in **ONLINE** mode, programs and other data can be transferred between the PLC and the computer.

If you are using a Workmaster or CIMSTAR I industrial computer, you may configure the Logicmaster 90-30/20/Micro software to use the keyswitch to select the operating mode. For those computers without a keyswitch, or if the keyswitch is not enabled, mode selection can be made by:

- Pressing the **ALT** and **M** keys simultaneously (i.e., press the **Alt** key and hold, then press the **M** key). Repeatedly pressing **ALT** and **M** switches the operating mode from **OFFLINE** to **MONITOR** to **ONLINE** and then back to **OFFLINE**.
- Going to the Programmer Setup screen (**Shift-F7**) and selecting an operating mode. For more information on using the Programmer Setup screen to select the operating mode, refer to chapter 6, "Programmer Setup."

# Chapter 3

## Program Editing

This chapter describes the program edit features of Logicmaster 90-30/20/Micro programming software. Program display and editing may be protected by passwords. If your system has been set up to use passwords and is in **ONLINE** mode, you may have to enter a password in order to use the functions described in this chapter. Chapter 3 contains the following sections:

Section	Title	Description	Page
1	LadderLogic Program Elements	SummarizestheSeries90-30/20/Microinstruction set. Refer to the <i>Series90-30/20/MicroProgrammableControllersReference Manual</i> GFK-0467, for additional information on each of these instructions.	3-2
2	ProgramFormat	Describes the format of a ladder logic rung.	3-18
3	ProgramEntry	Describes how to insert logic elements and functions in a program and three ways to exit from a rung.	3-23
4	Program Annotation	Describes program annotation and how to enter nicknames and reference descriptions.	3-30
5	Variable Declaration Table	Describes how to enter annotation using the variable declaration table. Also describes how to import and export files in SNF format.	3-33
6	RungComments	Describes how to insert rung comments.	3-46
7	Changing the DisplayMode	Describes how to show reference description information at each symbol or reference address in the program.	3-52
8	SubroutineBlocks	Describes how to use block declarations in a main program block.	3-57
9	RungEdit	Describes how to modify an existing program by adding rungs, deleting rungs, and editing rungs. This section also describes how to select, cut, paste, include, and write rungs or declarations, and how to use the open space functions to open element spaces in the rung.	3-67
10	Editor Options	Describes options in the program editor, including multiple coil use and automatic insertion of references.	3-92
11	Search Function	Describes how to search for a reference or instruction using the search function.	3-98
12	OnlineEditing/ Monitoring	Describes how to make changes to or debug a program while the program is running.	3-104

### Note

When the program editor is entered, the Logicmaster 90-30/20/Micro software checks the amount of space on the disk containing the program folder. If insufficient space exists to support changes to the program, the folder is temporarily locked. Program logic may be viewed, but not changed. If space is freed up on the disk using MS-DOS, the lock state is automatically cleared when you run Logicmaster 90-30/20/Micro software.

## Section 1: Ladder Logic Program Elements

The tables in this section summarize the programming instructions available for Series 90-30 PLCs, Series 90-20 PLCs, and Micro PLCs. You can access all of these instructions from either the **Insert (F1)** or **Edit (F2)** function by selecting one of the main functions displayed in the upper line shown below. Press the **Shift** key and the function key for the function you wish to select. For example, to select one of the math functions, first press **Shift-F3** to display the math functions listed in table 3-3 in this section. Then, select the appropriate math function by pressing the function key for that particular function.

RELAY	TMRCTR	MATH	RELATN	BITOP	DATAMV	TABLES	CONVRT	CONTRL	OPN SP
1—	2— /	3—	4—	5—( )	6—(SM)	7—(RM)	8vert	9horz	10more

These function keys provide access to the instructions required to edit ladder diagram rungs:

Function Key	Function	Description	Page
F1	Relay Functions	Select contacts, coils, and links in ladder logic rungs.	3-3
F2	Timer/Counter Functions	Select on-delay and stopwatch-type timers, up counters, and down counters.	3-5
F3	Math Functions	Select addition, subtraction, multiplication, division, modulo division, square root, sine, cosine, tangent, inverse sine, inverse cosine, and inverse tangent functions.	3-6
F4	Relational Functions	Select functions which can compare two numbers: equal, not equal, greater than, greater than or equal, less than, and less than or equal. Or, test a value against a range of numbers.	3-9
F5	Bit Operation Functions	Select functions which can perform comparison and move operations on bit strings: Logical AND, OR, and exclusive OR; logical invert; shift left or right; rotate left or right; bit test, bit set, bit clear, and bit position.	3-10
F6	Data Move Functions	Select basic data move functions: move, block move, block clear, shift register, bit sequencer, and communications request.	3-11
F7	Table Functions	Copy from one array to another; and search for all array values which are equal, not equal, greater than, less than, greater than or equal, or less than or equal to a specified value.	3-11
F8	Conversion Functions	Convert a data item from one number type to another, such as signed integer to 4-digit BCD format or BCD-4 to signed integer format.	3-12
F9	Control Functions	Limit program execution, call a subroutine, enter a comment in a program, and alter the way the CPU executes the program.	3-13



Appendix D, “Instruction Mnemonics,” lists the complete mnemonics for each programming instruction. Please refer to the *Series 90-30/20/Micro Programmable Controllers Reference Manual*, GFK-0467, for additional information on each instruction.

**Table 3-1. Relay Functions**

Key	Instruction	Mnemonic	Function	Description
<i>Selecting Relay Functions (Shift-F1) displays these function key assignments:</i>				
F1	-   -	&NOCON	Normally Open Contact	A normally open contact passes power if the associated reference is ON.
F2	-  /  -	&NCCON	Normally Closed Contact	A normally closed contact passes power if the associated reference is OFF.
F5	-( )-	&NOCOIL	Normally Open Coil	The associated reference is set ON if the coil receives power.
F6	-(SM)-	&SMLAT	Retentive SET Coil	The associated reference is set ON if the coil receives power. The reference remains set until reset by an -(RM)-coil. Its state is retained through power failure and <b>STOP-TO-RUN</b> transition.
F7	-(RM)-	&RMLAT	Retentive RESET Coil	The associated discrete reference is reset OFF if the coil receives power. The reference remains reset until set by an -(SM)- coil. Its state is retained through power failure and <b>STOP-TO-RUN</b> transition.
F8	vert	&VE	Vertical Link	A vertical link with no contact function or reference.
F9	horz -	&HO	Horizontal Link	A shunt; also acts as a delete function.
F10	more			Additional relay functions you can select.

Table 3-1. Relay Functions (Continued)

Key	Instruction	Mnemonic	Function	Description
<i>Pressing More (F10) displays these additional relay function key assignments:</i>				
F1	---<+>	&COILCTD	Continuation Coil	If power to the coil is ON, the continuation coil sets the next continuation contact ON. If power is OFF, the continuation coil sets the next continuation contact OFF.
F2	<+>---	&CONCTD	Continuation Contact	The continuation contact passes power to the right if the preceding continuation coil is set ON.
F3	-(/M)-	&NCMCOIL	Negated Retentive Coil	The associated discrete reference is set ON if the function does not receive power. The state is retained through power failure and <b>STOP-TO-RUN</b> transition.
F4	-(/)-	&NCCOIL	Negated Coil	The associated discrete reference is set ON if the coil does not receive power.
F5	-(M)-	&NOMCOIL	Retentive Coil	The associated discrete reference is set ON if the coil receives power. The state is retained through power failure and <b>STOP-TO-RUN</b> transition.
F6	-(S)-	&SLAT	SET Coil	The associated discrete reference is set ON if the coil receives power. It remains set until reset by an -(R)- coil.
F7	-(R)-	&RLAT	RESET Coil	The associated discrete reference is set OFF if the coil receives power. It remains reset until set by an -(S)- coil.
F8	-(↑)-	&PCOIL	Positive Transition Coil	If the associated discrete reference is OFF when the coil receives power, the reference will be set ON for one logic scan. This coil can be used as a one-shot.
F9	-(↓)-	&NCOIL	Negative Transition Coil	If the associated discrete reference is ON and the coil is not receiving power, the reference will be set ON for one logic scan.
F10	more			Return to the first level of relay functions.

Table 3-2. Timer and Counter Functions

Key	Instruction	Mnemonic	Function	Description
<i>Selecting Timer/Counter Functions (Shift-F2) displays these function key assignments.</i>				
F1	ondtr	&ONDTR	On-Delay Stopwatch Timer	The ONDTR function accumulates time while receiving power. It passes power if the current value exceeds the preset value. The current value is reset to zero when the reset (R) input receives power.
F2	oftd	&OFTD	Off-Delay Timer	The OFDT function increments while power flow is off, and resets to zero when power flow is on. Time may be counted in tenths of seconds (the default selection), hundredths of seconds, or thousandths of seconds. The range is 0 to +32,767 time units. The state of this timer is retentive on power failure; no automatic initialization occurs at power-up.
F3	tmr	&TMR	On-Delay Timer	The current value of the TMR function is set to zero when the function transitions on. The function accumulates time while receiving power, and passes power if the current value is greater than or equal to a preset value.
F4	upctr	&UPCTR	Up Counter	The UPCTR function increments by 1 each time the function receives transitional power. If the current value stored in the counter is greater than or equal to a preset value, the function passes power. The R input is used to reset the counter to zero.
F5	dnctr	&DNCTR	Down Counter	The DNCTR function counts down from a preset value every time the function receives transitional power. If the current value of the counter is zero, the function passes power. The R input is used to set the current value to equal the preset value.
F10	tmbase			Select the time base of a timer. Time may be counted in tenths of seconds or hundredths of seconds.
<i>Pressing Timebase (F10) displays these function key assignments:</i>				
F2	0.1s	_TEN	0.1 second	Time is counted in tenths of a second.
F3	0.01s	_HUN	0.01 second	Time is counted in hundredths of a second.
F4	0.001s	_TH	0.001 second	Time is counted in thousandths of a second.
F10	instrs			Return to the timer and counter functions.

### Note

The Off-Delay timer and the thousandth of a second timebase are available to Release 4.5 and later of all models of CPUs, and when using Logicmaster 90-30/20 4.5 or higher. These features are not available to earlier releases of CPUs, nor earlier versions of Logicmaster.

Table 3-3. Math Functions

Key	Instruction	Mnemonic	Function	Description
<i>Selecting Math Functions (Shift-F3) displays these function key assignments:</i>				
F1	add	&ADD	Addition	Add two numbers. The ADD function passes power if the operation does not result in an overflow.
F2	sub	&SUB	Subtraction	Subtract one number from another. The SUB function passes power if the operation does not result in an overflow.
F3	mul	&MUL	Multiplication	Multiply two numbers. The MUL function passes power if the operation does not result in an overflow.
F4	div	&DIV	Division	Divide one number by another, yielding a quotient. The DIV function passes power if the operation does not result in an overflow and if there is no attempt to divide by zero.
F5	mod	&MOD	ModuloDivision	Divide one number by another, yielding a remainder. The MOD function passes power unless there is an attempt to divide by zero.
F6	sqrt	&SQRT	SquareRoot	Find the square root of an integer or real value. When the function receives power flow, the value of the output Q is set to the square root of the input IN.
F9	more			Pressing <b>more (F9)</b> displays the Trig functions plus the degree/radian functions (see next page).
F10	types			Select a data type for the function. Pressing <b>Types (F10)</b> displays the function keys described in Table 3-10 on page 3-15.

### Note

Math functions can generally use the Real data type, but Real data requires a floating-point CPU (all releases of 352 CPUs; for all other 350 and higher CPUs, floating point support began with Release 9).

**Table 3-3. Math Functions (Continued)**

Key	Instruction	Mnemonic	Function	Description
<i>Pressing More (F9) displays these additional math function key assignments:</i>				
F1	sin	&SIN	Trigonometric Sine	Find the trigonometric sine of the input. When the function receives power flow, it computes the sine of IN, whose units are radians, and stores the result in output Q.
F2	cos	&COS	Trigonometric Cosine	Find the trigonometric cosine of the input. When the function receives power flow, it computes the cosine of IN, whose units are radians, and stores the result in output Q.
F3	tan	&TAN	Trigonometric Tangent	Find the trigonometric tangent of the input. When the function receives power flow, it computes the tangent of IN, whose units are radians, and stores the result in output Q.
F4	asin	&ASIN	Inverse Sine	Find the inverse sine of the input. When the function receives power flow, it computes the inverse sine of IN and stores the result in output Q, whose units are radians.
F5	acos	&ACOS	Inverse Cosine	Find the inverse cosine of the input. When the function receives power flow, it computes the inverse cosine of IN and stores the result in output Q, whose units are radians.
F6	atan	&ATAN	Inverse Tangent	Find the inverse tangent of the input. When the function receives power flow, it computes the inverse tangent of IN and stores the result in output Q, whose units are radians.
F7	deg	&DEG	Convert to Degrees	When the function receives power flow, a RAD_TO_DEG conversion is performed on the real radian value of IN and the result is placed in output degree real value Q.
F8	rad	&RAD	Convert to Radians	When the function receives power flow, a DEG_TO_RAD conversion is performed on the real degree value in input IN and the result is placed in output real radian value Q.
F9	more			Changes the menu to additional math functions.

**Note**

The above math functions require a floating-point CPU (all releases of 352 CPUs; for all other 350 and higher CPUs, floating point support began with Release 9).

Table 3-3. Math Functions (Continued)

Key	Instruction	Mnemonic	Function	Description
F1	log 10	&LOG	Base 10 Logarithm	When the function receives power flow, it finds the base 10 logarithm of the real value in input IN and places the result in output Q.
F2	ln	&LN	Natural Logarithm	When the function receives power flow, it finds the natural logarithm base ( $e$ ) of the real value in input IN and places the result in output Q.
F3	exp	&EXP	Power of e	When the function receives power flow, the natural logarithm base ( $e$ ) is raised to the power specified by IN and the result is placed in Q.
F4	expt	&EXPT	Power of X	When the function receives power flow, $X$ is raised to the power specified by IN and the result is placed in Q.

### Note

The above math functions require a floating-point CPU (all releases of 352 CPUs; for all other 350 and higher CPUs, floating point support began with Release 9).

**Table 3-4. Relational Functions**

Key	Instruction	Mnemonic	Function	Description
<i>Selecting Relational Functions (Shift-F4) displays these function key assignments:</i>				
F1	eq	&EQ	Equal	Test for equality between two numbers. The EQ function passes power if the two inputs are equal.
F2	ne	&NE	Not Equal	Test for non-equality between two numbers. The NE function passes power if the inputs are not equal.
F3	gt	&GT	Greater Than	Test for one number greater than another. The GT function passes power if the first parameter is greater than the second parameter.
F4	ge	&GE	Greater Than or Equal To	Test for one number greater than or equal to another. The GE function passes power if the first parameter is greater than or equal to the second parameter.
F5	lt	&LT	Less Than	Test for one number less than another. The LT function passes power if the first parameter is less than the second parameter.
F6	le	&LE	Less Than or Equal To	Test for one number greater than or equal to another. The LE function passes power if the first parameter is less than or equal to the second parameter.
F7	range	&RANG	Range	Test the input value against a range of two numbers. This instruction is only available for release 4.50 or higher CPUs (4.02 of the 341) and in Logic-master90-30/20 Version 4.5 and above.
F10	types			Select a data type for the function. Pressing <b>Types (F10)</b> displays the function keys described in Table 3-10 on page 3-15.

**Note**

You can use the Real data type for Relational functions except Range, but Real data requires a floating-point CPU (all releases of 352 CPUs; for all other 350 and higher CPUs, floating point support began with Release 9).

Table 3-5. Bit Operation Functions

Key	Instruction	Mnemonic	Function	Description
<i>Selecting Bit Operation Functions (Shift-F5) displays these function key assignments:</i>				
F1	and	&AND	Logical AND	Logical AND of two bit strings.
F2	or	&OR	Logical OR	Logical OR of two bit strings.
F3	xor	&XOR	Logical Exclusive OR	Logical Exclusive OR of two bit strings.
F4	not	&NOT	Logical Invert	Logical inversion of a bit string.
F5	shl	&SHL	Shift Left	Shift a bit string left.
F6	shr	&SHR	Shift Right	Shift a bit string right.
F7	rol	&ROL	Rotate Left	Rotate a bit string left.
F8	ror	&ROR	Rotate Right	Rotate a bit string right.
F9	more			Additional bit operation functions you can select.
F10	types			Select a data type for the function. Pressing <b>Types (F10)</b> displays the function keys described in Table 3-10 on page 3-15.
<i>Pressing More (F9) displays these additional bit operation function key assignments:</i>				
F1	bittst	&BTST	Bit Test	Test a bit within a bit string.
F2	bitset	&BSET	Bit Set	Set one bit within a string to true.
F3	bitclr	&BCLR	Bit Clear	Set one bit within a string to false.
F4	bitpos	&BPOS	Bit Position	Locate a bit set to true within a bit string.
F5	mskcmp	&MCM	Masked Compare	Perform a masked compare of two arrays (available only for Release 4.5 or higher CPUs and in Logicmaster 90-30/20 (version 4.5 and above.)).
F9	more			Return to the first level of bit operation functions.



**Table 3-6. Data Move Functions**

Key	Instruction	Mnemonic	Function	Description
<i>Selecting Data Move Functions (Shift-F6) displays these function key assignments:</i>				
F1	move	&MOV	Move	Move one or more bits of data within PLC memory.
F2	blkmov	&BLKMOV	Block Move	Move a block of up to 7 constants to PLC memory.
F3	blkclr	&BLKCLR	Block Clear	Clear (0) one or more bytes/ words of PLC memory.
F4	shfreg	&SHFR	Shift Register	Shift one or more words or bits of data through a block of PLC memory.
F5	bitseq	&BITSEQ	Bit Sequencer	Sequence a 1 through a group of bits in PLC memory.
F7	comreq	&COMMREQ	Communication Request	Send a communications request to a smart module in the PLC.
F10	types			Select a data type for the function. Pressing <b>Types (F10)</b> displays the function keys described in table 3-11 on page 3-15.

**Note**

You can use the Real data type with Move and Block Move functions, but Real data requires a floating-point CPU (all releases of 352 CPUs; for all other 350 and higher CPUs, floating point support began with Release 9)

**Table 3-7. Table Functions**

Key	Instruction	Mnemonic	Function	Description
<i>Selecting Table Functions (Shift-F7) displays these function key assignments:</i>				
F1	srh eq	&SRCHEQ	Search Equal	Search array for values equal to a specified value.
F2	srh ne	&SRCHNE	Search Not Equal	Search array for values not equal to a specified value.
F3	srh gt	&SRCHGT	Search Greater Than	Search array for values greater than a specified value.
F4	srh ge	&SRCHGE	Search Greater Than or Equal	Search array for values greater than or equal to a specified value.
F5	srh lt	&SRCHLT	Search Less Than	Search array for values less than a specified value.
F6	srh le	&SRCHLE	Search Less Than or Equal	Search array for values less than or equal to a specified value.
F8	arrmov	&ARRMOV	Array Move	Copy a specified number of data elements from a source array to a destination array.
F10	types			Select a data type for the function. Pressing <b>Types (F10)</b> displays the function keys described in Table 3-10 on page 3-15.

Table 3-8. Conversion Functions

Key	Instruction	Mnemonic	Function	Description
<i>Selecting Conversion Functions (Shift-F8) displays these function key assignments:</i>				
F3	→bcd-4	&TO_BCD4	Convert to BCD-4 (From INT)	Convert a value to 4-digit BCD format. The →BCD-4 function passes power unless the number to be converted is out of range (greater than 9999), and no conversion is performed.
F5	→word	&TO_WORD	Convert to Word (From REAL)	Convert a Real value to Word format. The [→] WORD function passes power unless the number to be converted is out of range (0 to 65,535); if so, the conversion is then set to the maximum or minimum (based on exceeding or going below the limits of the range), and power flow is not passed.
F6	→int	&TO_INT	Convert to INT (From BCD-4 or REAL)	Convert a value to signed integer format. The [→] INT function passes power unless the number to be converted is out of range (-32,768 to +32,767); if so, the conversion is then set to the maximum or minimum (based on exceeding or going below the limits of the range), and power flow is not passed.
F7	→tdint	&TO_DINT	Convert to DINT (From BCD-4 or REAL)	Convert a value to double precision integer format. The [→] DINT function passes power unless the number to be converted is out of range (-2,147,483,648 to +2,147,483,647); if so, the conversion is then set to the maximum or minimum (based on exceeding or going below the limits of the range), and power flow is not passed.
F8	→real	&TO_REAL	Convert to Real (From INT, DINT, BCD-4 or WORD)	Convert a value to real value format. The [→] REAL function always passes power.
F10	types			Select the type of data for the function. Pressing <b>Types (F10)</b> displays the function keys described in Table 3-10 on page 3-15.
<i>Pressing More (F9) displays these additional conversion function key assignments:</i>				
F1	→→int	&TRINT	Truncate to INT (from REAL)	Truncate to a 16-bit signed number. The range is -32,768 to +32,767.
F2	→→dint	&TRDINT	Truncate to Double Precision INT (from REAL)	Truncate to a 32-bit signed number. The range is -2,147,483,648 to +2,147,483,647.
F9	more			Return to the first level of conversion functions.
F10	types			Select the type of data for the function. Pressing <b>Types (F10)</b> displays the function keys described in Table 3-10 on page 3-15.

**Table 3-9. Control Functions**

Key	Instruction	Mnemonic	Function	Description
<i>Selecting Control Functions (Shift-F9) displays these function key assignments:</i>				
F1	call	&CALL	Call	Cause a program execution to go to a specified subroutine block. The CALL function always passes power.
F2	do io	&DOIO	DoI/O	Service a specified range of inputs or outputs immediately (all inputs or outputs on a module will be serviced if any addresses on that module are included in the function – partial I/O module updates are not performed). Optionally, a copy of the scanned I/O can be placed in internal memory.
F3	SER	&SER	Sequential EventRecorder	Capture a series of events in the program execution into an array.
F4	pidisa	&PIDISA	ISA PID Algorithm	Select the standard IDS PID algorithm.
F5	pidind	&PIDIND	Independent PIDAlgorithm	Select the non-interacting independent PID algorithm.
F6	sfcrs	&SFCRES	SFC Reset	Force an SFC block to start from the initial step on power-up or after a STOP-to-RUN transition.
F7	end	&END	Temporary End ofLogic	The program executes from the first rung to the last rung or the END instruction, whichever is encountered first. This instruction is useful for debugging purposes.
F8	commnt	&COMMENT	Comment	A rung explanation. After programming the instruction, the text can be typed in by zooming into the instruction.
F9	svcreq	&SVCREQ	ServiceRequest	A special PLC service function. This function passes power if power is received and the function executes properly.
F10	more			Additional control functions you can select.

Table 3-9. Control Functions (Continued)

Key	Instruction	Mnemonic	Function	Description
<i>Pressing More (F10) displays these additional control function key assignments:</i>				
F1	mcrn	&MCRN	Nested Master Control Relay	Start a master control relay range. This is the nested form of the MCR instruction. An MCRN causes all rungs between the nested MCRN and its subsequent ENDMCRN to be executed with no power flow. There can be nothing after an MCRN in a rung.
F2	endmcrn	&ENDMCRN	Nested End Master Control Relay	End a nested master control relay range. This is the nested form of the ENDMCR instruction. There can be nothing after a nested MCR in a rung.
F3	jumpn	&JUMPN	Nested Jump	Jump to a specified location indicated by a LABELN in the logic. This is the nested form of the JUMP instruction.
F4	labeln	&LABELN	Nested Label	The target location of a JUMP instruction. This is the nested form of the LABEL instruction.
F6	mcr	&MCR	Non-Nested Master Control Relay	Start a non-nested master control relay range. This is the non-nested form of the MCR instruction. A non-nested MCR causes all rungs between the non-nested MCR and its subsequent non-nested ENDMCR to be executed with no power flow. There can be nothing after a non-nested MCR in a rung.
F7	endmcr	&ENDMCR	Non-Nested End Master Control Relay	End a non-nested Master Control Relay range. This is the non-nested form of the ENDMCR instruction. There can be nothing after a non-nested ENDMCR in the rung.
F8	jump	&JUMP	Non-Nested Jump	Jump to a specified location indicated by a LABEL in the logic. This is the non-nested form of the JUMP instruction.
F9	label	&LABEL	Non-Nested Label	The target location of a JUMP instruction. This is the non-nested form of the LABEL instruction.
F10	more			Return to the first level of control functions.

**Table 3-10. Data Types**

Key	Instruction	Mnemonic	Function	Description
<i>Selecting Data Types (F10) displays these function key assignments:</i>				
F1	bit	_BI	Bit	A Bit data type is used with instructions which operate on bit strings that are not multiples of 16 bits, or whose reference address is not on a byte boundary (e.g., MOV or SHFREG).
F2	byte	_BY	Byte	A Byte has an 8-bit value.
F3	word	_W	Word	A Word data type uses 16 consecutive bits of data memory; but, instead of the bits in the data location representing a number, the bits are independent of each other. Each bit represents its own binary state (1 or 0), and the bits are not looked at together to represent an integer number. The valid range of word values is 0 to +65,535.
F4	dword	_DW	Double Word	A Double Word data type has the same characteristics as a single word data type, except that it uses 32 consecutive bits in data memory instead of only 16 bits.
F5	bcd-4	_BCD4	Four-Digit Binary Coded Decimal	Four-digit BCD numbers use 16-bit data memory locations. Each BCD digit uses four bits and can represent numbers between 0 and 9. This BCD coding of the 16 bits has a legal value range of 0 to 9999.
F7	real	_REAL	Floating Point	Real numbers use 32 consecutive bits (actually two consecutive 16-bit memory locations). The range of numbers that can be stored in this format is from $\pm 1.401298E-45$ to $\pm 3.402823E+38$ .
F8	int	_INT	Signed Integer	Signed integers use 16-bit memory data locations, and are represented in 2's complement notation. The valid range of an INT data type is -32,768 to +32,767.
F9	dint	_DI	Double Precision Integer	Double precision integers are stored in 32-bit data memory locations (actually two consecutive 16-bit memory locations) and are always signed values. (Bit 32 is the sign bit.) The valid range of a DINT data type is -2147483648 to +2147483867.
F10	instrs			Return to the functions displayed on the screen.

## Data Zoom

The data zoom feature supports the PID functions by providing a display window that expands the parameters of the function block and displays them in real time with labels in a format consistent with their use. For example, a word of data may contain several boolean flags. Each flag is labeled and displayed separately. The boolean inputs and outputs to the function are also displayed.

The data zoom feature is available in either **ONLINE** or **OFFLINE** mode. In **ONLINE** mode when the program folder is identical to the PLC, changes made to values are only stored to the PLC. Real-time updates are maintained within the data zoom window. In **OFFLINE** mode, changes made to values are only stored to disk.

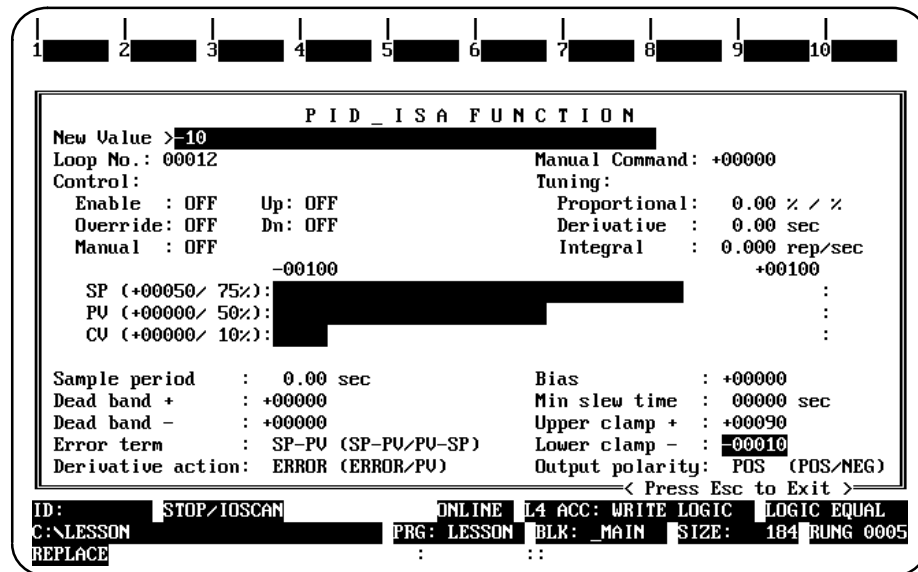
### Note

If you try to write-protect a floppy disk while the Data Zoom screen is displayed, any changes made on the Data Zoom screen will be lost when you exit the screen.

### Using the Data Zoom Feature

The data zoom feature is only available in the program editor. To use this feature:

1. Position the cursor within a PID function block and press **Zoom (F10)**. The following screen shows an example screen for the PID\_ISA function.



2. The PID function is displayed as a window, with the cursor positioned on the first field whose value can be changed.

Each field in the window is displayed in a format consistent with its usage in the function. In the example above, **LoopNo.** indicates the number this PID is within a loop structure and is, therefore, displayed as a signed integer; and **Min slew time** is a timing parameter displayed in seconds. Fixed point numbers, like those displayed in the **Sample period** field, are truncated upon entry.

While the data zoom window is active, the function softkeys normally displayed at the top of the screen will be blank.

3. The **New Value** field functions like a command line. To change the value of any field (provided that field can be changed), use the arrow keys to highlight the field, enter a new value in the **New Value** field, and press the **Enter** key. The value of the field will be changed when the **Enter** key is pressed. You can also use the **Tab** key to change the value of a field.
4. Most field values associated with explicit parameters can be changed. The software will not allow you to highlight those fields whose values cannot be changed.

In the PID function, the **CV**, **PV**, and **Enable** fields cannot be changed. The following table lists the formulas used to compute the percentages in these fields.

Parameter	Formula
SP	<p>Percentage = (quantity - min value) / (max value - min value), where quantity is the current value of SP.</p> <p>Using this formula with the values displayed on the Data Zoom screen shown on the previous page results in a value of 75% for SP:</p> $\begin{aligned} \text{SP} &= [50 - (-100)] / [+100 - (-100)] \\ &= 150/200 \\ &= .75 \text{ or } 75\% \end{aligned}$
PV	<p>Percentage = (quantity - min value) / (max value - min value), where quantity is the current value of PV.</p> <p>Using this formula with the values displayed on the Data Zoom screen shown on the previous page results in a value of 50% for PV:</p> $\begin{aligned} \text{PV} &= [0 - (-100)] / [+100 - (-100)] \\ &= 100/200 \\ &= .50 \text{ or } 50\% \end{aligned}$
CV	<p>Percentage = (CV - lower clamp) / (upper clamp - lower clamp).</p> <p>Using this formula with the values displayed on the Data Zoom screen shown on the previous page results in a value of 10% for CV:</p> $\begin{aligned} \text{CV} &= [0 - (-10)] / [+90 - (-10)] \\ &= 10/100 \\ &= .10 \text{ or } 10\% \end{aligned}$

Percentages are calculated to the nearest 1% on the screen. Percentages less than zero are set to 0%. Percentages greater than 999 are set to 999%. The bar will display up to 100%.

Whenever a bar graph is displayed in data zoom, minimum and maximum scaling values will also need to be displayed. For the PID function, the recommended default minimum and maximum values for the SP and PV bars are -32,000 and +32,000. The minimum and maximum scaling values for the CV bar are the lower and upper clamp values. In the screen shown above, the -100 and +100 displayed immediately above the bar graphs correspond to the minimum and maximum SP and PV scaling values.

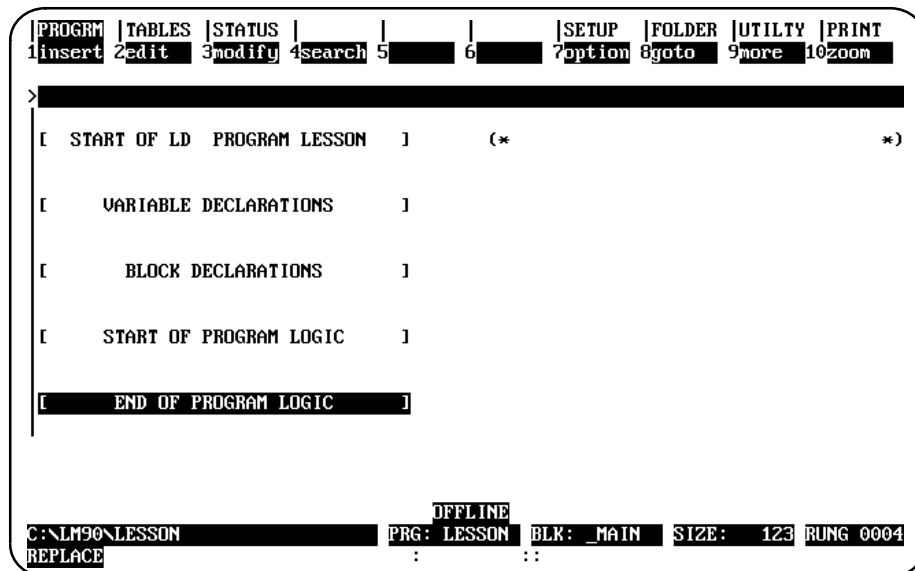
5. Press the **Escape** key to exit the data zoom window and return to your original position within the editor before the data zoom feature was begun.

For more information on the PID function and its parameters, refer to the *Series 90-30/20/Micro Programmable Controllers Reference Manual*, GFK-0467.

## Section 2: Program Format

Program elements are combined to form rungs of ladder logic. A ladder diagram has a symbolic power source. Power is considered to flow from the left rail through a contact to the coil or function block connected to the right.

From the main menu, select **Program Display/Edit (F1)**. The screen displays a list of markers which represent parts of a program.



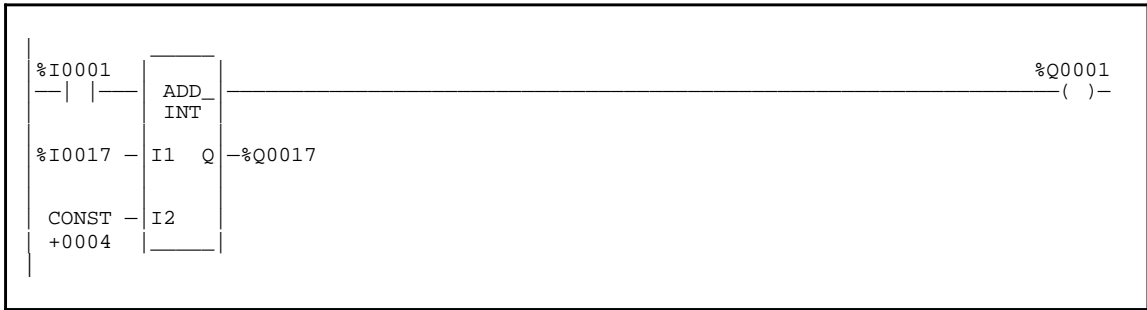
Marker	Description
Variable Declarations	To access the variable declaration table, move the cursor to this marker and press <b>Zoom (F10)</b> . Nicknames and reference descriptions can then be entered in the table.
Block Declarations	A program can include more than one block of logic. Additional blocks, known as subroutine blocks, can be called from other blocks. When that is done, blocks must be declared before they are called.  The main block has a block declaration table. This table lists all blocks which are part of the complete program.  Blocks do not have block declaration tables. However, blocks can be called from the main block or from any block in the program.
Start/End of Program Logic	All logic is placed between these two markers. To enter logic, place the cursor on the [ <b>END OF PROGRAM LOGIC</b> ] marker and press <b>Insert (F1)</b> .

The cursor keys are used to highlight the area of the program to be displayed or edited.



## Creating or Editing Program Logic

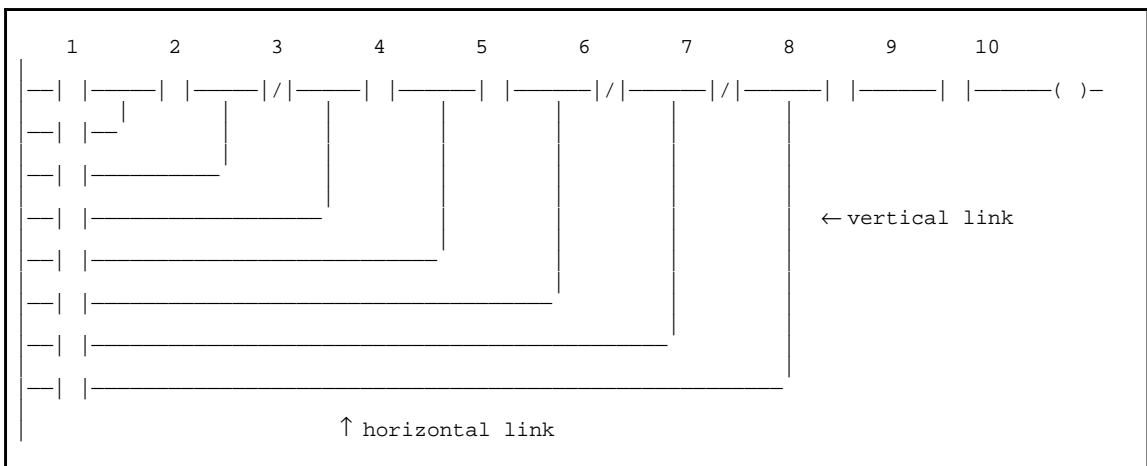
Program logic consists of various elements such as relays, timers, math functions, and other functions, placed together to form rungs of logic.



## Structure of a Ladder Logic Rung

The programmer allows great flexibility in entering program elements; however, it will not allow you to enter a rung with incorrect format or syntax.

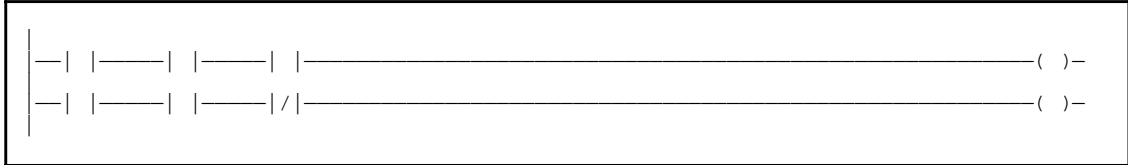
Each rung may contain up to eight parallel lines; each line may have up to ten elements connected in series. Examples of an element include a normally open contact, a normally closed contact, or a coil. Horizontal and vertical links are used to carry power around an element, or to place elements in parallel or series with one another.



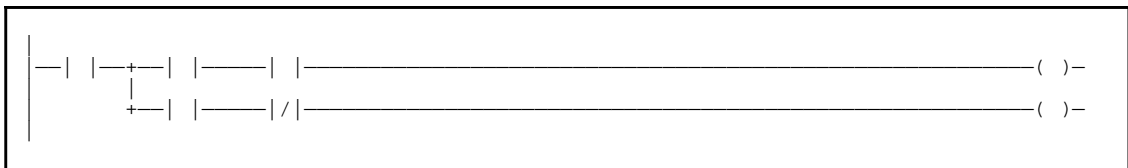
### Note

Programs created using the Hand-Held Programmer must conform to this format to be totally compatible with Logicmaster 90-30/20/Micro software.

The following example shows two separate rungs, which must be entered and accepted separately.



In the next example, two rung lines are connected by a vertical link, forming only one rung.



The last element of a group of rung elements in series must be a coil, a jump, or a function. Nothing may be to the right of a coil or a jump. The tenth position of a rung line is reserved for coils and jumps. A call instruction may occupy columns 9 and 10. A rung may contain up to eight coils.

A rung line is not required to have elements in each column.

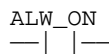
### Ladder Logic Language Rules

These guidelines should be followed when creating or editing ladder logic:

1. If a rung has a transitional coil, it must be the only coil in the rung.
2. There can be only one JUMP or MCR per rung. It must be the last instruction in the rung, and there cannot be a coil in the same rung.
3. A rung must contain at least one contact before any coil, jump, MCR, function, or vertical link. Contacts **must** be entered and cannot be left blank. Function blocks cannot be tied directly into the power rail.
4. Short circuits are not allowed.

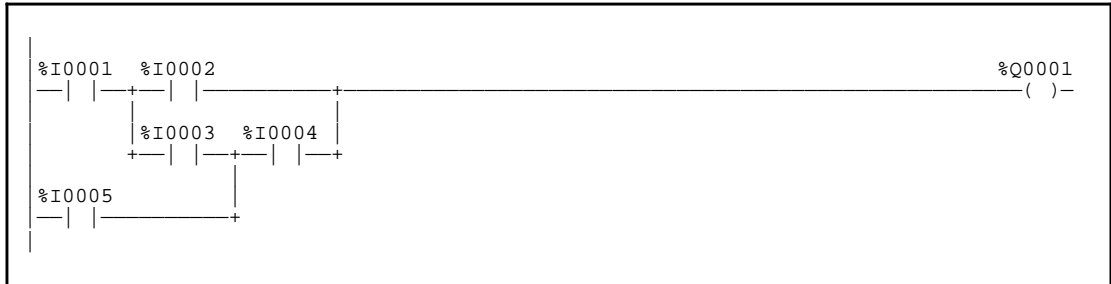
#### Note

The ALW\_ON contact, shown below, may be used to satisfy rules 3 and 4 above.

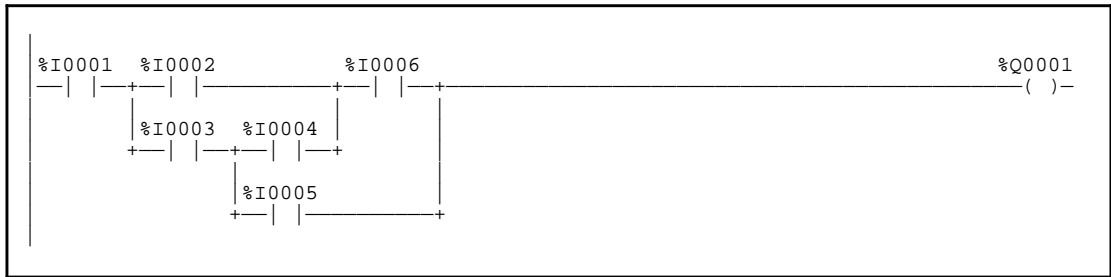


5. A rung must be composed of properly nested sub-expressions. There can be no branches either into or out of another branch. The following examples contain improperly nested rungs.

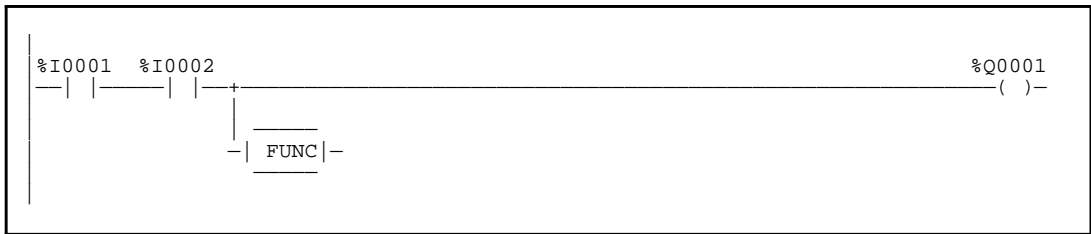
A. In this example, the rung line containing the %I0005 contact branches into the middle of the sub-expression (%I0002 OR (%I0003 AND %I0004)).



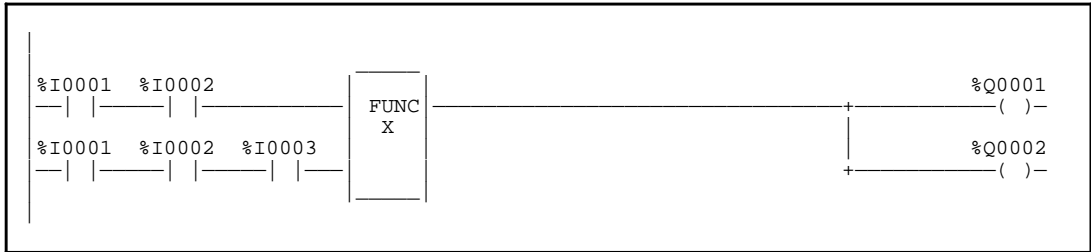
B. In this example, the rung line containing the %I0005 contact branches out of the middle of the sub-expression (%I0002 OR (%I0003 AND %I0004)).



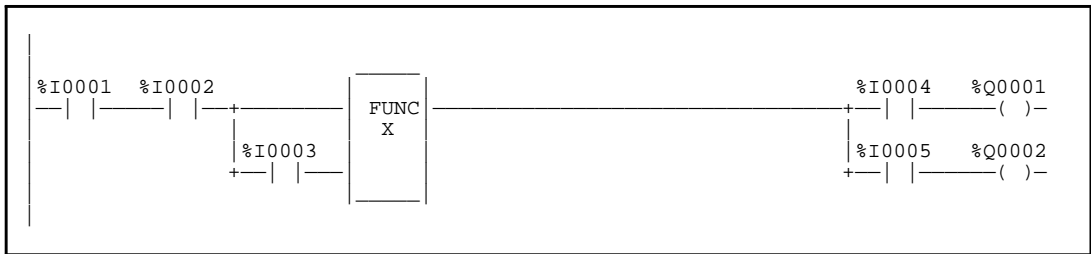
6. There can be no branch around (above or below) a function in a rung. The following rung is not allowed.



- 7. There can be no sub-paths starting from a vertical in a rung containing a function, except for sub-paths leading directly to coils.
  - A. The following rung is allowed because the first sub-path comes directly off the power rail and the second leads directly to coils.



- B. The next rung is not allowed. It has a sub-path starting from a vertical and leading into the function. It also has a sub-path that does not lead directly to coils; it goes through contacts first.



- 8. There can be no contacts following a function in a rung. Note that the rung in the last example above fails this rule, too.
- 9. In general, execution order of rung elements is left-to-right. Within a group of parallel branches, the first (lowest rung line) parallel branch is executed first. The first of multiple sub-paths is executed first.

---

## Section 3: Program Entry

Logicmaster 90-30/20/Micro software was designed to allow for rapid entry of relay ladder diagram programs. By allowing entry of program elements with either function keys or mnemonics, both frequent and occasional users can be satisfied.

Annotation (nicknames, reference descriptions, and rung comments) can be input either prior to logic entry or as each logic element is entered. Rung comments can be entered as logic is created or inserted after the logic has been debugged. For information on annotation, refer to chapter 3, section 4, "Program Annotation."

### Note

Program folders on write-protected floppy disks are automatically locked. Remove the write-protect tab and unlock the folder using the lock/unlock current program folder function (see chapter 7).

### Using Mnemonics

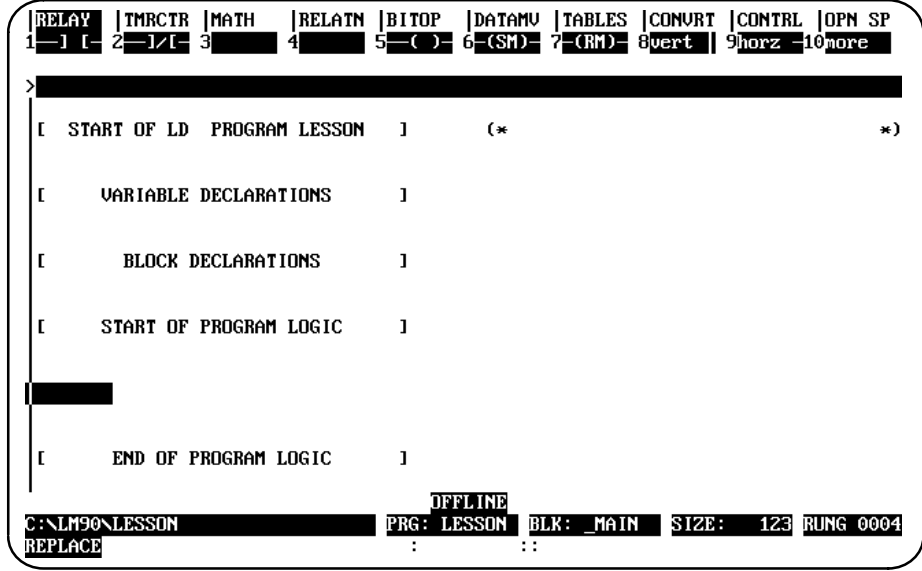
Mnemonic entry enables you to enter an instruction by typing its mnemonic on the command line. For example, to enter the ADD function, you would type `%ADD` on the command line and press the **Enter** key.

For some instructions, it is not necessary to type the entire mnemonic, just enough characters for the entry to be unique. For example, instead of typing `&ADD` to enter the ADD function, you could simply type `&AD` and press the **Enter** key. Appendix D, *Instruction Mnemonics*, lists the mnemonics of all program instructions. While programming, you can also display a list of mnemonics by pressing **ALT-I**.

For many functions, you can also specify a data type or reference address. For example, `&ADD_DINT` would enter the double precision integer version of the ADD function at the current cursor position in the rung.

## Inserting Logic Elements

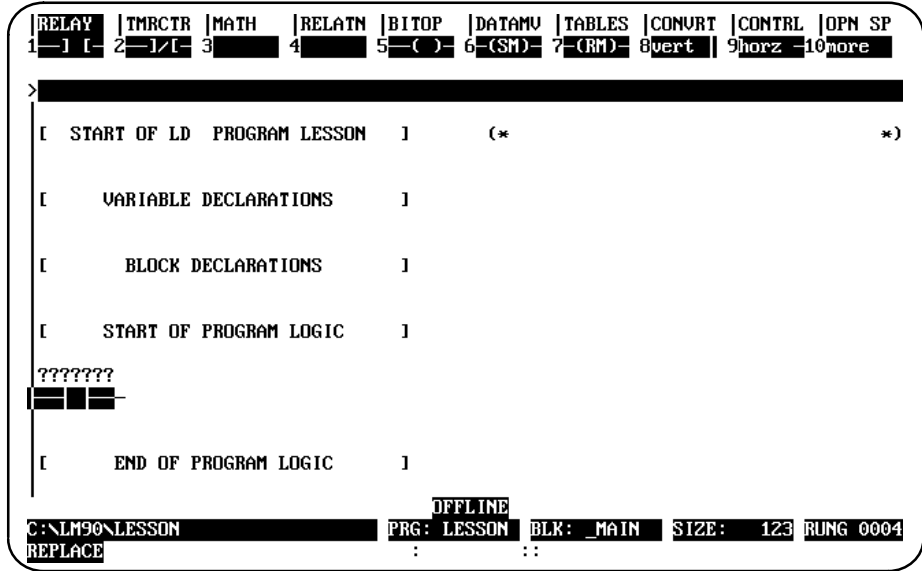
1. With the cursor on the [ END OF PROGRAM LOGIC ] marker, press **Insert** (F1). Rungs are always inserted before the rung the cursor is on.



```
RELAY | TMRCTR | MATH | RELATN | BITOP | DATAMV | TABLES | CONVRT | CONTRL | OPN SP
1-1 | 2-1/I | 3- | 4- | 5-( ) | 6-(SM) | 7-(RM) | 8vert | 9horz | 10more
>
[ START OF LD PROGRAM LESSON ] (* *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]
[ END OF PROGRAM LOGIC ]

OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: _MAIN SIZE: 123 RUNG 0004
REPLACE :
```

2. To enter a relay element at the cursor location, press the desired function key. For example, to enter a normally open contact, press **F1** with the relay functions displayed for the softkeys at the top of the screen. Or, you may enter the normally open contact by typing the mnemonic **&NOCON** on the command line and pressing the **Enter** key.



```
RELAY | TMRCTR | MATH | RELATN | BITOP | DATAMV | TABLES | CONVRT | CONTRL | OPN SP
1-1 | 2-1/I | 3- | 4- | 5-( ) | 6-(SM) | 7-(RM) | 8vert | 9horz | 10more
>
[ START OF LD PROGRAM LESSON ] (* *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]
[ END OF PROGRAM LOGIC ]

??????
REPLACE
```

- 3. A reference may now be entered on the command line. For each reference, include both a user reference and location. There are two ways to do this:
  - A. By entering the reference type and then the address (e.g., %I1), or
  - B. By entering them in reverse order (e.g., I1). The software automatically places the entry in the correct order and format when you press the **Enter** key. Entering the number before the reference type eliminates having to enter the % character.

Then, press the **Enter** key. The cursor automatically advances to the next position, ready for entry of the next element.

Some program functions require references that begin on a word or byte boundary. The Logicmaster 90-30/20/Micro software will automatically adjust the entries to be properly aligned.

You may also combine the previous step and this step into one operation by typing the mnemonic `&NOCON I1` on the command line and pressing the **Enter** key.

```

|RELAY |IMRCTR |MATH |RELATN |BITOP |DATAMV |TABLES |CONVRT |CONTRL |OPN SP
1-1 | 2-|/| 3- 4- 5-( ) 6-(SM) 7-(RM) 8vert | 9horz |10more
>
[ START OF LD PROGRAM LESSON 1 (*) *)
[ VARIABLE DECLARATIONS 1
[ BLOCK DECLARATIONS 1
[ START OF PROGRAM LOGIC 1
%I0001
| | |
[ END OF PROGRAM LOGIC 1
OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: _MAIN SIZE: 123 RUNG 0004
REPLACE : ::

```

### Note

The previous two operations can be combined into one by entering the reference address before pressing the contact function key.

- This process can be continued until a rung is completed and is ready to be accepted. A rung can be accepted by pressing the **Enter** key with the command line empty, or the Plus (+) key on the numeric keypad. If there is an error in the rung, the rung is not accepted and the cursor is placed on the incorrect element for correction.

```

|RELAY| |TMRCTR| |MATH| |RELATN| |BITOP| |DATAMV| |TABLES| |CONVRT| |CONTRL| |OPN SP|
1-| | 2-| | 3-| | 4-| | 5-( )-| 6-(SM)-| 7-(RM)-| 8vert | 9horz | 10more
(E137) Disconnected instruction
>
[ START OF LD PROGRAM LESSON ] (* *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]
%I0001
---
[ END OF PROGRAM LOGIC ]
OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: _MAIN SIZE: 123 RUNG 0004
REPLACE %I0001 : ::

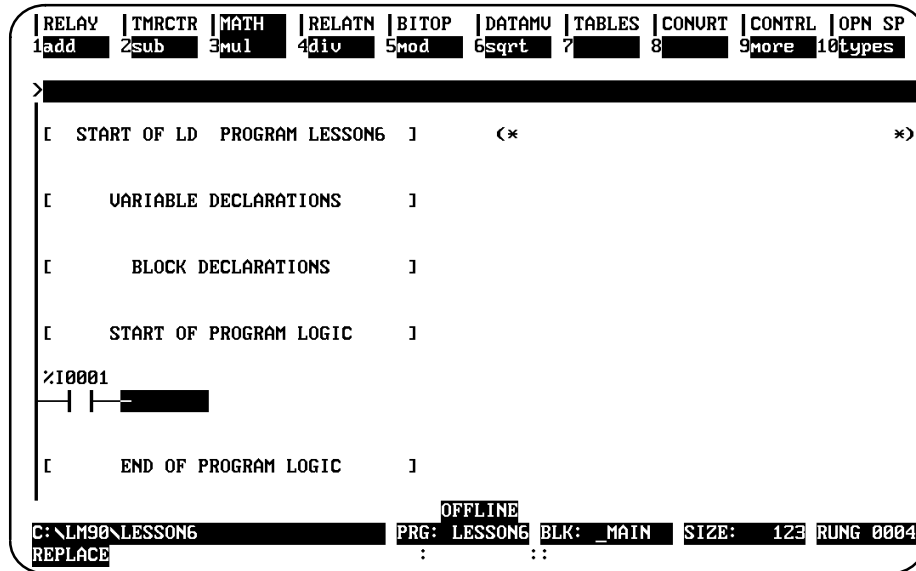
```



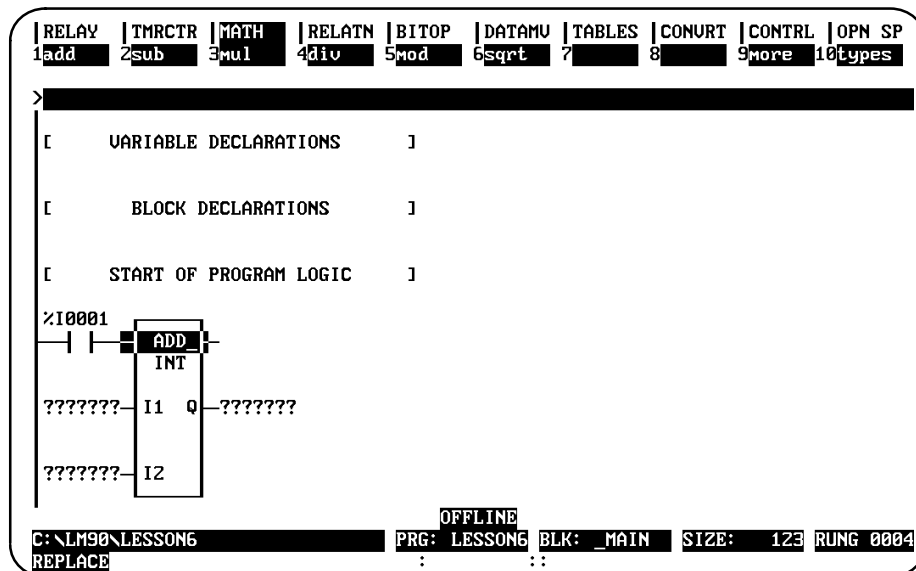
## Inserting Functions

Functions can be entered as easily as relay elements.

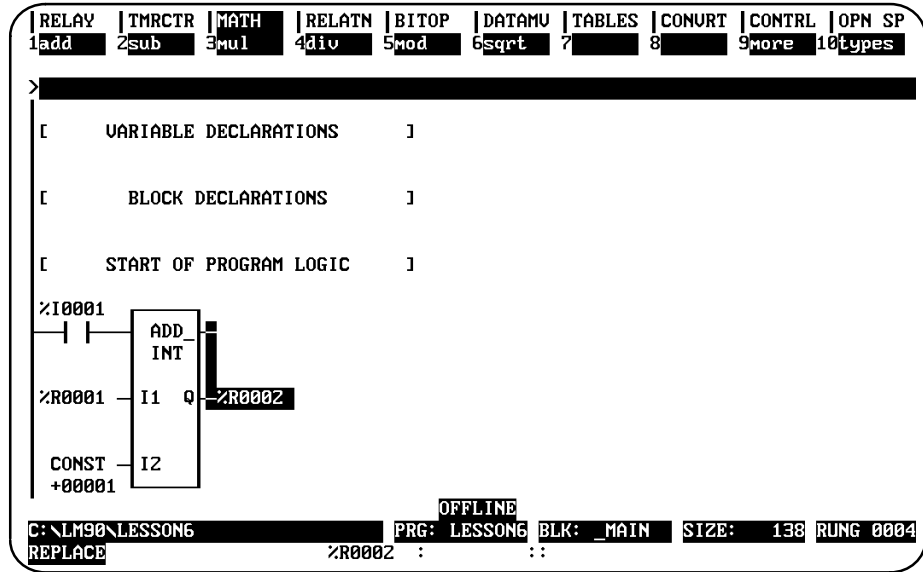
1. First, use the relay function keys to enter the enabling logic. In the first example screen shown below, a normally open contact with reference address %I1 is entered at the enabling logic.
2. Select the type of function using the shift-function keys. For example, to select math functions, press **Shift-F3**.



3. Select the function desired using the function keys. For example, to select the ADD function, press **Add (F1)**. Or, you may enter the ADD function by typing the mnemonic: **&ADD** on the command line and pressing the **Enter** key.



4. The Tab key can now be used to move the cursor to each parameter position around the function.
5. When the cursor is on the first input, simply type the value to be entered in this position on the command line (for example, %R0001 or 1R), and press the Tab key or Enter key to place this entry into its position. Pressing the Tab key will move the cursor to the next entry position. This process can be continued until the rung is completed and accepted.



For information on entering comments, refer to chapter 3, section 6, “Rung Comments.”  
 For information on zooming into entries, refer to chapter 3, section 1, “Ladder Logic Program Elements.”

## Exiting Rung Entry

There are three ways to exit from a rung.

1. Press the **Escape** key to attempt to accept the current rung. If the rung passes the software validity test, the original function key selections (shown below) are restored and the new logic is added to the program. After accepting a rung, the cursor moves to the next rung, ready for the next rung to be entered. However, in **INSERT** mode a new rung is automatically opened below the newly created rung.

PROGRAM	TABLES	STATUS				SETUP	FOLDER	UTILITY	PRINT
1insert	2edit	3modify	4search	5	6	7option	8goto	9more	10zoom

If the logic fails the check, an error message is displayed and the cursor is positioned on the location where the error occurred.

Informational messages relating to the executability of the logic may also be displayed.

2. Press the **Enter** key with the command line empty (or the **Plus (+)** key on the numeric keypad). This causes the same response as pressing the **Escape** key.
3. Press **ALT-A** to exit the insert or edit function without modifying the existing ladder diagram logic. Confirmation is required. Pressing **ALT-A** the first time will restore the rung to its original state. In **INSERT** mode, this would be an empty rung. Pressing **ALT-A** a second time exits the insert function.

Each rung is not saved to disk as it is accepted. To update the disk, press the **Escape** key to exit the program editor and return to the Programming Software main menu, or press **ALT-U**.

## Using the Cursor to Select a Reference Table

You may go directly to the reference table of a reference (e.g., of a function block operand, contact, coil, etc.) under the cursor by pressing **ALT-F2**. Then, press **Shift-F1** from the reference table to return to the same place in the program.

## Section 4: Program Annotation

Annotation is optional explanatory text in a program. This text makes the program easier to read and to understand.

### Note

Files containing program annotation exist only in the folder, not in the PLC.

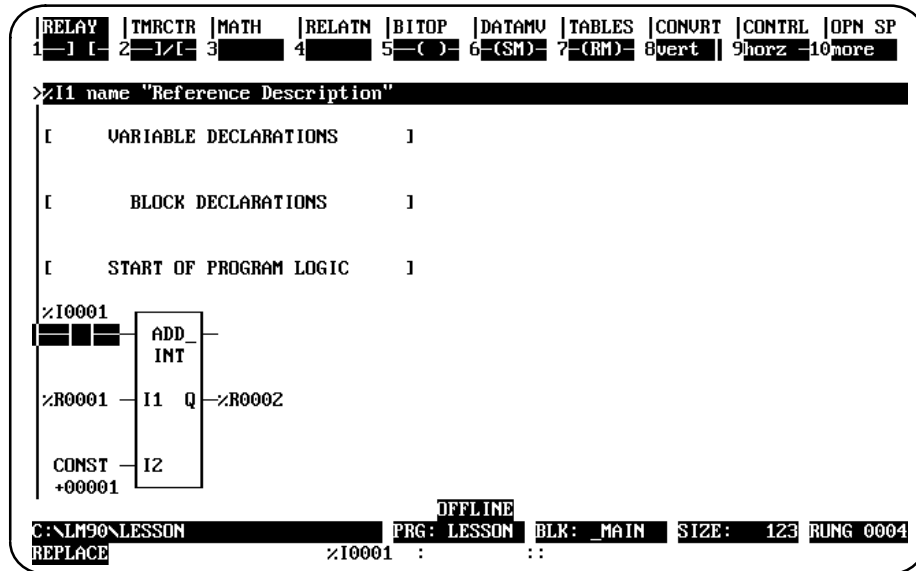
Logicmaster 90-30/20/Micro software provides the following types of program annotation:

Table 3-11. Program Annotation

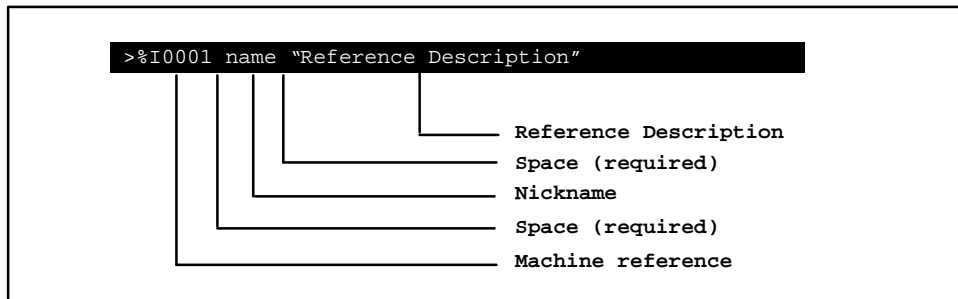
Type	Description
Nickname	<p>An optional 1- to 7-character identifier, which can be used for each program reference. The characters used in a nickname may be a letter from A through Z, a numeral from 0 through 9, an underscore, or the special characters +, -, %, #, @, &lt;, &gt;, =, and &amp;. The first character of the nickname must be a letter.</p> <p>A nickname is case-insensitive, unless a global nickname has been reassigned in a subroutine. When a global nickname (e.g., SWITCH1) has been reassigned in a subroutine, the local use of that nickname will remain in upper-case letters; however, the global use of that nickname will be displayed in lower-case letters (e.g., switch1).</p>
ReferenceDescription	<p>An optional text description of up to 32 characters which is associated with a machine reference or with implicit identifiers (e.g., program name, subroutine block, or JUMP/LABEL/MCR/ENDMCR). A reference description can be used with or without a nickname.</p>
Comment	<p>Longer blocks of text (rung explanations). A comment consists of up to 2048 characters of text. On the screen, the text of a comment can be read by pressing <b>Zoom (F10)</b> with the cursor located at the comment rung. The comment can also be printed as part of the ladder logic.</p>

## Entering Nicknames and Reference Descriptions

Nicknames and reference descriptions can be entered in two different ways. The first way is to create a nickname while programming. This is done by entering the nickname and/or reference description on the command line as the reference is used.



The order of entry is not important. For example:



Separate the reference, nickname, and reference description by a space. Use quotation marks before and after the reference description. A double colon (:) may also be inserted between the nickname and the reference description. If a double colon is used, quotation marks are not required around the reference description.

When you press the **Enter** key, if there are no conflicts, the nickname is displayed above the program element. It is also temporarily included in the variable declaration table for that program. While inserting or editing a rung, you can press **ALT-V** to view the variable declaration table. Temporary entries are marked with an asterisk. When the rung is accepted, the nickname is permanently added to the table. If the rung is aborted before it is accepted, this information is removed from the table.

If a conflict or error occurs when you press the **Enter** key, the contents of the command line are not applied to the instruction and a message is displayed. Press **CTRL-Home** to recall the last entry made on the command line. Then, press the **CTRL** key and the Left or Right cursor key to move within the command line to correct the entry.

Depending on which display mode is active, if you enter a nickname for a reference, it is displayed in the program instead of the reference. For example, if you entered the reference `%I0104` for a normally closed contact in the logic, the display would look like this:

```
|
| %I0104
| ---|/|---
|
```

If you entered a nickname and (optionally) a reference description for the reference on the command line:

```
%I0104 XWATMOV "'x' APM waiting move"
```

The display would look like this instead:

```
|
| XWATMOV
| ---|/|---
|
```

## Note

Use **ALT-N** to toggle between reference address display, nickname display, reference description display, and compressed rung display. Refer to chapter 6, "Programmer Setup," for more information on specifying which modes will be displayed when **ALT-N** is pressed.

The second and simplest way to enter nicknames and reference descriptions, however, is to use the variable declaration table, described in the next section. As each entry is made in the variable declaration table, the reference description for the current entry will appear in a window in the upper right portion of the table. This window consists of 4 lines, each 7 characters in length, and will show the way the description will look above a reference when printed or displayed in expanded mode.

## Section 5: Variable Declaration Table

Program annotation can be entered in a program using the variable declaration table, as described in this section.

You may enter nicknames in the program or any subroutine. Nicknames in the program's `_MAIN` (global) variable declaration table are known to all subroutine blocks. Nicknames in a subroutine block's (local) declaration table are known only to that subroutine.

Each subroutine block may have its own local use of nicknames. The same reference may have different local nicknames in different subroutine blocks, as shown in these examples:

```
BLOCK A    %R1    Light_1  
BLOCK B    %R1    Light_2
```

Two subroutine blocks may have the same nickname for different references, as shown in these examples:

```
BLOCK A    %L1    RESET  
BLOCK B    %L2    RESET
```

When using a reference address in a program block, Logicmaster 90-30/20/Micro software retrieves the nickname from the local table. If it does not have a local nickname, the software retrieves the nickname from the program's `_MAIN` table. If the nickname is not in the program's `_MAIN` table, the software then looks at the reserved nicknames (e.g., `FST_SCN`, `%S0001`).

### Note

5000 declarations (variables and identifiers) are allowed in the program's `_MAIN` variable declaration table, and 256 declarations (variables and identifiers) are allowed in each subroutine block's variable declaration table. Each table is always arranged in sorted reference address order.

As more nicknames are used, performance in the program editor may be affected. To enhance performance, 736K of Expanded Memory (LIM/EMSVersion 3.02 or later) or SMARTDRV may be used.

## Displaying the Variable Declaration Table

To display the variable declaration table:

1. Move the cursor to highlight the [ VARIABLE DECLARATIONS ] marker.

```

PROGRAM | TABLES | STATUS |  |  |  | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3modify 4search 5 6 7option 8goto 9more 10zoom
>
[ START OF LD PROGRAM LESSON ] (* *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]
NAME
| | ADD_
| | INT
| |
%R0001 | I1 Q | %R0002
OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: _MAIN SIZE: 138 RUNG 0001
REPLACE : ::
  
```

2. Press Zoom (F10).

```

PROGRAM | TABLES | STATUS |  |  |  | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3delete 4search 5copy 6 7 8goto 9region 10switch
>
VARIABLE DECLARATION TABLE
REFERENCE NICKNAME REFERENCE DESCRIPTION Reference
Description
%I0001 NAME Reference Description
OFFLINE
C:\LM90\LESSON2 PRG: LESSON2 BLK: _MAIN ENTRY 0001
REPLACE : ::
  
```



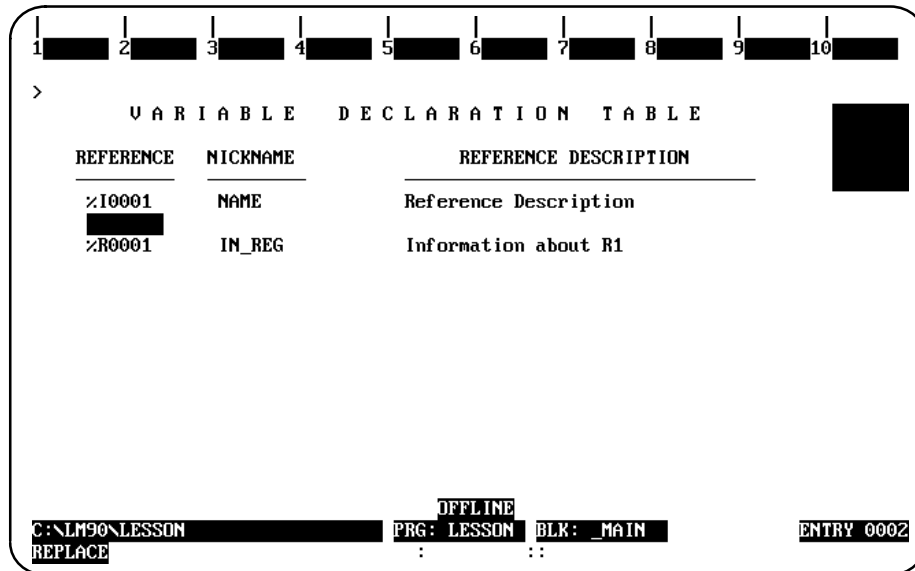
- 
3. The identifier table can be displayed from this screen by pressing **switch (F10)**. This table lists the program name, JUMPs, LABELs, MCRs, ENDMCRs, and subroutine blocks declared in this folder. You cannot insert new entries on this display screen; however, you can edit entries already displayed. For example, you could assign an identifier description to the program name.
  4. Variable declarations may be viewed in a window on the screen, without leaving the insert or edit function, by pressing **ALT-V** (**ALT-V** once for local variables, **ALT-V** twice for global variables). The variable declarations displayed in this window, however, cannot be edited. Entries temporarily listed (those created or modified during the current editing session) in the variable declarations table can be easily identified by an asterisk (\*). Use the Up/Down cursor keys or PageUp/Downkeys to scroll through the entries in this window; then, press the **Escape** key to exit the window.

## Entering Variable Declarations

To enter new nicknames and reference descriptions as you go:

1. Press **Insert (F1)**. A field will appear in the reference column.
2. Enter the reference in this field (e.g., %R2 or 2R), and then press the **Enter** or **Tab** key to move to the **Nickname** field.
3. Enter a nickname (e.g., IN\_REG), and then press the **Enter** or **Tab** key to move to the **ReferenceDescription** field.
4. Enter a 1- to 32-character description of this reference. The Logicmaster software breaks reference descriptions on boundaries after every seventh character for display in a 4-line x 7-character window. Only 28 characters of the 32-character reference description are displayed in the window. You may want to add extra space in the reference description in order to have the words correctly separated in the window.

To accept this entry, press the **Enter** key, or the **Plus (+)** key on the numeric keypad. The variable declaration table is automatically sorted each time you press the **Enter** key to accept a new entry. The cursor then moves to the **Reference** field on the next line.



5. To enter a nickname for the next reference, simply press the **Enter** or **Tab** key and the next reference in sequence (in this example, %R0002) will be inserted into the **Reference** field. The cursor will move to the **Nickname** field. To enter a nickname for a different reference, simply type in this reference and press the **Enter** key.
6. You can continue this process until all references have been defined.
7. To exit **INSERT** mode, press the **Escape** key. To exit the variable declaration table and return to the logic entry screen, press **Escape** again.

## Copying a Variable Declaration

You can copy text from the NICKNAME and REFERENCE DESCRIPTION of another variable. This editing feature is particularly useful when you are creating a program that has several similar variables using similar nicknames and reference descriptions as in the example shown on this page.

To use this feature, follow these steps:

- Enter the line you wish to copy in the standard way.
- After you press **Enter** at the end of the reference description, press the **Escape** key to accept that variable and change the selections available through the function keys.
- Press the **Up Arrow** key (i.e., the upward cursor mover key) once to move the cursor back to the line you just entered.
- Press the **Copy** function key (**F5** as can be seen in the sample shown below).

```
|PROGRAM|TABLES|STATUS| | | |SETUP|FOLDER|UTILITY|PRINT
1|insert 2|edit 3|delete 4|search 5|copy 6| |7| |8|goto 9|region10|switch
>
VARIABLE DECLARATION TABLE STARTUP
REFERENCE NICKNAME REFERENCE DESCRIPTION FOR
STATION
2
%I0001 ESTOP IMMEDIATE POWER-DOWN
%I0002 START1 STARTUP FOR STATION1
I0003 START2 STARTUP FOR STATION2

OFFLINE
C:\LM90\STATMNG PRG: STATMNG BLK: _MAIN ENTRY 0003
REPLACE :
```

The line you copied moves down one, and a copy of it appears where your cursor is located (awaiting your modifications). First enter a new reference number, then edit the nickname and description.

The sample shown on the next page shows the screen that appears immediately after pressing the **Copy** key (**F5**) from the screen shown above.

```

1 2 3 4 5 6 7 8 9 10
>
   V A R I A B L E   D E C L A R A T I O N   T A B L E
REFERENCE  NICKNAME      REFERENCE DESCRIPTION  STARTUP
FOR
STATION
Z
  ZI0001    ESTOP        IMMEDIATE POWER-DOWN
  ZI0002    START1          STARTUP FOR STATION1
  [REDACTED] START2          STARTUP FOR STATION2
  ZI0003    START2          STARTUP FOR STATION2

OFFLINE
C:\LM90\STATMNG      PRG: STATMNG  BLK: MAIN      ENTRY 0003
REPLACE              :           ::
  
```

Notice that the cursor is resting in the REFERENCE (i.e., reference number) field so that you can assign a unique reference number to it. You will also need to assign a unique NICKNAME and REFERENCE DESCRIPTION. Remember, you can use **Ctrl-Right Arrow**, i.e., hold the **Control** key down and press the **Right Arrow** (or Right cursor key), to move the cursor across the letters or words you want to keep. Then key over or add to the text you want to change; e.g., in the example shown above, you would only need to key over the last digit on both the nickname and the description.

**Note**

Make sure you enter a unique reference number. If you just press the **Right Arrow** key to advance to the NICKNAME field, the reference number from the line you copied will drop in by default which will force you to change it to an unused number when you press **Enter** at the end of that line.

## Editing Variable Declarations

To change the content of a variable declaration:

1. Place the cursor at the declaration to be changed, and press **Edit (F2)**.
2. Use the **Enter** key, cursor keys, or **Tab** key to move from field to field. Type over the entry as needed.
3. After changing the entry, press the **Enter** key or the **Plus (+)** key on the numeric keypad to accept the changes and move to the next table entry to continue editing. Press the **Escape** key to accept the changes and terminate the editing session.

The region functions, described in the rung edit section of this chapter, can also be used to select, cut, paste, include, write, and delete variable declarations. For more information on these functions, refer to chapter 3, section 9, "Rung Edit."

## Deleting Variable Declarations

To remove one or more entries from the variable declaration table, place the cursor at the first declaration to be deleted and press **Delete (F3)** or **ALT-D**. Another way to delete entries from the table is to use the **Select (F1)** softkey to select the variable declaration you wish to delete, then press the **Delete (F6)** edit softkey. Repeat this procedure until all the entries you wish to delete have been removed from the table.

To undo the delete, press **ALT-A** before leaving the table. Once you leave a table, you *cannot* undo a deletion made within that table.

## Searching for Variable Declarations

The variable declaration table may be searched for a reference or nickname; it cannot be searched for a reference description. To initiate a search in the variable declaration table:

1. Press **Search (F4)** to display the search function window.
2. Enter either the reference or its associated nickname into the **Search for** field. You cannot enter a reference description in this field. Then, press the **Enter** key.

## Using Goto

The **Goto (F8)** function key may be used to move the cursor within the variable declaration table. To move to the *n*th variable declaration, enter *n* on the command line, and press **Goto (F8)**. For example, to move the cursor to the first variable declaration, enter **1** on the command line, and press **Goto (F8)**.

## Cut/Pasting Variable Declarations

The region functions, described in the rung edit section of this chapter, can also be used to select, cut, paste, include, write, and delete variable declarations. For more information on these functions, refer to chapter 3, section 9, "Rung Edit."

### Note

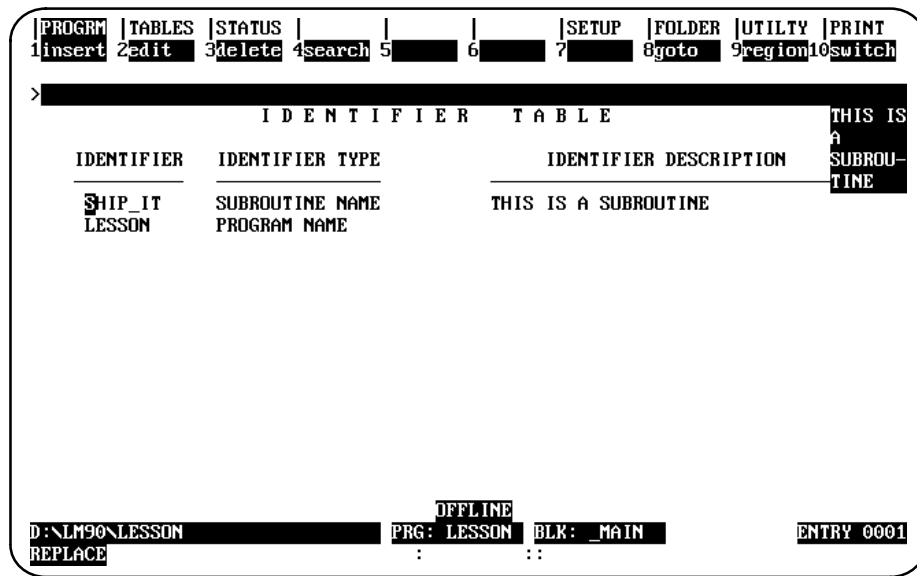
You may not paste or include variable declarations if a reference address in the paste buffer or include file is already in the variable declaration table. However, you may edit the variable declaration table before the paste or include operation to eliminate any reference address conflicts.

## Automatically Inserting References

References which do not have nicknames or reference descriptions can be automatically inserted into the variable declaration table as the program is being developed. Then, at a later time, you can go to the variable declaration editor and enter just the annotation. For information on automatically inserting references, refer to chapter 3, section 10, "Editor Options."

## Viewing the Identifier Table

The identifier table may be displayed by pressing **Switch (F10)** from the variable declaration table.



The identifier table in the `_MAIN` program contains the program name, subroutine block names, JUMPs, LABELs, MCRs, and ENDMCRs. The identifier table in a subroutine block contains JUMPs, LABELs, MCRs, and ENDMCRs.

You cannot insert new entries on this display screen. The program name cannot have its identifier edited. A subroutine block can have its identifier edited if it is not called. All other identifiers and descriptions can be edited.

The identifier table may be searched for a reference or nickname; it cannot be searched for a reference description. To initiate a search in the identifier table:

1. Press **search (F4)** to display the search function window.
2. Enter either the reference or its associated nickname into the **Search for** field. You cannot enter a reference description in this field. Then, press the **Enter** key.

## Importing to and Exporting from the Variable Declarations Table

### Comma Separated Variable (CSV) Format

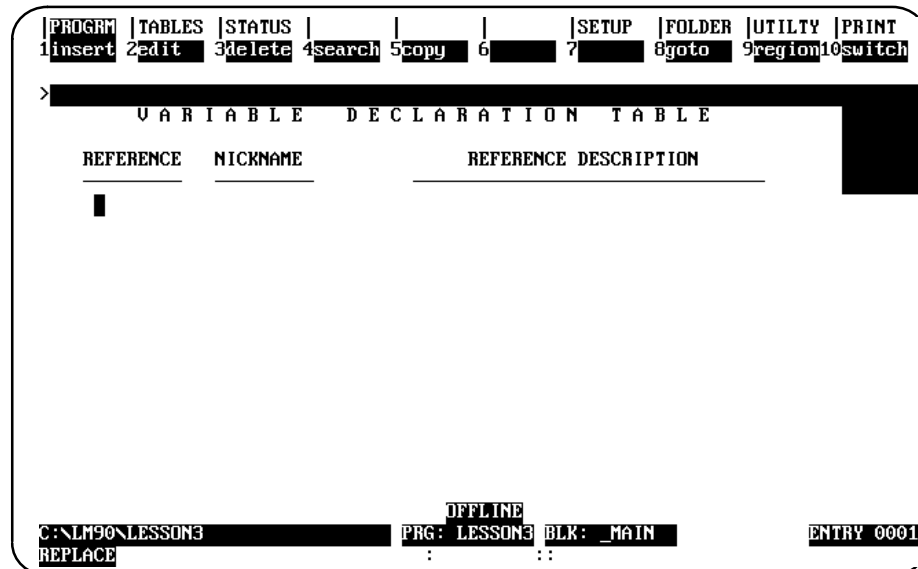
For Logicmaster uses an extension of the industry-standard Comma Separated Variable (CSV) format called Shared Name File (SNF) format. Importing an SNF into the Variable Declarations Table gives you the ability to define nicknames ahead of time in a spreadsheet program. Exporting an SNF from the Variable Declarations Table gives you the ability to use the exported file with CIMPLICITY® and third party operator interfaces.

The steps for importing and exporting files that use the SNF format are shown here. For information about SNF format, see Appendix I.

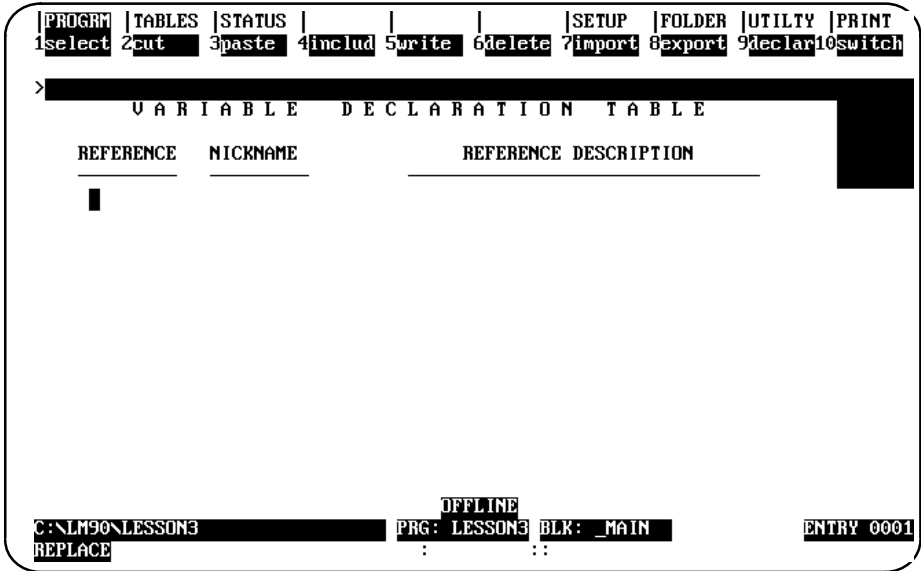
### Importing SNF Formatted Files

When importing Shared Name File (SNF) format files, the SNF files must adhere to the standards discussed in Appendix I of this manual. If your file does adhere to those standards, then follow these steps to import:

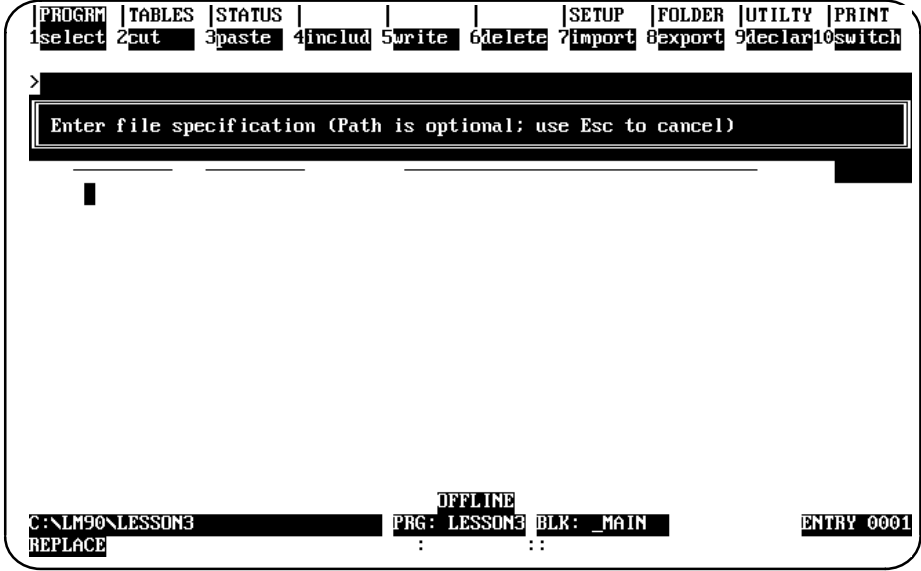
1. Before you begin the next steps (which are all from within the programming software), make sure that your SNF is in the same directory as the program into which you are going to import.
2. Move the cursor to highlight the [ VARIABLE DECLARATIONS ] marker as shown on page 3-34.
3. Press **Zoom (F10)**.



4. Press **region (F9)**. Your screen display will change to the one shown on the following page.



5. Press **import** (F7). You will see the message shown in the following sample screen.



6. Type the file name. The path is optional because in Step 1 you ensured that the SNF was in the same directory as the folder.

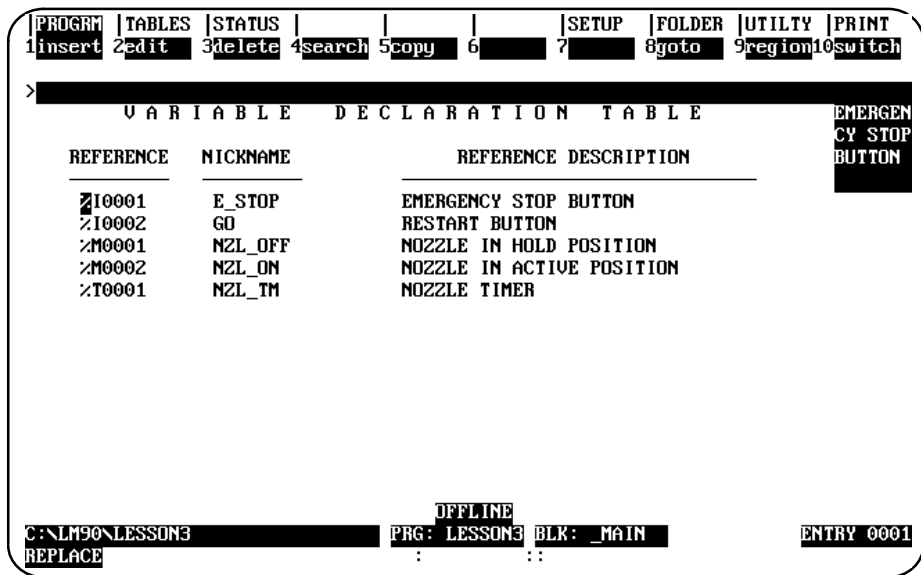
As long as your SNF follows the guidelines discussed in Appendix I, it will import correctly into the Variable Declarations Table.



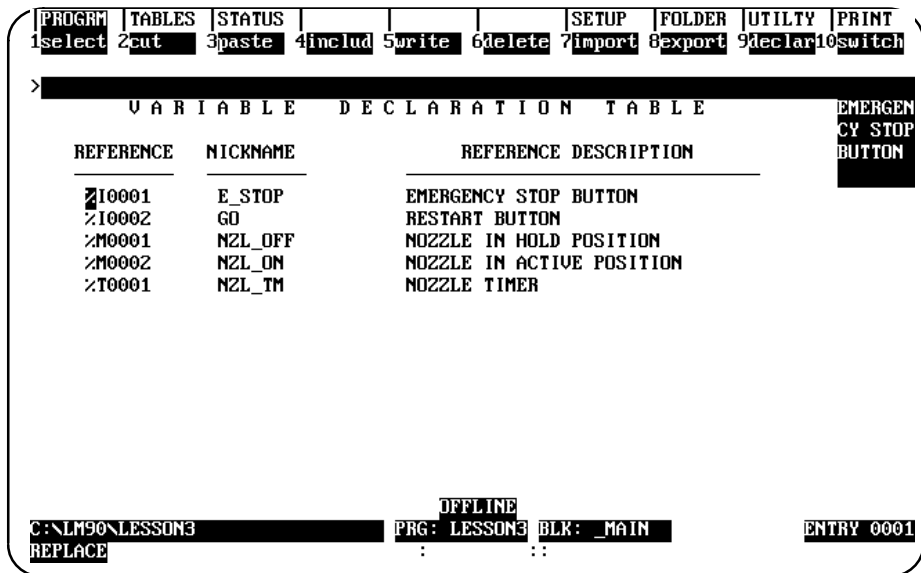
### Exporting SNF (CSV) Formatted Files

SNF is an extension of the industry-standard Comma Separated Variable (CSV) format. For more information about CSV and SNF formats, refer to page 3-41. When exporting your Variable Declarations Table to SNF format, Logicmaster will put the Variable Declarations into Shared Name File (SNF) format. For information about SNF format, refer to Appendix I of this manual.

1. Move the cursor to highlight the [ VARIABLE DECLARATIONS ] marker as shown on page 3-34.
2. Press Zoom (F10).



3. Press region (F9). The selections at the top of the screen will change to the ones shown below.



- Press **select** (F1). Then press the **Cursor Down** key (the **Down Arrow** key) until you have highlighted the portion of the Variable Declaration Table you wish to export.

The selected region will appear in reverse video as represented in the sample shown below.

```

|PROGRAM|TABLES|STATUS| | | |SETUP|FOLDER|UTILITY|PRINT
1select 2cut 3paste 4includ 5write 6delete 7import 8export 9declar10switch
>
VARIABLE DECLARATION TABLE
REFERENCE NICKNAME REFERENCE DESCRIPTION
%I0001 E_STOP EMERGENCY STOP BUTTON
%I0002 GO RESTART BUTTON
%M0001 NZL_OFF NOZZLE IN HOLD POSITION
%M0002 NZL_ON NOZZLE IN ACTIVE POSITION
%T0001 NZL_TM NOZZLE TIMER
OFFLINE
C:\LM90\LESSON3 PRG: LESSON3 BLK: _MAIN ENTRY 0006
REPLACE : ::
  
```

- Press **export** (F8); then type the name you wish to call the SNF that Logicmaster will create from your Variable Declarations. You do not have to add an extension; Logicmaster will automatically add the extension .SNF. (The recommended method of nomenclature is to use the same name for your SNF as the folder from which it came.) If you do not enter a path, the file will save in the same directory where your program resides. Refer to Appendix I for more information about SNF format.

```

|PROGRAM|TABLES|STATUS| | | |SETUP|FOLDER|UTILITY|PRINT
1select 2cut 3paste 4includ 5write 6delete 7import 8export 9declar10switch
>
Enter file specification (Path is optional; use Esc to cancel)
%I0001 E_STOP EMERGENCY STOP BUTTON
%I0002 GO RESTART BUTTON
%M0001 NZL_OFF NOZZLE IN HOLD POSITION
%M0002 NZL_ON NOZZLE IN ACTIVE POSITION
%T0001 NZL_TM NOZZLE TIMER
OFFLINE
C:\LM90\LESSON3 PRG: LESSON3 BLK: _MAIN ENTRY 0006
REPLACE : ::
  
```

- 
6. After typing in a name, press **Enter**. The words “Export completed” will appear in the upper left portion of your screen just below the menu.
  7. Press the **Escape** key after seeing the “Export completed” to return to the program.

## Section 6: Rung Comments

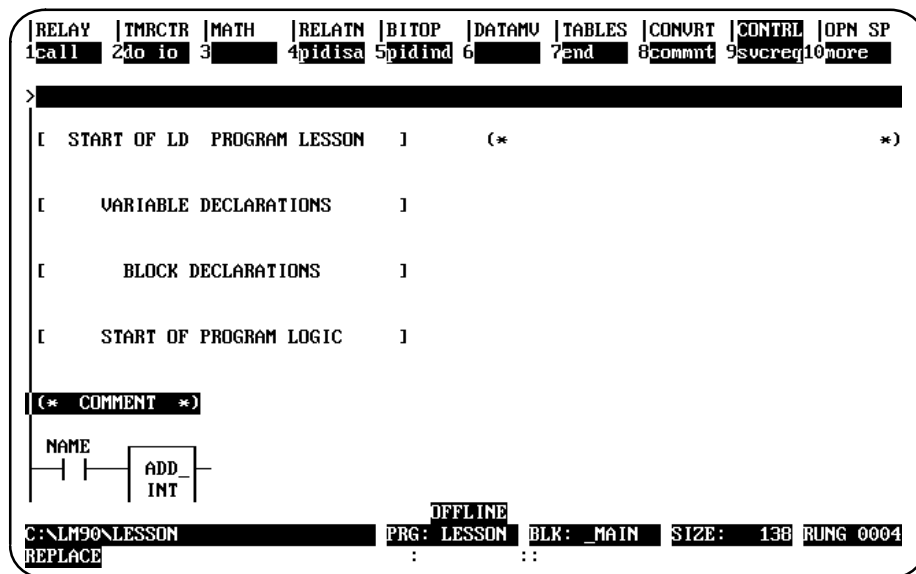
Each COMMENT instruction has a unique text block associated with it. Pasting/including a comment will create a copy of the text block for each COMMENT instruction pasted. The text for any COMMENT instruction is edited independently of any other COMMENT instruction. This allows you to copy a COMMENT instruction by cutting and pasting the text; then, you can edit the pasted text. Refer to chapter 3, section 5, "Variable Declaration Table," for information on cutting and pasting variable declarations.

Release 1 and Release 2 of Logicmaster 90-30/20/Micro software permitted two or more comments to be associated with one block of text. If you have such duplicate comments and want to separate them, write the entire block contents to a side file and delete all the logic. Then, include the side file into the empty block. Now, each of the duplicate comment locations will have its own copy of the text.

### Inserting a Rung Comment

To insert rung comments, move the cursor to the rung you wish to insert a comment before.

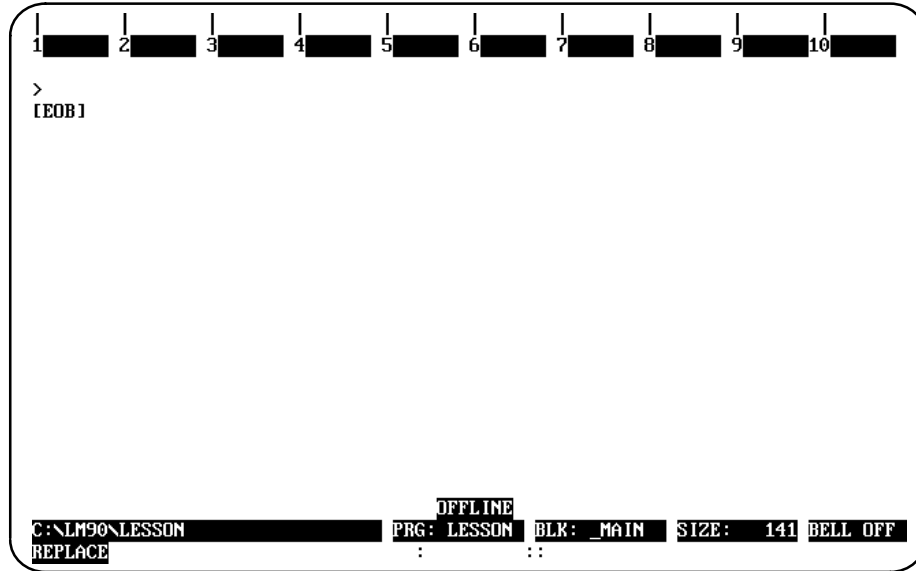
1. Press **Insert (F1)** and then **Control (Shift-F9)** to select the control functions.
2. Press **F8** to select the COMMENT function, or use the mnemonic by typing: **&COMMENT** on the command line and pressing the **Enter** key. The screen should appear as shown below:



3. Press the **Escape** key to accept the rung and exit **INSERT** mode.

## Adding Text

Text can be entered into the COMMENT instruction by positioning the cursor on the COMMENT instruction and pressing **Zoom (F10)**.



This is a simple full-screen editor, which allows you to input your rung description. Up to 2048 characters of text are permitted.

The available keys in the comment editor include:

Key	Description
Cursor keys	Move the cursor within the rung.
Page Up key	Move the cursor up one page.
Page Down key	Move the cursor down one page.
Insert key	Change the text editing mode ( <b>INSERT</b> or <b>REPLACE</b> ).
Delete key	Delete the character at the cursor position.
Backspace key	Delete the character to the left of the cursor position.
Home key	Position the cursor on the first character of the first line of the comment text.
End key	Position the cursor at the end of the comment text.

### User-Defined Footers in Listings

You can define up to four (4) lines of text to be printed at the bottom of each listing page generated by the Logicmaster 90-30 Programmer printing utility. Specify these lines of text using comment directives similar to the existing title, subtitle and border directives (\T\S, and \B respectively).

To use create a footer for a program file, follow these steps:

1. Follow the steps described on the previous pages to create a "COMMENT."
2. On a separate line at the beginning of the comment text, key in \F1 (for Footer 1); then key in the text you wish to appear in the footer. (You can enter up to four footers in this manner, using F1, F2, F3, and F4 as the footer line directives for footers 1 through 4 respectively. You can enter them in any order, but each must be on a separate line and must be at the beginning of the comment text in which they are entered. Regardless of the order entered, they will always print with Footer 1 before Footer 2, Footer 2 before Footer 3, etc.)

A sample of this is shown below:

```

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
>
\F1   Issued by:  DB           Introduced:  EC Number: 14012
\F2   Approved:  LM           Latest:     EC Number: 14013
\F3   xxxx This is a sample program xxxx
[EOB]

OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: _MAIN SIZE: 126 BELL OFF
REPLACE : :
```

3. When you are finished with your footers, press the **Escape** key to return to your program and accept that rung.

#### Note

Footer 4 supercedes the standard LM90 listing page footer line. If \F4 is not specified, the standard Logicmaster™ 90 footer is printed. If the \F4 directive is specified without text, the LM90 footer line will not be printed.

The following text shows the footer that resulted from the comment/footer settings shown on the previous page:

```
          Issued by: DB          Introduced: EC Number: 14012
          Approved: LM          Latest:   EC Number: 14013
          xxxx This is a sample program xxxx
Program: STATMNG          C:\LM90\STATMNG          Block: _MAIN
```

The following restrictions should be observed:

- Any blank footer lines will be printed in the listing as blanks.
- If text for a specified footer line has been previously defined and that footer line directive is specified again, but without text, that footer line will no longer be printed.
- Once specified, footer lines will be printed on each page of the listing until they are changed with user-defined footer line directives in subsequent comment text.
- For an 80-column listing, you can enter text up to 80 characters in length. For a 132-column listing, you can enter up to 132 characters.
- Footer line text less than the page width is centered to the page.
- To start user-defined footers on the pages prior to the logic, the comment containing the user-defined footers must be the first instruction in the main block of the program.





## Starting a New Page of Comments

To print a new page of comments:

1. Create the text for the comment.
2. Enter `\P` or `\p`, as lowercase alphanumeric characters are also recognized, on a line by itself.

The text following the `\P` character will then begin on a new page.

The `\P` can be used several times within a comment, with text before and after each one.

## Printing a Title

When the printer output is defined as described in chapter 9, a title and subtitle for the printout are created. This title and subtitle will appear on every page, unless changed as described below:

1. At the place in the program where the new title and subtitle should begin, enter a comment in the logic. This will be printed as the beginning of a new page.
2. On the first line of the comment, enter the following:
  - A. `\T` or `\t` with the new title (up to 62 characters) on the same line.
  - B. `\S` or `\s` with the new subtitle (also up to 62 characters) on the same line.

The new title and subtitle will appear on each subsequent page. `\T` or `\S` alone can be specified. `\B`, `\T`, or `\S` must be on separate lines. Those lines must be first in a comment, but they can be in any order.

## Creating Longer Comments

The maximum number of characters that can be included in a comment is 2048. Longer text can be included in printouts (but not displayed in program function or mode) using an annotation text file, as described below:

1. Create the comment as described on the previous page.
  - A. Enter text to the point where the text from the other file should begin.
  - B. Move the cursor to the beginning of a new line and enter `\I` or `\i`, the drive followed by a colon, the subdirectory or folder, and the file name, as shown in this example:

```
\I d:\text\commnt1
```

The drive designation is not necessary if the file is located on the same drive as the program folder.

- C. Continue editing the program, or exit to MS-DOS.
2. After exiting the programmer, create a text file using any MS-DOS compatible software package. Give the file the file name in the comment, and place it on the drive specified in the comment.

## Section 7: Changing the Display Mode

The view mode setup function, described in chapter 6, “Programmer Setup,” enables you to specify which modes are displayed when you press **ALT-N**. These view modes range from showing only rung references to showing reference names and reference descriptions in an expanded rung form (display all mode). You can also view the maximum amount of program logic on a screen by selecting a compressed rung mode.

Display all mode shows reference description information in four 7-character segments at each occurrence of an identifier or reference address in the program editor. The Variable Declaration Editor displays a 4 x 7 character window with the reference description for the current entry.

### Note

The display mode (ALT-N) cannot be changed during **RUNG EDIT** or **RUNG INSERT** mode.

Each time **ALT-N** is pressed, the editor display will move to the next display mode. The five display modes are listed below.

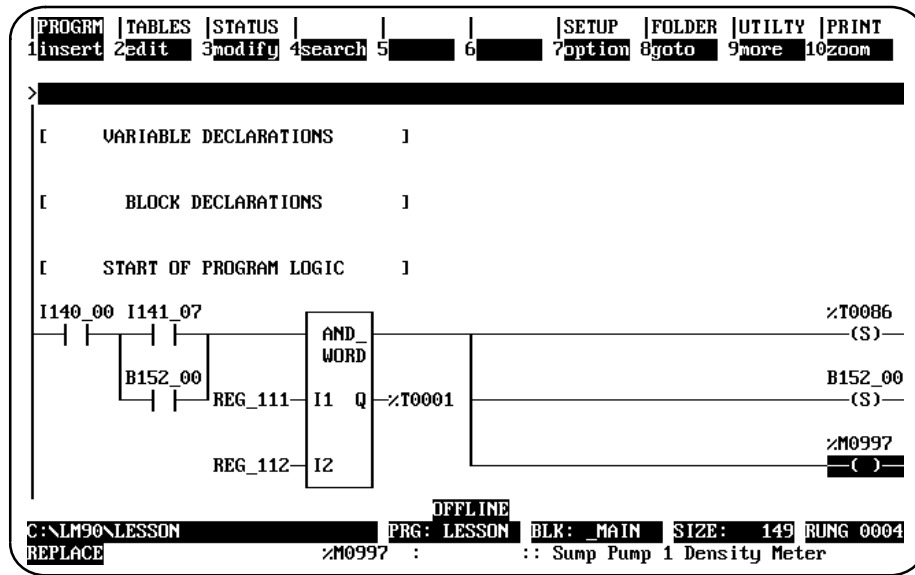
Display All Mode	Description	Lines / Rung
A	Reference descriptions off, nicknames on.	3 screen lines per rung line.
B	Reference descriptions off, nicknames off.	3 screen lines per rung line.
C	Reference descriptions on, nicknames on.	7 screen lines per rung line.
D	Reference descriptions on, nicknames off.	7 screen lines per rung line.
E	Compressed rung mode.	1 screen line per rung line.

Mode A is the default mode. Each time **ALT-N** is pressed, the mode changes to the next mode (e.g., mode A to B). If **ALT-N** is pressed while mode E is displayed, the display cycles back to mode A. Refer to the information on “View Modes Setup (ALT-N)” in chapter 6, “Programmer Setup,” in order to select the modes displayed when **ALT-N** is pressed.

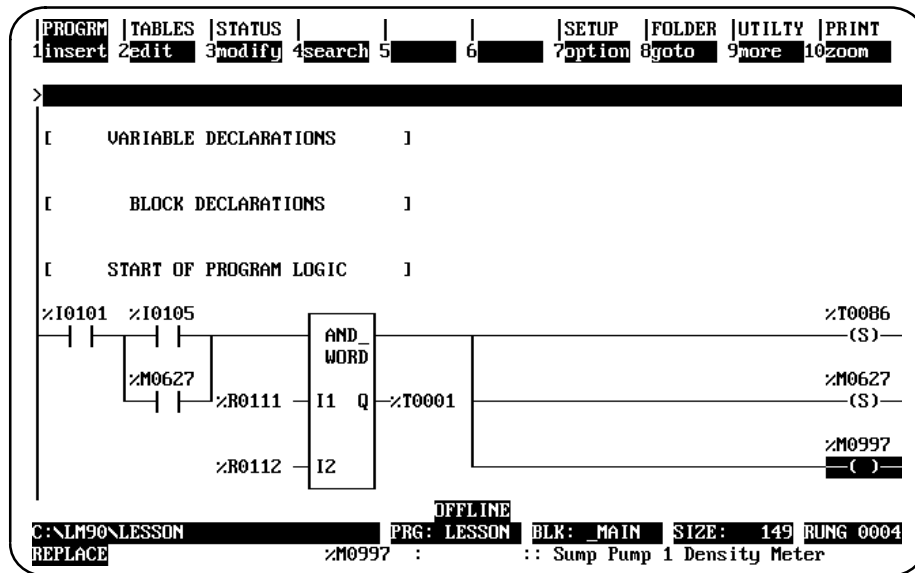
### Note

The mode identifier letters A, B, C, D, and E are used for reference in this document only.

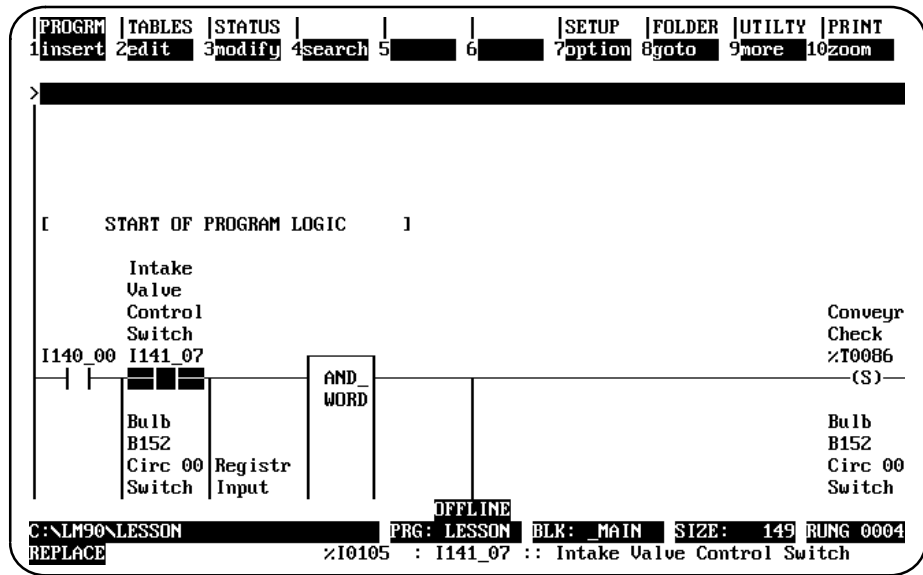
Display Mode A: Reference descriptions off, nicknames on.



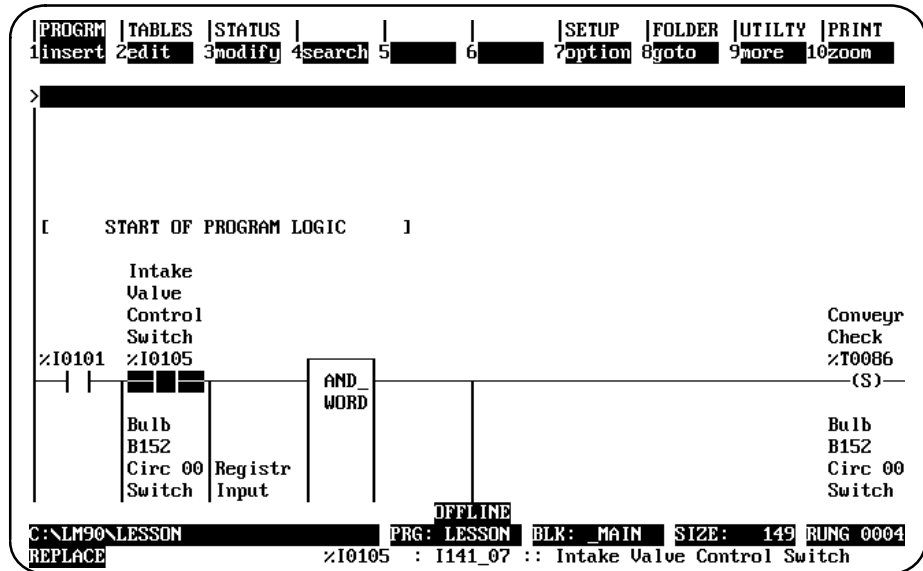
Display Mode B: Reference descriptions off, nicknames off.



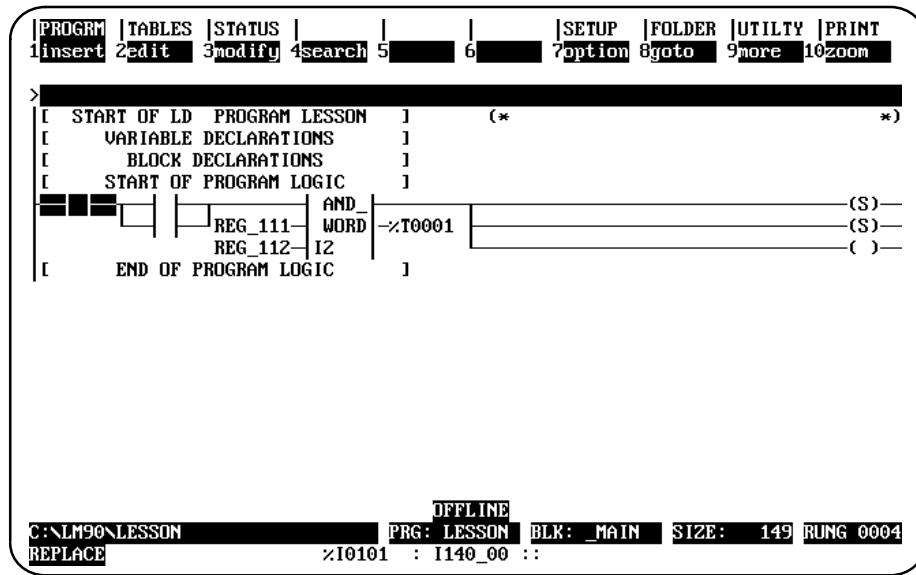
Display Mode C: Reference descriptions on, nicknames on.



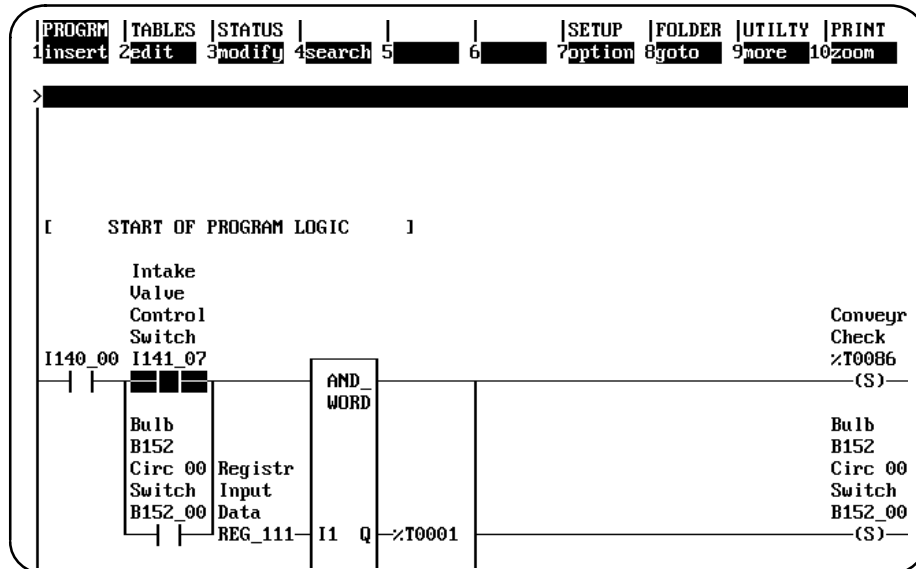
Display Mode D: Reference descriptions on, nicknames off.



Display Mode E: Compressed rung mode.



Display Mode D without status lines (press ALT-E to remove the status lines):



The following screen shows how the reference description breaks within the 4-line x 7-character window when the cursor is on %I0105.

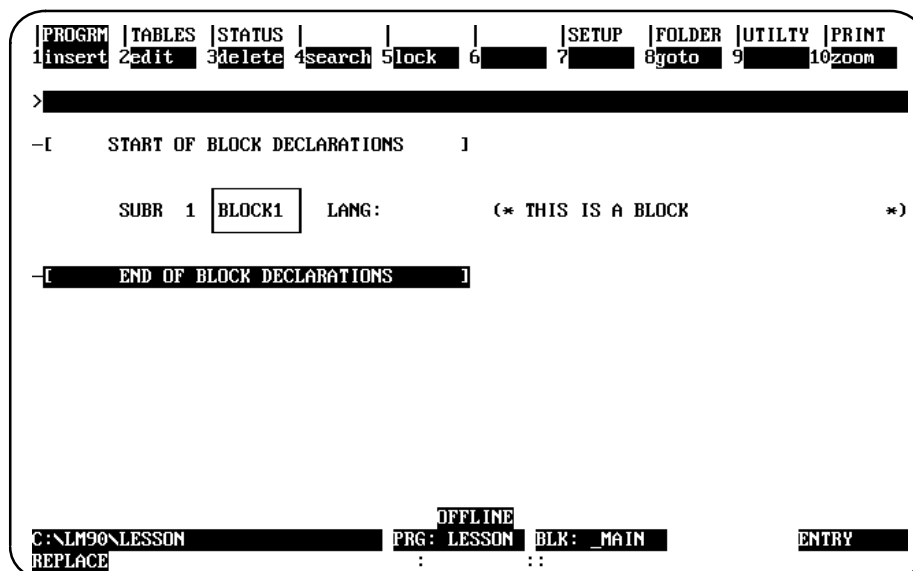
```

|PROGRAM | TABLES | STATUS | | | | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3delete 4search 5sort 6 7 8goto 9region10switch
>
  V A R I A B L E   D E C L A R A T I O N   T A B L E
  REFERENCE  NICKNAME  REFERENCE DESCRIPTION
  -----
  %I0001     DWELL_T   weld dwell timer
  %I0033     LIMIT_T
  %I0037     FREE       spare input; not wired
  %I0101     I140_00
  %I0102     I140_01   Air Intake Valve Switch
  %I0105     I141_07   Intake Valve Control Switch
  %Q0202     START     machine start coil
  %M0627     B152_00   Bulb B152 Circ 00 Switch
  %M0997     Sump Pump 1 Density Meter
  %T0086     ConveyrCheck
  %R0111     REG_111  Registr Input Data
  %R0112     REG_112  Pointer to Read Data
  OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: _MAIN ENTRY 0010
REPLACE : ::
  
```

## Section 8: Subroutine Blocks

Subroutine blocks in Logicmaster 90-30/20/Micro software provide structured programming for the Series 90-30 PLC. Subroutine blocks are *not* available for the Series 90-20 PLC nor for Micro PLCs. For 90-30 PLCs, up to 64 subroutine block declarations are allowed.

Subroutines are declared through the block declaration editor. To create or modify subroutine declarations, place the cursor on the [ BLOCK DECLARATIONS ] marker. Then, press **zoom** (F10) to display the block declarations screen.



Subroutine block declarations consist of:

- A label identifying the block as being used with a subroutine instruction.
- The number corresponding to the Hand Held Programmer’s subroutine number.
- A graphic box containing a subroutine name of up to 7 alphanumeric characters.
- The language the subroutine was programmed in. Initially, this field is blank. It will remain empty until you have entered some logic for the subroutine. Once you have zoomed into the block to enter logic, the letters **LD** (ladder diagram language) are displayed after **LANG:** beside the block name.
- A 32-character descriptive explanation.
- An indicator that it has errors. This indicator is only displayed if the block is not executable. (See screen capture at top of the next page for an example of the error message.)

The following screen shows an example of two subroutine block declarations, one without errors and one with errors.

```

|PROGRAM | TABLES | STATUS | | | | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3delete 4search 5lock 6 | 7 | 8goto 9 | 10zoom
>
-[ START OF BLOCK DECLARATIONS 1
    SUBR 2 SHIP_IT LANG: LD (* prepare for shipping *)
    SUBR 1 MAKE_IT LANG: LD (* make widgets *)
    HAS ERRORS
-[ END OF BLOCK DECLARATIONS 1

OFFLINE
C:\MLM90\LESSON PRG: LESSON BLK: MAIN ENTRY 0001
REPLACE : SHIP_IT :: prepare for shipping

```

## Adding Subroutine Block Declarations

To add a subroutine block declaration, place the cursor at the desired location, enter the block name on the command line, and press **Insert** (F1). A 32-character explanation can also be inserted at this time. If you press F1 with the command line blank, a box containing seven question marks (???????) will be displayed for the subroutine block.

When the subroutine block declaration is inserted in the software, the next available Hand Held Programmer number (from 1 to 64) is assigned to it.

## Editing Subroutine Block Declarations

To change a subroutine block declaration, place the cursor at the declaration to be changed and press **Edit** (F2). Type over the entry as needed. After changing the entry, press the **Enter** key (or Plus (+) key on the numeric keypad) to move to the next table entry and continue editing; or press the **Escape** key to exit editing.

Use the **Tab**, **Back Tab**, **Previous**, or **Next** key, or the cursor keys to move the cursor. Use the **Page Up** and **Page Down** keys to scroll the display up and down.

When you leave the subroutine block declarations, any changes made are automatically stored to the current program folder. You can also update the program folder while working on the screen by pressing **ALT-U**.



## Deleting Subroutine Block Declarations

A subroutine block declaration can be deleted if there are no CALL instructions to that subroutine block in the program logic. Deleting a subroutine block also deletes its associated logic. To delete a subroutine block and declaration, place the cursor at the declaration to be deleted and press **Delete** (F3).

## Searching for Subroutine Block Declarations

In order to search for a subroutine block declaration, you must have the Block Declaration screen displayed on your programmer. Then, press **Search** (F4). Enter the name of the subroutine block in the **Search for** field, set the **Scope** to LOCAL, and press the **Enter** key.

You can search for a subroutine CALL instruction from either the logic or the Block Declaration screen by pressing **Search** (F4). Then, enter the name of the subroutine block in the **Search for** field, or enter **&CALL** to search for all subroutine CALL instructions, and press the **Enter** key.

## Using Goto

The **Goto** (F8) function key may be used to move the cursor. To move to a particular subroutine block declaration, enter the number of that declaration on the command line, and press **Goto** (F8). For example, to move the cursor to the first subroutine block declaration, enter **1** on the command line, and press **Goto** (F8).

You may also go to a subroutine block by simply entering the block name on the command line and pressing **Goto** (F8).

## Zooming into Subroutine Block Logic

To display the subroutine block logic, place the cursor on the block name and press **Zoom** (F10). You can edit the logic on this screen.

## Locking/Unlocking Subroutines

The block locking feature allows you to lock subroutines. Four types of locks are available:

Type of Lock	Description
View	Once locked, you cannot zoom into that subroutine.
Edit	Once locked, the information in the subroutine cannot be edited.
Perm View	The subroutine is permanently locked and cannot be unlocked.
Perm Edit	The subroutine is permanently locked and cannot be unlocked.

In addition to the locking capability, locked subroutines can also be unlocked, unless they are permanently locked.

A search or search and replace function may be performed on a view-locked subroutine. If the target of the search is found in a view-locked subroutine, one of the following messages is displayed, instead of logic:

For view-locked subroutines:

`Found in locked block. (Continue/Quit)`

For edit-locked subroutines:

`Cannot write to locked block. (Continue/Quit)`

You may continue or abort the search. If you decide to continue, the locked subroutine is skipped and the search continues from the next subroutine. If you decide to quit, the search is aborted. For more information on search and search/replace, refer to chapter 3, section 11, "Search Function."

Folders that contain locked subroutines may be cleared or deleted. If a folder contains locked subroutines, these blocks remain locked when the Logicmaster 90-30/20/Micro software copy, backup, and restore folder functions are used. For more information on program folders, refer to chapter 7, "Program Folders."

## Locking a Subroutine

To lock a subroutine:

1. Move the cursor to the [ BLOCK DECLARATIONS ] marker and press Zoom (F10).

```
PROGRAM | TABLES | STATUS | | | | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3delete 4search 5lock 6 7 8goto 9 10zoom
>
-[ START OF BLOCK DECLARATIONS ]
      SUBR 1 BLOCK1 LANG: (* THIS IS A BLOCK *)
-[ END OF BLOCK DECLARATIONS ]

OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: MAIN ENTRY 0001
REPLACE : BLOCK1 :: THIS IS A BLOCK
```

2. In the block declaration editor, move the cursor to the desired subroutine and press Lock (F5) to display the Lock/Unlock Block screen. The subroutine name is included as part of the title on this screen.

```
PROGRAM | TABLES | STATUS | | | | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3delete 4search 5lock 6 7 8goto 9 10zoom
>
-[ S Lock/Unlock Block : BLOCK1 *)
      Lock State : UNLOCK
      Password :
      << Press Enter key to begin lock function >>
      << Press Esc key to leave the lock screen >>

OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: MAIN ENTRY 0001
REPLACE : BLOCK1 :: THIS IS A BLOCK
```

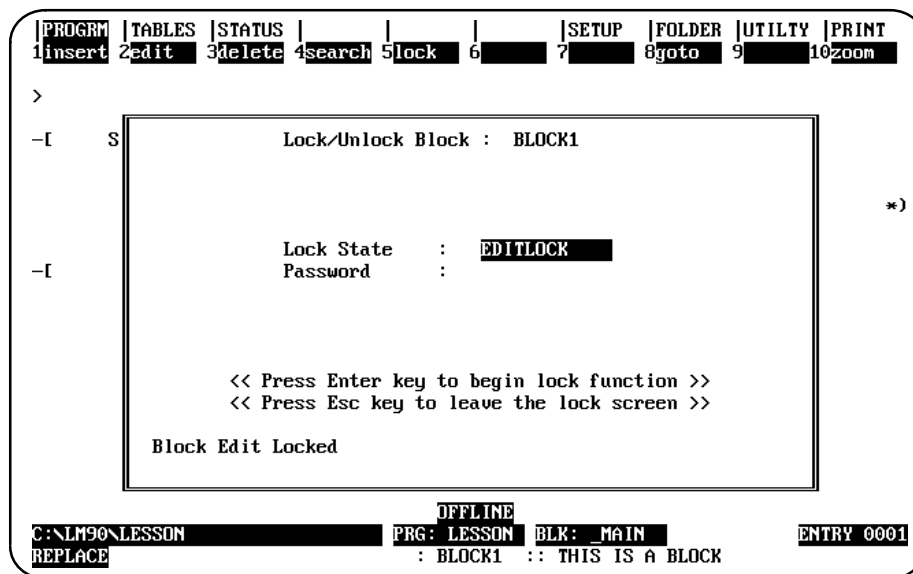
Field	Description
Lock State	<p>Specify the operation to be performed on the current subroutine. Use the <b>Tab</b> key to view the values for this field:</p> <ul style="list-style-type: none"> <li>• <b>UNLOCK</b>: The current subroutine is unlocked.</li> <li>• <b>VIEWLOCK</b>: A view lock is set on the subroutine, and you cannot zoom into the subroutine. If you try to zoom into the subroutine, the error message “Zoom denied because the block is locked” is displayed.</li> <li>• <b>EDITLOCK</b>: An edit lock is set on the subroutine, and the subroutine cannot be changed. If you try to edit the subroutine, the error message “Edit denied because the block is locked” is displayed.</li> <li>• <b>PERMVIEWLOCK</b> or <b>PERMEDITLOCK</b>: The subroutine is <b>permanently</b> locked and cannot be unlocked once it is locked.</li> </ul> <p>When the desired value is displayed in the <b>Lock State</b> field, press the <b>Enter</b> key. If you try to enter an incorrect value or leave the field empty, a message is displayed in the message area of the screen and the field remains active.</p> <p>If you try to lock a subroutine that is already locked, the message “Block already locked” is displayed in the message area and the screen remains displayed.</p>
Password	<p>Specify a password of up to four characters to lock a subroutine. Valid characters include A through F and 0 through 9. If a lowercase letter is entered, it is converted to an uppercase letter. Once a password is set, the same password must be entered before the subroutine can be unlocked. The characters of the password are displayed as they are typed into the <b>Password</b> field.</p> <p>For <b>PERMVIEWLOCK</b> and <b>PERMEDITLOCK</b>, the <b>Password</b> field is ignored. Passwords are not required and cannot be used with permanently locked blocks.</p>

3. If no lock is set on the current subroutine, the **Lock State** field is set to **UNLOCK** when the screen is displayed. If a lock is set on the current subroutine, then the **Lock State** field is set to the type of lock imposed on the block. Initially, the **Password** field is empty.
4. To set the lock, enter the value in the **Lock State** field. Move the cursor to the **Password** field, and enter a password. Then, press the **Enter** key. The software will prompt you to confirm the locking process with the message “Is logic block to be locked?(Y/N)”.
5. If you enter **Y** (Yes) after the confirmation prompt, the locking process begins. Entering **N** (No) aborts the process. Once the subroutine is successfully locked, the message “Block locked” is displayed in the message area of the screen, the type of lock and password are written to the subroutine lock header, and the **Password** field is cleared.
6. To quit the Lock/Unlock Block screen, press the **Escape** key. To restore the value of the fields to their original values, press **ALT-A**.

## Unlocking a Subroutine

A previously view-locked or edit-locked subroutine may be unlocked in the block declaration editor, unless it is permanently view locked or permanently edit locked.

1. Move the cursor to the desired subroutine block and press **Lock (F5)** to display the Lock/UnlockBlockscreen.



The current subroutine name is included as part of the title on this screen. If the current subroutine is locked, the **LockState** field indicates the type of lock. If the subroutine is not locked, the field is set to **UNLOCK**. The **Password** field is initially empty.

2. To unlock the subroutine, set the value of the **LockState** field to **UNLOCK**.
3. Move the cursor to the **Password** field, and enter the correct password. As each character of the password is typed, an asterisk is displayed in the field.
4. Then, press the **Enter** key. The password you entered is compared with the password last set. If the two passwords are identical, the subroutine is unlocked and the message “Block unlocked” is displayed in the message area of the screen. Once the subroutine is unlocked, the **LockState** field is set to **UNLOCK** and the **Password** field contains the password of the block just unlocked. In addition, the password is cleared in the subroutine header.
5. If the passwords do not match, the subroutine is not unlocked, the error message “Incorrect password given for unlock” is displayed in the message area of the screen, the **Password** field is cleared, and the Lock/Unlock Block screen remains displayed.
6. To quit the Lock/Unlock Block screen, press the **Escape** key. To restore the value of the fields to their original values, press **ALT-A**.

## Permanently Locking a Subroutine

In addition to **VIEWLOCK** and **EDITLOCK**, there are two types of permanent locks. If a **PERMVIEWLOCK** lock is set, all zooms into a subroutine are denied. If a **PERMEDITLOCK** lock is set, all attempts to edit the block are denied. Therefore, passwords are not required and cannot be used with permanently locked subroutines.

### Warning

**Permanent locks differ from the regular **VIEWLOCK** and **EDITLOCK** in that once set, they cannot be removed.**

Once a **PERMEDITLOCK** is set, it can only be changed to a **PERMVIEWLOCK**; you cannot unlock the block first and then set a **PERMVIEWLOCK**. A **PERMVIEWLOCK** cannot be changed to any other type of lock.

When you press the **Enter** key to initiate the locking, the software will prompt you to confirm any permanent lock.

## Displaying the Lock Status of the Subroutine

The display zoom level function (**ALT-X**) can be used to display the lock status of the subroutine in the block declaration editor. Move the cursor to the desired block, and press **ALT-X**. The following example screen shows a block that is locked for view.

```

|PROGRAM |TABLES |STATUS | | | | |SETUP |FOLDER |UTILITY |PRINT
1insert 2edit 3delete 4search 5lock 6 7 8goto 9 10zoom
[ZOOM LEVEL 1] [Block Edit Locked]
>
-[ START OF BLOCK DECLARATIONS ]
      SUBR 1 [BLOCK1] LANG:      (* THIS IS A BLOCK *)
-[ END OF BLOCK DECLARATIONS ]

OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: MAIN ENTRY 0001
REPLACE : BLOCK1 :: THIS IS A BLOCK

```

## Periodic Subroutines

Model 340 and higher CPUs support the use of a periodic subroutine. A periodic subroutine is a single subroutine with a unique name in the form `1Tiiii`, which will execute periodically during `RUN` mode. If a subroutine with this name is present in the PLC when it transitions from `STOP` to `RUN` mode, this subroutine will be executed at periodic intervals while the PLC is in `RUN` mode. If a `RUN MODE STORE` is performed, a periodic subroutine will be stopped while the PLC is in `PAUSE` mode. (For more information on `RUN MODE STORE` and `PAUSE` mode, refer to chapter 8, “Program Utilities.”)

### Note

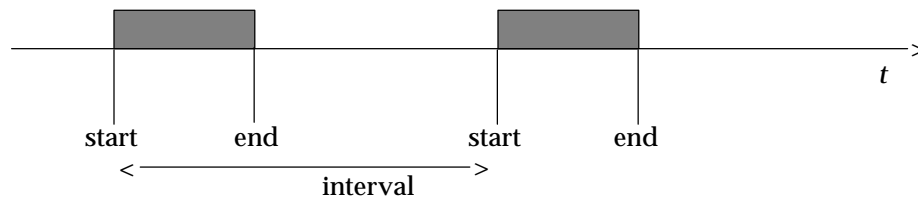
**Only** Model 340 and higher Series 90™ -30 PLCs support use of a periodic subroutine. More specifically, Model 341 CPUs, Version 4.20 or later, all Model 340 CPUs, and all 350 and higher CPUs support periodic subroutines.

You can specify a periodic subroutine by giving the block a name in the form: `1T0001`, where the four digits after the `T` indicate the number of time intervals between executions of the block. If the leading zeros of the interval are not specified, the Logixmaster 90-30/20/Micro software will fill them in so that the total number of characters is six. If the time interval specified is too small for execution of the subroutine and the rest of the normal PLC sweep, the PLC watchdog timer may be activated. The time per interval is .001 seconds. The maximum time allowed between executions is 10 milliseconds.

When you enter a subroutine name with the correct format for a periodic subroutine, the timebase, interval value, and “INTR” will be displayed beside the subroutine’s name and number, instead of “SUBR.”

### Executing a Periodic Subroutine

Each execution of the periodic subroutine will occur interval seconds after the previous start, as shown below:



### Note

The latency for the periodic subroutine (i.e., the maximum interval between the time the periodic subroutine should have executed and the time it actually executes) can be around .35 milliseconds if there is no PCM, CMM, or ADC module in the main rack. If there is a PCM, CMM or ADC module in the main rack — even if it is not configured or used — the latency can be almost 2.25 milliseconds. For that reason, use of the periodic subroutine with PCM-based products is not recommended.

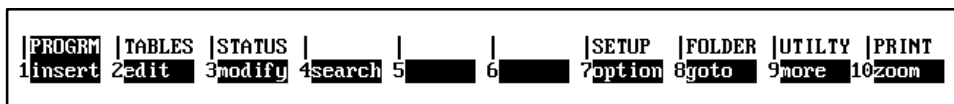
### Restrictions on Use of the Periodic Subroutine

- Only one periodic subroutine per program is allowed.
- Periodic subroutines cannot be called by the main program or by another subroutine.
- If you call another subroutine from the periodic subroutine, the subroutine being called must not be called from anywhere else in the program.
- If you attempt to load a program with a periodic subroutine into a version of Logicmaster 90-30 software prior to Release 4.01, you will not be able to display or edit the program. Likewise, you cannot store a periodic subroutine to a PLC prior to Version 4.20.
- If a DOIO function block whose I/O reference range includes an intelligent module is executed within a periodic subroutine, communication with the module may be lost.
- Timer (TMR, ONDTR, and OFDTR) function blocks will not execute properly within a periodic subroutine. A DOIO function block within a periodic subroutine whose reference range includes references assigned to a Smart I/O Module (HSC, APM, Genius, etc.) will cause the CPU to lose communication with the module. The FST\_SCN and LST\_SCN contacts (%S1 and %S2) will have an indeterminate value during execution of the periodic subroutine. A periodic subroutine cannot call or be called by other subroutines.
- The PID function block depends in its implementation on the sweep time of the main program. Consequently, it will not function properly when called from a periodic subroutine.



## Section 9: Rung Edit

After a logic program has been entered, there is always a need to make modifications, either to correct logic errors or to add new capabilities. Logicmaster 90-30/20/Micro software has an array of features to make this process easy. The same function keys used to initially enter a rung are available to insert new rungs. In addition, there are function keys to edit or modify existing rungs.



Function Key	Function	Description
F1	Insert	Add one or more rungs to the program. Selecting the insert function opens a new space above the rung where the cursor is positioned and displays the Edit Rung keys, which are used to select program elements. This allows you to create a new rung.
F2	Edit	Edit the rung at the cursor location. The edit function activates the current rung, enabling you to modify that rung. When <b>F2</b> is pressed, the Edit Rung keys are displayed at the top of the screen. These keys are used to select program elements.
F3	Modify	Initiate word-for-word instruction changes using the rung edit softkeys instead of mnemonics.
F4	Search	Locate a program element.
F7	Option	Access coil checking and other editor options.
F8	Goto	Go to the specified rung. To use this feature, type the rung number you want to go to, then press <b>F8</b> .
F9	More	Access additional rung edit softkeys.
F10	Zoom	Go to a more detailed level. To return to the original level, press the <b>Escape</b> key.

Pressing **More (F9)** displays these additional rung edit softkeys.



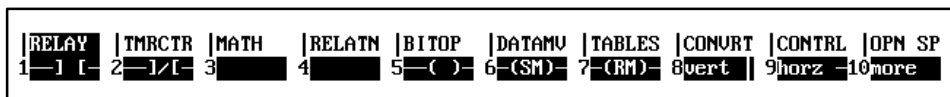
Function Key	Function	Description
F1	Select	Select or deselect a rung or group of rungs. Rungs or declarations may also be deselected by pressing <b>F1</b> again, by pressing <b>ALT-A</b> , or by pressing the <b>Escape</b> key. After deselecting the rungs or declarations, the message "Select mode cancelled" is displayed on the message line and the cursor remains on the last rung or declaration that had been selected.
F2	Cut	Delete the selected section of rungs or declarations from a program and put them in the cut buffer. After a cut operation, the cursor will appear on the rung or declaration immediately after the selected region.
F3	Paste	Insert previously cut rungs or declarations from the cut buffer. Cut rungs and declarations may be pasted any number of times, until they are replaced by new information in the cut buffer, or until the program editor is exited.
F4	Include	Insert previously written rungs or declarations back into the original program, or into any other program. Written rungs and declarations may be included any number of times.
F5	Write	Copy the selected section of rungs or declarations from a program into a special file. After a write, the cursor remains on the last rung or declaration selected.
F6	Delete	Delete the rung at the cursor location or a range of selected rungs.
F8	Goto	Enter the rung number on the command line, and then press <b>F8</b> to "go to" a specific rung in the ladder diagram logic. You can also specify a subroutine number and rung number in the subroutine to "go to" a specific rung in that subroutine.
F9	More	Return to the first level of rung edit softkeys.
F10	Zoom	Zoom into the item the cursor is on.

### Note

The function keys listed above can also be applied to the variable declaration table. Using the file write function, a group of reference descriptions can be written to a file to be used by another program.

## Editing a Rung

When either the insert, edit, or modify function is activated by pressing **F1**, **F2**, or **F3**, respectively, the following softkey selections are displayed at the top of the screen.



These function keys provide access to the instructions required to edit ladder diagram rungs.

### Note

Only one rung is active at a time in either the insert or edit function. Each rung must be completed **and** accepted by the software before the next rung can be edited.

## Entering Insert or Edit Mode

Enter either **INSERT** or **EDIT** mode by:

1. Pressing **F1** to enter **INSERT** mode. In this mode, the new rung is inserted before the rung on which the cursor is positioned. Therefore, make sure the cursor is on the rung following the location for the new rung before you press **F1**.
2. Pressing **F2** to enter **EDIT** mode. Then, move the cursor to the rung element you want to change.

## Entering Instructions

Select instructions by:

1. **After entering Insert or Edit mode** (as discussed above), typing the mnemonic for the instruction, preceded by an ampersand character (&). (Refer to appendix D, *Instruction Mnemonics*, for a listing of the mnemonics for Logicismaster 90-30/20/Micro programming software.) Then, press the **Enter** key.
2. **After entering Insert or Edit mode** (as discussed above), pressing the **Shift** key and a function key to display a specific group of instructions. Then, select a specific instruction within that group by pressing its function key.

In this example, the ADD function is selected by typing the mnemonic: **&ADD** on the command line.

```

|RELAY |TMRCTR |MATH |RELATN |BITOP |DATAMU |TABLES |CONVRT |CONTRL |OPN SP
1-1 L- 2-1/L- 3- 4- 5-( )- 6-(SM)- 7-(RM)- 8vert | 9horz -10more
>&ADD
[ START OF PROGRAM LOGIC ]
(* COMMENT *)
NAME
| | | ADD
| | | INT
IN_REG - I1 Q -%R0002
CONST - I2
+00001
%I0002
| | |
OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: MAIN SIZE: 141 RUNG 0005
REPLACE : ::

```

After pressing the **Enter** key, the screen adjusts to display the complete new instruction and the ADD function is inserted.

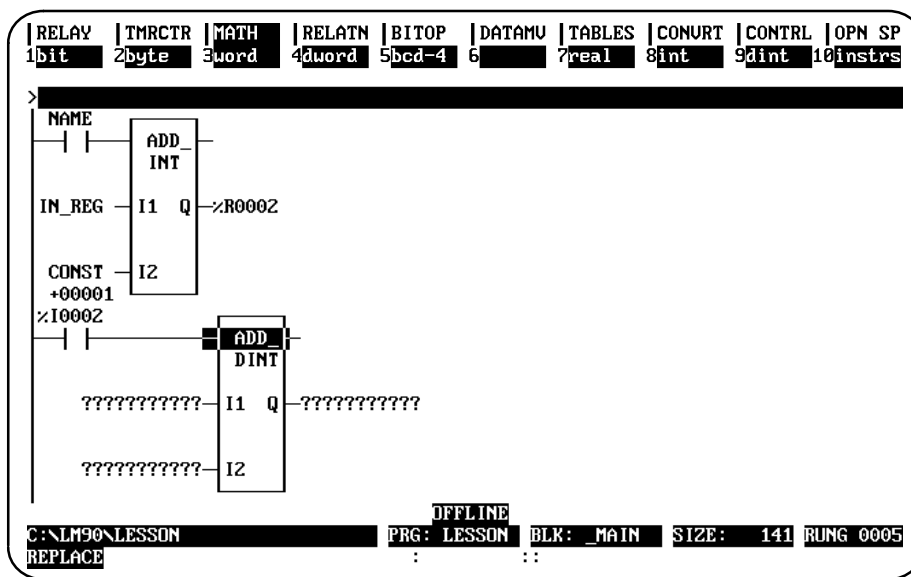
```

|RELAY |TMRCTR |MATH |RELATN |BITOP |DATAMU |TABLES |CONVRT |CONTRL |OPN SP
1-1 L- 2-1/L- 3- 4- 5-( )- 6-(SM)- 7-(RM)- 8vert | 9horz -10more
>
NAME
| | | ADD
| | | INT
IN_REG - I1 Q -%R0002
CONST - I2
+00001
%I0002
| | | ADD
| | | INT
???????? I1 Q -????????
???????? I2
OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: MAIN SIZE: 141 RUNG 0005
REPLACE : ::

```

## Entering/Modifying Data Types

Some instructions support different data types. To change ADD\_INT to ADD\_DINT, press **Types** (F10) with the math functions still displayed on the screen. Then, press **DINT** (F9) with the cursor on the ADD function block to create the following screen. An alternative way to select ADD\_DINT is to type `&ADD_DINT` on the command line, and then press the **Enter** key.



## Moving the Cursor within a Rung

Use the keys listed in the table below to move within the rung.

Key	Description
Cursor keys Tab and Shift-Tab Home key End key	Move the cursor within the rung. Move among the inputs and outputs of a function. Position the cursor on column 1 in row 1. Position the cursor on the last defined row.
<i>For Command Line Entry</i>	
Insert key Delete key Backspace key CTRL-Left Cursor or CTRL-Right Cursor	Change the text editing mode ( <b>INSERT</b> or <b>REPLACE</b> ). Delete the character at the cursor position. Delete the character to the left of the cursor position. Move the cursor within the command line.

### Entering a Reference Address

Enter a reference address on the command line. Then, press the **Enter** key to apply the reference to the operand at the current cursor location. Repeat this step for each operand in the instruction. For multiple operand instructions, you may find that pressing the **Tab** key is more efficient because doing so causes Logicmaster to apply the reference to the operand at the current cursor location *and* automatically moves the cursor to the next operand location awaiting your input.

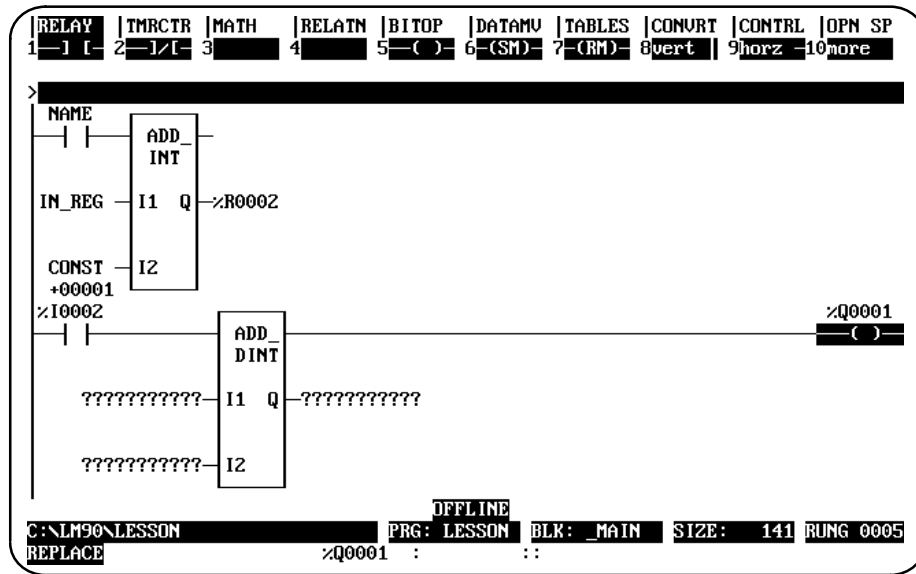
For single operand instructions, the reference address can be entered on the command line before pressing the function key (described in the previous step). Then, when the function key is pressed, the reference address is automatically applied to the instruction.

### Entering Nicknames

Nicknames can also be created or modified on the command line. To do this, enter the nickname together with its machine reference and any associated reference description. (Refer to chapter 3, section 4, "Program Annotation," for more information on entering nicknames and reference descriptions.)

### Using Vertical and Horizontal Links

Use **Vertical Links (F8)** and **Horizontal Links (F9)** to connect the instructions within a rung. These links are available from the relay functions, but may also be entered without returning to the relay function menu by pressing the Vertical bar (|) key to enter a vertical link or the Tilde (~) key to enter a horizontal link.

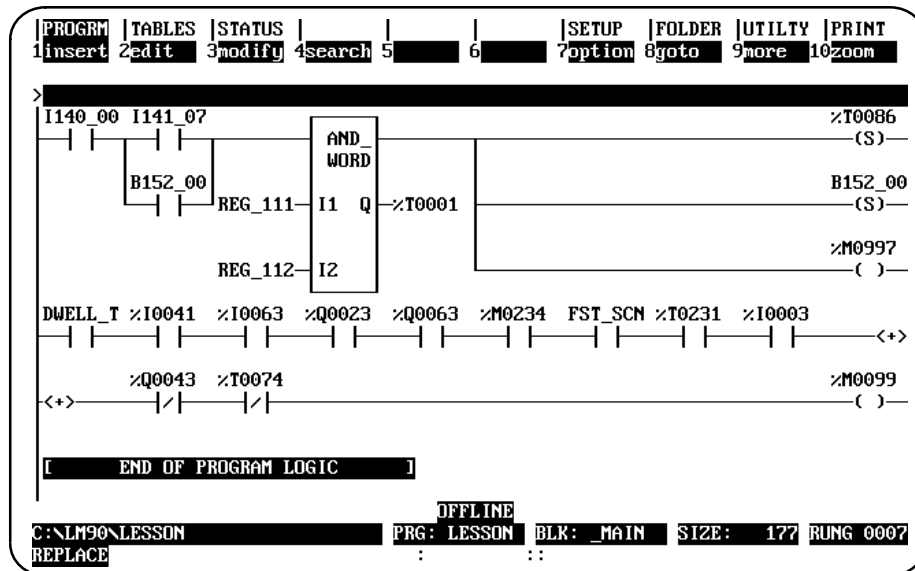


## Using Continuation Coils and Contacts

Use the **Continuation Coil (F1)** and the **Continuation Contact (F2)** to continue relay ladder rung logic beyond the limit of ten columns. These keys are available from the relay functions by pressing **More (F10)**; they may also be entered using their mnemonics (&COILCTD and &CONCTD). There can be only one continuation coil and/or one continuation contact per rung.

The **Continuation Coil (F1)** is used to carry the current rung's status to the rung which has a continuation contact. The continuation coil can only be placed in column 10 of the rung. It does not require the use of a continuation contact in following rungs for editing.

The **Continuation Contact (F2)** is used to continue the status of the continuation coil rung's logic on the continuation contact's rung.

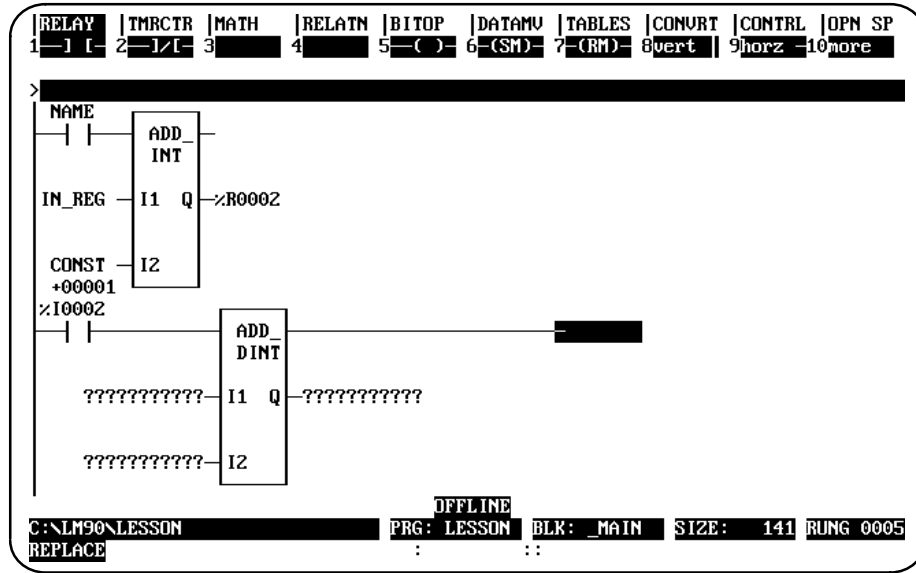


The state of the last executed continuation coil is the flow state that will be used on the next executed continuation contact. The continuation contact does not require the use of a continuation coil in previous rungs for editing. However, if the flow of logic does not execute a continuation coil before it executes a continuation contact, the state of that contact will be no flow.

## Deleting an Element

To delete an operand or instruction from a rung while in **INSERT** or **EDIT** mode, press **ALT-D** or **Open Space (Shift-F10)** and then press **Delete Instruction (F10)**. Enter another instruction in the space left by the deleted instruction, or enter a horizontal link. An instruction may also be deleted by replacing it with another instruction.

To remove the horizontal links and coil that follow the **ADD\_DINT** instruction, place the cursor on each logic segment and press **ALT-D**. The **Tilde (~)** key or **Horizontal Link (F9)** softkey can also be used to clear the link.



## Using Open Space Functions

To add open space to a rung in **INSERT** or **EDIT** mode, select **Open Space (Shift-F10)**. Then, select the specific function key described below:

Function Key	Function	Description	Page
F1	Move Right	Move the element at the cursor position, and all the elements to the right of the cursor position, one position to the right in all lines. Shunts are automatically inserted into the new column in the rung.	3-83
F3	Move Down	Create room for an additional line of logic above the line the cursor is on.	3-84
F5	Delete Column	Delete an entire column of instructions (including verticals) and operands within a rung. All elements to the right of the deleted column will automatically move left.	3-85
F7	Delete Row	Delete an entire row of instructions (including verticals) and operands within a rung.	3-86
F9	Delete Nickname	Remove nicknames from the variable declaration table. Enter the nickname to be deleted on the command line before pressing <b>F9</b> .	N/A
F10	Delete Instruction	Delete an operand or instruction from a rung. (You may also press <b>ALT-D</b> to delete an operand or instruction.)	3-87



### Completing (Accepting) Rung Entry

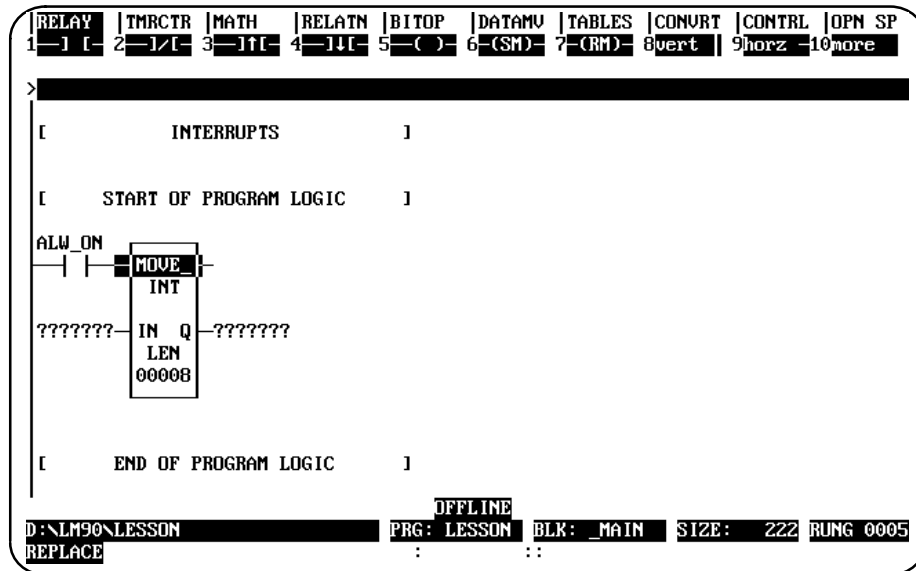
After changing the rung, press the **Enter** key with the command line empty (or **Plus** (+) key on the numeric keypad) to accept the edited rung and move the cursor to the next rung for editing. In **EDIT** mode, you must press the Edit (F2) key after the rung is accepted in order to edit the next rung. In **INSERT** mode, it is not necessary to press **Edit** (F2). You will still be in **INSERT** mode after the rung is accepted and can enter the next rung. You will remain in **INSERT** mode until you press the **Escape** key.

Press **ALT-A** to quit the rung without saving any changes in the program.

### Entering an Instruction Length

To change the length of the instruction (in this example to 8), do the following:

- Position the cursor on the function.
- Type 8 (i.e., the length appropriate for your situation—8 in this example) on the command line.
- Press the **Enter** key.



## Viewing Variable Declarations

To view the current set of variable declarations from anywhere in the Program Editor, press **ALT-V** to invoke the variable declaration window. The window allows cursor and page key scrolling. Pressing **ALT-V** again will show the next table if it exists (local or global).

In **RUNG INSERT** or **EDIT** mode, an asterisk (\*) is displayed next to the variable declarations that were created or modified during the current editing session. These new or modified declarations will remain declared if the rung is accepted.

The screenshot shows a terminal-style interface with a menu bar at the top containing options like RELAY, TMRCTR, MATH, RELATN, BITOP, DATAMV, TABLES, CONVRT, CONTRL, and OPN SP. A central window titled "Variable Declarations for Folder 'LESSON'" displays the following table:

REF	NAME	REF DESCRIPTION
%I0105	I141_07	Intake Valve Control Switch
%R0111	REG_111	Registr Input Data
%R0112	REG_112	Pointer to Read Data
*%R0010	DAYS	day of week
*%R0030	HOLD#	temporary result
*%R0020	EXTRA	overtime

Below the table, there is a note: "(Use cursor keys, or ESC)". The status bar at the bottom of the window displays: "OFFLINE", "C:\LM90\LESSON", "PRG: LESSON", "BLK: MAIN", "SIZE: 149", "RUNG 0005", and "REPLACE %R0020 : EXTRA :: overtime".

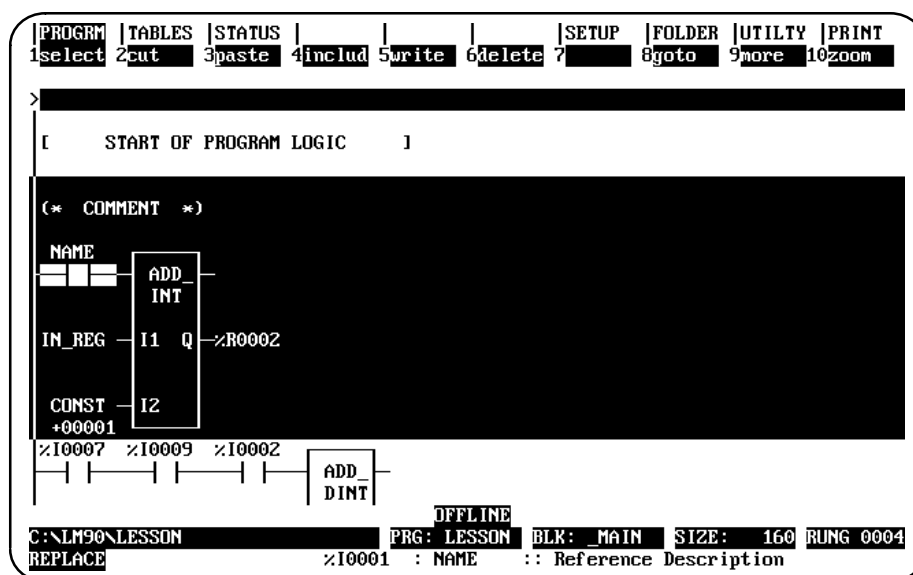
## Deleting a Rung

To remove a single rung of logic, place the cursor at the rung, press **More (F9)**, and then press **Delete (F6)**; or simply press **ALT-D** (while not in **INSERT** or **EDIT** mode).

## Selecting Rungs

The cutting and file writing of rungs requires that a section of rungs first be selected. To select the rungs:

1. Press **More (F9)** to display the cut/paste function keys.
2. Move the cursor to the rung at the beginning or end of the section to be selected. Then, press **select (F1)**. The message "Select mode initiated ..." displayed on the message line indicates that the select function has been activated. Once this function is active, the only functions available are cursor, page, cut, write, help, and goto.
3. The current rung is automatically selected. By pressing the Up/Down cursor keys, Prev, Next, Page Up/Down, Home, or End keys, you can add subsequent or previous rungs into the selected region. Selected rungs are shown in reverse video on the display.



4. The Select (F1) softkey acts as a toggle between the select and deselect functions. To deselect the selected rungs, press the **select (F1)** key or **ALT-A**.
  - A. Press **Cut (F2)** to cut the selected rungs.
  - B. Press **Write (F5)** to write the selected rungs to a file.
  - C. Press **Delete (F6)** or **ALT-D** to delete the selected rungs.
5. To deselect the selected rungs, press **select (F1)**. The **F1** key acts as a toggle between the select and deselect functions.

### Note

The select function can also be used to select variable declarations and interrupt declarations.

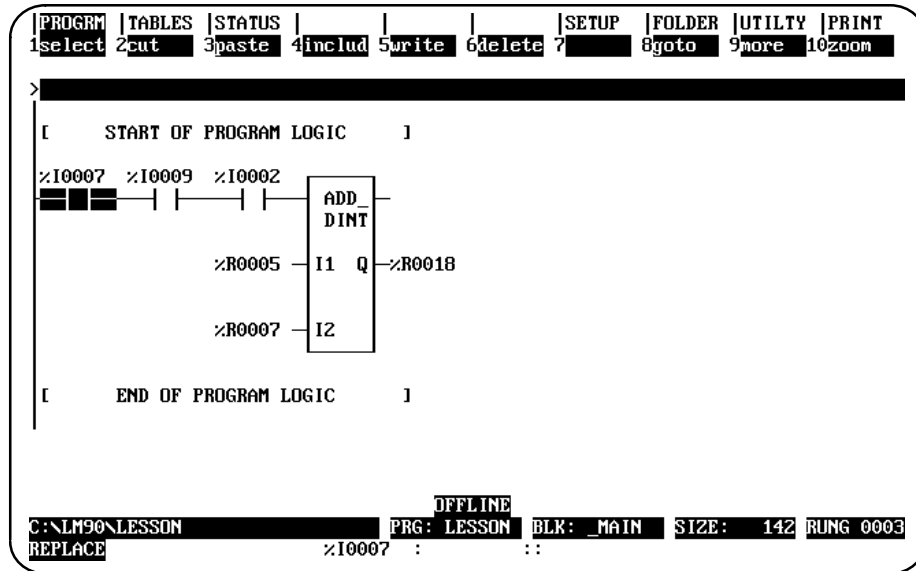
## Cutting Selected Rungs

The cut function enables you to remove a section of previously selected rungs from the current program. Any nicknames, reference descriptions, or comment annotation used within the selected rungs are also copied with the cut rungs for later use in a paste operation. However, nicknames and reference descriptions remain in the variable declaration table.

This function can be used to reorder logic within the program; it is also useful for deleting a section of rungs.

To cut a section of rungs from the current program's logic:

1. Select one or more rungs, as previously described.
2. Press **Cut (F2)** to cut the selected rungs from the program. The select function is automatically exited as part of the cut operation. The cursor will be on the rung after the cut section of rungs.



If the cut buffer becomes full, repeat the operation, selecting a smaller amount of logic.

3. All rungs below the selected section will scroll up to fill the gap of the just-cut section.

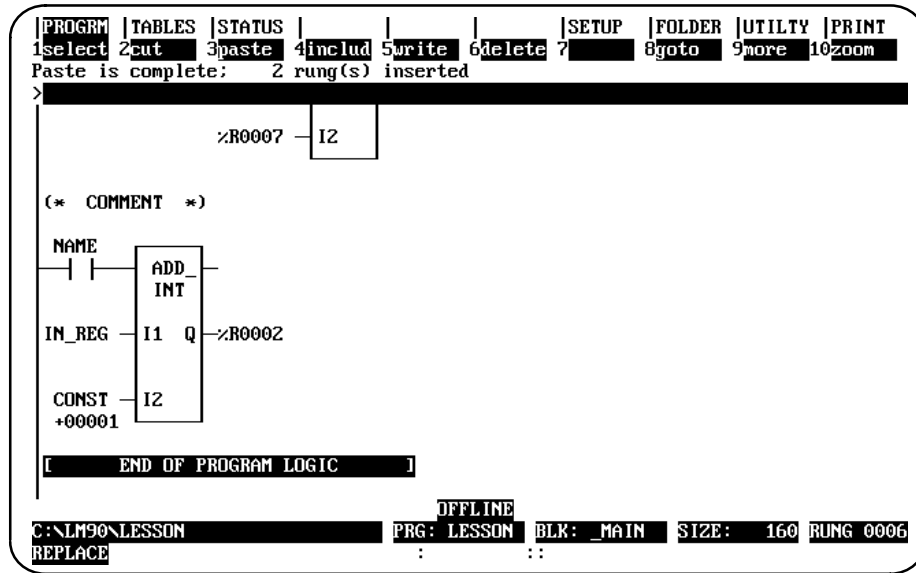
### Note

Cut rungs are saved only as long as you remain in the program editor.

## Pasting Previously Cut Rungs

The paste function enables you to insert a copy of a section of previously cut rungs before the current rung. Any unique reference address nickname assignments and any identifier name will be added to the current program's variable declaration table.

Once the cut operation is complete and the logic has been placed in the buffer, press **Paste (F3)** to initiate the paste function. The position of the cursor remains the same. All entries below the inserted section will scroll down to make room for the just-pasted rungs.



Conflicts may occur between declarations in the section of rungs to be pasted (buffer) and declarations in the existing (target) program. A nickname in the buffer that is identical to one in the target program is simply passed over and ignored during the paste operation.

When a declaration in the buffer has the same reference address but a different nickname than a declaration in the target program, the declaration from the buffer is not added to the existing program's declaration table.

A name conflict occurs when a declaration in the buffer has the same name as a declaration in the target program, but a different meaning. For a nickname, the different meaning would be a different reference address.

Name conflicts are resolved by automatically generating a unique system name for the conflicting declaration in the buffer, and then adding it to the target program. The presence of system names (any name beginning with the \$ character, e.g., \$LA00001) in the pasted logic indicates that name collisions have occurred during the paste operation.

Implicit declarations, such as JUMPs, LABELs, and MCR names, are handled in the same way.

### Note

If coil checking is set to SINGLE, rung(s) may not be pasted should a coil-use conflict occur. %T references should be used on coils to allow rungs to be pasted, or use the WARN MULTIPLE coil check option. (Refer to the information on coil checking later in this section.) %T references may later be replaced with %Q or %M references.

## Writing Selected Rungs to a File

The file write function enables you to store a copy of a section of previously selected rungs from the program logic to a disk file called a side file or program segment. This function is useful for creating a file of commonly used rungs which can be used in different programs.

Only variable declarations used in the rungs that are selected are written to the file. To write the entire variable declaration table to a file, you should select the entire table from the variable declaration section of the program.

### Note

The file write function differs from the cut function in that selected rungs are not removed from the program.

1. To use the file write function, you must first select the rungs to be written to the side file. Press **F1** to select the current rung; then, select additional rungs required.
2. Enter the name of the side file on the command line, and press **Write (F5)**. The selected section of rungs is written out to the specified file.

Any valid file name, minus an extension, can be used for the side file. If the file specification includes a path, the specified directory must already exist. It is not created as part of a file write operation. If no path is specified, the current folder is assumed.

## Including Rungs from a File

The file include function enables you to insert a copy of a file of previously written rungs before the current rung in the program logic. All rungs below the inserted section will be scrolled down to make room for the just-included rungs.

### Note

Side files which contain subroutines cannot be inserted into folders using Logicmaster 90-30/20/Micro software releases prior to Release 3.

Any unique reference address nickname assignments and any identifier names are added to the current program's variable declaration table. If the program name in the include file declaration matches a program name in the target folder, the duplicate entry is not added. Conflicts are handled as previously described under "Pasting Previously Cut Rungs."

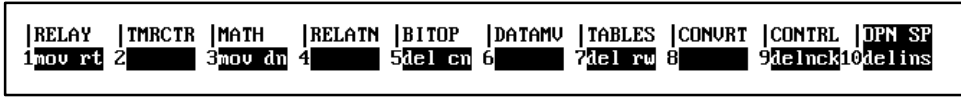
If the program name from the include file conflicts with a non-program name in the target folder, a unique system name is automatically generated for the program declaration before it can be added to the target folder. For example, an MCR has the name END\_OP in the include file, but END\_OP is also the name of the folder (or a nickname). A new name is generated for END-OP (\$MCR001). A corresponding initial logic block and data block are also created.

1. To use the file include function, the program logic to be included must have been previously stored to a particular disk file.
2. Press **Include (F4)** and enter the name of the side file on the command line. Then, press the **Enter** key. The contents of the specified disk file are included before the current rung in the program.

Any valid file name, minus an extension, can be used for the side file. The file specification may include a path; however, if no path is specified, the current folder is assumed.

## Open Space Functions

Open space functions are used, while editing a rung, to open element spaces in the rung. To display the open space function keys shown below, press **F10**.



Function Key	Function	Description	Page
F1	Move Right	Move the element at the cursor position, and all the elements to the right of the cursor position, one position to the right in all lines. Shunts are automatically inserted into the new column in the rung.	3-83
F3	Move Down	Create room for an additional line of logic above the line the cursor is on.	3-84
F5	Delete Column	Delete an entire column of instructions(includingverticals) and operands within a rung. All elements to the right of the deleted column will automatically move left.	3-85
F7	Delete Row	Delete an entire row of instructions(includingverticals) and operands within a rung.	3-86
F9	Delete Nickname	Remove nicknames from the variable declarations table. Enter the nickname to be deleted on the command line before pressing <b>F9</b> .	N/A
F10	Delete Instruction	Delete an operand or instruction from a rung. (You may also press <b>ALT-D</b> to delete an operand or instruction.)	3-87

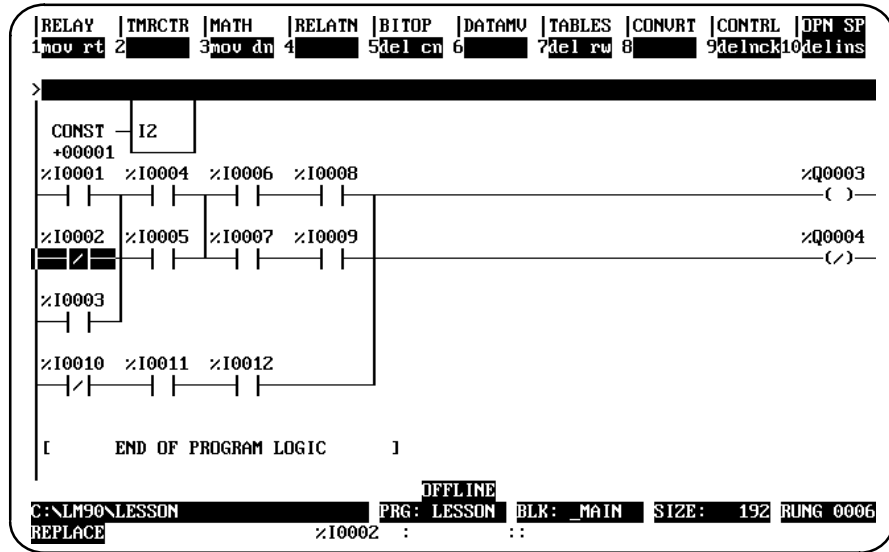


## Move Logic Right

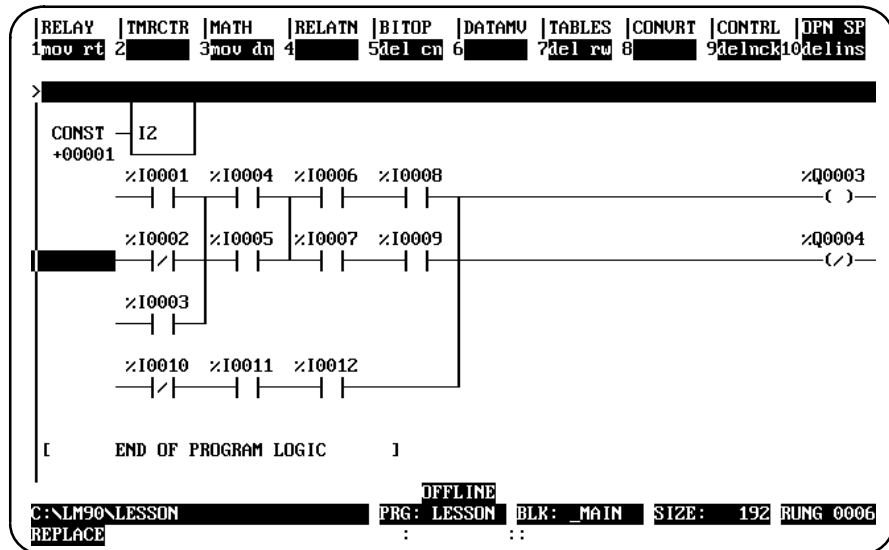
The Move Right (MOV RT) function moves the element at the cursor position, and all the elements to the right of the cursor position, one position to the right in all lines. Shunts are automatically inserted into the new column in the new rung.

In the following example, the column next to the power rail is moved right one column by positioning the cursor in column 1 and pressing **Move Right (F1)**.

Before:



After:

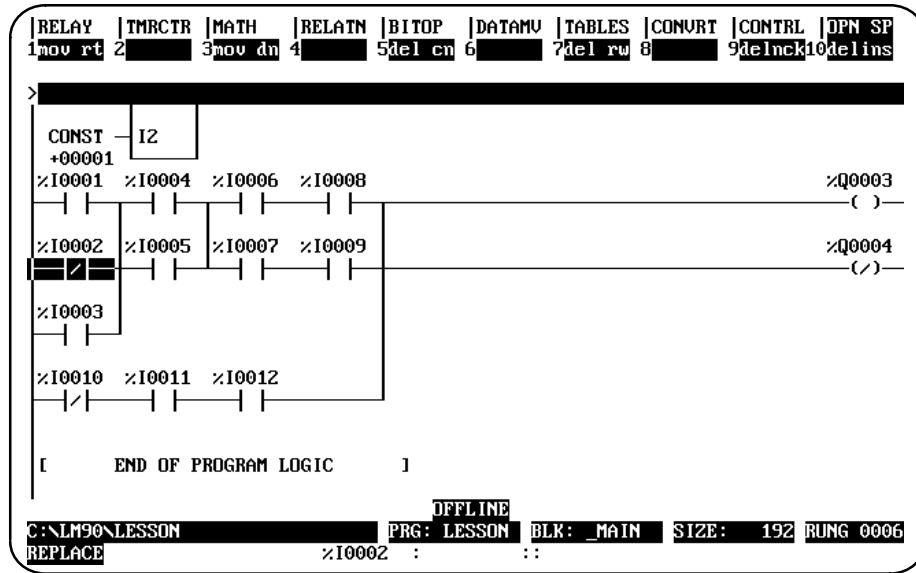


### Move Logic Down

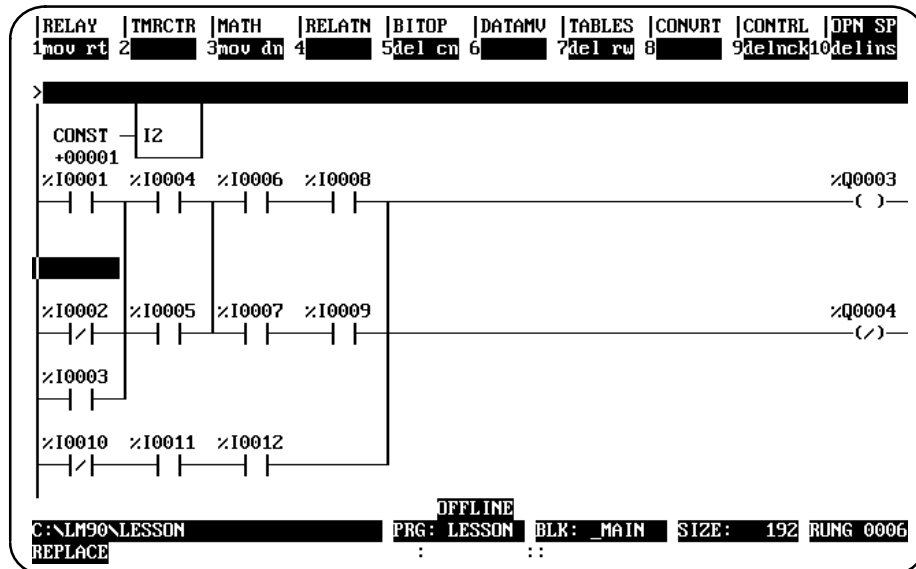
The Move Down (MOV DN) function is used to create room for an additional line of logic above the line the cursor is on. All elements in the region to be moved must be located entirely in the same row as the cursor, or in the rows below.

In the following example, the rung of logic beginning with %I0002 is moved down one row by positioning the cursor in row 2 and pressing **Move Down (F3)**.

Before:



After:

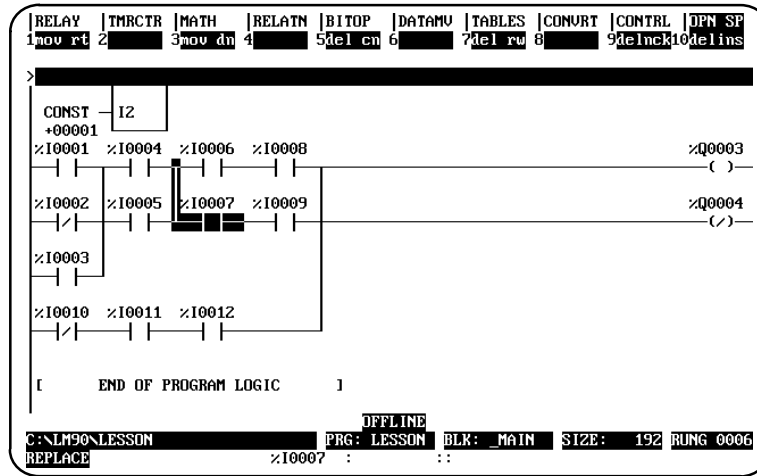


### Delete Column

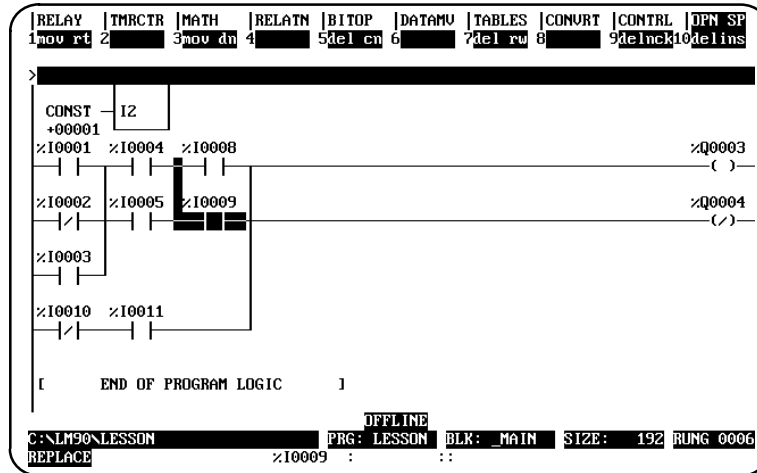
The Delete Column (DEL CN) function is used to delete an entire column of instructions (including verticals) and operands within a rung. It can be used to delete a column that contains functions as well as contacts or coils, as long as the function template itself is totally contained within the column.

When the column to be deleted is within a rung that contains no coil or jump instructions, all columns to the right of the deleted column are moved left one column. In the following example, the column which contains %I0007 is deleted by positioning the cursor in column 3 and pressing Delete Column (F5).

Before:



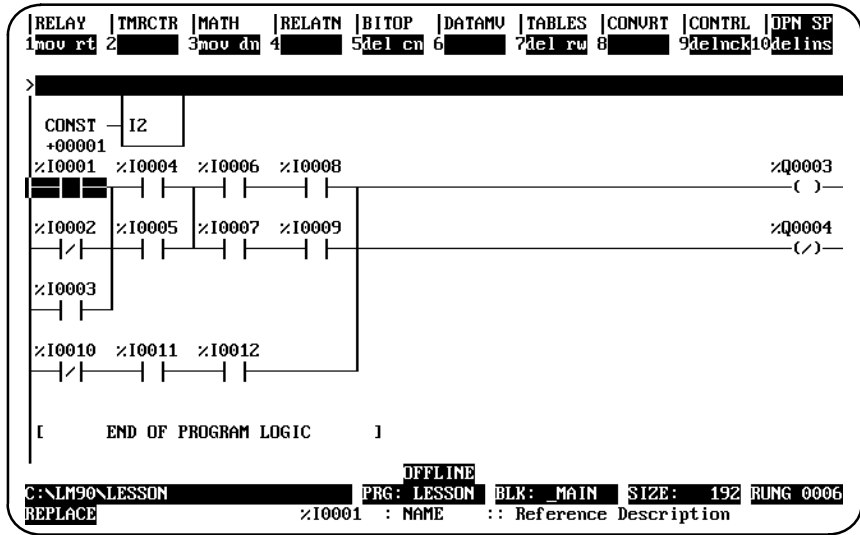
After:



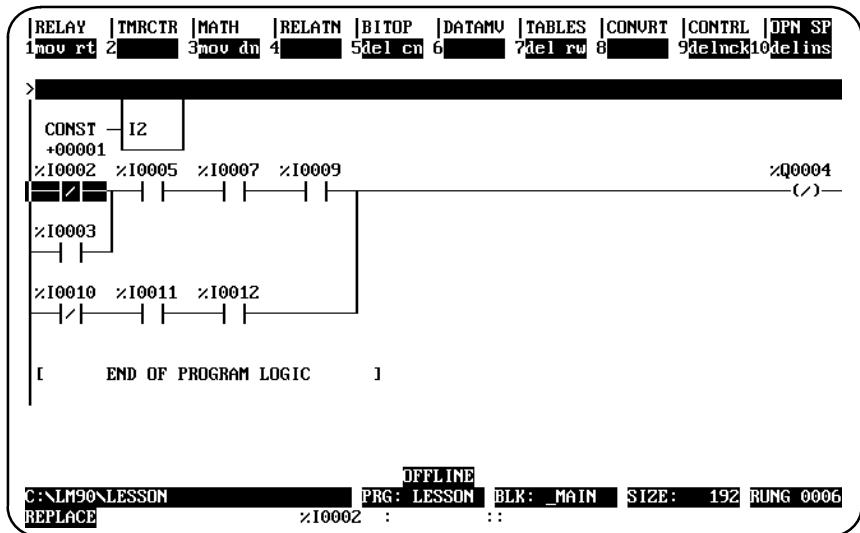
### Delete Row

The Delete Row (DEL RW) function is used to delete an entire row of instructions (including verticals) and operands within a rung. It can only be used to delete a row whose instructions, together with their operands, are totally contained within the row. For example, the DEL RW function can be used to delete a row with a function such as MCR, but it cannot delete a row containing an ADD\_INT function because the ADD\_INT function spans several rows. After the deletion, all rows beneath the deleted row move up one row. When a row is moved up to the first row position, its verticals are automatically deleted. In the following example, the row which contains %I0001 is deleted by positioning the cursor on that row and pressing Delete Row (F7).

Before:



After:

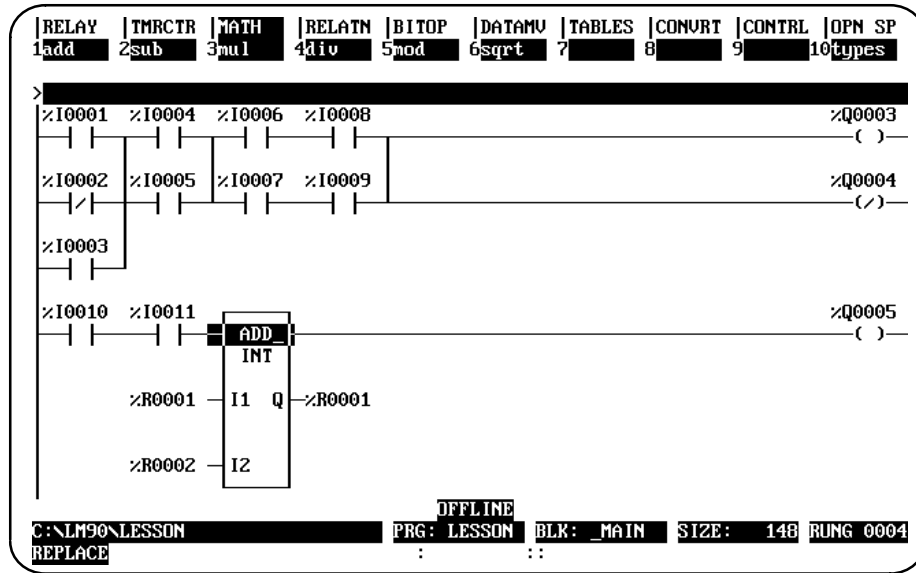


### Delete Instruction

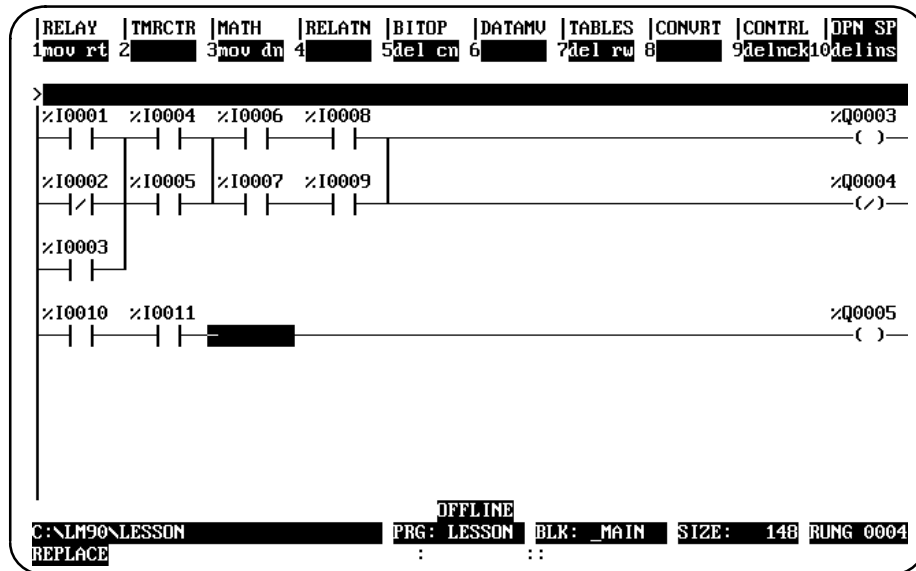
The Delete Instruction (DELINS) function is used to delete an operand or instruction from a rung. Another instruction or a horizontal link may then be entered in the space left by the deleted instruction. The ALT-D key sequence may also be used to delete instructions; however, the cursor will automatically move to the right after the deletion.

In the following example, the ADD instruction is deleted by positioning the cursor on the ADD function block and pressing **Delete Instruction (F10)**.

Before:



After:



## Increment/Decrement Reference Address

The increment/decrement reference address feature is available in either **INSERT** or **EDIT** mode in the program editor and from the variable declaration editor. It allows you to increment or decrement the reference address where the cursor is located by either one or the amount specified on the command line.

Key	Description
CTRL-U	Increment key sequence.
CTRL-D	Decrement key sequence.

### Note

For reference operands which must be byte-aligned, the reference address is incremented/decremented by one byte.

For example, a discrete input reference address of a MOVE\_INT function block will increment or decrement by one byte (e.g., %I0001 to %I0009). The same reference address on a normally open contact will increment or decrement by one (e.g., %I0001 to %I0002). A register reference address on the input of an ADD function block will also increment or decrement by one (e.g., %R0005 to %R0006).

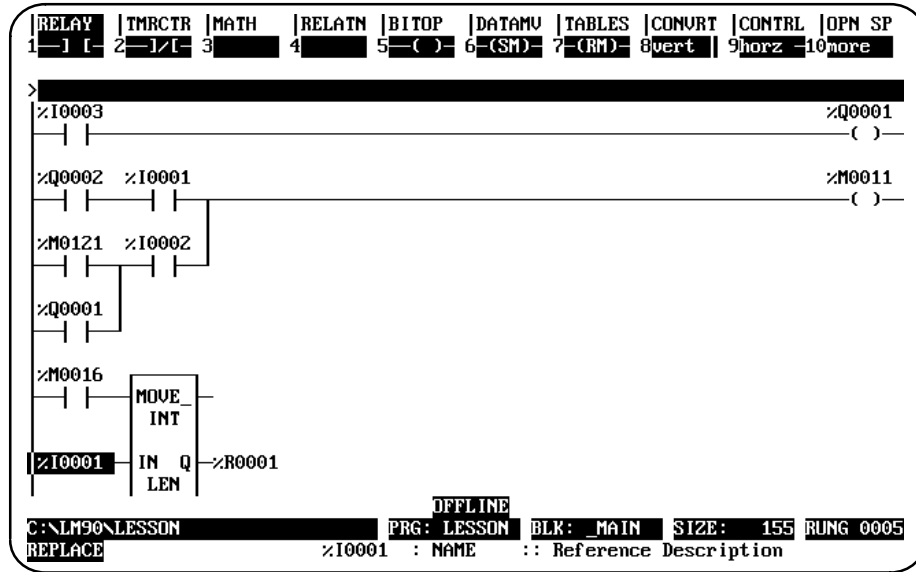
If there is a numeric value on the command line, it is used as the number of bits to add to or subtract from the reference address. For parameters that are byte-aligned, the increment/decrement value is one byte when the value on the command line is between 0 and 8, two bytes when the value is between 9 and 16, etc. The value will remain displayed on the command line after the function is completed so that the reference address can be incremented/decremented by the same value again.

If the reference address reaches the current configuration limit, an error message is displayed and the function is not performed. If the cursor is not on a reference address and you attempt to increment or decrement the address, an error message is also displayed.

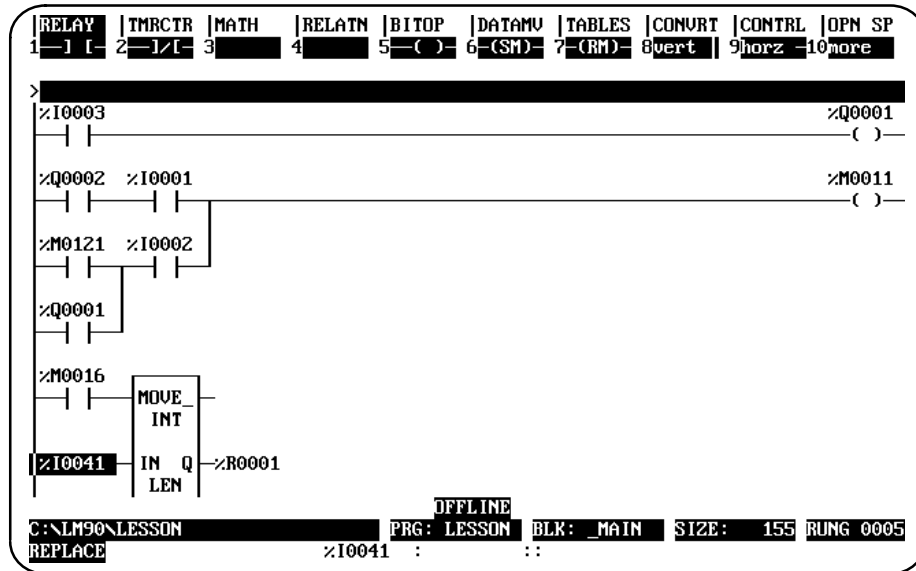
## Incrementing/Decrementing within a Rung

This example illustrates how to increment reference address %I0001 in a ladder diagram rung.

1. First, position the cursor on the reference address to be changed. In this example, that would be %I0001.



2. Press CTRL-U five times to display the reference address %I0041, as shown in this screen.



3. To decrement the reference address to %I0025, press **CTRL-D** twice. Or, you could enter the decrement amount 16 (2 bytes) on the command line and press **CTRL-D**.

The value 16 displayed on the command line does not disappear when the decrement is completed. It continues to be displayed on the command line so that you may continue decrementing by that same amount.

## Auto-Next Highest Reference

The auto-next highest reference address function automatically uses the next reference offset of the specified type, after the highest currently used in the folder during an editing session. If the parameter requires a byte-aligned reference offset, the next available aligned offset is automatically provided.

To use this function, enter the % character and the user reference (e.g., %I, %Q, %R, etc.) on the command line. The % character is used to distinguish a reference type from a nickname character. Then, press the **Enter** key.

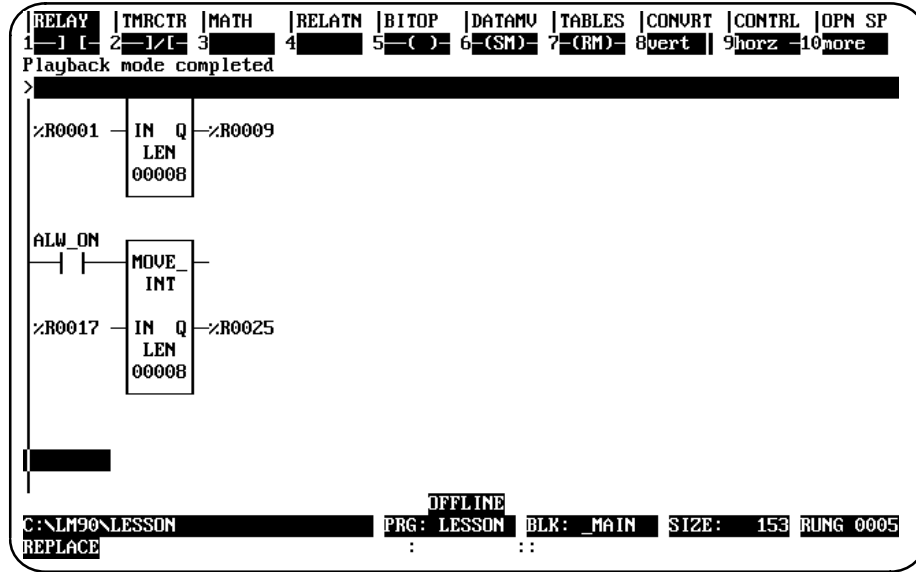
For example, if the highest %I reference already used during an editing session is %I0019 and %I is specified as the address for an input operand to an **ADD\_INT** function block, %I0024 (the next available aligned reference) is automatically used.

The following example illustrates how useful the auto-next highest reference address function can be, when used in conjunction with **TEACH** mode. It allows rungs to be entered and reference addresses automatically assigned to the next available reference address. (For more information on **TEACH** mode, refer to chapter 2, section 5, "Keyboard Functions").

1. Enter **INSERT** mode by pressing **Insert (F1)** from the program editor.
2. Press **ALT-T** to enter **TEACH** mode, and then press **ALT-N**, where **n** is the number to be used for the teach file. For this example, use **ALT-0**.
3. On the command line, type **ALW\_ON &NOCON** and press the **Enter** key.
4. Then, type **&MOV** on the command line and press the **Enter** key.
5. Use the **Tab** key to position the cursor on parameter **IN** of the **MOVE** function block.
6. Type **%R** on the command line, and press the **Tab** key.
7. Then, enter **%R** on the command line again (or press **CTRL-Home** to display the previous command line contents), and press the **Enter** key twice.
8. Press **ALT-Q** to exit **TEACH** mode.



- 9. Press **ALT-0** to play back the teach file. The following screen is displayed. Note that the reference address has automatically been changed to the next available address.



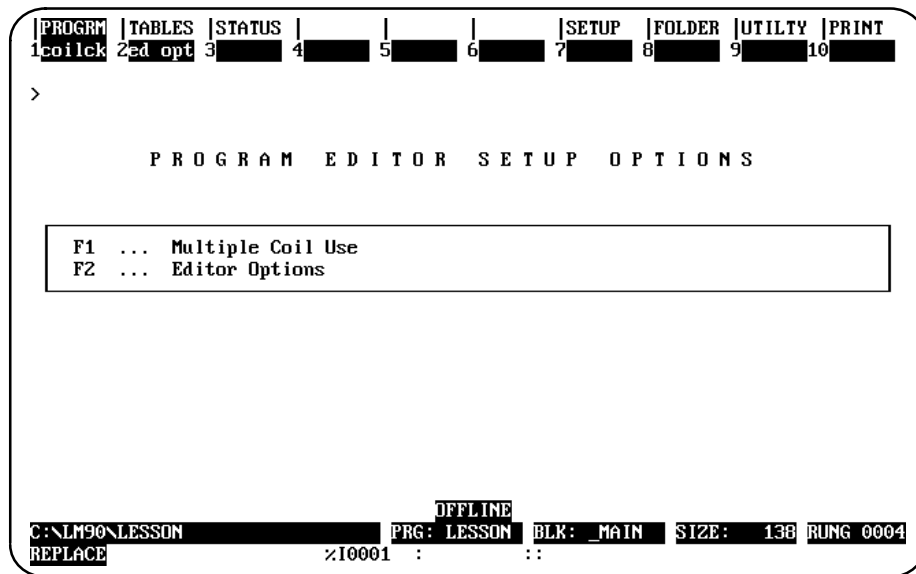
## Section 10: Editor Options

The Program Editor Options menu provides access to options in the program editor. These options include multiple coil use and automatically inserting references.

### Coil Checking

The coil check function of the programming software checks for multiple uses of %M or %Q references with relay coils or outputs on functions. Beginning with Release 3 of Logicmaster 90-30/20/Micro software, you can select the level of coil checking desired from a screen similar to the one shown below.

To access this screen, press **Program (F1)** from the Programming Software main menu. Then, press **Options (F7)** from the program editor. Instructions for selecting the desired level of coil checking begin on page 3-93.



Three levels of coil checking are available:

Level	Name	Description
SINGLE	Coil checking enabled	Single coil use only. When coil use is set to <b>SINGLE</b> , Logicmaster 90-30/20/Microsoft software does not allow multiple coil use. When the rung is accepted and multiple use is detected, an error message is displayed and the cursor is placed on the first conflict found. You cannot exit the rung edit until the conflicts are resolved. When this setting is accepted, the coil use map is rebuilt. A “checking coil use” message is displayed while the map is rebuilt. If conflicts are found, only the reference address of the conflict is listed. Only one screen full of conflicts is displayed. If conflicts are found, the <b>Current Coil Use</b> field is not updated; it is only updated when no conflicts are found.
WARN MULTIPLE*	Coil checking disabled with warning	When coil use is set to <b>WARN MULTIPLE</b> , Logicmaster 90-30/20/Microsoft software allows multiple coil use with warning messages. The <b>Current Coil Use</b> field is updated, and the coil use map is rebuilt. A message is displayed while the map is rebuilt and the conflicts are listed. Only one screen of conflicts is displayed.
MULTIPLE	Coil checking disabled	When coil use is set to <b>MULTIPLE</b> , Logicmaster 90-30/20/Microsoft software allows multiple coil use without any restrictions or messages. The <b>Current Coil Use</b> field is updated, and the coil use map is not rebuilt. No conflicts are displayed.

\* Default selection.

### Note

If the program folder is locked, the level of coil checking cannot be changed and the coil map cannot be rebuilt. Refer to the information on unlocking program folders described in chapter 7, “Program Folders.”

The Coil Check screen shows the current level of coil checking, allows you to change the level, and checks for and displays coil use conflicts on demand. You can change the level selected, regardless of which level was previously selected.

Coils can function as SET Coils or as RESET Coils with **MULTIPLE** or **WARN MULTIPLE** coil checking enabled. For details on the possible effects of **MULTIPLE** and **WARN MULTIPLE** checking on Coils, refer to the information on “SET Coils” and “RESET Coils” in Section 1, “Relay Functions,” of Chapter 3 of the *Series 90™ -30/20/Micro Reference Manual* (GFK-0467).

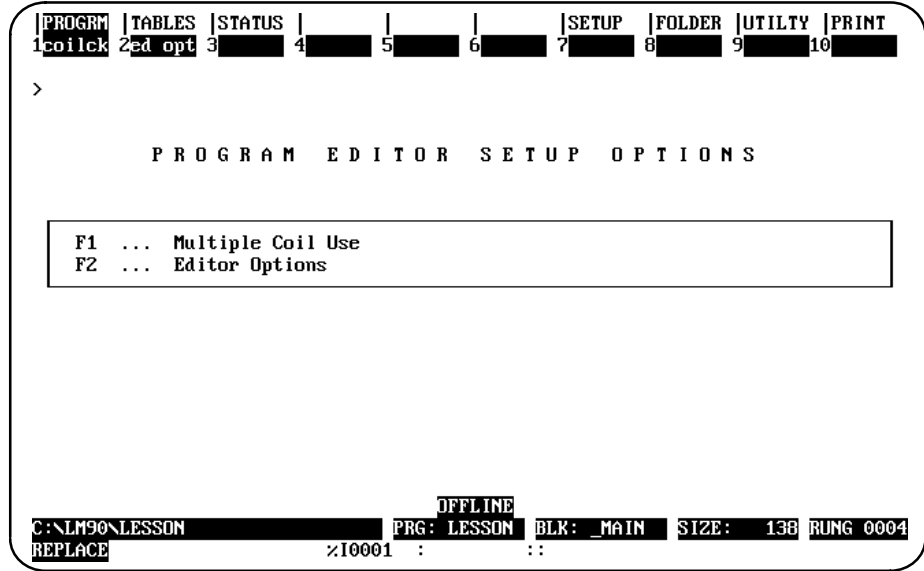
### Note

When the program mode is **MONITOR** or **ONLINE** and the program in your computer is identical to the program in the PLC (i.e., **LOGIC EQUAL** status is displayed), only the current level of coil checking is displayed. An error message is displayed if you try to change the coil check level.

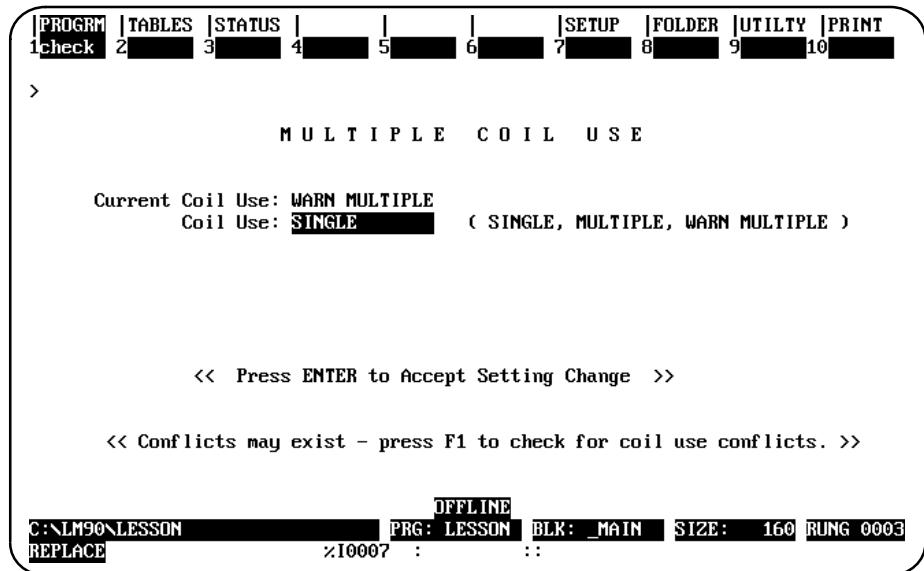
**Warn Multiple** can give misleading feedback for **Micro PLCs**: if coil references are added or deleted in the RLD program, the Logic-Not-Equal warning will be displayed. You can correct the inaccurate message by following the steps on the next page or you can avoid using the Warn Multiple setting with Micro PLCs.

To change the level and check for conflicts:

1. Press **Program (F1)** from the Programming Software main menu. Then, press **Options (F7)** from the program editor to display the Program Editor Options menu.



2. Press **Coil Check (F1)** to display the Multiple Coil Use screen.



- The screen shows the current level of coil checking. To select another level, repeatedly press the **Tab** key until the desired setting is displayed. Then, press the Enter key to accept the change. Or, you can type the desired setting into the **Coil Use** field, and then press the **Enter** key.
- The coil use map is automatically rebuilt when the **SINGLE** or **WARN MULTIPLE** level is accepted. To rebuild the coil map without changing the current setting, press **Check (F1)**. **F1** may be used regardless of the current coil use setting. The message "Checking Block . . ." is displayed as the map is built, along with the name of the block currently being checked.

The rebuilding of the map takes approximately 15 seconds per block. For example, a program consisting of 10 to 15 blocks may take up to two minutes to rebuild the map.

- Press **ALT-A** to abort the rebuild and restore the old coil use map. A message indicating that an abort has occurred is displayed.
- When the rebuild is complete, any conflicts found are listed. Both explicit and implicit use conflicts are displayed. Only one screen full of conflicts is displayed.

```

|PROGRAM|TABLES|STATUS| | | |SETUP|FOLDER|UTILITY|PRINT
1check 2 3 4 5 6 7 8 9 10
Check For Coil Use Conflicts Complete
>

          M U L T I P L E   C O I L   U S E

Current Coil Use: WARN MULTIPLE
Coil Use: WARN MULTIPLE ( SINGLE, MULTIPLE, WARN MULTIPLE )

Coil conflicts found :
%M0585      %M0586      %M0587      %M0588      %M0589
%M0590      %M0591      %M0592      %Q0001      %Q0002
%Q0003      %Q0004      %M0053      %M0057      %M0049
%M0055      %Q0121      %Q0122      %Q0123      %Q0124
%Q0125      %Q0126      %Q0127      %Q0128

OFFLINE
C:\LM90\LESSON PRG: LESSON SUB: COIL SIZE: 215 RUNG 0013
REPLACE %I0001 : DWELL_T :: weld dwell timer

```

Conflicts are not saved in memory. When you leave this screen, they will not be redisplayed if you re-enter this display screen. However, the **Check (F1)** softkey may be pressed to check for coil conflicts.

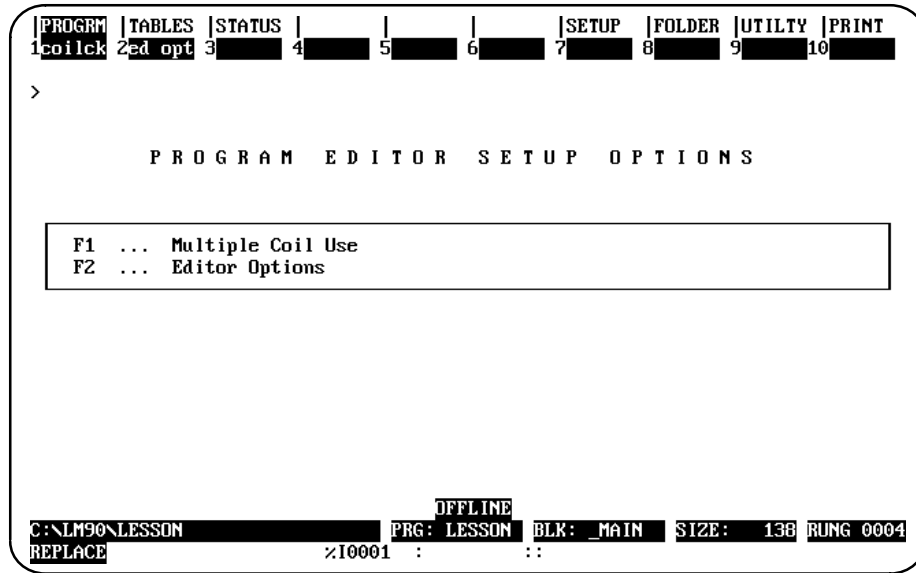
**ALT-P** may be used to print a screen of conflicts. For information on printing a screen display, refer to chapter 9, section 2, "Selecting a Screen Print Device."

## Automatically Inserting References

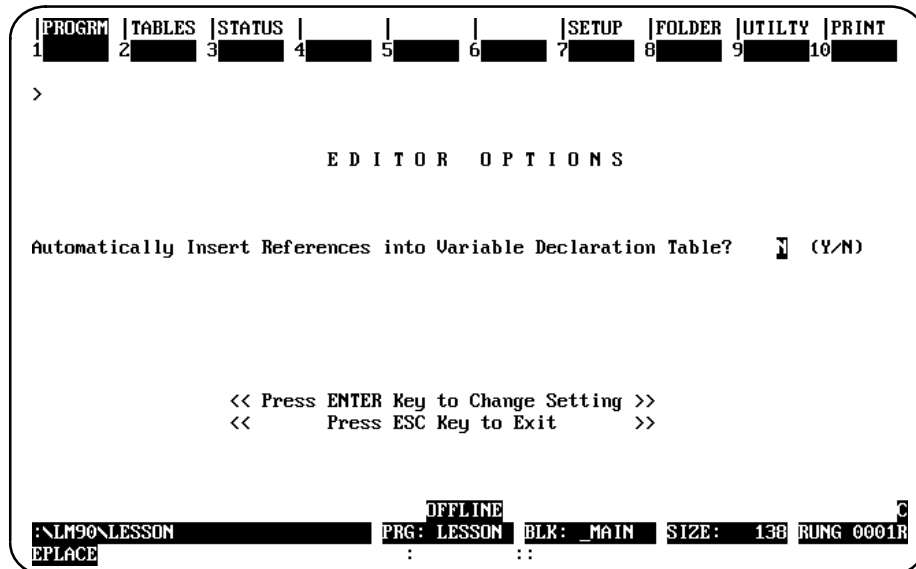
References which do not have nicknames or reference descriptions can be automatically inserted into the variable declaration table as the program is being developed. Then, at a later time, you can go to the variable declaration editor and enter just the annotation.

To enable the automatic insertion of references:

1. From the Programming Software main menu, press **Program (F1)** and then **Options (F7)** to display the Program Editor Options menu.



2. Press **Editor Options (F2)** to display the Editor Options screen.



- 3. Enter **Y** (Yes) and press the **Enter** key to enable automatic insertion. Explicit references entered on the command line during an edit session are automatically inserted in the variable declaration table; implicit references (those references not entered while programming logic) are not automatically inserted in the table.

In the following example, the explicit reference %I0001 is inserted automatically in the variable declaration table. %I0002 through %I0016 are implicit references and are not inserted automatically in the table.

```

%M0001      ADD      %Q0001
-| |----- ( )-
| |
| |      INT
%I0001 - I1  Q -%Q0017
CONST - I2
+0004

```

In addition, %S and %T references are not added to the table.

References associated with rungs that are pasted or included are added to the variable declaration table only if this operation were enabled when the rungs were cut or written. Removing rungs using either the cut/paste function or the file include function does not affect the table. Any reference to be removed from the variable declaration table must be deleted using the variable declaration table editor. References entered during a word-for-word operation are not automatically inserted into the variable declaration table.

- 4. If the variable declaration table becomes full, a message is continuously displayed for each reference entered until some of the references are deleted. Use the variable declaration table editor to delete some references in order to make room for more insertions.
- 5. To disable the automatic insertion feature, enter **N** (No). When this feature is disabled, references entered on the command line are not placed into the variable declaration table, unless they have a nickname or reference description associated with them.

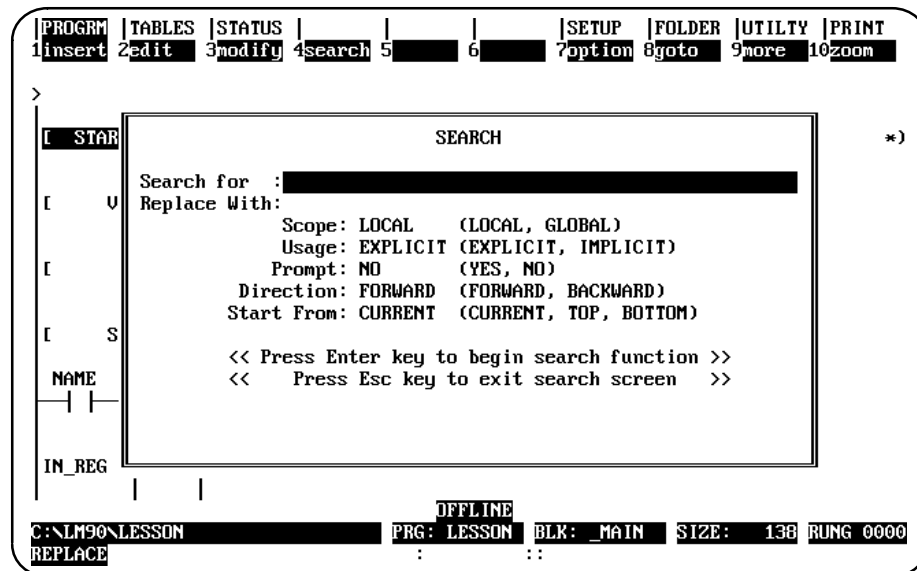
**Note**

The default selection for automatically inserting references is **N** (No). Every new folder defaults to **NO**.

## Section 11: Search Function

The search function enables you to locate an identifier name, reference address, nickname, instruction, instruction plus reference address, instruction plus nickname, instruction plus identifier name, and reference description anywhere in the program.

- To begin the search function, select **Search (F4)** from the program edit functions. When **F4** is pressed, a search function window is displayed on the screen.



- To continue searching for the same target, press **ALT-F4**.  
Pressing only the **Search** key accesses the search function again. Since all fields are already set as desired, simply press the **Enter** key to initiate the search.
- If the target is not found, a message indicating this is displayed. The search function will remain active.
- If the target is found in a block that is locked for view, the following message is displayed:  

```
Found in locked block <block_name>. (Continue/Quit)
```

If you continue, all remaining blocks are searched. If you decide to quit, the search is aborted. For more information on viewing locked blocks, refer to chapter 3, section 8, "Subroutine Blocks."
- The search can be aborted at any time by pressing **ALT-A** (abort). A message acknowledging the abort is displayed, and the search function will remain active.



Refer to the definitions in the following table when making entries in the search function window. To change a selection in one of the fields, other than the **Search for** and **Replacewith** fields, use the Tab key to toggle through the available choices.

Field	Description
Search for	<p>The target to be searched for. It may be an identifier name, a reference address (e.g., %I0012) or nickname (e.g., WIDGET), an instruction (e.g., &amp;COIL), an instruction plus reference address (e.g., &amp; COIL %Q0001), nickname (e.g., &amp;COIL WIDGET), or identifier name (e.g., &amp;JUMP BLK1), or a reference description entered in quotes. Enter the desired target into this field.</p> <p>Search allows you to locate all uses of a nickname. First, the software searches the local table if you are in a subroutine block. Next, it searches the main variable declarations table and finally the reserved table. The search operation stops when the first occurrence of the search target is found.</p>
Replace with	What will replace the target being searched for.
Scope	Modify the search by specifying whether the target should be searched for in the current subroutine block only or across all subroutine blocks in the program. Choices for this field include <b>LOCAL</b> for the current block only or <b>GLOBAL</b> for all blocks. The search order for all blocks is the subroutine block declaration order, beginning with the current block.
Usage	Modify the search by specifying whether only explicit usage of the reference is checked for ( <b>EXPLICIT</b> ) or both explicit and implicit usage ( <b>IMPLICIT</b> ). More information about implicit search can be found at the end of the search function.
Prompt	<p>Action to be taken if the search target is found. For the search function, selecting <b>Y</b> (Yes) indicates that the system will prompt you for confirmation before searching for the next target. When <b>Y</b> (Yes) is selected and the target is found, you may display the search target, disregard this instance of the target and continue searching, or terminate the actual search but remain in the search function.</p> <p>For the search and replace function, selecting <b>Y</b> (Yes) indicates that the system will prompt you for confirmation before the found target is replaced. There are four choices available:</p> <ol style="list-style-type: none"> <li>1. Replace the current found target and continue.</li> <li>2. Disregard the found target and continue.</li> <li>3. Starting from this found target, replace the rest of the found targets without prompting (i.e., change the prompt value to <b>N</b> (No)).</li> <li>4. Terminate the replace but remain in the function.</li> </ol>
Direction	The direction of the search, either <b>FORWARD</b> or <b>BACKWARD</b> .
Start From	The starting point of the search, either from the current position ( <b>CURRENT</b> ), from the top of the block ( <b>TOP</b> ), or from the bottom of the block ( <b>BOTTOM</b> ).

### Note

You should change the default Scope to Global and change Usage to Implicit when searching a program for address conflicts. Global will search all blocks including **\_MAIN**. Implicit will check for memory usage by functions with memory lengths greater than 1 as well as memory addresses explicitly visible in the logic.

The following table shows the legitimate replacement items for the different types of search items.

<b>Search Item</b>	<b>Replacement Item</b>
Reference address or nickname	Reference address or reference nickname.
Identifier	Identifier of the same type.
Instruction	Word-for-word equivalent instruction. Refer to chapter 3, section 12, "OnlineEditing/Monitoring," for more information on making word-for-word changes.
Instruction and reference	Word-for-word instruction or reference or nickname. Refer to chapter 3, section 12, "OnlineEditing/Monitoring," for more information on making word-for-word changes.
Instruction and identifier	Identifier of the same type.
Description	No replacement allowed.

### Note

If a program contains references that have the same reference description text, searching for the reference description always finds the first occurrence of the reference description.

## Using Search and Replace

The search and replace function is used to search and replace the target within the same variable declaration table. It cannot be used to search for the target in one variable table and replace it in another variable table.

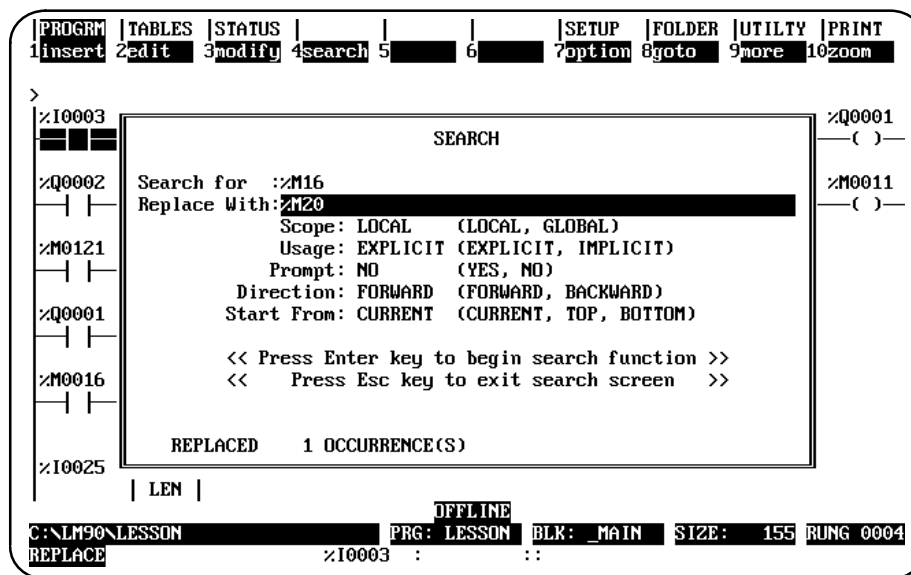
To use the search and replace function:

1. Enter the target to be searched for in the **Search for** field and the replacement item in the **Replace with** field. Other fields can be changed to further modify the search.
2. Press the **Enter** key to begin the search function, or press the **Escape** key to exit this screen. If the **Prompt** field is set to **No**, a “busy” prompt is displayed on the search screen with the current replacement count, indicating that the search and replace function is in progress.
3. If the target is found in a block that is locked for view, the following message is displayed:

Cannot write to locked block <block\_name>. (Continue/Quit)

You may continue the search and replace operation at the next block or terminate the search and replace. For more information on viewing locked blocks, refer to chapter 3, section 8, “Subroutine Blocks.”

In the following example, %M0016 will be searched for and replaced with %M0020. Since the **Prompt** field is set to **No**, no confirmation is required. This screen shows the parameters selected for this search.

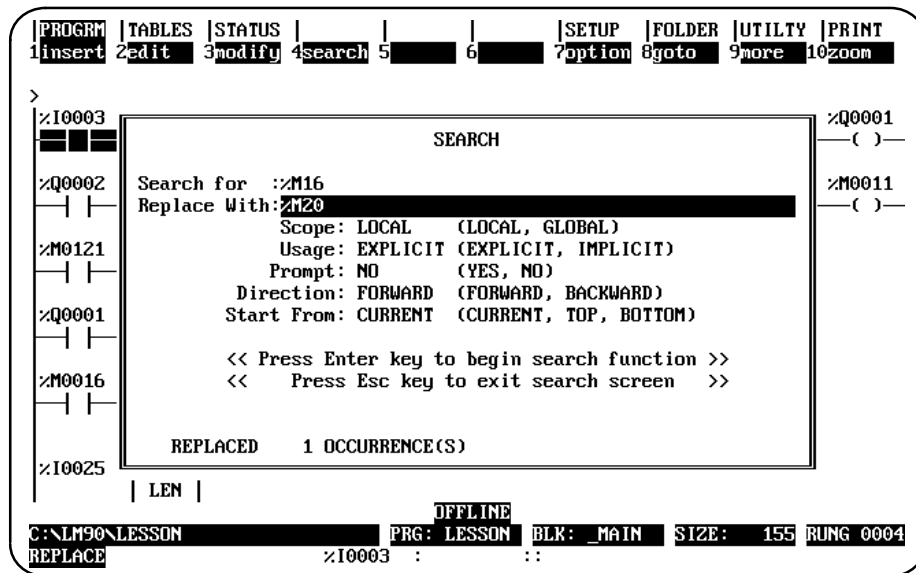


- If the **Prompt** field is set to **Y** (Yes), confirmation is required before each replacement. The following prompt is displayed each time the found target (in this example, %M0016) is found:

Replace %M0016 with %M0020 ? (Yes, No, All, Quit)

Press **Y** (Yes) to replace the current found target with the replacement item and then continue the search. Select **N** (No) to disregard this instance of the target and continue the search. Press **A** (All) to drop the confirmation process, beginning with this found target, and proceed to replace the rest of the found targets with the replacement item. Pressing **Q** (Quit) terminates the current search and replace process, but remains in the function.

- Assuming that the function replaced one occurrence of %M0016 with %M0020 without error, an appropriate message including the replacement count is displayed and the function remains active.



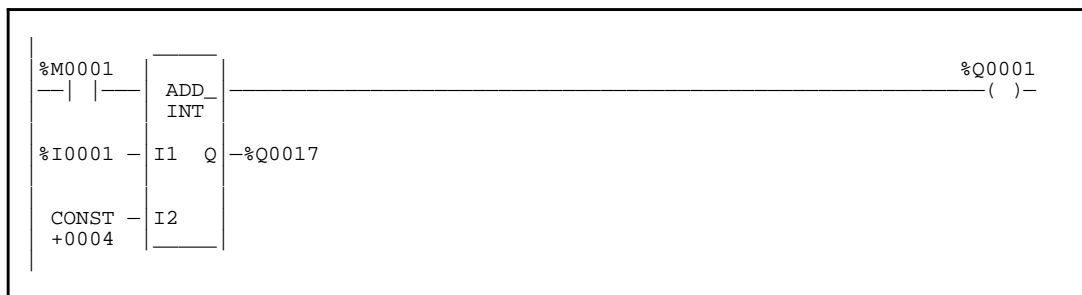
- If an error occurs while replacing one of the reference addresses, the rung number where the error occurred is displayed at the top of the ladder diagram screen with the cursor on the erroneous item. In addition, a prompt displayed on the message line will ask you to either stop the function or skip the erroneous replacement and continue the function. Press **Y** (Yes) to skip the erroneous replacement and continue with the search and replacement function, or press **N** (No) to stop. A search and replace function, however, cannot be aborted.

## Implicit Search

Implicit references are those references which are not directly programmed in the logic. However, due to the length of a function parameter, they are included. An implicit search enables you to locate these implicit references within a program.

Selecting **IMPLICIT** in the *Usage* field of the search function window means that **both** explicit **and** implicit references will be searched for. (Explicit references are those references you entered while programming the logic for your program or subroutine block.)

In the following example, %I0001 is an explicit reference and %I0002 through %I0016 are implicit references.



## Search by Reference Type

The search by reference type function allows you to locate all uses of a specific reference type within the program. To initiate this type of search, enter the % character and the user reference (e.g., %R for register references) in the **Search for** field. Then, press the **Enter** key. The % character is required in order to distinguish a user reference from a name character.

## Quick Search for a Coil

The quick search feature allows you to search for a coil with a particular reference address by entering the reference address on the command line and then pressing the **Search (F4)** softkey. The search will begin at the current location and continue to search for the reference address until a coil with the specified address is found. The reference address continues to be displayed on the command line so that you can find the next occurrence of a coil with the specified reference address by simply pressing **Search (F4)** again.

## Section 12: Online Editing/Monitoring

In addition to offline editing, Logicsmaster 90-30/20/Micro software supports several online functions. To make online changes, the programmer must be in **ONLINE** mode; and logic in the current folder and PLC must be **EQUAL**. Refer to chapter 8, "Program Utilities," for information on verifying the program with the PLC.

The functions available when online and equal include:

- Monitoring logic and registers.
- Forcing and overriding discrete references.
- Substituting instructions, constant values, and reference addresses.
- Online changes for data values.
- Block edit Run Mode store

### Note

Changes made when **ONLINE** and **EQUAL** are restricted to those of equal size. The PLC is updated as each change is completed.

### Inserting or Editing Rungs (Block Edit)

Logic in the program folder and the PLC may be inserted or edited from the program editor while the PLC is in **STOP** mode and the programmer is **ONLINE** and **EQUAL**. After pressing **INSERT (F1)** or **EDIT (F2)** to enter **INSERT** or **EDIT** mode and making a change, the status line will change from **LOGIC EQUAL** to **BLOCK EDIT**. One or more rungs may be inserted or modified. After accepting all the changes, the modified logic may be saved to both the program folder and PLC by pressing **ALT-S**.

**ALT-S** may also be used to save a single subroutine block while the PLC is in **STOP** mode.

### Block Edit ("Bumpless") Run Mode Store Function (6.61 or Later CPUs)

Press **ALT-S** (holding the **ALT** key and then press the **S** key) from the Program Logic Editor while in Block Edit mode to store changes to a program in the PLC. Prior to Release 6.61 of the CPU firmware and Release 6.60 of Logicsmaster 90-30/20/Micro, this operation was not allowed while the PLC is in **RUN** mode. However, starting with Release 6.61, editing (changing, adding to, or deleting from) a program while the PLC is in **RUN** mode is allowed.

Please note the following:

- Only one subroutine may be modified at a time, and there must be enough unused, unfragmented memory to store the subroutine that is being modified and the main program. Otherwise, the PLC will need to be stopped to perform the store.
- Note that stores while in **RUN** mode from the Program Utility Store screen in Logicsmaster will behave as before (i.e., those stores will "pause" the execution of the PLC while the store takes place).
- There must be enough PLC memory to store both the old block and the new block.

---

## Substitutions

If the programmer is in **ONLINE** mode and communicating with an operating PLC and if the program logic in the PLC and in the programmer are **EQUAL**, an instruction, constant, or reference address may be substituted with another. Every substitution updates the logic in the program folder and in PLC memory.

Instruction substitutions are permitted within the groups listed in the following table. Data types cannot be changed within each group; therefore, it is not necessary to include the data type.

**Table 3-12. Substitution Groups**

Function	Description	Type This
<i>Contacts</i>		
-   -	Normally open contact.	&NOCON
- / -	Normally closed contact.	&NCCON
<i>Coils</i>		
-( )-	Normally open coil.	&NOCOI
-(/)-	Negated coil.	&NCCOI
-(S)-	SET coil.	&SL
-(R)-	RESET coil.	&RL
-(M)-	Retentive coil.	&NOMC
-(/M)-	Negated retentive coil.	&NCM
-(SM)-	Retentive SET coil.	&SM
-(RM)-	Retentive RESET coil.	&RM
-(↑)-	Positive transition coil.	&PCOI
-(↓)-	Negative transition coil.	&NCOI
<i>Retentive On-Delay Timer Function</i>		
ONDTR_TENTHS	Tenth of a second time base.	&ON_TE
ONDTR_HUNDTHS	Hundredth of a second time base.	&ON_H
ONDTR_THSDTHS	Thousandth of a second time base.	&ON_TH
<i>On-Delay Timer Function</i>		
TMR_TENTHS	Tenth of a second time base.	&TM_TE
TMR_HUNDTHS	Hundredth of a second time base.	&TM_H
TMR_THSDTHS	Thousandth of a second time base.	&TM_TH
<i>Off-Delay Timer Function</i>		
TMR_TENTHS	Tenth of a second time base.	&OFDT_TE
TMR_HUNDTHS	Hundredth of a second time base.	&OFDT_H
TMR_THSDTHS	Thousandth of a second time base.	&OFDT_TH
<i>Counter Functions</i>		
UPCTR	Up counter.	&UP
DNCTR	Down counter.	&DN
<i>Integer Math Functions</i>		
ADD_INT	Signed integer addition.	&AD
SUB_INT	Signed integer subtraction.	&SUB
MUL_INT	Signed integer multiplication.	&MUL
DIV_INT	Signed integer division.	&DIV
MOD_INT	Signed integer modulo.	&MOD
<i>Double Integer Math Functions</i>		
ADD_DINT	Double precision integer addition.	&AD_DI
SUB_DINT	Double precision integer subtraction.	&SUB_DI
MUL_DINT	Double precision integer multiplication.	&MUL_DI
DIV_DINT	Double precision integer division.	&DIV_DI
MOD_DINT	Double precision integer modulo.	&MOD_DI



**Table 3-12. Substitution Groups (Continued)**

Function	Description	Type This
<i>Real/Floating-Point Math Functions</i>		
ADD_REAL	Real/floating-point addition.	&AD_R
SUB_REAL	Real/floating-point subtraction.	&SUB_R
MUL_REAL	Real/floating-point multiplication.	&MU_R
DIV_REAL	Real/floating-point division.	&DIV_R
EXPT_REAL	Real/floating-point power of X.	&EXPT_R
<i>Real/Floating-Point Trigonometric Functions</i>		
SIN_REAL	Sin.	&SIN
COS_REAL	Cosine.	&COS
TAN_REAL	Tangent.	&TAN
ASIN_REAL	Inverse sine.	&ASIN
ACOS_REAL	Inverse cosine.	&ACOS
ATAN_REAL	Inverse tangent.	&ATAN
<i>Real/Floating-Point Logarithmic Functions</i>		
SQRT_REAL	Square root.	&SQRT
LOG_REAL	Base 10 logarithm.	&LOG
LN_REAL	Natural logarithm.	&LN
EXP_REAL	Power of e.	&EXP
<i>Degree / Radian Conversion Functions</i>		
RAD_TO_DEG	Convert to degrees.	&DEG
DEG_TO_RAD	Convert to radians.	&RAD
<i>Integer Comparison Functions</i>		
EQ_INT	Test for equality between two integers.	&EQ
NE_INT	Test for no equality between two integers.	&NE
GT_INT	Test for one integer value greater than another.	&GT
GE_INT	Test for one integer value greater than or equal to another.	&GE
LT_INT	Test for one integer value less than another.	&LT
LE_INT	Test for one integer value less than or equal to another.	&LE
RANG_INT	Test the input value against a range of two numbers.	&RANG
<i>Double Integer Comparison Functions</i>		
EQ_DINT	Test for equality between two double precision integers.	&EQ_DI
NE_DINT	Test for no equality between two double precision integers.	&NE_DI
GT_DINT	Test for one double precision integer value greater than another.	&GT_DI
GE_DINT	Test for one double precision integer value greater than or equal to another.	&GE_DI
LT_DINT	Test for one double precision integer value less than another.	&LT_DI
LE_DINT	Test for one double precision integer value less than or equal to another.	&LE_DI
RANG_DINT	Test the input value against a range of two numbers.	&RANG_DI

Table 3-12. Substitution Groups (Continued)

Function	Description	Type This
<i>Real/Floating-Point Comparison Functions</i>		
EQ_REAL	Test for equality between two real/floating-point values.	&EQ_R
NE_REAL	Test for no equality between two real/floating-point values.	&NE_R
GT_REAL	Test for one real/floating-point value greater than another.	&GT_R
GE_REAL	Test for one real/floating-point value greater than or equal to another.	&GE_R
LT_REAL	Test for one real/floating-point value less than another.	&LT_R
LE_REAL	Test for one real/floating-point value less than or equal to another.	&LE_R
<i>Bit Operation Functions (Words)</i>		
AND_WORD	Logical “and” of two 16-bit word strings.	&AN
OR_WORD	Logical “or” of two 16-bit word strings.	&OR
XOR_WORD	Logical “exclusiveor” of two 16-bit word strings.	&XO
<i>Shift Bit String Functions (Words)</i>		
SHL_WORD	Shift bit string left.	&SHL
SHR_WORD	Shift bit string right.	&SHR
<i>Rotate Bit String Functions (Words)</i>		
ROL_WORD	Rotate bit string left.	&ROL
ROR_WORD	Rotate bit string right.	&ROR
<i>Bit Set/Clear Functions (Words)</i>		
BIT_SET_WORD	Set a bit within a bit string to 1.	&BS
BIT_CLR_WORD	Clear a bit within a bit string.	&BCL
<i>Integer Search Table Functions</i>		
ARRAY_MOVE	Copy from one array to another.	&AR
SRCH_EQ	Search for array values equal to a specified value.	&SRCHE
SRCH_NE	Search for array values not equal to a specified value.	&SRCHN
SRCH_GT	Search for array values greater than a specified value.	&SRCHGT
SRCH_GE	Search for array values greater than or equal to a specified value.	&SRCHGE
SRCH_LT	Search for array values less than a specified value.	&SRCHIT
SRCH_LE	Search for array values less than or equal to a specified value.	&SRCHLE
<i>Double Integer Search Table Functions</i>		
ARRAY_MOVE	Copy from one array to another.	&AR_DI
SRCH_EQ	Search for array values equal to a specified value.	&SRCHE_DI
SRCH_NE	Search for array values not equal to a specified value.	&SRCHN_DI
SRCH_GT	Search for array values greater than a specified value.	&SRCHGT_DI
SRCH_GE	Search for array values greater than or equal to a specified value.	&SRCHGE_DI
SRCH_LT	Search for array values less than a specified value.	&SRCHIT_DI
SRCH_LE	Search for array values less than or equal to a specified value.	&SRCHLE_DI

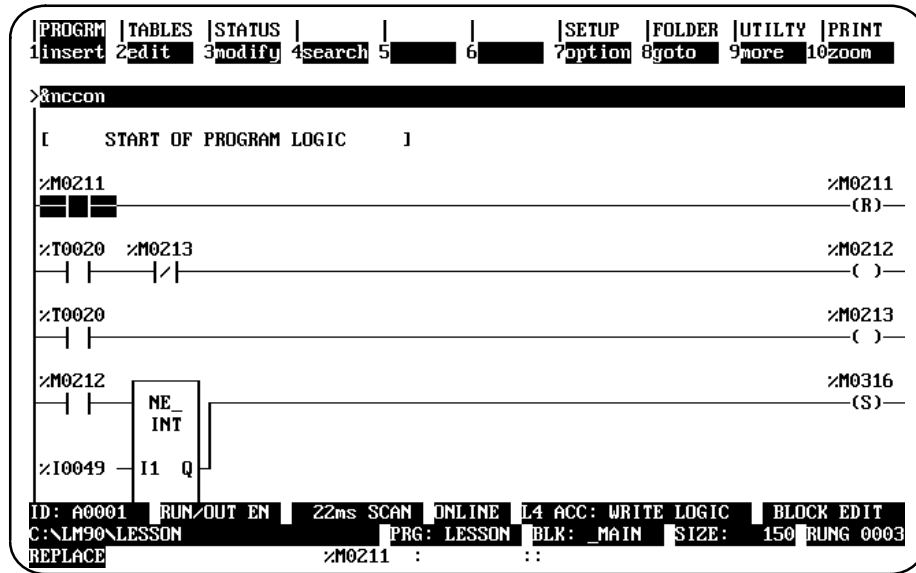
**Table 3-12. Substitution Groups (Continued)**

Function	Description	Type This
<i>Integer Search Table Functions(Words)</i>		
ARRAY_MOVE	Copy from one array to another.	&AR_W
SRCH_EQ	Search for array values equal to a specified value.	&SRCHE_W
SRCH_NE	Search for array values not equal to a specified value.	&SRCHN_W
SRCH_GT	Search for array values greater than a specified value.	&SRCHGT_W
SRCH_GE	Search for array values greater than or equal to a specified value.	&SRCHGE_W
SRCH_LT	Search for array values less than a specified value.	&SRCHLT_W
SRCH_LE	Search for array values less than or equal to a specified value.	&SRCHLE_W
<i>Integer Search Table Functions(Byte)</i>		
ARRAY_MOVE	Copy from one array to another.	&AR_BY
SRCH_EQ	Search for array values equal to a specified value.	&SRCHE_BY
SRCH_NE	Search for array values not equal to a specified value.	&SRCHN_BY
SRCH_GT	Search for array values greater than a specified value.	&SRCHGT_BY
SRCH_GE	Search for array values greater than or equal to a specified value.	&SRCHGE_BY
SRCH_LT	Search for array values less than a specified value.	&SRCHLT_BY
SRCH_LE	Search for array values less than or equal to a specified value.	&SRCHLE_BY

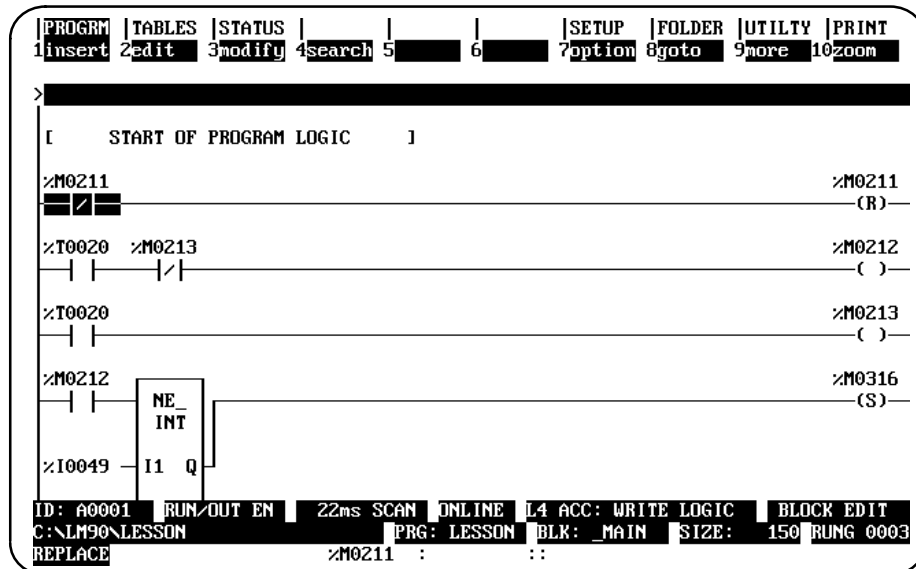
## Modifying Instructions

The following steps describe how to change relay ladder diagram elements and update the PLC while online and the PLC is running.

1. Place the cursor on the element to be changed.
2. Enter the new instruction mnemonic on the command line. The new instruction must be in the same instruction group as the existing instruction. (Instruction groups are listed in the table at the beginning of this section.)



3. Press the **Enter** key, and then type **Y** (Yes) in response to the confirmation prompt.



The same steps are used to replace functions for other functions in the same instruction group. For example, the NE\_INT function can be replaced with the LE\_INT function. With the cursor on the NE\_INT function, enter the new instruction mnemonic &LE\_INT on the command line and press the **Enter** key.

### Using the Modify Softkey

The **Modify (F3)** softkey enables you to substitute instructions from the rung editor environment by using the rung editor instruction softkeys. Position the cursor on the rung where the substitution is to be made and press **Modify (F3)**. Then, use the function softkeys to enter the new instruction. You can also use the **Modify (F3)** softkey to change a reference address or constant parameter and to create/modify nicknames.

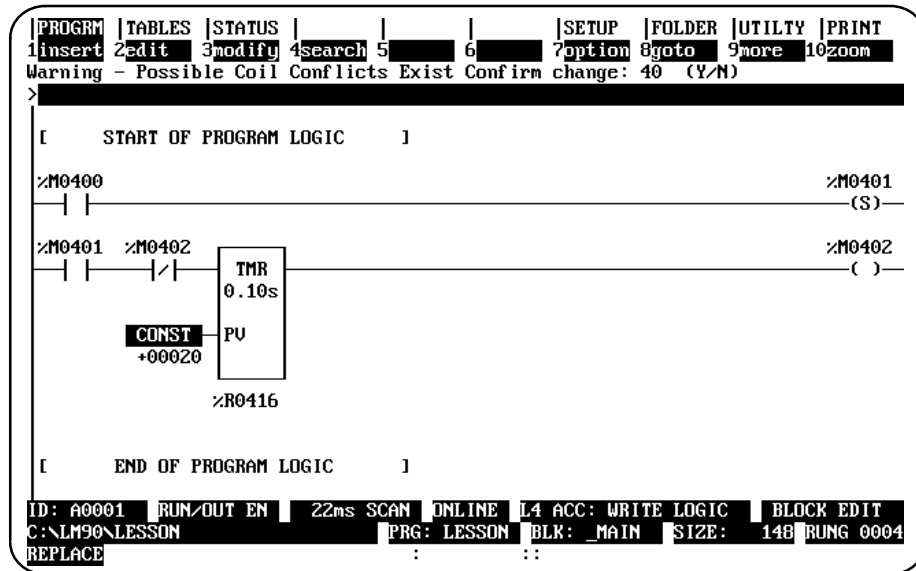
To restore the original values prior to the substitution, press **ALT-A**.

When the change is completed, press the **Escape** key.

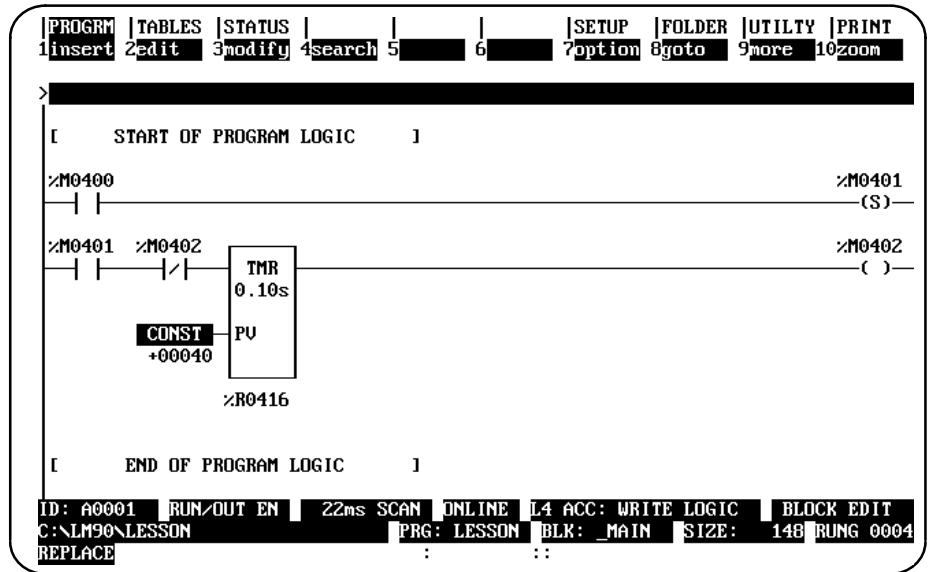
## Modifying a Reference Address or Constant

A reference address or other parameter can be changed while online and the PLC is running.

1. Place the cursor on the reference to be changed. In this example, the cursor is positioned on the TMR function's PV parameter, which currently contains a value of 20. Enter a new value of 40 on the command line, and press the **Enter** key.



- 2. Press Y (Yes) in response to the confirmation prompt to update the PLC with the new operand value.

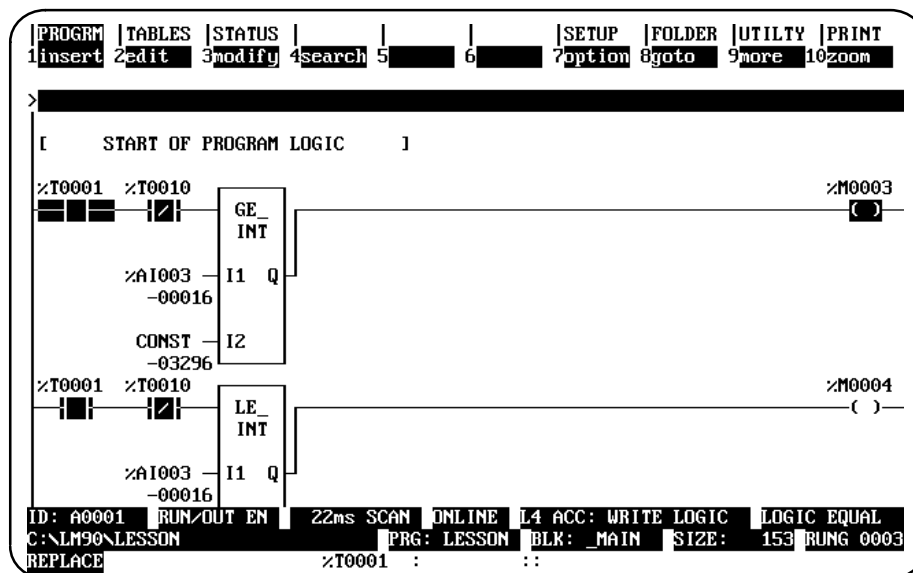


## Forcing and Overriding Discrete References

A value that is overridden can be protected from change by the program logic; however, an overridden value can be forced or toggled.

1. Discrete points can be forced or toggled by positioning the cursor on a contact or coil with the reference address to be modified, and pressing **F12** or the keypad (-) key.
2. For input points that are being scanned or output points that the logic program changes, you must first override the point by pressing **F11** or the keypad (\*) key before forcing the reference.
3. Once a point has been overridden, the first character of the reference address or nickname will flash.
4. To remove an override, position the cursor on the use of the reference address, and press **F11** or the keypad (\*) key again.

To force the discrete reference %T0001 in the following example, position the cursor on an element which uses that reference. Then, press **F12** to update the PLC.



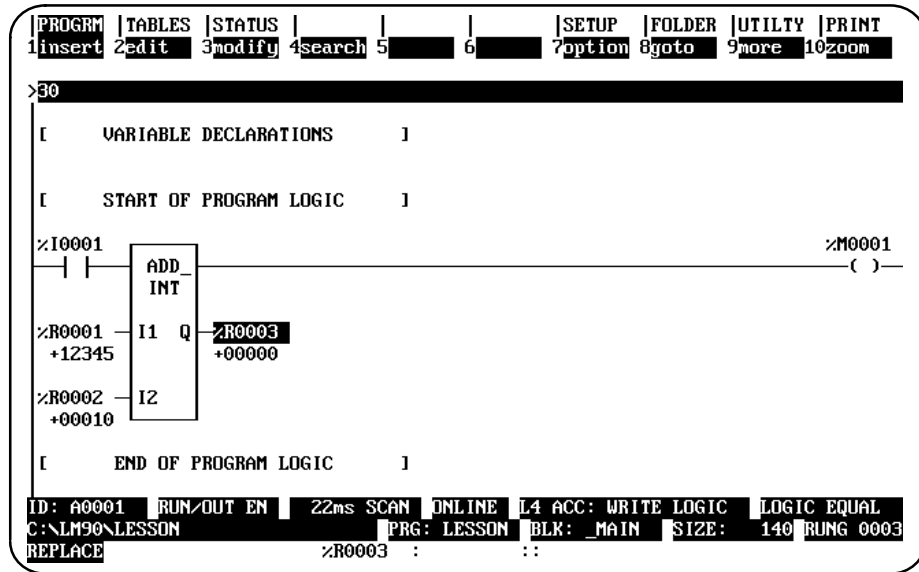
### Note

Only 331 and higher CPUs are capable of forcing and overriding discrete references.

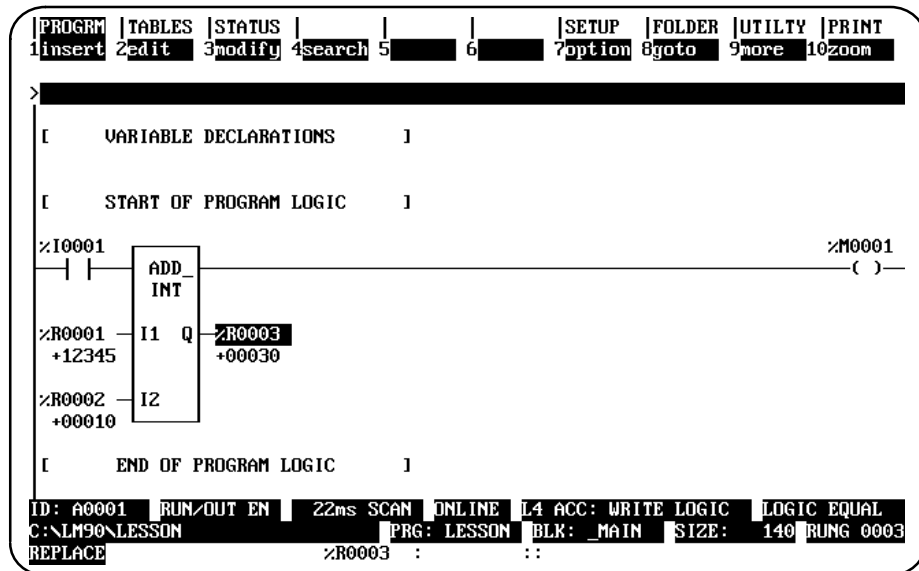
### Changing Register Values

The following steps describe how to change the value of a register at the current cursor position.

1. Place the cursor on the register to be changed. (For this example, place the cursor on %R0003.)
2. Enter the new value on the command line. (For this example, enter 30 on the command line.)



3. Press the **Enter** key. The register value for this example will change from 0 to 30, as shown in the screen below.





# Chapter 4

## Reference Tables

---

---

The display reference tables function is used to:

- Display tables of reference values.
- Set all the reference values in the table to zero.
- Change the formats in which reference tables are displayed or printed.
- Change reference values.
- Override discrete references (remove reference control from the PLC).

Chapter 4 contains the following sections:

Section	Title	Description	Page
1	Displaying Reference Tables	Explains how to display reference tables.	4-2
2	Changing Reference Table Values	Explains how to change reference values.	4-4
3	Using Overrides	Explains how to remove control of discrete references from the logic program.	4-9
4	Changing Display Formats	Explains how to make format changes.	4-11
5	Mixed Reference Tables	Describes mixed reference tables.	4-20

## Section 1: Displaying Reference Tables

Reference tables can be displayed directly from the main menu in any programming mode, or from any main menu function screen. While a table is displayed, you can change the format of any references in the table for a particular application.

This is a sample reference table used for discrete inputs (%I):

```

PROGRAM | TABLES | STATUS |          |          |          | SETUP | FOLDER | UTILITY | PRINT
1       | 2int    | 3dint  | 4real   | 5hex    | 6bin    | 7ascii | 8tmctr | 9mixed | 10hgall
>
          INPUT STATUS
          %I0001 NAME
00001  | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00065  | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00129  | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00193  | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00257  | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00321  | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00385  | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00449  | 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

          OFFLINE
C:\LM90\LESSON  PRG: LESSON
REPLACE          %I0001 : NAME      :: Reference Description
  
```

### Note

In **OFFLINE** mode, reference values from the current program folder are displayed. In **ONLINE** or **MONITOR** mode, references from the PLC are displayed.

Offline values from a reference table (values from the folder) can also be printed using the print function. These values will have the formats you set up on the display. To print values displayed online, the values must first be loaded from the PLC and then printed.

## Displaying a Reference Table

To display a reference table:

1. Enter one of the reference types listed in the following table:

User Reference	Reference Table
%I	Discrete input.
%Q	Discrete output.
%M	Discrete internal.
%T	Discrete temporary.
%G	Discrete Genius global data.
%S, %SA, %SB, %SC	Discrete system.
%AI	Analog input.
%AQ	Analog output.
%R	Register

- A. To view the lowest-numbered table of a particular type, enter just the reference type. For example, for the lowest-numbered analog input table, enter `%AI`.
  - B. To view a table containing a specific reference, enter the reference or its nickname. For example, `%AI123` or `123AI`.
2. Then, press **Tables (Shift-F2)**.
  3. Once in the reference tables function, you may go to a different reference table by entering the reference on the command line and pressing the **Enter** key.

## Using the Cursor to Select a Reference Table

You may go directly to the reference table of a reference (e.g., of a function block operand, contact, coil, etc.) under the cursor by pressing **ALT-F2**. Then, press **Shift-F1** from the reference table to return to the same place in the program.

## Moving the Cursor in a Reference Table

Follow these guidelines for moving the cursor in a reference table:

- Use the cursor keys to move one value horizontally or vertically in the table.
- To move the cursor to a specific reference in the same table, enter the reference on the command line and press the **Enter** key. To move to a specific reference in any table, enter the reference in the command line and press the **Enter** key, or use **Shift-F2**.
- To display the next screen of the same table, use the Page Down key. Use the Page Up key to display the previous screen.
- Use the CTRL-Page Down keys to move the cursor to the first reference of the next line. Use the CTRL-Page Up keys to move the cursor to the first reference of the previous line.
- Use the Home key to go to the first value in the table. Use the End key to go to the last value in the table.

## Section 2: Changing Reference Table Values

You can force (change) both discrete and register reference values with the reference tables function.

### Note

If the programmer mode is **OFFLINE**, changes will only be made to the current program folder. Changes made to the program folder may later be stored to the PLC by using the store utility function, described in chapter 8, “Program Utilities.” If the programmer mode is **ONLINE**, any changes will only be made to reference values in the PLC. Reference values in the PLC may then be loaded to the program folder by using the load utility function, also described in chapter 8. No changes to the PLC can be made while the programmer is in **MONITOR** mode.

### Warning

**Improper use of online program changes can damage equipment or cause personal injury.**

**Online program changes should always be made with extreme care. Online changes can have serious and unforeseen results on a control system, and on the process to which it applies, if they are improperly used. It is recommended that these functions not be used with people near the equipment. If possible, they should be done with direct visual control over the system and the process. Proper external power disconnects should be made to prevent undesired equipment operation.**

In order to make online changes, the status line at the bottom of the screen must show that the programmer is online to the CPU.

### Changing a Register Reference

You can enter or change the values of a register reference as described below. This method can be used to load a value into a register. To change a register reference value:

1. In the reference table that contains the reference, move the cursor to highlight the reference or enter the reference on the command line.
2. Enter a new value on the screen’s command line and press the **Enter** key. The new value will be interpreted in the format (for example, signed integer) currently assigned to the reference.

## ASCII String Entry

An ASCII string up to 79 characters, the size of the command line, can be entered in word-oriented tables. The string of text is entered the same as a name explanation and is displayed one character at a time, beginning with the reference the cursor is on.

The string is entered by typing the text, enclosed in quotes, on the command line and then pressing the **Enter** key. To include quotes within a string of text, you must begin the string with two colons. For example, to enter the string: **Use "ALT-N" to toggle** you must enter **::Use "ALT-N" to toggle** on the command line. A string can be accepted without a closing quote, but only to the last non-space character. Trailing spaces are lost without the closing quote.

The following screen shows the reference table display after the string "Registers in this table are displayed from left to right" is entered. (The display mode of any reference table can be changed from right to left or left to right, as explained on the following pages of this manual.)

```

|PROGRAM |TABLES |STATUS |      |      |      |SETUP |FOLDER |UTILITY |PRINT
1|        |2int   |3aint  |4real  |5hex   |6bin   |7ascii |8tnctr |9mixed |10hgall
>
REGISTER
%R0094 00000000
00001 R e g i s t e r s           i n           t h i
00011 s           t a b l e           a r e           d i s p
00021 l a y e d           f r o m           l e f t
00031           t o           r i g h t ^e ^e ^e ^e ^e ^e ^e ^e
00041 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00051 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00061 U s e           " A L T - N "           t o
00071 t o g g l e           t h e           d i r e c
00081 t i o n           o f           t h e           d i
00091 s p l a y .           ^C ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00101 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00111 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
OFFLINE
C:\LM90\LESSON           PRG: LESSON
REPLACE           %R0094 :           ::

```

Non-printable characters (e.g., "null" to terminate a string) may be included in the ASCII string by entering a backslash and the three-digit decimal value of the non-printable character (e.g., "\000"). If you wish to include a backslash (\) in the text, enter two backslashes. There is no limit, other than the size of the command line, on the number of non-printable characters a string may have. The software performs a check to verify that the value is a valid decimal number.

This example shows a null terminated string.

```

|PROGRAM |TABLES |STATUS | | | |SETUP |FOLDER |UTILITY |PRINT
1 | | 2|int  | 3|int  | 4|real | 5|hex  | 6|bin  | 7|ascii | 8|tmctr | 9|mixed |10|hgall
> "To enter a backslash, type \ like this\000
REGISTER
                                %R0151                                00000000
00061 U s e " A L T - N " t o
00071 t o g g l e t h e d i r e c
00081 t i o n o f t h e d i

00091 s p l a y . ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00101 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00111 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e

00121 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00131 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00141 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e

00151 ^c ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00161 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00171 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e

OFFLINE
C:\LM90\LESSON PRG: LESSON
REPLACE %R0151 : :

```

The next example shows the reference table after the null terminated string is accepted.

```

|PROGRAM |TABLES |STATUS | | | |SETUP |FOLDER |UTILITY |PRINT
1 | | 2|int  | 3|int  | 4|real | 5|hex  | 6|bin  | 7|ascii | 8|tmctr | 9|mixed |10|hgall
>
REGISTER
                                %R0151                                01010100
00061 U s e " A L T - N " t o
00071 t o g g l e t h e d i r e c
00081 t i o n o f t h e d i

00091 s p l a y . ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00101 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00111 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e

00121 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00131 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e
00141 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e

00151 T o e n t e r a b a c k s l a s h
00161 , t y p e \ l i k e t h i s ^e ^e
00171 ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e ^e

OFFLINE
C:\LM90\LESSON PRG: LESSON
REPLACE %R0151 : :

```

## Changing the Reference Display Mode (ALT-N)

The ALT-N key sequence enables you to display table data either right to left (lowest reference address on the right) or left to right (lowest reference address on the left). The display mode can be changed in any table (fixed or mixed) and in any program mode (OFFLINE, ONLINE, or MONITOR).

The Home End, Page Up and Page Down, Previous, Next, and cursor keys function the same, regardless of the view mode selected.

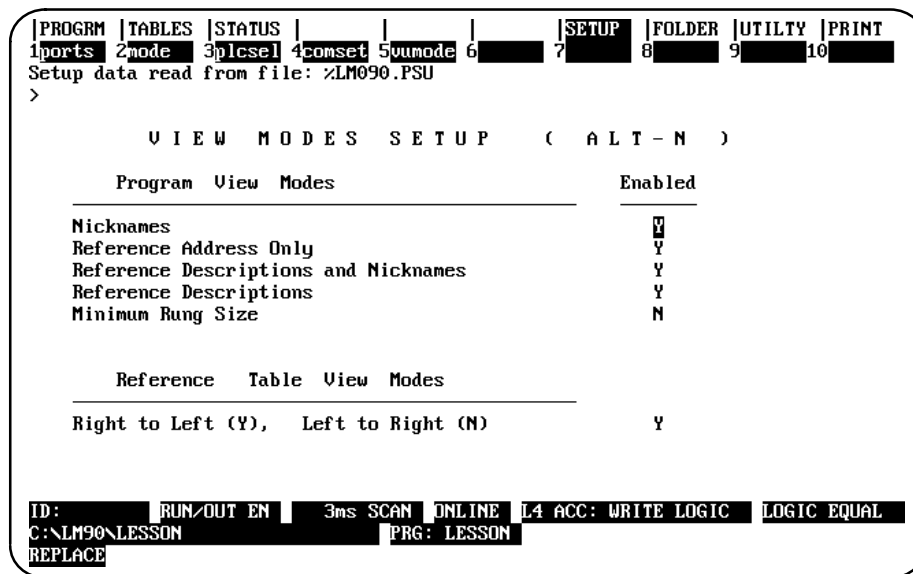
Data displayed in ASCII format differs slightly, in that the quotes are removed and the format is treated more as a byte (1 character) instead of a word (2 characters).

### Note

The print function cannot distinguish between these two display modes and prints all tables right to left.

The default display mode is the last mode selected with ALT-N. If the display mode has never been changed, the default set in the programmer setup is used (see chapter 6, "Programmer Setup"). Follow the steps below to change the display mode.

1. The default reference table display mode is displayed on the View Modes Setup screen. To display this screen, press **Setup (F7)** from the Programming Software main menu and then **View Modes Setup (F5)** from the Programmer Setup menu. The view mode in the following screen is right to left (the default display).



2. To change the default view mode to display left to right, move the cursor to the, **Enabled**, field for the, **Reference Table View Modes**, entry. Then, use the Tab key to toggle the selection to **N** (No) or enter **N**.
3. To validate and save the default view mode selection in the home directory file %LM090.PSU, press **ALT-U**, the **Escape** key, or the shifted function key.

### Forcing a Discrete Reference

In discrete reference tables, a reference can be forced on or off. If the reference being forced is currently overridden (described below), it retains its new status until forced again. If the reference is not overridden, it retains its new status until changed by some other function, such as rung solution or I/O servicing. This usually occurs within one sweep.

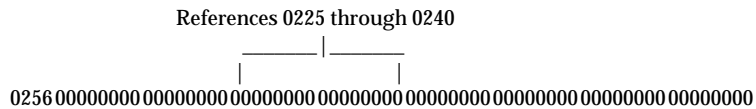
You can enter the desired state (0 or 1) on the command line, or toggle the reference state as described below. The following method is easier.

To toggle a reference:

1. With the table displayed on the screen, place the cursor on the reference to be forced.
2. Press the keypad minus (-) key or F12 key to change that reference to its opposite state. All logic elements in the program that use the reference will reflect the new status.

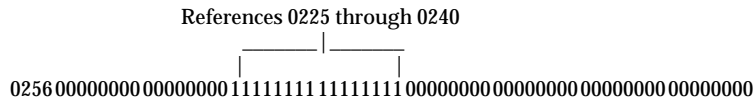
### Changing the Values of a Word of Discrete References

You can change the value of the word where the cursor is. To do that, first you must change the number base of the references. For example, suppose you want to convert discrete references (see below) 0225 through 0240 to ones.



Follow the steps below to change a word of discrete references.

1. The cursor indicates the rightmost (lower-numbered) byte of the word to be changed. (The software automatically adjusts the cursor position if it is incorrectly placed.) Use the function keys to change the number base to one of the following: signed integer (by pressing F2) or hexadecimal (by pressing F5).
2. On the command line, type the equivalent of the binary value in the selected format.
3. Press the Enter key to change the reference at the cursor position to the new value.
4. Press Binary (F6) to convert the base back to binary values.





## Section 3: Overrides

Discrete references in Models 331 or higher CPU can be overridden from the reference tables display. (This feature is not available in Models 323, 321, 311, 313, 211, and Micro CPUs.)

### Note

If you attempt to use overrides with a CPU below Model 331, you will see an error message on your screen stating that overrides are not allowed.

Discrete references that have been overridden are indicated by flashing digits on the reference table screen. An override removes control of the reference from its normal source. Overridden inputs ignore information from the devices wired to the I/O structure, such as limit switches or pushbuttons. Similarly, overridden outputs ignore programmed logic and internal power flow. Overrides are retained even when power is removed from the system. Non-relay functions such as timers, counters, math functions, and data move functions still work when a coil is overridden.

### Warning

**If overrides are applied to a reference associated with a transitional coil, the coil may pulse on for one sweep when the override is removed.**

The override is a very powerful tool for program checking and maintenance. You can test a program in a PLC that is not connected to I/O hardware by using overrides to simulate inputs. You can also check a program when I/O is connected, by using overrides to prevent coil operation.

After the I/O is wired up, it can be tested by activating each coil with an override to verify I/O communications, module operation, power to a device, wiring to a device, indicator lights, fuses, and other hardware.

After the control system is thoroughly checked and placed in operation, the override is useful in a monitored system. If a sensor or input module should fail while the process is in operation, that input can be overridden. Thus, the process can be continued until it can be shut down safely.

References should not be overridden when the programmer is removed from the process, or when making copies of a program. Use the reference tables function to verify all inputs and coils before removing them from the programmer, or copying the program.

## Using Overrides

Overrides should be used on an operating system only with extreme care.

### Warning

**Improper use of the override can damage equipment or cause personal injury.**

1. Place the cursor on the reference to be overridden.

### Caution

**The reference will be overridden throughout the program, not just at the cursor location.**

2. Press the keypad Asterisk (\*) key or **F11**. This toggles the state of the reference between overridden and not overridden. When overridden is selected as the state of reference, the value displayed in the table will flash.

## Removing Overrides

All discrete reference bits on a reference table screen, or starting character in a tag name on a displayed rung, will flash if they have been overridden. To remove an override from one reference, toggle it by placing the cursor on the reference and pressing the keypad Asterisk (\*) key or **F11**.

To remove all overrides shown on the current screen, including the last three lines which may be hidden beneath the status lines (Press **ALT-E** to remove the status lines.):

1. Press **Change All (F10)**.
2. Press the keypad Asterisk (\*) key or **F11**. The screen prompts: "Remove overrides from displayed references? (Y/N)".
3. Enter **Y** (Yes) to remove the overrides.

## Section 4: Changing Display Formats

The format of any reference in a table can be changed for a particular application while the table is displayed on the programmer screen. This section explains how to make format changes.

### Discrete Reference Tables

The sample reference table below is used for discrete inputs (%I), discrete outputs (%Q), discrete internal coils (%M), discrete temporary coils (%T), discrete system status references (%S, %SA, %SB, and %SC), and discrete global data (%G).

```

|PROGRAM |TABLES |STATUS |      |      |      |SETUP |FOLDER |UTILITY |PRINT
1|       |2int   |3dint  |4real  |5hex  |6bin  |7ascii |8tmctr |9mixed |10hgall
>
                                INPUT STATUS
                                %I0001 NAME
00001  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00065  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00129  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00193  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00257  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00321  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00385  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00449  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

                                OFFLINE
C:\LM90\LESSON          PRG: LESSON
REPLACE                %I0001 : NAME      :: Reference Description

```

The default format for discrete reference table displays is to have the values displayed or printed in binary. The format may be changed to signed integer or hexadecimal.

Any of these formats can be used for some of the values in the table, or for the entire table. Thus, each reference table can be uniquely formatted to be most meaningful for the type of information it contains. (Double precision integer and timer/counter are not allowed for discrete references.)

### Note

The format for system status references (%S, %SA, %SB, and %SC) cannot be changed. System status references can only be displayed in binary format. The %S reference table values cannot be cleared or changed. %S memory is read only and cannot be written. The bit values in the %SA, %SB, and %SC reference tables can, however, be cleared or toggled.

## Register Reference Tables

The sample register reference table below is used for system registers (%R), analog inputs (%AI), and analog outputs (%AQ).

```

|PROGRAM|TABLES|STATUS| | | |SETUP|FOLDER|UTILITY|PRINT
|-----|-----|-----| | | |-----|-----|-----|-----
1|      |Zint  |3|int  |4|real  |5|hex  |6|bin  |7|ascii |8|tmctr |9|mixed |10|hgall
>
                                     ANALOG INPUT
                                     %AI001
00001 +00000 +00000 +00000 +00000 +00000 +00000000 00000000
00011 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000
00021 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000

00031 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000
00041 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000
00051 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000

00061 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000
00071 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000
00081 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000

00091 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000
00101 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000
00111 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000

                                     OFFLINE
C:\LM90\LESSON          PRG: LESSON
REPLACE                %AI001 :      ::

```

The default format for register reference table displays is for the values to be shown or printed as signed integers. The format can be changed to double precision integer, hexadecimal, binary, ASCII, or timer/counter. The format can also be changed for some of the values in the table or for the entire table.

In addition, the value of the reference at the cursor is displayed at the top of the table in binary format.

## Display Formats

Use the function keys to change display formats. Changes are automatically saved when you leave a reference table display. The table will continue to be displayed in the new format until the format is changed again. If you want to save changes to the program folder **without** leaving the reference table display, press **ALT-U**.

Format	Function Key	Description
Signed Integer	F2	A 16-bit number from -32,768 to +32,767. In discrete reference tables, 16 consecutive references are used for the display, beginning at a multiple-of-8-plus-one boundary (e.g., 1, 9, 17, 25, 33, etc.). In register reference tables, a single reference is required (e.g., -21846).
Signed Double Integer	F3	A 32-bit number from -2,147,483,648 to +2,147,483,647. This format can be used for register reference tables only. Two consecutive references are required, beginning at a multiple-of-2-minus-one boundary (e.g., 1, 3, 5, 7, etc.).
Real	F4	32 consecutive bits (actually two consecutive 16-bit memory locations). This format can be used for register reference tables only. The range of numbers that can be stored in this format is from $\pm 1.401298E-45$ to $\pm 3.402823E+38$ . The Real format requires a 352 CPU.
Hexadecimal	F5	A 16-bit number from 0000 to FFFF. When entering hexadecimal numbers on the command line, a leading A through F must be preceded by a zero. In addition, type an <b>h</b> at the end of the numeric string to ensure that the entry is read as a hexadecimal number. In discrete reference tables, 16 consecutive references are used, beginning at a multiple-of-16 boundary (e.g., 1, 17, 33, etc.). In register reference tables, a single reference is required (e.g., 0AAAAh).
Binary	F6	A 1-bit number with a value of 0 or 1. This format can be used for discrete reference tables only. A single reference is used. Binary values are displayed in groups of 8 references (e.g., 10101010).  Register references cannot be displayed in binary format. However, the 16-bit binary equivalent of the value indicated by the cursor is automatically displayed at the top of the table.
ASCII	F7	8-bit encoded characters. This format can only be used in register reference tables. A single reference is required to make up 2 (packed) ASCII characters. The rightmost character of the pair corresponds to the low byte of the reference word. Bit 8 of each pair is a parity bit and is ignored. The remaining 7 bits in each section are converted as shown below. Command codes and non-displayable characters appear on the screen as the characters <b>^@</b> (e.g., X W).
TMRCTR	F8	Three contiguous reference addresses, which support timer/counter functions, display the current value, the preset value, and the control word. Regardless of whether the reference is assigned to a timer or counter, the format can be used as long as the reference type is %R. More information on this format is provided later in this chapter.
Mixed	F9	Select a user-defined table. Up to 99 user-defined tables may be created.
Change All	F10	Press <b>F10</b> to change the format of all elements in the table.

Table 4-1. ASCII Characters

Bit Pattern	Character	Bit Pattern	Character	Bit Pattern	Character	Bit Pattern	Character
X0000000	^@	X0100000	(blank)	X1000000	@	X1100000	'
X0000001	^A	X0100001	]	X1000001	A	X1100001	a
X0000010	^B	X0100010	"	X1000010	B	X1100010	b
X0000011	^C	X0100011	#	X1000011	C	X1100011	c
X0000100	^D	X0100100	\$	X1000100	D	X1100100	d
X0000101	^E	X0100101	%	X1000101	E	X1100101	e
X0000110	^F	X0100110	&	X1000110	F	X1100110	f
X0000111	^G	X0100111	'	X1000111	G	X1100111	g
X0001000	^H	X0101000	(	X1001000	H	X1101000	h
X0001001	^I	X0101001	)	X1001001	I	X1101001	i
X0001010	^J	X0101010	*	X1001010	J	X1101010	j
X0001011	^K	X0101011	+	X1001011	K	X1101011	k
X0001100	^L	X0101100	,	X1001100	L	X1101100	l
X0001101	^M	X0101101	-	X1001101	M	X1101101	m
X0001110	^N	X0101110	.	X1001110	N	X1101110	n
X0001111	^O	X0101111	/	X1001111	O	X1101111	o
X0010000	^P	X0110000	0	X1010000	P	X1110000	p
X0010001	^Q	X0110001	1	X1010001	Q	X1110001	q
X0010010	^R	X0110010	2	X1010010	R	X1110010	r
X0010011	^S	X0110011	3	X1010011	S	X1110011	s
X0010100	^T	X0110100	4	X1010100	T	X1110100	t
X0010101	^U	X0110101	5	X1010101	U	X1110101	u
X0010110	^V	X0110110	6	X1010110	V	X1110110	v
X0010111	^W	X0110111	7	X1010111	W	X1110111	w
X0011000	^X	X0111000	8	X1011000	X	X1111000	x
X0011001	^Y	X0111001	9	X1011001	Y	X1111001	y
X0011010	^Z	X0111010	:	X1011010	Z	X1111010	z
X0011011	^[	X0111011	;	X1011011	[	X1111011	{
X0011100	^	X0111100	<	X1011100	\	X1111100	
X0011101	^]	X0111101	=	X1011101	]	X1111101	}
X0011110	^^	X0111110	>	X1011110	^	X1111110	tilde
X0011111	^_	X0111111	?	X1011111	_	X1111111	delta

## Changing the Display Format

To change the format of one reference, move the cursor to that reference and press the appropriate function key. The new format will appear at the rightmost position of the reference(s). Within any standard reference table, you can change as many reference formats as you want, as shown by the example below.

```

|PROGRAM|TABLES|STATUS|_|_|_|SETUP|FOLDER|UTILITY|PRINT
1|      |2int  |3dint  |4real  |5hex  |6bin  |7ascii |8tmctr |9mixed |10chgall
>-----
                                INPUT STATUS
                                %I0257
00001          +00000          +00000          0000          0000
00065          0000          0000          0000          0000
00129  00000000 00000000 00000000 00000000 00000000 00000000 00000000

00193          +00000          +00000          +00000          +00000
00257  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00321  00000000 00000000 00000000 00000000 00000000 00000000 00000000

00385  00000000 00000000 00000000 00000000 00000000 00000000 00000000
00449  00000000 00000000 00000000 00000000 00000000 00000000 00000000

                                OFFLINE
C:\LM90\LESSON          PRG: LESSON
REPLACE          %I0257 :          ::
  
```

## Changing the Format of a Table

You can also change the format of an entire table, including the values not currently on the screen or portions of the table. To change the format of an entire table:

1. Press **Change All (F10)**.
2. Select the format for the display. For example, press **F2** to select signed integer.

```

|PROGRAM|TABLES|STATUS|_|_|_|SETUP|FOLDER|UTILITY|PRINT
1|      |2int  |3dint  |4real  |5hex  |6bin  |7ascii |8tmctr |9mixed |10chgall
>-----
                                INPUT STATUS
                                %I0497
00064          +08272          +08976          +22748          +23625
00128          +13352          +15920          +02078          +04712
00192          +00000          +07202          +00000          +28264

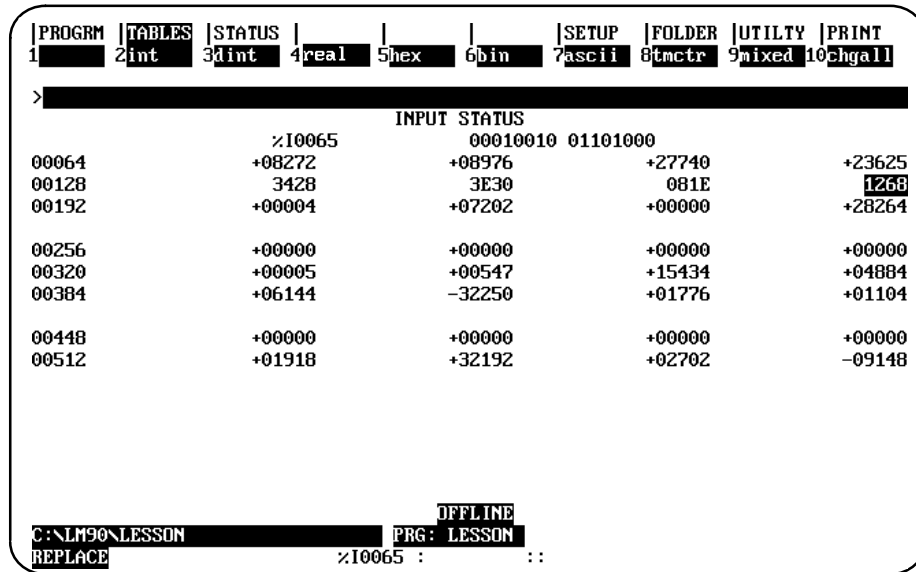
00256          +00000          +00000          +00000          +00000
00320          +00005          +00547          +15434          +04884
00384          -32250          +00000          +01776          +01104

00448          +00000          +00000          +00000          +06144
00512          +01918          +32192          +02702          -09148

                                OFFLINE
C:\LM90\LESSON          PRG: LESSON
REPLACE          %I0497 :          ::
  
```

The **Change All (F10)** key can also be used to change a specified block of range formats.

1. Enter a starting reference (the lower address) and an ending reference (the higher address) from the current table (e.g., 65i 128i).
2. Press **Change All (F10)**.
3. Select the format (e.g., hex).



Entering only a valid starting reference will cause the formats from the starting reference to the end of the table to change. Entering a valid starting reference followed by an invalid entry will also cause the formats from the starting reference to the end of the table to change.

To restore the previous format (and in **OFFLINE** mode, restore the data values stored at the time of the last disk update), press **ALT-A** to abort the change. Pressing **ALT-A** will undo **all** format changes and value changes made since the last disk update.

To save all the changes and remain in the reference table, press **ALT-U**. To save the changes and exit from the reference table, press the **Escape** key.



## Timer/Counter Format

To support the timer/counter function, a timer/counter format requiring three contiguous reference addresses will display the current value, the preset value, and the control word. In the control word, bit 15 contains the output status (Q), and bit 16 contains the enable status (EN). The rest of the bits in the control word are not used by a counter. For a timer, the rest of the control word bits contain the timer accuracy.

The preset value (PV) can be modified in both **ONLINE** and **OFFLINE** mode. If a reference or constant was assigned to the PV parameter of the timer/counter function, the contents of the reference or the constant are written to the PV reference of the timer/counterformat when in **RUN** mode.

Regardless of whether the reference is assigned to a timer or counter, the format can be used as long as the reference type is %R.

The example screen below shows the timer/counter format displayed, using the Timer/Counter (F8) function key.

```

|PROGRAM |TABLES |STATUS | | | |SETUP |FOLDER |UTILITY |PRINT
1|       |2int   |3dint  |4real  |5hex   |6bin   |7ascii  |8tmctr  |9mixed  |10hgall
>
REGISTER
%R0261 00000000 00000000
00261 +00000 +00000 +00000000000 +00000 +00000 0000 +00000 +00000 +00000
00271 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000
00281 +00000 +00000 +00000 +00000 +00000 +00000 CU: +00000 +00000
PU: +00000
EN: 0, Q: 0

00291 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000
00301 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000
00311 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000

00321 CU: +00000 +00000 +00000 +00000 CU: +00000 +00000
PU: +00000 PU: +00000
EN: 0, Q: 0 EN: 0, Q: 0

00331 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000
00341 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000 +00000
OFFLINE
C:\LM90\LESSON PRG: LESSON
REPLACE %R0261 : ::

```

## Returning to Default Values

After editing, you can return a standard reference table to its default format and fill the table locations with zeros. This will:

- Set all the reference values in the table to zero.
- Change all references in a discrete reference table back to binary format.
- Change all references in a register reference table back to signed decimal format.

To restore the table to its default content:

1. Select **Change All (F10)**.
2. Enter **0** on the command line, and press the **Enter** key. The screen prompts: "Initialize table data and formats to default? (Y/N)".
3. Enter **Y** (Yes) to restore the table's default content. For example:

```

|PROGRAM |TABLES |STATUS | | | |SETUP |FOLDER |UTILITY |PRINT
1|       |2int   |3dint  |4real  |5hex  |6bin  |7ascii |8tmctr |9mixed |10hgall
>
                                INPUT STATUS
                                %I0001 NAME
00064  00000000 00000000 00010000 00000000 00000000 00000000 00000000 00000000
00128  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00192  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00256  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00320  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00384  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00448  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00512  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

ID: A0001  RUN/OUT EN  22ms SCAN  ONLINE  L4 ACC: WRITE LOGIC  LOGIC EQUAL
C:\LM90\LESSON          PRG: LESSON
REPLACE                %I0001 : NAME      :: Reference Description
  
```

## System Reference Table

The sample system reference table below is used for system references (%S, %SA, %SB, and %SC), while in **ONLINE** or **MONITOR** mode. (No data is displayed in **OFFLINE** mode.) The system reference table resembles a mixed reference table; however, format changes are not allowed, the Change All (F10) function key cannot be used, and the cursor cannot be moved to blank lines.

```

|PROGRAM |TABLES |STATUS |      |      |      |SETUP |FOLDER |UTILITY |PRINT
1|       |2int  |3dint |4real |5hex  |6bin  |7ascii |8tmctr |9mixed |10hgall
>
          SYSTEM STATUS
          %SA001  PB_SUM
%S0032 |      |      |      |      |      |      |      |      |      |
%SA032 |      |      |      |      |      |      |      |      |      |
%SB032 |      |      |      |      |      |      |      |      |      |
%SC032 |      |      |      |      |      |      |      |      |      |
          00000000 00000000 00010000 01110010
          00000000 00000000 00000000 00000000
          00000000 00000000 00000000 00000000
          00000000 00000000 00011111 00000000

ID: A0001  RUN/OUT EN  22ms SCAN  ONLINE  L4 ACC: WRITE LOGIC  LOGIC EQUAL
C:\LM90\LESSON      PRG: LESSON
REPLACE          %SA001 : PB_SUM  ::

```

The system reference table is displayed with the cursor on the reference address entered, or the reference address associated with the nickname entered.

In the screen shown above, pressing the **Home** or **Page Up** key will position the cursor on %S0001. Pressing the **End** or **Page Down** key will place the cursor on %SC0032.

## Section 5: Mixed Reference Tables

A mixed table represents a collection of data from one or more of the fixed tables. Therefore, the identical information displayed in a mixed table could also be displayed on the related fixed tables.

A total of 99 user-defined mixed reference table displays can be supported. A number from 1 to 99 is associated with each table.

### Note

The mixed table function is used only for viewing PLC information; therefore, mixed reference tables are not valid when the programmer mode is **OFFLINE**. When **OFFLINE**, no cursor movement or other mixed table functions are allowed, except for changing tables, exiting reference tables, or changing modes.

To select a mixed table, enter a number from 1 through 99 on the command line and press **Mixed (F9)**. This is an example of a mixed reference table display:

```

|PROGRAM |TABLES |STATUS |      |      |      |SETUP |FOLDER |UTILTY |PRINT
1|      |2int  |3aint  |4real  |5hex  |6bin  |7ascii |8tctr  |9mixed |10hgall
>
TITLE: EMERGENCY SHUT DOWN
MIXED TABLE 01      %M0217      00111100 00010000
%R0010      v u t s r q o n m l k j i h f e d c b a
%R0050      +00000      CU: +00002 +00234 +00000      CU: +00000 +00000
      PU: +00000      PU: +00000
      EN: 0, Q: 0      EN: 0, Q: 0

%S0032      00000000 00000000 00010100 01101110

%M0256      00001111 10001000 11100000      +15376 01000100 10000100 10011000
%M0512      04FF 00000100 10000100 00100100      +17545 00100000

%I0192      01010010 00100100      +02853 00101001 00100010 01001000 10101000

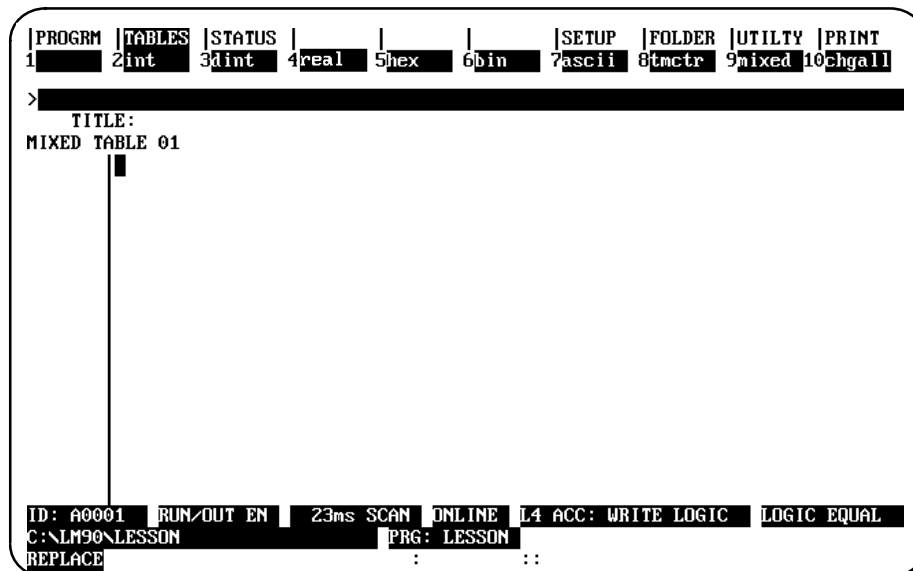
%AQ020      +00000 +06345 +00000      +0003764839 +08578 +04523 +00975 +04538 +00546

%T0064      +02345 00000101      +00054 00000100 00000001 01100100

ID: ABCDEF RUN/OUT EN      6ms SCAN ONLINE L4 ACC: WRITE LOGIC LOGIC EQUAL
C:\LM90\LESSON      PRG: LESSON
INSERT      %M0217 :      ::
  
```

Field	Description
User-Defined Table Title	A description of the mixed table. A maximum number of 63 characters is allowed. When the cursor is on the 63rd character, it remains fixed, overwriting the last character.
Mixed TableNumber	A number from 1 to 99, which indicates the mixed table currently displayed.
Cursor Item Line	Information about the address the cursor is on; namely, the current reference, reference nickname, and current value binary representation.
Table Data	16 display lines. Some may be program block headers or blank lines.

If the mixed table entered is not defined, a blank mixed table (no data, no title, only the mixed table number) is displayed, as shown below.



### Editing the Title

To edit the table title, position the cursor on the title of the display by moving the cursor to the topmost data display line and pressing the **Up** cursor key. While editing the title, the Update or Abort key can be used to save or restore a title. A message is displayed when the function is complete.

### Note

Since mixed reference tables are defined on a line-by-line basis, the abort function will restore the lines, formats, and title displayed upon entry. Unless an update to disk was done, any lines defined or title edited are lost.

The **Mixed (F9)** key is used to change from one mixed table to another. The **Tables (Shift-F2)** key sequence can also be used to get to any table. The **Change All (F10)** key is not supported for global format changes, as in fixed table displays.

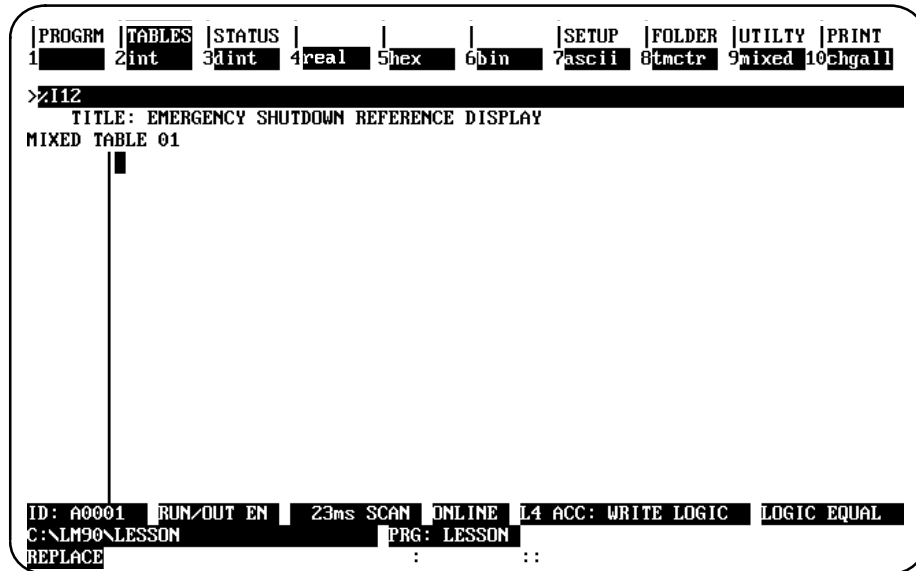
To return to the display area, press the **Enter** key or the **Down** cursor key. The cursor will return to the original position in the topmost data display line.

## Defining a Mixed Table

A mixed reference table display is created or changed by pressing the **Change Line (F10)** key.

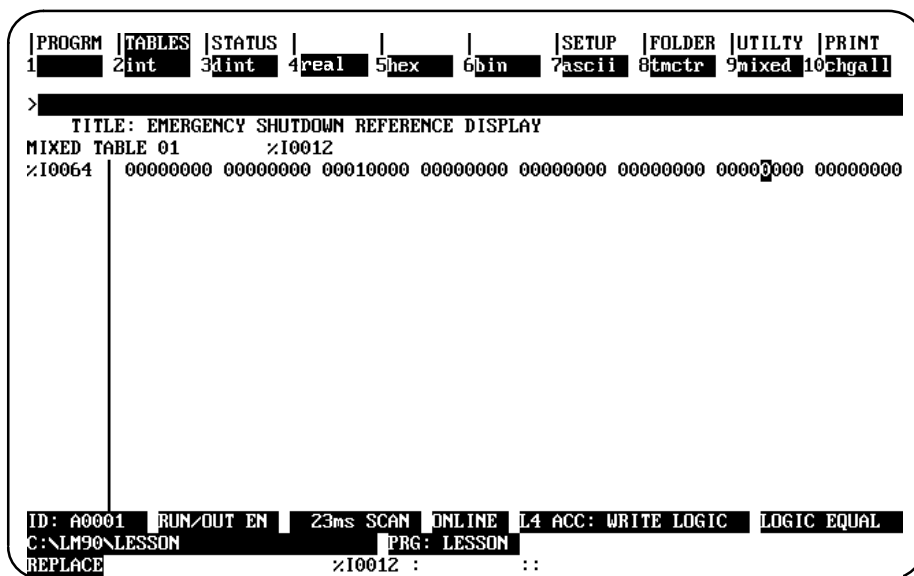
1. First, position the cursor on the display line where a certain data value should be displayed. The line may already be defined, or it may be a blank line.
2. Enter the reference address or nickname of the desired data value on the command line.

In the following example, a line is defined in mixed reference table 1. With the cursor positioned on the top line of the display, enter `%I1` onto the command line.



- 3. Press **Change Line (F10)** to display the desired information. The cursor will be positioned on the requested data value. Any information currently displayed is replaced with the new requested information.

In this example, when **F10** is pressed, the data value positioned at reference %I0012 is entered into the topmost display line.



The default display format for data extracted from a discrete table is binary; for register tables, it is signed integer.

The timer/counter format can be used for a %R line in a mixed table if enough exists for the format. If the two lines below the %R line are defined or are beyond the table size, an error message is displayed.

### Deleting a Line

If the command line is blank when the **Change Line (F10)** key is pressed, the line is cleared and the cursor is automatically moved to the next line.

If the current line is already blank, moving the cursor to the next line will produce the same result as pressing **F10** with the command line blank.

## Moving the Cursor

Cursor movement is the same in mixed tables as in fixed tables. Since blank lines can be defined in a mixed table, the cursor is allowed on blank lines.

Cursoring up is allowed when the cursor is on the topmost line, in order to access the title. You can cursor to lines which contain a message, indicating that the reference is out of range. This allows you to delete or re-define the line.

The same line of references can also be defined twice on the same mixed table display. If the Enter key is used to move the cursor to a duplicate reference address, the search for the reference address begins with the next reference and continues to the end of the table. If the reference address is not found in that section, the search begins with the first non-blank line, lowest reference. If the reference address is still not found, an error message is displayed.

The Enter key cannot be used to change tables.

## Timer/Counter Format

The timer/counter format in a mixed table is the same as in a fixed table, except for the first timer/counter format of the line. In the timer/counter format, the next two lines must be blank so that there is sufficient room for the format. Otherwise, an error message is displayed.



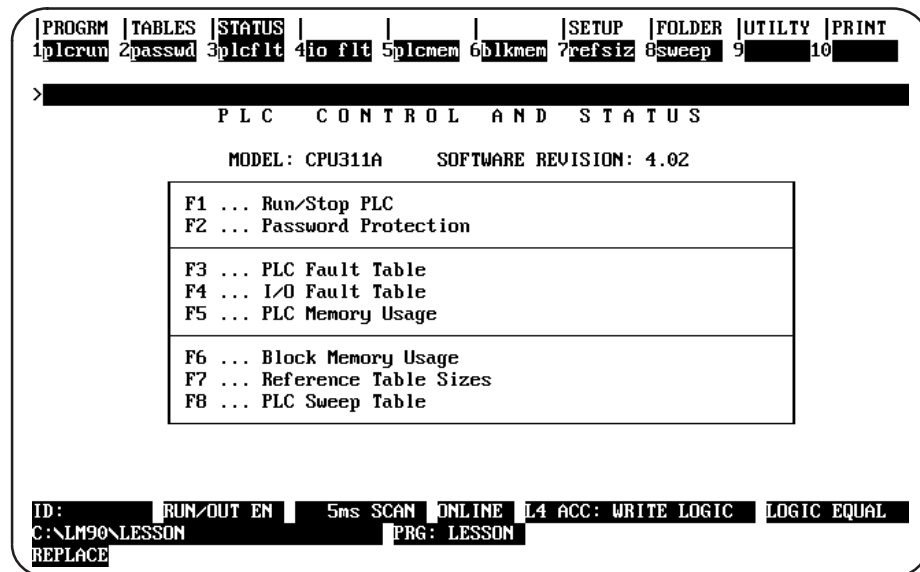
# Chapter 5

## PLC Control and Status

The programmer can interact with an operating PLC in many ways. Monitoring program execution and reference tables was described in previous chapters. Several additional features involve interaction between the programmer and an operating PLC. To use most of them, the programmer must be connected to the PLC and must be in **ONLINE** or **MONITOR** mode.

If the programmer is not connected or is in **OFFLINE** mode, asterisks may be displayed in place of values in many fields when the function screens are displayed. In addition, some of these features may be protected by passwords.

To use these features, press **STATUS (F3)** from the main menu, or **Shift-F3** from any other main menu function.



### Note

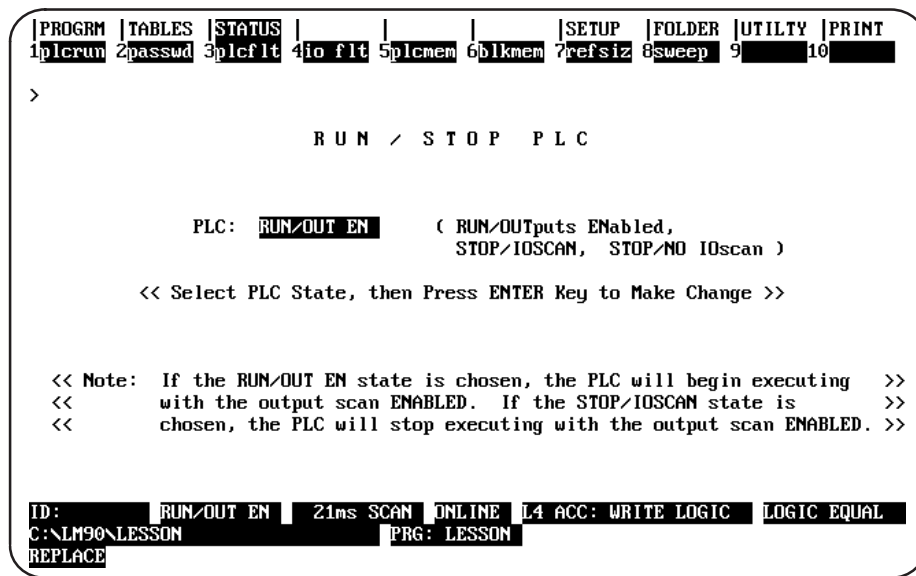
The screen displayed above identifies the CPU model which is attached and its software revision.

The **F9** and **F10** softkeys are now blank when the PLC Control and Status menu is displayed. **F9**, which had previously been designated as the OEM softkey, is not present unless the active function is OEM protection. When the function is OEM protection, F9 will become the OEM softkey.

F10, which had previously been designated as the Clear softkey, is not used unless the active function is either the PLC fault table or the I/O fault table. When either fault table function is active, **F10** will become the **Zoom** softkey and **F9** will become the **Clear** softkey.

## Run/Stop the PLC

To start or stop program execution in the PLC, and to determine the state of outputs if the PLC is started, press **PLC Run (F1)** from the PLC Control and Status menu or from another PLC functions screen. Communications between the programmer and the PLC must already be established.



If the programmer is in **OFFLINE** mode, default values are displayed on the screen. Press **ALT-M** to change the programmer operating mode to **ONLINE** or **MONITOR** mode. (For the Workmaster and CIMSTAR I computers, use the keyswitch. Refer to appendix C, "Programmer Environment Setup," for instructions on using the keyswitch.) **ALT-R** can also be used to toggle the PLC mode between **RUN** and the configured **STOP** mode.

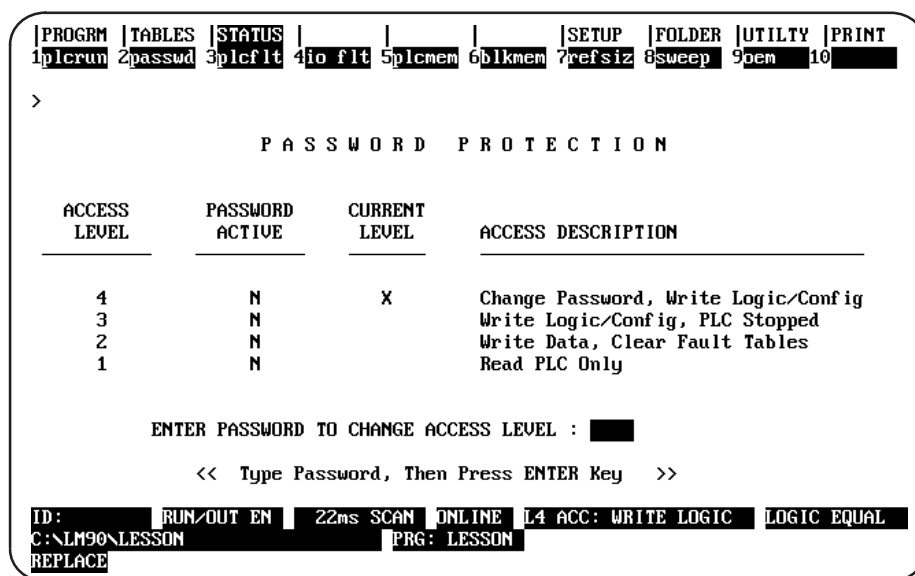
The **PLC** field on the screen shown above indicates whether the attached PLC is to be started or stopped. When the programming software is initially executed, this field is initialized to **STOP**. Use the **Tab** key to select one of the modes listed below; then, press the **Enter** key.

Mode	Description
Run/Outputs Enabled	The PLC is running the logic program with outputs enabled.
Stop/IOScan	The PLC is stopped, not executing the logic program, but is scanning I/O.
Stop/No IOScan	The PLC is stopped, not executing the logic program, no I/O scan.

## PLC Password Protection

PLC password protection can be used to restrict access to selected PLC functions. After passwords have been set up, a protected function cannot be used unless the proper password has been entered. Password protection is not intended to restrict access to the programming or configuration software. If PLC communications are suspended, protection level will automatically return to the highest unprotected level.

To display or change the current protection level of the PLC, the programmer must be in **ONLINE** or **MONITOR** mode and communicating with the intended PLC. Select **Password (F2)** from the PLC Control and Status menu or from another PLC functions screen.



If the screen appears without entries, the programmer is in **OFFLINE** mode. Press **ALT-M** to change the programmer operating mode to **ONLINE** or **MONITOR** mode. (For the Workmaster and CIMSTAR I computers, use the keyswitch.)

Field	Description
Access Level	Each access level includes all the privileges of lower levels: <ul style="list-style-type: none"> <li>• <b>Level 4:</b> Write to all configuration or logic. Configuration may only be written in <b>STOP</b> mode; logic may be written in <b>STOP</b> or <b>RUN</b> mode. Display, set, or delete passwords for any level. (This is the default if no passwords are assigned.)</li> <li>• <b>Level 3:</b> Write to any configuration or logic, including word-for-word changes, the addition/deletion of program logic, and the overriding of discrete I/O.</li> <li>• <b>Level 2:</b> Write to any data memory, except overriding discrete I/O. The PLC can be started or stopped. PLC and I/O fault tables can be cleared.</li> <li>• <b>Level 1:</b> Read any PLC data except passwords; no PLC memory may be changed.</li> </ul>
Password Active	A <b>Y</b> (Yes) displayed in this column indicates that a password has been assigned to this access level in the PLC.
Current Level	An <b>X</b> displayed in this column indicates the current protection level of the PLC.
Access Description	Access allowed at each level. Privileges accumulate as the level increases; at any given level, all privileges at lower levels are permitted.

### Changing the Privilege Level

To access any level, the programmer must be in **ONLINE** or **MONITOR** mode and communicating with the PLC.

1. Move the cursor to the line at the bottom of the screen which prompts you to enter a password to change the access level.
2. Type a password, consisting of up to 4 ASCII hexadecimal digits (0 – 9, A – F), and press the **Enter** key. Characters are not displayed as you type them.

A valid password enables you to access the protection level for that password, and also all protection levels below that level.

## Creating, Changing, or Removing Passwords

Passwords are created in the configuration software, using the status function as described below. They are then transmitted to the PLC, where they are stored.

### Caution

**The PLC may not save passwords through a loss of power, if the CPU battery is not attached. After restarting the PLC, passwords must be sent to the PLC again.**

The use of passwords is an optional feature; it is not necessary to use any passwords at all. You can also use passwords to restrict access to some PLC features but not to others. Note, however, that the PLC always defaults to the highest unprotected level, so there should not be any “gaps” in protection levels.

To enter or change passwords, the computer must be in **ONLINE** mode and communicating with the PLC. Entering or changing passwords requires access to the highest level. If no passwords have been set up for the system, this level is automatically available. Once passwords have been entered, they can **only** be changed by performing one of the following tasks:

- Enter the correct password to access the highest-level privileges.

**OR**

- In the configuration software, by place the master diskette No. 1 in the system disk drive of the computer and press the **ALT** and **O** keys. Because this allows passwords to be overridden or changed without entering the correct password, it is important to keep the original software master diskettes in a secure location.

To create, change, or remove passwords:

1. Access the Password Protection screen in the configuration software by pressing **Password (F2)** from the PLC Control and Status menu. A column titled “Password” appears on this screen; this column is not displayed in the programming software. Use this column to enter new passwords.
2. If there is not already a password set for the highest level, begin by creating one. This will protect the passwords you enter against unauthorized changes.
3. In the password column, locate the cursor at a level you want to protect or unprotect. Then:
  - A. To create or change a password, enter a password consisting of up to 4 ASCII hexadecimal digits (0 - 9, A - F).
  - B. To remove an existing password previously stored to the PLC, enter four blank characters or press **ALT-C** to clear the field.
  - C. Press the **Enter** key to validate your entry.
4. To save each new password to the PLC, press the **Enter** key again. Respond to the prompt that appears by pressing the **Y** (Yes) key.

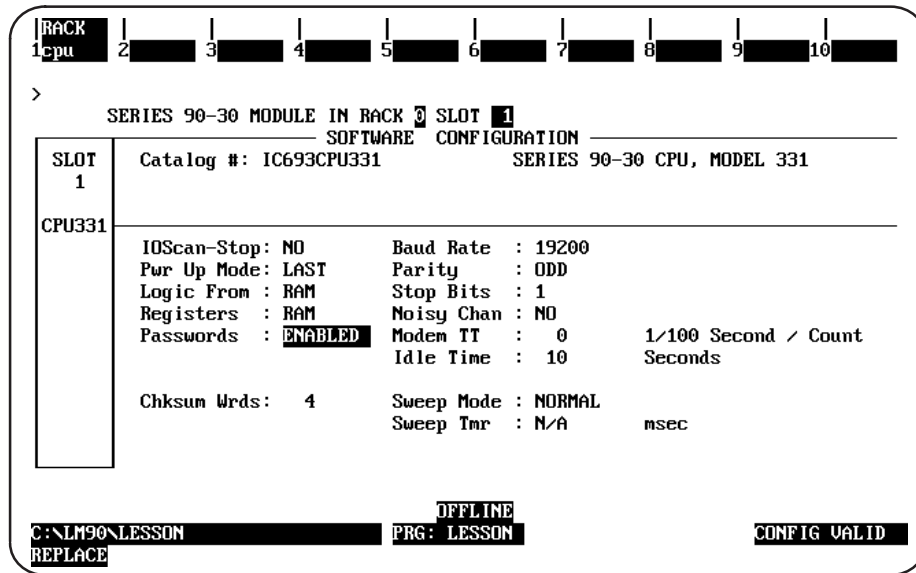
If you want to quit without saving any passwords to the PLC, press the **N** (No) key.

5. To exit, use the **Escape** key or any main menu function key.

## Enabling/Disabling Passwords

The ability to use passwords may be enabled or disabled in the configuration software. You may want to disable this feature to prevent someone from setting passwords in the CPU.

A sample CPU module detail screen displaying the *Passwords* field that would be used to enable or disable the password feature is shown below. To access this screen, press **Zoom (F10)** with the cursor positioned on the CPU-configured slot on the Rack Configuration screen.



The values for the *Password* field are **ENABLED** (default selection) or **DISABLED**. Entering **ENABLED** will allow the password feature to be used; entering **DISABLED** will prevent the password feature from being used.

### Caution

In order to re-enable passwords once they have been disabled, PLC memory must be cleared with an HHP. The HHP needs to be connected. Then power off the PLC. Then hold both the <CLR> and <M/T> keys down while powering the PLC back up. If you do not have an HHP, call the GE Fanuc Technical Service Hotline (1-800-828-5747) for assistance.

Please refer to chapter 10, section 3, "Configuring the CPU Module," for more information on configuring this module.

## OEM Protection

The OEM protection feature provides a higher level of security than password protection. It may be used to further restrict access to program logic and configuration parameters.

To display the OEM Protection screen, press **OEM (F9)** from the Password Protection screen.

```

PROGRAM | TABLES | STATUS |          |          | SETUP | FOLDER | UTILITY | PRINT
1|plcrun 2|passwd 3|plcfl 4|io flt 5|plcmem 6|blkmem 7|refsiz 8|sweep 9|          10|
>
O E M   P R O T E C T I O N

OEM PROTECTION STATE = UNLOCKED

ENTER OEM KEY TO CHANGE PROTECTION STATE : █

<< Type OEM Key, Then Press ENTER Key >>

ID:          RUN/OUT EN  22ms SCAN  ONLINE  L4 ACC: WRITE LOGIC  LOGIC EQUAL
C:\LM90\LESSON          PRG: LESSON
REPLACE

```

OEM protection is enabled and disabled by entering the OEM key. The OEM key is a 1 to 4 hexadecimal digit password. If the OEM key has never been set, it will equal the NULL (blank) string. Definition of the OEM key may only be performed in the configuration software; however, OEM protection may be locked or unlocked in the programming software.

The OEM protection state is toggled between unlocked and locked each time the OEM key is correctly entered. The OEM protection state cannot be toggled to locked until the OEM key is set to something other than the NULL (blank) string. Once the OEM protection is locked, it will remain locked until the OEM key is correctly entered to toggle the protection state.

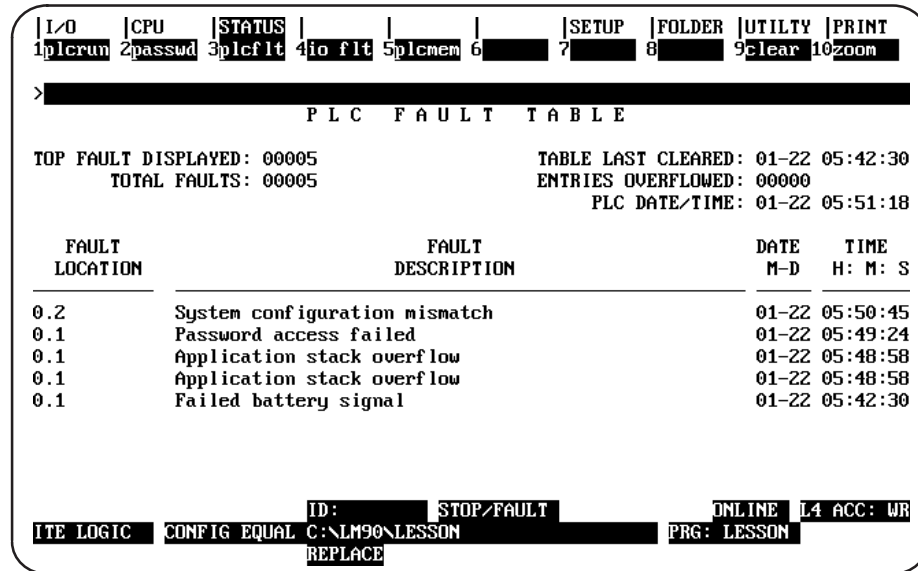
The OEM protection state is retentive across a power cycle. If the OEM protection is locked in the PLC and power is lost, the OEM protection state will remain locked when power is restored to the PLC if the CPU battery is attached.

### Note

Beginning with version 3.0 of Logicmaster, Subroutine View and Edit Lock features have provided a more flexible method for limiting access to OEM logic than the OEM Password feature which locks both configuration and all logic. Moving logic to a block and locking only that block allows the end user to access other parts of the PLC. Refer to page 4-60 for a description of each of the Lock options and page 4-61 for directions on how to use the Lock feature.

## PLC Fault Table

The PLC Fault Table screen lists PLC faults such as password violations, PLC/configuration mismatches, parity errors, and communications errors. For example:



To display the PLC Fault Table screen, press **PLC Fault (F3)** from the PLC Control and Status menu or from another PLC functions screen. The programmer may be in any operating mode. However, if the programmer is in **OFFLINE** mode, no faults are displayed. In **ONLINE** or **MONITOR** mode, PLC fault data is displayed. In **ONLINE** mode, faults can be cleared (this may be password protected).

Field	Description
Top Fault Displayed	The index of the PLC fault currently at the top of the fault display is shown on the first line of this screen.
Total Faults	The total number of faults since the table was last cleared.
Table Last Cleared	The date and time faults were last cleared from the fault table. This information is maintained by the PLC.
Entries Overflowed	The number of entries lost because the fault table has overflowed since it was cleared. The PLC fault table can contain up to 16 faults.
PLC Time/Date	The current date and time. This is also maintained by the PLC.

### Note

Because the Model 323, 321, 311, 313, 211, and Micro CPUs do not support the time-of-day clock, entries for date and time in the fault tables are displayed as 00-00 00:00:00.



## Fault Table Entries

For each fault, the display shows the date and time the fault occurred, and the following information about the fault:

Field	Description
Fault Location	The location of the fault (rack/slot address). For example, 3.2 refers to rack 3, slot 2.
Fault Description	The fault that has occurred: <ul style="list-style-type: none"> <li>Loss of, or missing, I/O module.</li> <li>Loss of, or missing, option module.</li> <li>Addition of, or extra, rack.</li> <li>Addition of, or extra, I/O module.</li> <li>System configuration mismatch.</li> <li>PLC CPU hardware failure.</li> <li>Non-fatal module hardware failure.</li> <li>Option module software failure.</li> <li>Program checksum failure.</li> <li>Low battery signal.</li> <li>Constant sweep time exceeded.</li> <li>PLC system fault table full.</li> <li>User application fault.</li> <li>No user program present.</li> <li>Corrupted user program on power-up.</li> <li>Password access failure.</li> <li>PLC CPU system software failure.</li> <li>Communications failure during store.</li> </ul>

Different faults will result in different reactions; e.g., some faults will stop the CPU; others will not. For more information about each type of fault, refer to the *Series 90-30/20/Micro Programmable Controllers Reference Manual*, GFK-0467.

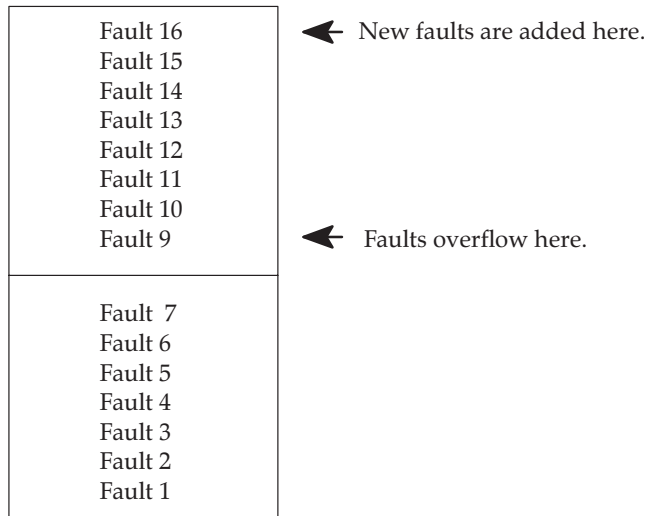
For diagnostic faults, the CPU sets fault references. For fatal faults, the CPU sets fault references and places the CPU in **STOP** mode. Informational faults do neither.

If there are more faults than will fit on one screen, you can display them using the Home, End, Page Up and Page Down, and cursor keys. Refer to the *Series 90-30/20/Micro Programmable Controllers Reference Manual*, GFK-0467, for more fault information.

### Number of Faults in the PLC Fault Table

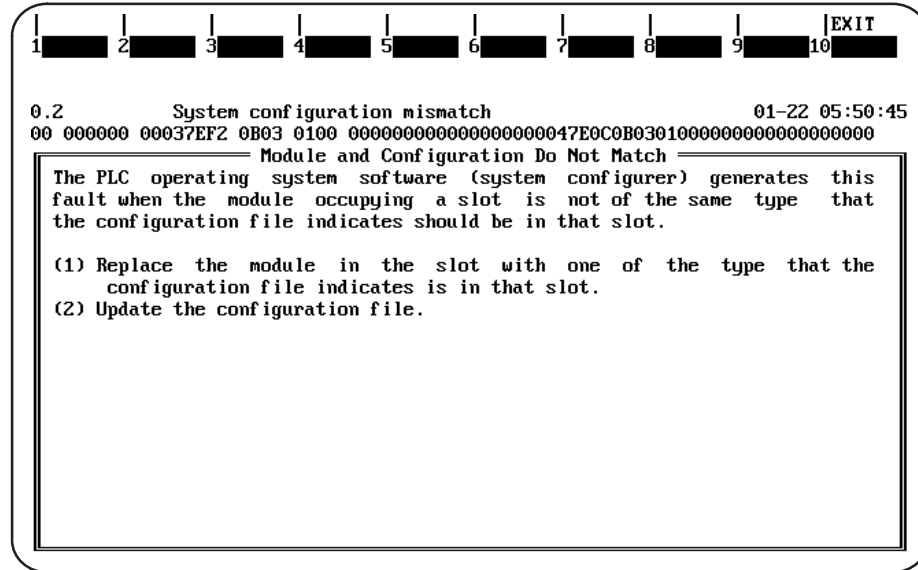
The PLC fault table can contain up to 16 faults. Additional faults cause the table to overflow, and faults are lost. The system reference SY\_FULL (%S0009) is set to indicate that the fault table is full.

As faults occur, the first 8 faults are logged into the table and remain there until the table is cleared. None of these eight faults is dropped if the table overflows. For faults 9 through 16, however, the fault table operates as a First-In-First-Out queue. When fault 17 occurs, fault 9 is dropped from the table. Clearing the fault table removes all the fault entries.



### Zooming into the PLC Fault Table

The **Zoom (F10)** softkey enables you to obtain additional information pertaining to each fault listed in the PLC fault table. By moving the cursor to a particular fault and pressing **Zoom (F10)**, information about the error code, default action, description of the error, and appropriate corrective action is displayed on a screen similar to this one for a low battery signal.



### Note

A hexadecimal dump of the fault can be displayed by pressing **CTRL-F** from this screen. For more information on fault explanations and correction and **CTRL-F**, please refer to the *Series 90-30/20/Micro Programmable Controllers Reference Manual*, GFK-0467.

All softkeys are disabled on this screen, except for **Exit (Shift-F10)**. The **Abort** key (**ALT-A**) and the **Escape** key may also be used to exit from this screen and return to the PLC fault table.

### Clearing the Fault Table

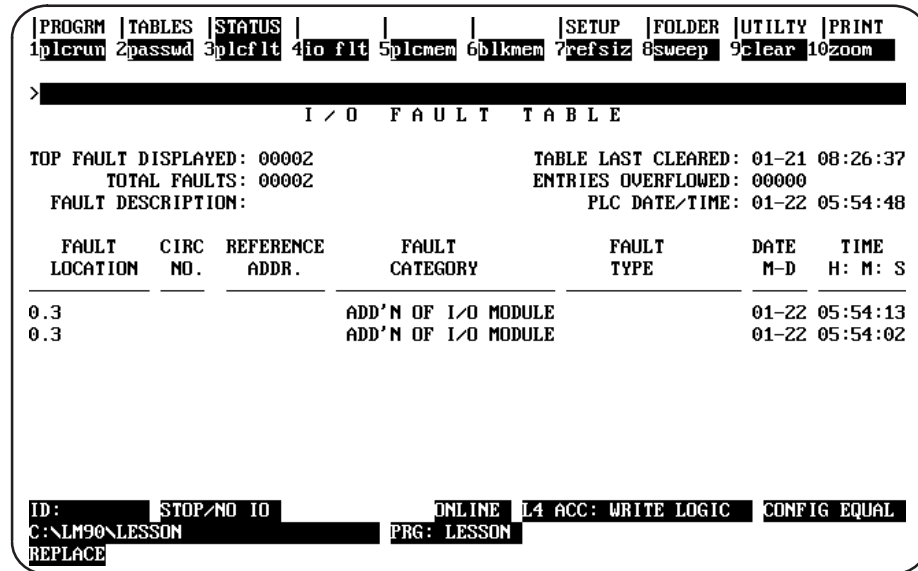
Pressing **Clear (F9)** in **ONLINE** mode clears all faults from the fault table. Faults can also be cleared from the program logic using the **SVCREQ** function, as described in the *Series 90-30/20/Micro Programmable Controllers Reference Manual*, GFK-0467. If passwords have been enabled in the PLC, you must be at level 2 or higher in order to clear the fault table.

If a printer is connected to the programmer, you may want to print a copy of the fault table screen by pressing the **Print Screen** key before clearing the fault table.

Clearing the fault table, of course, does not clear fault conditions in the system. If the condition that caused a fault still exists, the fault may be reported again after storing the configuration, cycling power to the PLC, or during a **STOP-TO-RUN** transition.

## I/O Fault Table

The I/O Fault Table screen lists I/O faults such as circuit faults, address conflicts, forced circuits, and I/O bus faults. For example:



To display the I/O Fault Table screen, press **I/O Fault (F4)** from the PLC Control and Status menu or from another PLC functions screen. The programmer may be in any operating mode. However, if the programmer is in **OFFLINE** mode, no faults are displayed. In **ONLINE** or **MONITOR** mode, PLC fault data is displayed. In **ONLINE** mode, faults can be cleared (this feature may be password protected).

Field	Description
Top Fault Displayed	The index of the I/O fault currently at the top of the fault display is shown on the first line of this screen.
Total Faults	The total number of faults since the table was last cleared.
Fault Description	An explanation of the fault that is currently highlighted in the I/O fault table.
Table Last Cleared	The date and time faults were last cleared from the fault table. This information is maintained by the PLC.
Entries Overflowed	The number of entries lost because the fault table has overflowed since it was cleared. The I/O fault table can contain up to 32 faults.
PLC Time/Date	The current date and time. This is also maintained by the PLC.

### Note

Because the Model 311, 313, 211 and Micro CPUs do not support the time-of-day clock, entries for date and time in the fault tables are displayed as 00-00 00:00:00.

## Fault Table Entries

For each fault, the display shows the date and time the fault occurred, and the following information about the fault:

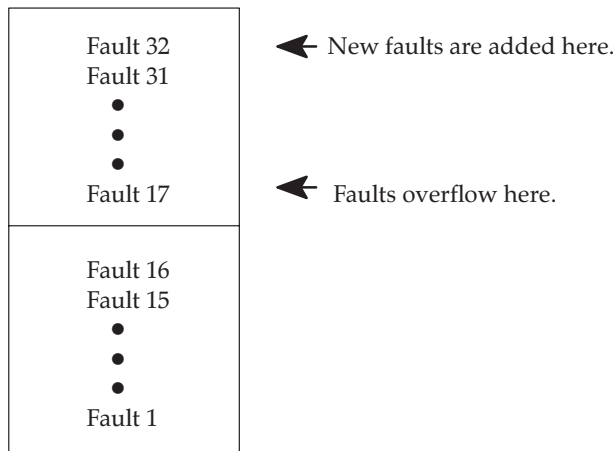
Field	Description
Fault Location	The location of the fault (rack/slot address). For example, 3.2 refers to rack 3, slot 2.
Circuit Number	The relative position of a point within its module. The value may be from 0 to 1023.
Reference Address	The I/O reference address where the fault was detected. The address consists of a two or three character identifier (%I, %Q, %IQ, %AI, %AQ) specifying the memory type and a five-digit offset within the memory type.
Fault Category	The general type of fault that has occurred. For more information, refer to GFK-0467.  For diagnostic faults, the CPU sets fault references. For fatal faults, the CPU sets fault references and places the CPU in <b>STOP</b> mode.
Fault Type	This further explains the fault category. For more information, refer to GFK-0467.

If there are more faults than will fit on one screen, you can display them using the Home, End, Page Up and Page Down, and cursor keys.

## Number of Faults in the I/O Fault Table

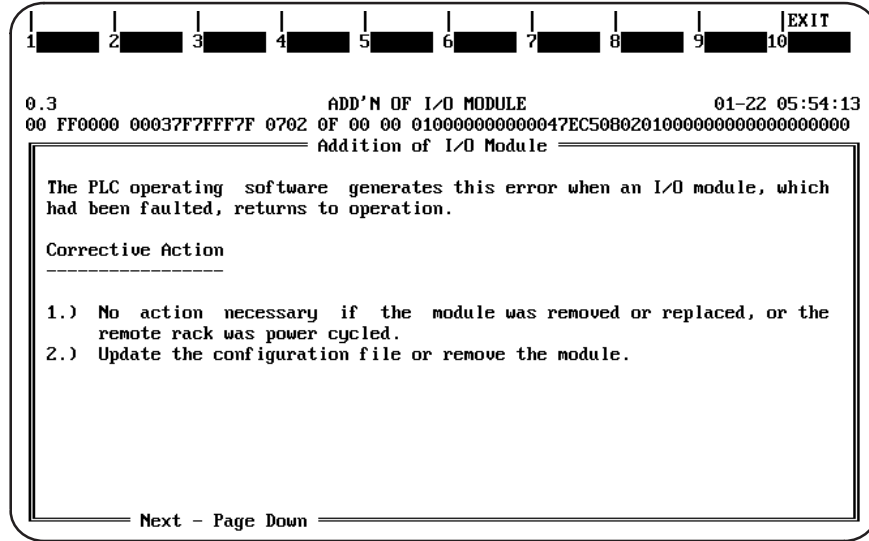
The I/O fault table can contain up to 32 faults. Additional faults cause the table to overflow, and faults are lost. The system reference IO\_FULL (%S0010) is set to indicate that the fault table is full.

As faults occur, the first 16 faults are logged into the table and remain there until the table is cleared. None of these 16 faults is dropped if the table overflows. For faults 17 through 32, however, the fault table operates as a First-In-First-Out queue. When fault 33 occurs, fault 17 is dropped from the table. Clearing the fault table removes all the fault entries.



### Zooming into the I/O Fault Table

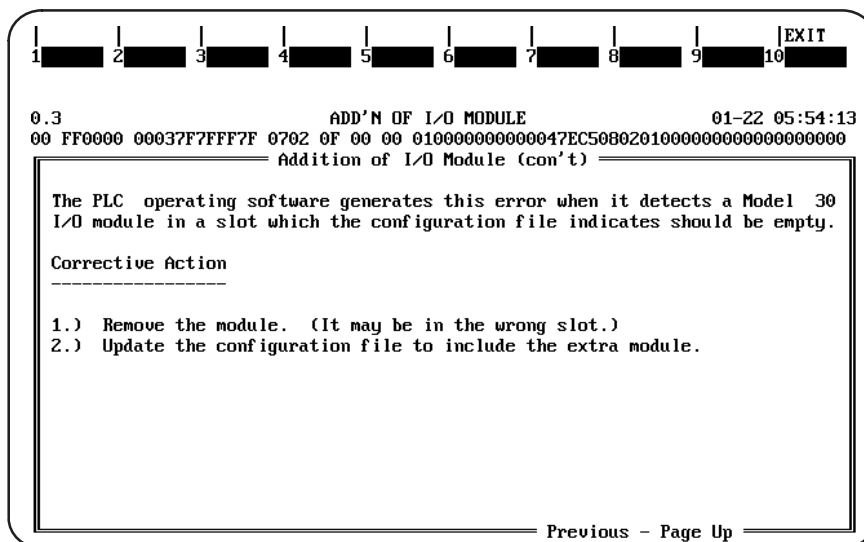
The **Zoom (F10)** softkey enables you to obtain additional information pertaining to each fault listed in the I/O fault table. By moving the cursor to a particular fault and pressing **Zoom (F10)**, information about the error code, default action, description of the error, and appropriate corrective action is displayed on a screen similar to this one for an Addition of I/O Module error.



### Note

A hexadecimal dump of the fault can be displayed by pressing **CTRL-F** from this screen. For more information on fault explanations and correction and **CTRL-F**, please refer to the *Series 90-30/20/Micro Programmable Controllers Reference Manual*, GFK-0467.

Press the **Page Down** key to display additional data pertaining to the fault.



For more information on fault explanations and correction, refer to chapter 3, "Fault Explanation and Correction," in the *Series 90-30/20/Micro Programmable Controllers Reference Manual*, GFK-0467.

All softkeys are disabled on this screen, except for **EXIT (Shift-F10)**. The **Abort** key (**ALT-A**) and the **Escape** key may also be used to exit from this screen and return to the PLC fault table.

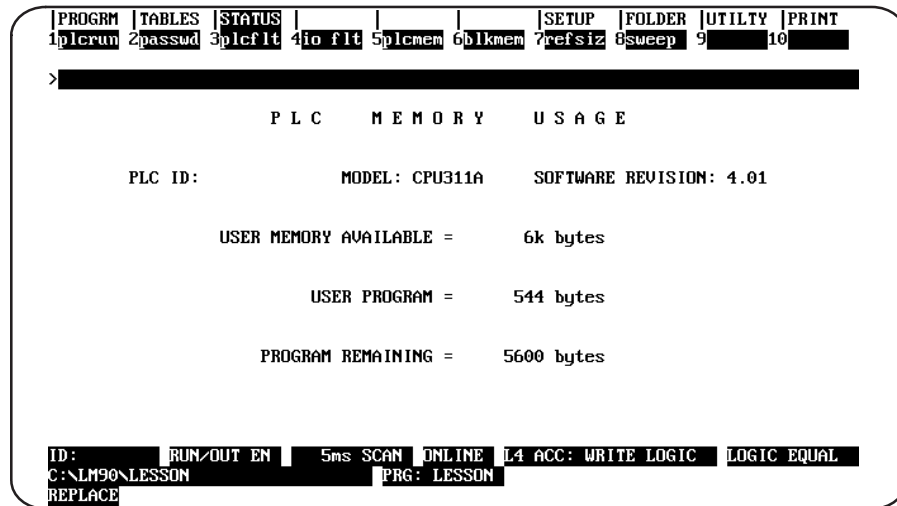
### Clearing the Fault Table

You can clear the fault table in **ONLINE** mode by pressing **Clear (F9)**. (This may be password protected.) Faults can also be cleared from the program logic.

Clearing the fault table removes the faults it contains. Clearing the fault table does not clear fault conditions in the system. If the condition that caused a fault still exists, the fault may be reported again after storing the configuration, cycling power to the PLC, or during a STOP-to-RUN transition.

## PLC Memory Used

The PLC Memory Usage screen shows the amount of PLC memory available and the amount used for application program information. To display this screen, press **PLC Memory (F5)** from the PLC Control and Status menu or from another PLC functions screen. This is an example of the PLC Memory Usage screen:



Field	Description
PLC ID	In <b>ONLINE</b> or <b>MONITOR</b> mode, the name of the PLC being communicated with. It will always be the same as that shown in the status area of the screen. The ID name can be set on the Assign PLC ID screen in the configuration software.
Model	The model number of the attached PLC: <ul style="list-style-type: none"> <li>● CPU 211: Series 90-20 PLC Model 211</li> <li>● UDR1/2: Series 90 Micro PLC Model UDR001 or UDR002</li> <li>● UAA003: Series 90 Micro PLC Model UAA003</li> <li>● UDR005: Series 90 Micro PLC Model UDR005</li> <li>● UAL006: Series 90 Micro PLC Model UAL006</li> <li>● UAA007: Series 90 Micro PLC Model UAA007</li> <li>● CPU 311: Series 90-30 PLC Model 311 (5-slot) or 321* (10-slot).</li> <li>● CPU 313: Series 90-30 PLC Model 313 (5-slot) or 323* (10-slot).</li> <li>● CPU 331: Series 90-30 PLC Model 331 with a 10-slot rack.</li> <li>● CPU 340: Series 90-30 PLC Model 340</li> <li>● CPU 341: Series 90-30 PLC Model 341</li> <li>● CPU 350: Series 90-30 PLC Model 350</li> <li>● CPU 351: Series 90-30 PLC Model 351</li> <li>● CPU 352: Series 90-30 PLC Model 352</li> <li>● CPU 360: Series 90-30 PLC Model 360</li> <li>● CPU 363: Series 90-30 PLC Model 363</li> <li>● CPU 364: Series 90-30 PLC Model 364</li> </ul>
Software Revision	The revision of PLC software.
User Memory Available	The amount of memory in the PLC that is available for application program information.
User Program	The amount of program memory occupied by the logic program.
Program Remaining	The amount of program memory remaining.

\*PLC model 321 uses the CPU311 in a 10-slot rack. Likewise, PLC model 323 uses the CPU313 in a 10-slot rack. This field displays the actual CPU model, not the PLC model.



## Block Memory Usage

The Block Memory Usage screen shows the amount of memory used for different parts of the application program. For example:

```

|PROGRAM |TABLES |STATUS |   |   |   |SETUP |FOLDER |UTILITY |PRINT
1|plcrun 2|passwd 3|plcflt 4|ioflt 5|plcmem 6|blkmem 7|refsiz 8|sweep 9|   |10|
>
BLOCK MEMORY USAGE
BLOCK NAME: _MAIN
BLOCK CHECKSUM: EQUAL
PROGRAMMER      PLC
-----
PROGRAM (bytes) =      170      170
DECLARATIONS (entries) =      ****      ****
EXPLANATIONS (bytes) = ***** *****

<< To Display Information About Another Block, Type the >>
<< Block Name on the Command Line, Then Press ENTER Key >>

ID:  RUN/OUT EN  3ms SCAN  ONLINE  L4 ACC: WRITE LOGIC  LOGIC EQUAL
C:\LM90\LESSON  PRG: LESSON
REPLACE

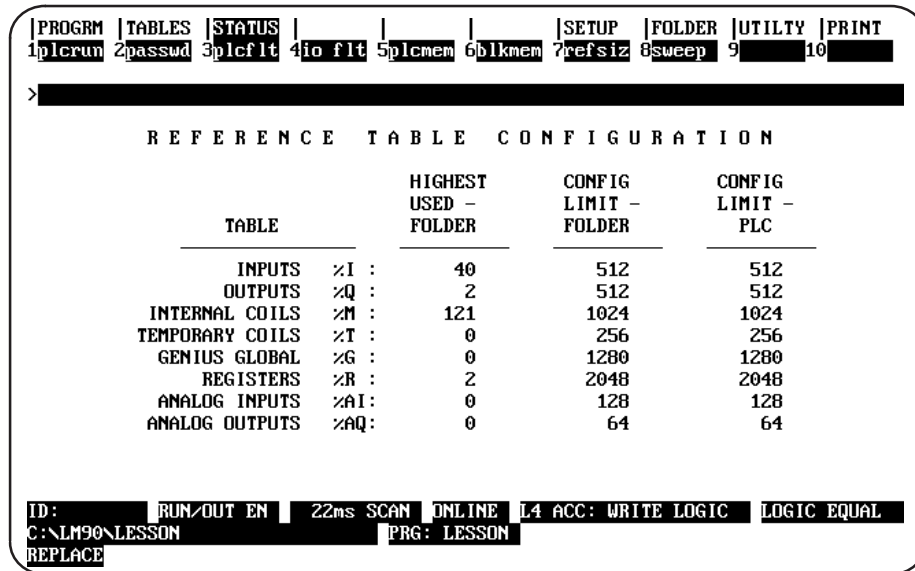
```

To display the Block Memory Usage screen, press **Block Memory (F6)** from the PLC Control and Status menu or from another PLC functions screen. The first block for which information appears is the `_MAIN` block. To display information for another subroutine block, enter its name on the command line and press the **Enter** key.

Field	Description
Block Name	The name of the block for which values are displayed. Values can also be displayed for locked blocks.
Block Checksum	The result of a comparison of the code checksums for the block from the programmer and from the PLC. This is either EQUAL or NOT EQUAL.
Program	The block's logic memory size.
Declarations	The block's symbol table size.
Explanations	The block's explanation text size. This field will always display asterisks since no explanations are currently stored in the PLC.

## Configured Reference Sizes

To see how much memory has been used for program references, press **Reference Size (F7)** from the PLC Control and Status menu or from another PLC functions screen.



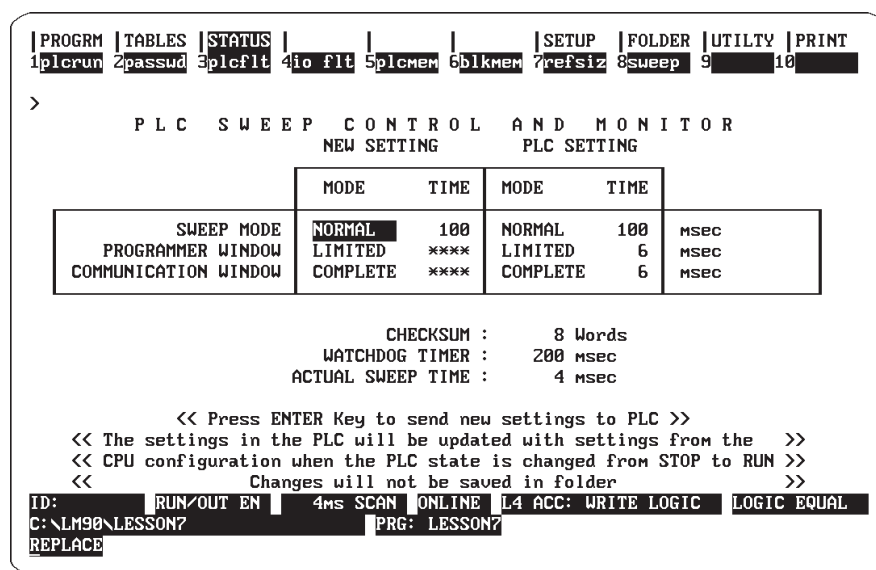
This function is available in all programmer operating modes and also to locked blocks. No values are displayed for the PLC if the programmer is in **OFFLINE** mode. No values are displayed for the configuration file if there is no configuration file in the current folder. Values that do not match are highlighted.

Field	Description
Highest Used – Folder	Lists the highest value used.
Config Limit – Folder	Lists a configuration value for each reference type in the folder.
Config Limit – PLC	Lists a configuration value for each reference type in the PLC.

## PLC CPU Sweep Control

You can set the Sweep while the PLC is in RUN mode, and it only affects the PLC during that RUN; i.e., each time the PLC goes from STOP to RUN mode, the default sweep mode takes effect. No matter what the default sweep setting is, you can change it for the current RUN and have the resulting effects immediately applied. You can use the Active Constant Sweep Mode Parameter to toggle the sweep mode of the PLC without having to change the configured sweep settings, or you can change the amount of time for the constant sweep on this same screen. (This may be very useful when you need to fine tune the sweep time while the PLC is running a program.)

To display or change PLC timers, press **sweep (F8)** from the PLC Control and Status menu or from another PLC functions screen.



In **ONLINE** or **MONITOR** mode, the software shows the current CPU sweep time information.

### Note

The default window mode for the programmer window is “LIMITED”; i.e., if a request takes more than 6 milliseconds to process, it is processed over multiple sweeps, and no one sweep is impacted by more than 6 milliseconds. For CPU models 313, 323, and 331, the sweep impact may be as much as 12 ms. during a RUN-Mode store. If you change the programmer window mode to “COMPLETE,” then the sweep time may be impacted by as much as 50 milliseconds, but the response to external programmer devices will be quicker.

To change the programmer window mode from “LIMITED” (the default) to “COMPLETE,” press the down arrow (cursor) key once to highlight the programmer window field. Then press the **Tab** key to change the first selection.

To change the Active Constant Sweep Mode Parameter, press the **Tab** key to change the first selection (“Mode”) from “NORMAL” to “CNST SWP” (if necessary—if you are just changing the time of the Constant Sweep, it will already say “CNST SWP”). Press the

**Right Cursor** (arrow) key to advance the cursor to the “Time” selection. Then enter the desired number of milliseconds (5 to 200 with a default of 100 for most CPUs—5 to 500 for the 350 and higher CPUs).

### Note

Remember that the number of seconds entered here cannot exceed the value of the Watchdog Timer. Also remember that the change you make through the Active Constant Sweep Mode Parameter will have immediate effect but will be replaced by the default sweep mode each time the PLC goes from STOP to RUN mode.

Press the **Escape** key when done.

### Note

Remember, settings from this screen are only stored in the PLC, not in the folder. These settings may be configured and stored in the folder using the configuration function. For more information, refer to chapter 10, section 3, “Configuring the CPU Module.”

## Active Constant Sweep Mode Setting

No matter what the default sweep setting is, you can change it when the PLC is in **RUN** mode and have the resulting effects immediately applied. You can use the Active Constant Sweep Mode Parameter to toggle the sweep mode of the PLC without having to change the configured sweep settings, or you can change the amount of time for the constant sweep on this same screen. (This may be very useful when you need to fine tune the sweep time while the PLC is running a program.)

To change the constant sweep time (unless it is password-protected) or the mode, Logicmaster must be in **ONLINE** mode. To set the Active Constant Sweep Mode Parameter, after entering the PLC Sweep Control screen, press the **Tab** key to change the first selection “Sweep Mode” from “NORMAL” to “CNST SWP” (if necessary—if you are just changing the time of the Constant Sweep, it will already say “CNST SWP”). Press the Right Cursor (arrow) key to advance the cursor to the “Time” selection. Enter the desired number of milliseconds (5 to 200 with a default of 100).

### Note

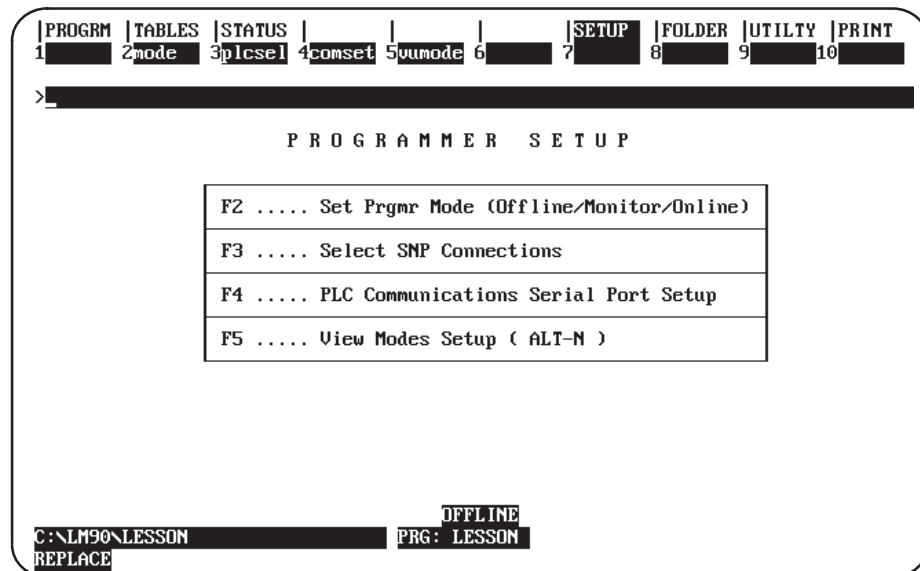
Remember that the number of seconds entered here cannot exceed the value of the Watchdog Timer. Also remember that the change you make through the Active Constant Sweep Mode Parameter will have immediate effect but will be replaced by the default sweep mode each time the PLC goes from STOP to RUN mode.

For an explanation of the PLC sweep, refer to chapter 2, “System Operation,” in the *Series 90-30/20/Micro Programmable Controllers Reference Manual*, GFK-0467.

# Chapter 6

## Programmer Setup

This chapter explains how to set up the programmer for communication with the PLC and how to select the programmer operating mode. To use the programmer setup features, press **SETUP (F7)** from within the programming software.



### Note

On computers with a keyswitch, such as the Workmaster or CIMSTAR I industrial computer, the **Mode Selection (F2)** key does not appear on the Programmer Setup screen, unless the keyswitch was disabled during programmer setup.

## Section 1: Programmer Operating Mode

During configuration and programming, the computer is always in one of three operating modes:

Mode	Description
OFFLINE	<b>OFFLINE</b> mode is used for program development. The programmer does not communicate with the PLC in <b>OFFLINE</b> mode; power flow display and reference values are not updated.
ONLINE	<b>ONLINE</b> mode provides full CPU communications, allowing data to be both read and written.
MONITOR	<b>MONITOR</b> mode allows programs to be examined and real-time status to be displayed, but no changes of logic, reference values or I/O overrides are allowed. For Workmaster and CIMSTAR I computers, <b>MONITOR</b> mode is the only mode which allows the key to be removed from the keyswitch.

Many functions require the computer to be in either **MONITOR** or **ONLINE** mode. To use either of these operating modes, communications must have been established between the computer and the CPU.

### Mode Selection

Both the Workmaster and CIMSTAR I computers have a keyswitch which must be used to select the programmer operating mode. The presence of this keyswitch must be indicated in the programmer setup file, as described in appendix C.

For computers without a keyswitch (e.g., the Workmaster II computer), the programmer mode can be selected by pressing **Mode (F2)** from the Programmer Setup menu or the Serial Port Setup screen. The programmer mode can also be changed by pressing **ALT-M** from another programming function.

```

|PROGRAM |TABLES |STATUS | | | |SETUP |FOLDER |UTILITY |PRINT
|1ports  |2mode  |3plcse| 4comset| 5uomode| 6 | 7 | 8 | 9 |10
>

      S E T   P R O G R A M M E R   M O D E

      M O D E =  ONLINE  (OFFLINE, MONITOR, ONLINE)

      << Press ENTER Key to Invoke Mode Change >>

ID:      RUN/OUT EN  22ms SCAN  ONLINE  L4 ACC: WRITE LOGIC  LOGIC EQUAL
C:\LM90\LESSON    PRG: LESSON
REPLACE

```

To change the programmer operating mode, use the Tab key to toggle choices at the cursor position or type in a new value. Press the **Enter** key.

## Section 2: Selecting SNP Connections

The “Select SNP Connections” screen provides a log of SNP IDs defined by the user. Logicmaster 90-30/20/Micro software does not create or use the information in these fields, but it may be helpful to refer to this listing when selecting a specific SNP ID.

```

PORTS |MODE |PLCSEL |COMSET |UUMODE |          |          |          |
1show p 2show f 3default 4          5          6setup 7save 8          9          10
>
                S E L E C T   S N P   C O N N E C T I O N S
FILE NAME      C:\LM90\%PLC030.PSU
SELECTED SNP ID
PORT CONNECTION  DIRECT          (DIRECT, MULTIDROP)
                S N P   I D S   ( I n f o r m a t i o n a l   U s e   O n l y )
-----
1: A0001      2: A0002      3: A0003      4: B0001      5: B0002
6: B0003      7:            8:            9:            10:
11:          12:          13:          14:          15:
16:          17:          18:          19:          20:
21:          22:          23:          24:          25:
26:          27:          28:          29:          30:
31:          32:
ID:          RUN/OUT EN  3ms SCAN  ONLINE  L4 ACC: WRITE LOGIC  LOGIC EQUAL
C:\LM90\LESSON  PRG: LESSON
REPLACE

```

Field	Description
File Name	The name of a disk file containing the PLC selection information. If no file name is entered, the default name %PLC030.PSU is used. This field allows you to save the setup parameters to a file other than the default file name. You can then recall the setup from this saved file and perform the setup within that screen.
Selected SNP ID	The name used to identify the PLC to be communicated with. A null string may be specified as the SNP ID by leaving the field blank. This allows communication to any PLC using the currently active serial port setup. The null string SNP ID should be used only for direct (point-to-point) PLC connections, since all PLCs will respond in a multidrop configuration. Prior to Release 6.6, the maximum length was six characters. Beginning with Release 6.6 CPUs, the SNP ID may be up to seven characters long.
Port Connection	The serial link configuration. The serial port can be set up for direct (point-to-point) communication with one PLC, or multidrop configuration with several PLCs connected to the WSI Board serial port. Direct connection should not be used with a multidrop configuration. In a multidrop configuration, all PLCs must be set up with the same parity and number of stop bits. They can, however, have a different baud rate.
SNP ID Numbers	These fields allow you to enter the SNP IDs which have already been defined.

## Section 3: PLC Communications Serial Port Setup

In the standard serial communications version of the software package, the PLC Communications Serial Port Setup screen takes the place of the Programmer WSI Serial Port Setup screen used in the WSI-based version. If you are using the WSI Board for serial communications, skip this information and go to the information on WSI serial port setup which follows.

The Standard Serial COM Port version of Logicismaster 90-30/20/Micro software provides logic programming and configuration for the Series 90-30, 90-20, or Micro PLC, using the COM1, COM2, COM3, or COM4 serial port instead of a Work Station Interface (WSI) board. This version is available as a separate programming package. It also supports offline programming.

### Standard Serial COM Port

The contents of the Standard Serial COM Port version of Logicismaster 90-30/20/Micro programming package include:

- Two 5.25-inch High Density diskettes, three 3.5-inch Double Density diskettes.
- *Logicismaster 90-30/20/Micro Programming Software User's Manual*, GFK-0466.
- *Series 90-30/20/Micro Programmable Controller Reference Manual*, GFK-0467.
- *Important Product Information for the Standard Serial COM Port Version of Logicismaster 90-30/20/Micro Software*, GFK-0683.
- If the programming package is ordered as IC641SWC306 (or IC641SWC307 depending on the type of diskettes used), a Miniconverter Kit, IC690ACC901, is included. The kit consists of an RS-422 (SNP) to RS-232 mini-converter, a 6-foot (2 meter) serial extension cable, and a 9-pin to 25-pin converter plug assembly.

The Standard Serial COM Port version of Logicismaster 90-30/20/Micro software uses a software communications driver to perform the functions of the WSI board. This communications driver, when added to the Logicismaster 90-30/20/Micro software, may not fit in the available memory of most computers *unless* MS-DOS Version 5.0 ( or higher) or a commercially available memory manager is used.

The following steps must be performed in order to establish communications with the Series 90-30/20/Micro PLC:

1. Configure the computer memory for the communications driver. Logicismaster 90-30/20/Micro software uses a communications driver that may be loaded into different memory areas. Please refer to "Configuring Memory for the Communications Driver" on page 6-6.
2. If both versions of the software package (WSI or Standard Serial COM Port) are installed on your computer, you must select which version to run from the Start-up Menu's Setup File Editor. (For directions on how to set up your startup menu to handle both versions, refer to page 3-16 and appendix C, "Programmer Environment Setup.")
3. Connect the cable between the computer's serial port (COM1, COM2, COM3, or COM4), the RS-422/RS-485 Converter, and the Series 90-30/20/Micro PLC serial port.



### Note

If you have the Miniconverter Kit, IC690ACC901, please refer to GFK-0682 for more information. If you wish to make your own cable, refer to the *Series 90-30/20/Micro Programmable Controller Installation Manual*, GFK-0356, for more information.

4. Set up the computer's serial port to be used for PLC communications. It should match the characteristics of the Series 90-30/20/Micro PLC serial port. Please refer to "Setting up the Computer's Serial Port," on page 6-10.

## Requirements

To run the Standard Serial COM Port version of Logicmaster 90-30/20/Micro software, you will need:

Requirement	Description
MS-DOS	MS-DOS Version 5.0 or higher.
Memory Manager	MS-DOS Version 5.0 (or higher) or a memory manager that complies with one of the following specifications: <ul style="list-style-type: none"> <li>• The LIM/EMS (Lotus Intel Microsoft/Expanded Memory Specification) Version 3.2 or later.</li> <li>• The XMS (eXtended Memory Specification) Version 2.0 or later.</li> <li>• A list of some compliant memory managers is included at the end of this section.</li> </ul>
Computer	A Workmaster II, Zenith Mastersport SL Notebook, or other computer with a 80386 or higher processor and at least one available COM port.
Hard Disk	At least 4 Megabytes of hard disk space.
RAM	At least 564 KB (577,536 bytes) of available DOS conventional memory; or 520 KB of available DOS conventional memory <i>and</i> 42 KB of High Memory Area, Upper Memory Block, or Expanded Memory, for the COM port driver.
Miniconverter Kit	A Miniconverter Kit, IC690ACC901, may be ordered. This kit contains the necessary cable and converter to connect from your computer to the Series 90 PLC.

## Port Requirements

The Standard Serial COM Port version requires at least one Standard Serial COM Port in the host computer in order to provide communications with the PLC. This version will support either the COM1, COM2, COM3, or COM4 port in common use by IBM-compatible machines. These ports must use the INS8250 UART chip, or functional equivalent. The COM port I/O addresses and interrupt requests are shown in the following table:

Port	Port Address	IRQ
COM1	3F8	IRQ4
COM2	2F8	IRQ3
COM3	3E8	IRQ4
COM4	2E8	IRQ3

While the Standard Serial COM Port version of Logicmaster 90-30/20/Micro software is active, the serial port assigned to PLC communications will be used exclusively for PLC communications. If a serial printer is being used, a second serial port is required; or you may direct print output to disk files, and then use MS-DOS to print the disk files while outside the Logicmaster environment.

The Standard Serial COM Port version of Logicmaster 90-30/20/Micro software monitors Clear-to-Send (CTS) to determine a cable disconnect. Therefore, Logicmaster 90-30/20/Micro software must use a connection with the Series 90-30/20/Micro PLC that will keep its Request-to-Send (RTS) active in **ONLINE** or **MONITOR** mode. The RTS at the PLC must be connected to CTS at the Logicmaster computer serial port. Half-duplex modems cannot be used with Logicmaster 90-30/20/Micro software since they rely on RTS/CTS flow control, which is not supported by the software.

The Standard Serial COM Port version of Logicmaster 90-30/20/Micro software supports multi-drop connections, using RS-422 4-wire hookups or modems. RS-422 multi-drop will work as long as the Series 90 PLC slaves transmit only in response to requests from an attached Logicmaster host.

## Configuring Memory for the Communications Driver

The Standard Serial COM Port version of Logicmaster 90-30/20/Micro software requires a communications driver that may be loaded in different memory areas. The communications driver requires 42 KB of memory. A memory manager may be required, depending on the computer and the amount of installed memory. That is to say, you can load the communications driver (requiring 42 KB of memory) into conventional memory if you have enough available conventional memory to do so; if not, then you will need to ensure that it can be loaded into one of the other areas of memory described on the next page and discussed in detail on page 6-13 and following.

If you are using MS-DOS version 6.0 or later, try the following techniques for increasing the amount of free memory.

First, at a DOS prompt, type the command `MEM /C /P` and press Enter to list all programs and how much Conventional and Upper Memory each program is using as well as showing you the total amount of free memory. Then remove any programs not needed.

MS-DOS also includes the MEMMAKER program to help optimize usage of upper memory. To use it, type **MEMMAKER** at the DOS prompt; select the Custom option using the space bar. Choose **YES** on the questions about using EMS memory. Select **YES** for "Scan Upper Memory Aggressively" and for "Use Monochrome Area" if you have a color monitor. MEMMAKER will reboot your computer several times as it tries and tests CONFIG.SYS and AUTOEXEC.BAT memory options. If MEMMAKER fails to provide enough memory for the serial comm driver, use the `MEM /D /P` command to display which regions of upper memory have free space and use the `/L:region_number` switches in CONFIG.SYS to load programs manually.

## MS-DOS Memory Areas Defined

One of the following memory areas is used to load the communications driver:

- Extended Memory, consisting of the High Memory Area and extended memory blocks.
- MS-DOS System Memory, consisting of Conventional Memory and Upper Memory.
- Expanded Memory.

Type	Description
High Memory Area	<p>65,520 bytes of memory space just above the 1 Megabyte boundary (addresses 0FFFFF to 10FFFF). This area can only be accessed by computers with 80286/80386/80486 microprocessors. PCs with 8086/8088/80188/80186 microprocessors cannot address this memory space.</p> <p>Use of the High Memory Area (HMA) requires either HIMEM.SYS or QEMM-386™ memory manager. The memory manager must comply to the Extended Memory Specification XMS 2.0. For more information on using the High Memory Area on an 80386 (Workmaster II) or higher computer, refer to page 6-14.</p>
Upper Memory Block	<p>Memory space between 640 KB and 1 Megabyte. This memory space is usually used for BIOS or video RAM. On computers with 80386/80486 microprocessors and extended memory installed, extended memory can be mapped into this area with the aid of a memory manager or MS-DOS Version 5.0 or higher. If you want the communications driver loaded into this area, you can configure the memory manager to make this area available. For more information on using the Upper Memory Block on an 80386 (Workmaster II) or higher computer, refer to page 6-15.</p>
Conventional Memory	<p>The first 640 KB of memory addresses. It is sometimes referred to as low memory (RAM) or base memory (RAM). For more information on using conventional memory on an 80386 (Workmaster II) or higher computer, refer to page 6-16. For more information on using video RAM on an 80386 (Workmaster II) or higher computer, refer to page 6-17.</p>
Expanded Memory	<p>The user-installed memory option that can be accessed with an expanded memory manager through a page frame area between the MS-DOS 640 KB and 1024 KB addresses (e.g., GEXMEM). The communications driver may be loaded to the page frame area. This memory area is suitable for computers with Intel 80xx family processors. Some laptop personal computers with 8086/8088 microprocessor and 1 Megabyte of memory come with an expanded memory manager driver (for example, Toshiba 1200). The memory managers must comply to the Lotus/Intel/Microsoft LIM Expanded Memory Specification 3.2 or later. For more information on using Expanded Memory on an 80386 (Workmaster II) or higher computer, refer to page 6-16.</p>

™ QEMM-386 is a trademark of Quarterdeck Office Systems.

## Communications Driver Load Order

To specify the area for the communications driver, you must enter Setup from the main menu. Refer to appendix C, "Programmer Environment Setup," for instructions on how to change the memory allocation for this driver.

The recommended setting for this driver memory allocation is "Automatic," i.e., you let Logicmaster decide how best to set up the memory allocation. If the option selected from the PLC Communications Options screen is automatic, communications driver is loaded into the first available memory area found, according to the following sequence:

1. If the High Memory Area is free and the computer supports the High Memory Area, the driver is installed in the High Memory Area.
2. If the High Memory Area is not available, Logicmaster 90-30/20/Micro software will look for the Upper Memory Block area. If there is enough space for the driver, the driver can be installed in the Upper Memory Block space.
3. Then, if neither the High Memory Area nor the Upper Memory Block is available, Logicmaster 90-30/20/Micro software will examine the MS-DOS conventional memory. If the MS-DOS conventional memory space is large enough for Logicmaster 90-30/20/Micro software and the communications driver to coexist, the driver will be installed in the MS-DOS memory space as a Terminate-and-Stay-Resident (TSR) program.
4. Next, Logicmaster 90-30/20/Micro software will examine the Expanded Memory. If the expanded memory manager is installed, the loader will try to allocate an expanded memory page frame area and install the driver in the page frame area.

### Note

If the COMM driver has to load here in the EMS page frame, EMS memory will be disabled for storing the Variable Table. You will get a warning that Logicmaster needs 1024 KB of EMS memory even though EMS memory is present.

5. If all these steps fail, then the driver is not installed and a message will appear on your screen stating, "PLC communications driver was not installed." In addition, the following prompt will be displayed: "Do you wish to continue? (Y/N)" If you answer: **Y**, you may program offline without PLC communication capability.
6. If you still wish to communicate with the PLC, you must ensure that one of the memory areas listed in steps 1 through 4 above is available. Instructions for memory management are listed on page 6-13 and following of this manual and are also included in your DOS manual. If you have an additional memory manager, such as QEMM, you may wish to refer to that manual as well.

### Note

The PLC Communications Serial Port Setup screen in the setup function of Logicmaster 90-30/20/Micro software will display where the communications driver was loaded. Refer to "Setting Up the Computer's Serial Port" on page 6-10 for more information.

## Running Logicmaster 90 Software

When entering the Standard Serial COM Port version of the software package, the software will attempt to install the correct communications driver in the available memory space. As mentioned previously, if the driver cannot be installed, the message, "PLC communications driver was not installed," will appear on the Initializing screen.

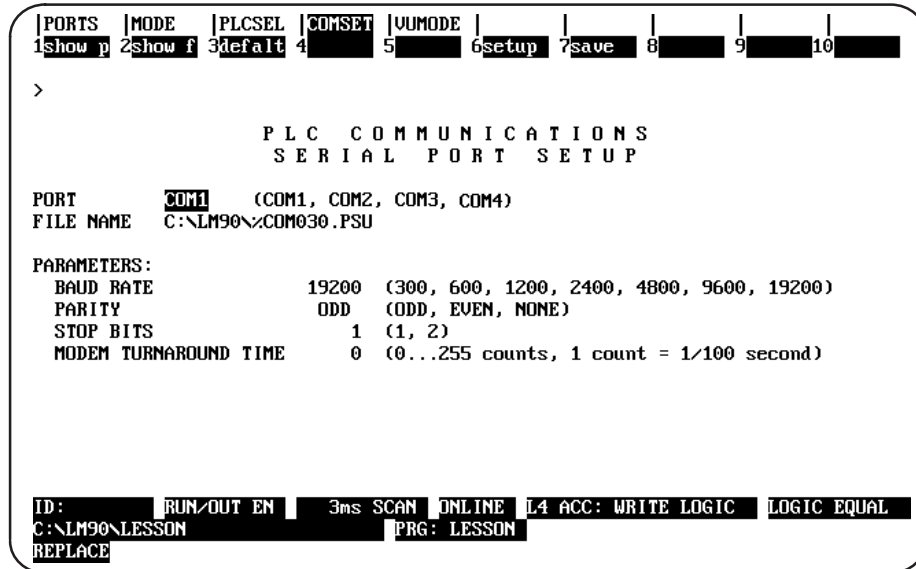
The software will prompt you to continue. If you enter **N** after the prompt, the Logicmaster 90 main menu is displayed. You may then exit back to MS-DOS to correct the memory problem.

If you enter **Y**, the software will proceed without PLC communication capability. (You may program offline.) The main menu screen is then displayed. For example, the programmer software would display this screen:

```
|PROGRAM |TABLES |STATUS | | | |SETUP |FOLDER |UTILITY |PRINT
1program 2tables 3status 4 5 6 7setup 8folder 9utility10print
>
SERIES 90-30 / 90-20 / MICRO PROGRAMMING SOFTWARE
Version 5.00 Direct Serial - COM
F1 ..... Program Display/Edit
F2 ..... Reference Tables
F3 ..... PLC Control and Status
F7 ..... Programmer Mode and Setup
F8 ..... Program Folder Functions
F9 ..... Utility: Load/Store/etc.
F10 ..... Print Functions
<< Press ALT-K at any time to see special key assignments >>
C:\LM90\LESSON OFFLINE
REPLACE PRG: LESSON
```

## Setting Up the Computer's Serial Port

The computer's serial port (COM1, COM2, COM3, or COM4) may be set up from the PLC Communications Serial Port Setup screen in the setup function. Press **COMSET (F4)** (COMSET is the abbreviation for Communications Setup) from the Programmer Setup menu to display the PLC Communications Serial Port Setup screen shown below:



### Note

The PLC CPU defaults to the parameters displayed on this screen (e.g., 19200 baud, odd parity, etc.). These parameters must match the CPU slot configuration that was last stored to the PLC before communications can be established (see Chapter 10, Section 3, "Configuring the CPU Module").

The table on the next page (i.e., the facing page) provides detailed information about the fields in the screen shown above.

### Note

This section of the manual discusses setting up the ports on your PC. For information about configuring the ports on the PLC, refer to Chapter 10, Section 3, "Configuring the CPU Module." Of special significance, the 351 and 352 PLC CPUs have two configurable ports; for information on configuring those ports, refer to page 4-18.

Field	Description
Port Selection	Select the serial COM port (COM1, COM2, COM3, or COM4) to be used for communicating with the PLC.
Setup File Name	The default file name for saving the setup parameters is changed from %WSI030.PSU to %COM030.PSU, to avoid any confusion with the WSI-based version. The %COM030.PSU will contain additional information, such as the COM port selection, which is not present in the %WSI030.PSU file.
Parameters	<p>Press <b>Default (F3)</b> to set the parameters on this screen to their default values. The PLC CPU defaults to these parameters, as displayed on the screen shown above (e.g., 19200 baud, odd parity, 1 stop bit, etc.). These parameters must match the CPU slot configuration that was last stored to the PLC before communications can be established (see chapter 11, section 3, “Configuring the CPU Module”).</p> <p><b>Baud Rate:</b> The transmission rate, in bits per second, of data through the port.</p> <p><b>Parity:</b> Specify whether the parity bit is <b>odd</b> or <b>even</b>; or, if no parity bit is needed, select <b>none</b>. To toggle the selections, place the cursor on the field and press <b>Tab</b> to toggle forward (<b>Shift-Tab</b> to toggle backwards.)</p> <p><b>Stop Bits:</b> All serial communication uses at least one stop bit. Slower devices may use two stop bits.</p> <p><b>Modem Turnaround Time:</b> The time required for the modem to start data transmission after receiving the transmit request. If no modem is used, specify zero. When the PLC is connected through a modem, the value <b>must</b> be one or greater.</p>
PLC Communication Driver Information	<p>Information on the communications driver, including:</p> <ul style="list-style-type: none"> <li>• Version.</li> <li>• Where the driver was installed: High Memory Area (HMA), Expanded Memory (EMS), Conventional MS-DOS Memory (DOS), or Upper Memory Block (UMB) Area.</li> <li>• MS-DOS segment address at which the driver is loaded.</li> <li>• MS-DOS INT vector that it will use.</li> </ul>

## Setting Up the Port

To set up the serial port, enter the appropriate values in the fields on the Serial Port Setup screen. Use the **Up** and **Down Arrow** (cursor mover) keys to move from field to field. In each field, you can use the **Tab** key to cycle through the valid selections, **Shift-Tab** to reverse the direction of the selection display. Once all the values have been entered, press **Setup (F6)**.

## Displaying Port Settings

Press **Show Port (F1)** to display the current settings for the Serial Port Setup screen.

Pressing **F1** while completing the screen will delete all the changes that have been made. The previous settings of the current driver will then be re-displayed on the screen.

### Note

The Standard Serial COM Port version of Logicmaster 90-30/20/Micro software requires either the COM1, COM2, COM3, or COM4 as the communications port for communicating with the Series 90-30/20/Micro PLC. *Do not use the same COM port for printing* that you are using for communicating with the Series 90-30/20/Micro PLC. For additional information, see the “Serial Printer Setup” section at the end of this chapter.

## Saving the Port Setup

Press **Save (F7)** to save a copy of the current serial port settings to a disk file. The data will be saved to a separate file. If you do not specify a file name, a default name will be assigned.

To create a file of settings which is not automatically invoked, enter a file name other than the default name. If no path is given, the current default directory is used. If no file extension is provided, .PSU is used.

After you press **Save (F7)**, the settings are stored to a file and also set up on the port.

## Displaying File Settings

The settings currently stored in a file can be viewed by first entering the file name and then pressing **Show File (F2)**. If no path is specified, the current default directory is used. If no file extension is provided, .PSU is used. If the *File Name* field is blank, the default setup file settings are displayed.

## Port Usage Conflicts

The standard serial communications version will reserve one of the COM ports for PLC communications. To avoid any conflicts, you should follow these guidelines:

Conflict	Description
Setup Files	The setup file looked for during system initialization is %COM030.PSU. This file is used to initialize the PLC communications port. If more than one setup file is found for the designated PLC communications port, the PLC communications file %PLC030L.PSU will take precedence.
User Setup	If the Printer Serial Port Setup screen is displayed and you attempt to perform a setup on the designated PLC communications port, a warning is displayed and the setup is not done.
Print Destinations	The Print utility provides several screens for selecting the print output device. Any attempt to select the designated PLC communications port as the output device is denied.

### Note

Logicmaster only supports standard I/O ports (3F8, 2F8, 3E8 and 2E8) and interrupts (IRQ4 and IRQ3) for COM1 to COM4. The selected I/O port must also be defined in the standard BIOS data area at hex offset 40:0. The BIOS in some computers may not initialize the BIOS data area at power up, specially for COM3, COM4 or serial PCMCIA cards. In this case, Logicmaster will indicate that the COM port is not available. You can check if I/O ports are defined by starting DEBUG at the DOS prompt or in a DOS window under Windows or Windows 95. Enter the command, D 40:0 at the dash (-) prompt and look at the first 8 hex bytes on the first line. If you reverse the bytes (Intel style), you will see the I/O port addresses defined for COM1 to COM4.



## Memory Manager Specifications

Any memory manager used to support the Standard Serial COM Port version of Logicmaster 90-30/20/Micro software must comply with one of the following specifications:

1. LIM/EMS (Lotus Intel Microsoft/Expanded Memory Specification) Version 3.02 or later.
2. XMS (eXtended Memory Specification) Version 2.0 or later.

### Compliant Memory Managers

The following are some drivers which comply with the memory manager specifications listed above:

3. QEMM-386 (Quarterdeck Expanded Memory Manager for 386). This driver can be used on computers with 80386/80486 microprocessors. It is available from:

**Quarterdeck Office Systems**  
150 Pico Boulevard  
Santa Monica, CA 90405  
(213) 392-9701

4. QEMM (Quarterdeck Expanded Memory Manager). This driver can be used on computers with 8086/8088 microprocessors. It can be ordered from same address listed above.
5. HIMEM.SYS (MS-DOS Version 5.0 or higher) from Microsoft Corporation. This Microsoft memory manager complies with XMS specification. It can be used on computers with 80386/80486 microprocessors.
6. EMM386.SYS (MS-DOS Version 5.0 or higher) from Microsoft Corporation. This Microsoft memory manager simulates expanded memory while using extended memory. It can be used on computers with 80386/80486 microprocessors.
7. BlueMAX™ for PS/2 computers and 386MAX™ for 80386 AT-based computers. Both are available from:

**Qualitas, Inc.**  
7101 Wisconsin Avenue, Suite 1386  
Bethesda, MD 20814  
(301) 907-6700

### Restrictions

To avoid conflicts of memory usage between the Standard Serial COM Port version of Logicmaster 90-30/20/Micro software and MS-DOS programs, the following rules must be followed:

1. If the computer is set up to use the High Memory Area memory area, the VDISK program must not be used. Any program using VDISK must remove VDISK before running the Logicmaster 90-30/20/Micro software.
2. Another application cannot use Expanded Memory if the communications driver is loaded into Expanded Memory.

™ BlueMAX and 386MAX are trademarks of Qualitas, Inc.

## Using High Memory Area on an 80386 (Workmaster II) Or Higher Computer

### Physical Extended Memory for High Memory Area Operation

If not already present, you must add and configure physical memory on your computer to be accessed in memory addresses over 1 Megabyte (FFFFF).

### 386 Memory Manager

You must install a 386 high memory manager such as Microsoft HIMEM.SYS (MS-DOS Version 5.0 or higher), Quarterdeck QEMM-386 or 386MAX. The memory manager should be compliant with the Extended Memory Specification (XMS) Version 2.0 or later.

You can install HIMEM to allow use of the High Memory Area.

1. Verify that your CONFIG.SYS file contains a device command for the HIMEM.SYS extended memory manager (or another memory manager that conforms to the XMS specification). For example:

```
device=c:\dos\himem.sys
```

2. If you have MS-DOS 5.0 (or higher) and wish to use the High Memory Area, do not load MS-DOS into the High Memory Area. (The CONFIG.SYS file should not have a DOS=HIGH command.)
3. Restart your computer by pressing **CTRL-ALT-Delete**.

## Using Upper Memory Block on an 80386 (Workmaster II) Or Higher Computer

### Physical Extended Memory for Upper Memory Block Operation

If not already present, you must add and configure physical memory on your computer to be accessed in memory addresses over 1 Megabyte (FFFFF).

### 386 Memory Manager

You should install a 386 memory manager such as Microsoft EMM386 (MS-DOS Version 5.0 or higher), Quarterdeck QEMM-386 or 386MAX. The memory manager must be compliant with the Extended Memory Specification (XMS) Version 2.0 or later.

With MS-DOS 5.0 (or higher), you can install HIMEM and EMM386 to allow use of the Upper Memory Block.

1. Verify that your CONFIG.SYS file contains a device command for the HIMEM.SYS extended memory manager (or another memory manager that conforms to the XMS specification). For example:

```
device=c:\dos\himem.sys
```

2. Add a device command for EMM386 to your CONFIG.SYS file. The device command for EMM386 must come after the device command for HIMEM and before any commands for device drivers that use expanded memory. For example:

```
device=c:\dos\emm386.exe 1024 ram
```

### Note

With MS-DOS 6.0, more upper memory can be obtained from B000 – B7FF that is reserved for monochrome VGA display memory; however, this address space is unused on most computers. To enable extra UMBs, include this address range using the following command line instead of the one above:

```
device=c:\dos\emm386.exe 1024 ram i=b000-b7ff
```

3. Add a DOS=HIGH command to the CONFIG.SYS file. For example:

```
dos=high,umb
```

4. Restart your computer by pressing CTRL-ALT-Delete.

## Using Conventional Memory on an 80386 (Workmaster II) Or Higher Computer

### Extended Memory and MS-DOS 5.0 or Higher

If you have MS-DOS 5.0 (or higher), you can load MS-DOS into high memory. This will free more conventional memory for Logicmaster 90-30/20/Micro software and the communications driver. You must have at least 564 KB (577,536 bytes) of available DOS conventional memory.

1. Verify that your CONFIG.SYS file contains a device command for the HIMEM.SYS (MS-DOS Version 5.0 or higher) extended memory manager and the command to load MS-DOS into high memory. For example:

```
device=c:\dos\himem.sys
dos=high
```

2. Restart your computer by pressing CTRL-ALT-Delete.

## Using Expanded Memory on an 80386 (Workmaster II) Or Higher Computer

### Extended Memory

You must install a 386 memory manager such as Microsoft EMM386 (MS-DOS Version 5.0 or higher), Quarterdeck QEMM-386 or 386MAX. These memory managers use extended memory to simulate expanded memory. The memory manager should be compliant with the Extended Memory Specification (XMS) Version 2.0 or later.

### 386 Memory Manager

To install EMM386 as an expanded-memory emulator:

1. Verify that your CONFIG.SYS file contains a device command for the HIMEM.SYS extended memory manager (or another memory manager that conforms to the XMS specification). For example:

```
device=c:\dos\himem.sys
```

2. Add a device command for EMM386 to your CONFIG.SYS file. The device command for EMM386 must come after the device command for HIMEM and before any commands for device drivers that use expanded memory. For example:

```
device=c:\dos\emm386.exe 1024 ram
```

3. Disable or remove any other device commands for expanded memory managers.
4. Restart your computer by pressing CTRL-ALT-Delete.

## Using Video RAM on an 80386 (Workmaster II) or Higher Computer

As described above, the QEMM VIDRAM program from Quarterdeck Office Systems can use the EGA/VGA graphic display memory to gain additional MS-DOS conventional memory for use by the communications driver. The graphic display memory (if your EGA/VGA has 64 KB to 96 KB graphic memory) can be turned into 64 KB to 96 KB contiguous memory following the 640 KB conventional memory.

1. Verify that your CONFIG.SYS file contains a device command for the QEMM memory manager. For example:

```
device=c:\qemm\qemm386.sys
```

2. Verify that your AUTOEXEC.BAT file contains a command for the VIDRAM program. For example:

```
c:\qemm\vidram on
```

## Tested Configurations

The configurations listed below have been tested using the Standard Serial COM Port version of Logicmaster 90-30/20/Micro software. GE Fanuc offers this list as a guide to selecting compatible hardware for this version of software. These products have demonstrated compatibility with GE Fanuc via in-house testing and/or customer reports; however, hardware manufacturers' modifications to their products may affect compatibility. This list is not exhaustive and is offered as a guide only. No endorsement of any particular product is intended. If you have any questions, please contact the GE Fanuc Hotline, 1-800-828-5747.

Computer	MS-DOS / Memory Manager	CONFIG.SYS File *
Zenith Data Systems Z-Note 433 Lnc+ (Driver in MS-DOS)	MS-DOS 6.0 Microsoft HIMEM	device=c:\dos\himem.sys device=c:\dos\emm386.exe 1024 ram dos=high
Zenith Data Systems Z-Note 433 Lnc+ (Driver in UMB)	MS-DOS 6.0 Microsoft HIMEM	device=c:\dos\himem.sys device=c:\dos\emm386.exe 1024 ram dos=high,umb
Zenith Data Systems Z-Note 433 Lnc+ (Driver in EMS)	MS-DOS 6.0 Microsoft HIMEM	device=c:\dos\himem.sys 1024 ram device=c:\dos\emm386.exe dos=high
Gateway 2000 4DX-33 (Driver in MS-DOS)	MS-DOS 6.0 Microsoft HIMEM	device=c:\dos\himem.sys 1024 ram device=c:\dos\emm386.exe dos=high
Gateway 2000 4DX-33 (Driver in UMB)	MS-DOS 6.0 Microsoft HIMEM	device=c:\dos\himem.sys device=c:\dos\emm386.exe 1024 ram i=b000-b7ff dos=high,umb
Gateway 2000 4DX-33 (Driver in EMS)	MS-DOS 6.0 Microsoft HIMEM	device=c:\dos\himem.sys device=c:\dos\emm386.exe 1024 ram dos=high
Workmaster II (Driver in MS-DOS)	MS-DOS 5.0 Microsoft HIMEM	device=c:\dos\himem.sys device=c:\dos\emm386.exe 1024 ram dos=high
Workmaster II (Driver in UMB)	MS-DOS 5.0 Microsoft HIMEM	device=c:\dos\himem.sys device=c:\dos\emm386.exe 1024 ram dos=high,umb
Workmaster II (Driver in EMS)	MS-DOS 5.0 Microsoft HIMEM	device=c:\dos\himem.sys device=c:\dos\emm386.exe 1024 ram dos=high
ZEOS 486 (Driver in MS-DOS)	MS-DOS 6.0 QEMM 6.0	device=c:\dos\qemm386.sys dos=high
ZEOS 486 (Driver in UMB)	MS-DOS 6.0 QEMM 6.0	device=c:\dos\qemm386.sys ram=D000-Dfff dos=high

\* Each CONFIG.SYS file contains the lines: files=20 and buffers=48.

### Note

If your computer has a WSI card, the switch **x=ce00-cfff** should be added to the **device=c:\dos\emm386.exe** lines above to reserve this area of memory for its use. For optimum performance, SMARTDRV should be used.

### TCP/IP Communications

If you are using a TCP/IP Ethernet version of Logicmaster, refer to Chapter 3 of *Logicmaster™ 90-30 and Logicmaster 90-70 TCP/IP-Ethernet User's Manual Supplement* (GFK-1029B or later).

## Section 4: WSI Serial Port Setup

The serial port on the Work Station Interface (WSI) Board in the programmer provides serial communication between the programmer and the attached PLC. The WSI serial port setup functions are used to configure the WSI serial port, and to save or recall those configurations from disk files. If you do not have a WSI Board for serial communications, turn to the information on PLC communications serial port setup which preceded this information.

To display the WSI Serial Port Setup screen, press **COMSET (F4)** from the Programmer Setup menu.

```

|PORTS |MODE |PLCSEL |COMSET |UUMODE | | | | | |
1show p 2show f 3default 4 5 6setup 7save 8 9 10
>
          P R O G R A M M E R  W S I
          S E R I A L  P O R T  S E T U P

FILE NAME  C:\LM90\WSI030.PSU

PARAMETERS:
BAUD RATE          19200 (300, 600, 1200, 2400, 4800, 9600, 19200)
PARITY             ODD (ODD, EVEN, NONE)
STOP BITS          1 (1, 2)
MODEM TURNAROUND TIME 0 (0...255 counts, 1 count = 1/100 second)

          OFFLINE
C:\LM90\LESSON  PRG: LESSON
REPLACE

```

### Note

The PLC CPU defaults to the parameters displayed on this screen (e.g., 19200 baud, odd parity, one stop bit, etc.). These parameters must match the CPU slot configuration that was last stored to the PLC before communications can be established (see chapter 10, section 3, "Configuring the CPU Module").

The WSI serial port settings may be viewed, changed, saved to a disk file, or recalled from a disk file in all modes of operation by pressing the appropriate function key.

Field	Description
File Name	The name of a disk file containing the WSI port setup parameters. If no file name is entered, the default name %WSI030.PSU is used. This field allows you to save the setup parameters to a file other than the default file name. You can then recall the setup from this saved file and perform the setup within that screen.
Baud Rate	The transmission rate, in bits per second, of data through the port.
Parity	Specify whether the parity bit is <b>odd</b> or <b>even</b> ; or, if no parity bit is needed, select <b>none</b> . To toggle the selections, place the cursor on the field and press <b>Tab</b> to toggle forward ( <b>Shift-Tab</b> ) to toggle backward.
Stop Bits	All communications use one stop bit. Slower devices may use two stop bits.
Modem Turnaround Time	The time required for the modem to start data transmission after receiving the transmit request. If no modem is used, specify zero. When the PLC is connected through modem, the value <b>must</b> be one or greater.

### Setting Up a Port

To set up the serial port, enter the appropriate values in the fields on the WSI Serial Port Setup screen. Once all the values have been entered, press **Setup (F6)**.

### Displaying Port Settings

The current settings for the WSI Serial Port Setup screen can be displayed by pressing **Show Port (F1)**.

Pressing **F1** while completing the screen will delete all the changes that have been made. The previous settings of the current driver will then be redisplayed on the screen.

### Saving the Port Setup

A copy of the current serial port settings may be saved to a disk file by pressing **Save (F7)**. The data is saved to a separate file. If no file name is specified, a default name is assigned.

To create a file of settings which will not be automatically invoked, a file name other than the default name can be specified. If no path is given, the current default directory is used. If no file extension is provided, .PSU is used.

When **save (F7)** is pressed, the settings are stored to a file and also set up on the port.

### Displaying File Settings

The settings currently stored in a file can be viewed by first entering the file name and then pressing **Show File (F2)**. If no path is specified, the current default directory is used. If no file extension is provided, .PSU is used. If the *File Name* field is blank, the default setup file settings are displayed.

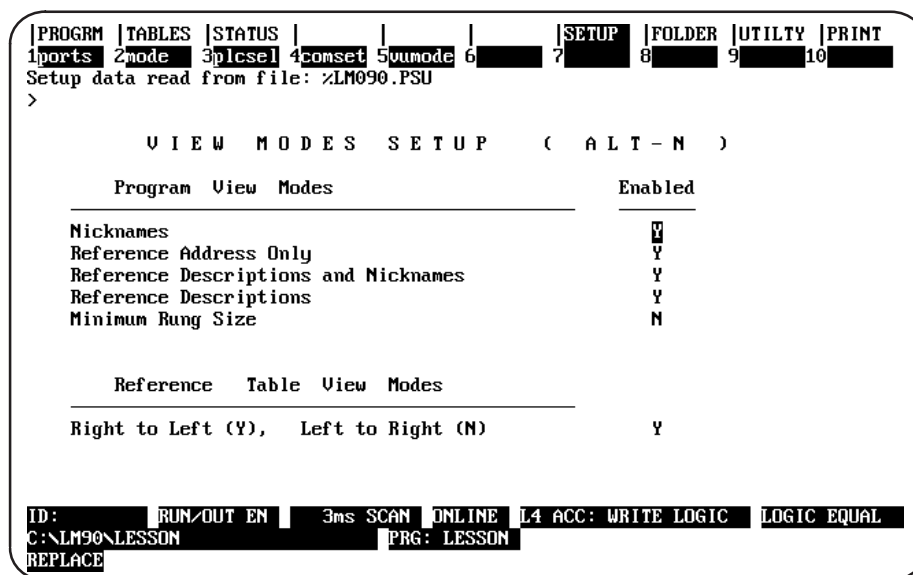


## Section 5: View Modes Setup (ALT-N)

The view mode setup function enables you to specify which modes are displayed when you press **ALT-N**. These view modes range from showing only rung references to showing reference names and reference descriptions in an expanded rung form (display all mode). You can also view the maximum amount of program logic on a screen by selecting a compressed rung mode.

The **Home**, **End**, **Page Up** and **Page Down**, **Previous**, **Next**, and cursor movement keys function the same, regardless of the view mode selected.

Display modes are selected from the View Modes Setup screen. To display this screen, press **View Mode (F5)** from the Programmer Setup menu.



When the View Modes Setup screen is displayed, the mode selections which were last saved are shown. The default selections are **Y** (Yes) for the programmer view modes and **N** (No) for minimum rung size mode.

Use the Up and Down cursor keys to move between selection fields. Then, use the Tab key to toggle each selection between **Y** (Yes) and **N** (No), or enter **Y** for the view modes you want to sequence through when you press **ALT-N**. Enter **N** to disable the modes you do not want to sequence through.

To validate and save the view mode selections in the home directory file %LM090.PSU, press **ALT-U** or the **Escape** key. At least one selection must be **Y** (Yes) in order for the selections to be valid. If all view mode selections are **N** (No), an error message is displayed and the selections are not saved.

To abort the selections and use the last saved setup, press **ALT-A**.

## Changing the View Mode

The **ALT-N** key sequence also enables you to display table data either right to left (lowest reference address on the right) or left to right (lowest reference address on the left). The view mode can be changed in any table (fixed or mixed) and in any program mode (**OFFLINE**, **ONLINE**, or **MONITOR**).

### Note

The print function does not distinguish between these two display modes and prints all tables right to left.

The default view mode is the last view mode selected with **ALT-N**. If the view mode has never been changed, the default set in the programmer setup is used.

1. The default reference table view mode is displayed on the View Modes Setup screen. To display this screen, press **View Modes Setup (F5)** from the Programmer Setup menu. The view mode in the following screen is right to left (the default display).

```

|PROGRAM |TABLES |STATUS |      |      |      |SETUP |FOLDER |UTILITY |PRINT
|ports  |mode   |plcset |comset|vumode|      |      |      |      |
Setup data read from file: %LM090.PSU
>

      V I E W   M O D E S   S E T U P   (   A L T - N   )

Program View Modes                                     Enabled
-----
Nicknames                                             Y
Reference Address Only                               Y
Reference Descriptions and Nicknames                 Y
Reference Descriptions                               Y
Minimum Rung Size                                    N

Reference Table View Modes
-----
Right to Left (Y), Left to Right (N)                 Y

ID:      RUN/OUT EN  3ms SCAN  ONLINE  L4 ACC: WRITE LOGIC  LOGIC EQUAL
C:\LM90\LESSON      PRG: LESSON
REPLACE

```

2. To change the default view mode to display left to right, move the cursor to the *Enabled* field for the *Reference Table View Modes* entry. Then, use the Tab key to toggle the selection to **N** (No) or enter **N**.
3. To validate and save the default view mode selection in the home directory file %LM090.PSU, press **ALT-U** or the **Escape** key.

## Section 6: Serial Printer Setup

The COM1 and COM2 serial ports can be used for serial printers. Beginning with Logicmaster 90-30/20 Release 4.50, the serial printer port *must* be configured with the MS-DOS mode command to match the printer settings.

### Note

This is a significant change from the previous versions. If you are printing to a serial printer, make sure you observe the following guidelines:

- Either COM1 or COM2 can be used for serial printers; however, the serial printer port must be configured with the MS-DOS mode command to match the printer settings. The following printer settings must be configured: baud rate, parity, data bits, and stop bits. An example mode command for a serial printer operating on COM2 at 1200 baud, with 8 bits per character, 1 stop bit per character, and no parity bits would be:

```
mode com2:12,n,8,1
```

In the example mode command displayed above, *COM2:* is the port, *12* specifies 1200 baud, *n* specifies no parity, *8* specifies 8 bits per character, and *1* specifies 1 stop bit.

- The mode command can be entered as an MS-DOS command or placed in the AUTOEXEC.BAT file. The MS-DOS mode command must be used to configure the serial printer port before entering the Logicmaster 90-30/20/Micro software package.
- If persistent errors occur when printing listings to COM1 or COM2 from within the Logicmaster 90-30/20/Micro software package, the buffer size on the printer may have to be increased and the serial baud rate configured for the port may have to be lowered. Read the user manuals provided with your printer and the MS-DOS sections in the manual concerning serial ports, printing, and the mode command.

### Note

The Standard Serial COM Port version of Logicmaster 90-30/20/Micro software requires either COM1, COM2, COM3, or COM4 as the communications port for communicating with the Series 90-30/20/Micro PLC. *Do not use the same COM port* for printing that you are using for communicating with the Series 90-30/20/Micro PLC.

# Chapter 7

## Program Folders

---

---

Series 90 programs and configuration data are stored in folders. A folder is an MS-DOS subdirectory which contains all information about one PLC program. To store more than one PLC program, you need to have one folder for **each** program. Folders must be created using Logicmaster 90-30/20/Micro software. An existing MS-DOS directory cannot be used as a folder.

PROGRAM  
FOLDER

- ← CPU Configuration
- ← I/O Configuration
- ← Program Logic
- ← Reference Tables
- ← Rung Comments
- ← Backup
- ← Teach Files

### Drawer

A drawer is an MS-DOS directory path which contains one or more folders.

### Note

Drawers must be created using MS-DOS before they can be used by Logicmaster90-30/20/Microprogrammingsoftware.

When Logicmaster 90-30/20/Micro software is entered, the current MS-DOS directory is the default drawer. The default drawer may be changed from the Select screen by pressing **ALT-C** to clear the **Program Folder** field and then entering the MS-DOS directory path (ending in \) of the subdirectory containing the folders you want, for example:

**C:\LM90\FOLDERS\.**

Refer to appendix G for an explanation of the files created with Logicmaster 90-30/20/Micro software.

## Program Folder Names

Every program will reside in its own program folder. The name of the program in the folder and the program folder will be the same (except for the “TEMP” folder, described below). The folder name is the program name in the PLC. The software will check these names to be sure they match. Programming functions, such as loading a program from the PLC to the computer, are not allowed if the names do not match.

## TEMP Program Folder

The special program folder TEMP does not require the folder name and program name to match. The TEMP folder can be used if you need to view a user program and monitor its operation when you do not have the original folder on your machine.

The TEMP folder can also be used to load a copy of a program and modify it, without overwriting the previous version. Temp folders created for the Series 90-30 and Series 90-70 are *not* compatible. If you have a 90-70 TEMP folder, you must exit to DOS and delete the TEMP directory before creating a 90-30 TEMP folder. You can use the DOS DELTREE command (DELTREE \LM90\TEMP) or delete all files in TEMP and remove the directory with the RD command, (CD \LM90, DEL TEMP\\*.\*, then RD TEMP).

### Note

Annotation files (nicknames, reference descriptions, and comment text) remain in the folder and are not stored to the PLC. Therefore, when loading a program from the PLC to the TEMP folder, the annotation will be missing. Those files are still in the original folder where the program was developed. If another program was developed in the TEMP folder prior to the download operation, the annotation in the recently loaded program will actually belong to the previous program.

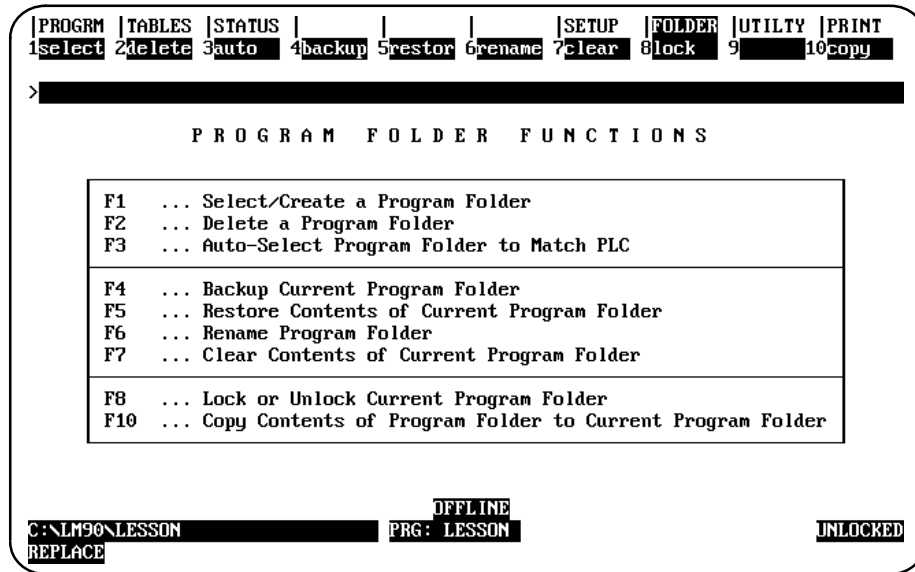
## Using Program Folder Functions

Both the configuration software and programming software provide a group of program folder functions. These functions can be used to create, select, modify, or delete program folders.

### Caution

**Do not use MS-DOS to copy individual files from one folder to another or to delete files. Doing so may produce unexpected results. MS-DOS may only be safely used to copy an entire program folder to another program folder of the same name. If MS-DOS functions have been used to place program files into a program folder with a different name, you will not be able to select the program folder.**

To use a program folder function, press **Folder (F8)** from the main menu.



Function Key	Function	Description	Page
F1	Select	Create a new program folder, or select a previously created folder.	7-4
F2	Delete	Remove a program folder that is no longer needed.	7-7
F3	Auto	Automatically select the correct program folder and attempt to verify equality with the PLC program.	7-5
F4	Backup	Make a backup copy of the current program folder.	7-8
F5	Restore	Restore a program folder with its backup copy.	7-9
F6	Rename	Rename a program folder to a new folder name.	7-11
F7	Clear	Clear the contents of the current program folder.	7-12
F8	Lock	Lock or unlock a program folder. Locking a program folder prevents its files from being changed or deleted.	7-13
F10	Copy	Copy a program folder into the current program folder. Use this function to make a copy of a program folder that can be modified while retaining the original.	7-14

### Note

If a folder contains locked subroutines, these blocks remain locked when the Logicmaster 90-30/20/Microsoft software copy, backup, and restore folder functions are used. For more information on locking and unlocking subroutines, refer to chapter 3, section 8, “Subroutine Blocks.”

## Selecting/Creating Program Folders

To create a new program folder or use one that already exists, press **select** (F1) from the Program Folder Functions menu.

```

|PROGRAM |TABLES |STATUS |      |      |      |SETUP |FOLDER |UTILITY |PRINT
1select  2delete 3auto   4backup 5restor 6rename 7clear 8lock  9      10copy
>
      S E L E C T   O R   C R E A T E   A   P R O G R A M   F O L D E R

Program Folder: ████████████████████████████████████████████████████████████
PLC Program Name: *****

Folders in Drawer: C:\LM90
  LESSON

<< Type a folder name, or use the cursor keys to select an existing folder. >>
<< Use PgUp/PgDn to page through folders. Press ENTER to start selection. >>

OFFLINE
C:\LM90\LESSON  PRG: LESSON  UNLOCKED
REPLACE

```

### Note

Once you edit a program folder using Release 3 or later of Logicmaster 90-30/20 software, you cannot use that folder with an earlier release of software.

### Selecting a Previously Created Folder with a New 351 or 352 (or Later) Configuration

If you are using a 351 or 352 CPU hardware configuration and select a folder created previously when you had a different CPU configuration, you will be prompted to convert the folder to one that uses the code written for 351 or 352 CPUs.

### Note

As mentioned above, Logicmaster prompts you if you are using 351 or 352 or later hardware configuration and then select a folder created under a different CPU configuration. If you plan to use that folder again with lower model CPUs, Logicmaster does allow that; but make sure you stay within the memory limit and program size limits of the lower model CPU.

### Caution

**Do not use a pre-Release 7 version of Logicmaster 90-30/20/Micro to store folders to a 352 CPU. There is the potential of having that folder translated to 331 CPU format.**

The name of the last folder selected is shown in reverse video in the **Program Folder** field. Other program folders in the current drawer are also listed on the screen. If the attached PLC contains a user program, the name of this program appears in the **PLC Program Name** field. If the attached PLC does not contain a user program, this field is blank. If the programmer is offline or not connected to a PLC, this field contains asterisks. An entry in this field may not be edited.

To select a folder in the current drawer, move the cursor to the desired folder name or type the name of the program folder in the **Program Folder** field, and press the **Enter** key. (This is also the name of the program.) The name can have up to seven characters. If a second floppy disk is inserted after a folder has already been selected or created on the first floppy disk, the list of folders is not updated. In order to have the folder list updated, you must reselect the drawer (e.g., enter **A:\**).

To select a folder in a different drawer, first select the drawer by entering the drawer MS-DOS path, ending with a “\”, and pressing the **Enter** key. **For example, enter C:\LM90\FOLDERS\**. When the **Enter** key is pressed, the names of the folders in the newly selected drawer are displayed. Position the cursor on the name of the program folder you wish to select or type the name of the program folder, and press the **Enter** key.

If the program folder already exists, select whether or not to create a backup copy. If the program folder does not already exist, the software will prompt you for confirmation to create a new program folder. This prevents you from accidentally creating a program folder due to a typing mistake. To cancel any changes made to this screen, press **ALT-A** (abort). To return to the Program Folder Functions menu, use the **Escape** key.

## Auto-Select Function

The **Auto (F3)** softkey is only active when the select screen is displayed. When **F3** is pressed in **OFFLINE** mode or if the programmer is not communicating with the PLC, a message is displayed indicating that the auto-select function is not available in that mode.

When **F3** is pressed in **MONITOR** or **ONLINE** mode, the programming software checks to see if a folder exists under the current directory whose name matches the name of the program in the PLC. If the folder does exist, it is automatically selected. The software then attempts to verify the program or configuration in the folder with that in the PLC, and updates the **equality** field on the status line based on the result. After the select screen is exited, the main menu is displayed.

In the programming software, if a matching folder does not exist, the following message is displayed: “Folder matching PLC name not found; load program to TEMP folder? (Y/N)”. If **N** (No) is entered, no folder is selected and the select screen remains displayed on the screen.

If **Y** (Yes) is entered, the TEMP folder is selected or created if one does not exist. Program logic and configuration are loaded from the PLC into the TEMP folder. If the TEMP folder already contains logic and/or configuration files, those files are backed up prior to the load. The status lines are updated to show that the folder program is equal to the version in the PLC.

In the configuration software, the message displayed when a matching folder does not exist is “load configuration to TEMP folder?”.

If an error is encountered verifying or loading data from the PLC, an error message is displayed and the select screen continues to be displayed.



## Automatic Folder Selection

The automatic folder selection feature allows Logicmaster 90-30/20/Micro software to automatically select the correct program folder and attempt to verify equality with the PLC program.

During power-up initialization, when the initializing screen is displayed, Logicmaster 90-30/20/Micro software attempts to establish point-to-point communications with an attached PLC. If this attempt is not successful, the initial folder selection screen is displayed. The programmer mode will default to **OFFLINE** mode on computers without a programmer mode keyswitch.

If a connection exists, computers without a keyswitch default to **MONITOR** mode. Logicmaster 90-30/20/Micro software will then check to see if a folder whose name matches the PLC program name exists. If the folder does exist, that folder is automatically selected and an equality check is performed. The main menu screen is displayed after the equality check is completed.

If a matching folder does not exist, the initial folder selection screen is displayed. You can select a folder or press **Auto (F3)** to load to the TEMP folder, or select a new drawer and press **Auto (F3)** to try again to select the matching folder and then to check to see if they are both the same.

### Note

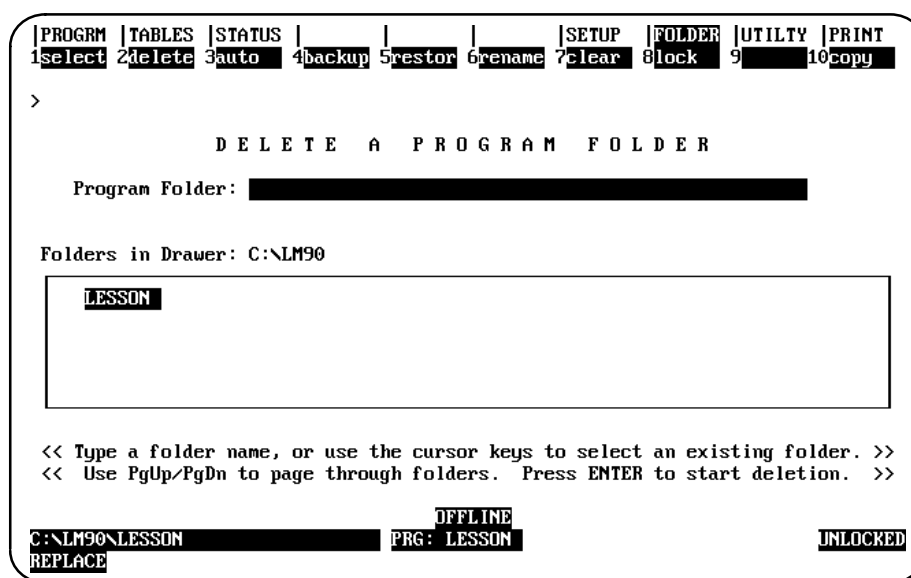
If the baud rate with the serial version of Logicmaster 90-30/20/Micro software is set to 1200 or less, the programmer does not automatically go to **MONITOR** mode.

## Deleting Program Folders

Use this function to remove a program folder that is no longer needed. If the program folder has a backup, that copy is also deleted automatically. If you have created any directories using MS-DOS beneath the program folder to be deleted, you must remove them before using the delete function. You may not be able to delete a program folder residing on a virtual drive.

If a program folder is locked (lock status is shown in the lower right corner of your screen), you cannot delete it. Please refer to the information on unlocking program folders provided later in this chapter.

Press **Delete (F2)** from the Program Folder Functions menu.



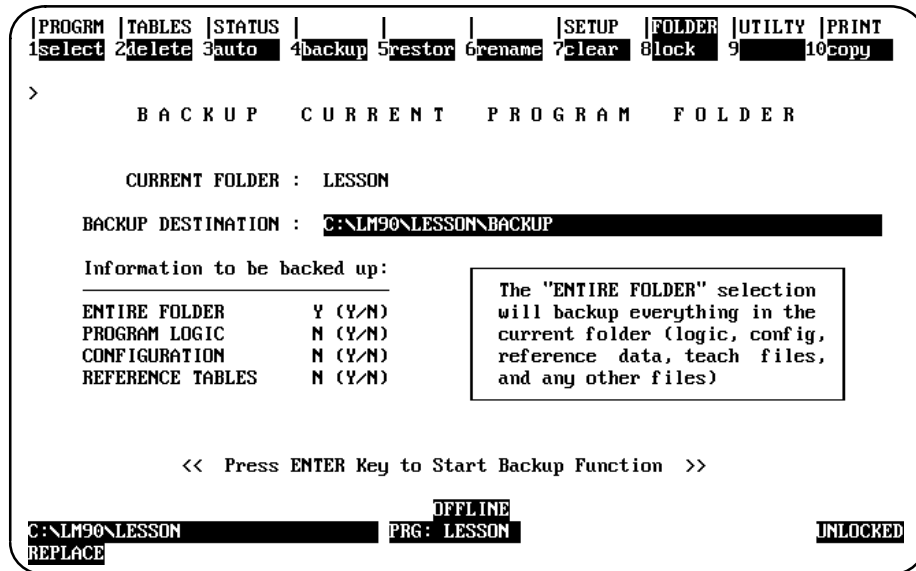
To delete a folder in the current drawer, type the name of the program folder in the **Program Folder** field or move the cursor to the desired folder name. Then, press the **Enter** key. The software will not allow the currently selected folder to be deleted.

To delete a folder in a different drawer, enter the full path specification or specify the drawer in the **Program Folder** field. The window is changed to display the folders in this drawer.

Respond to the confirmation prompt to continue with the deletion. **Once started, the delete operation cannot be aborted.** To return to the Program Folder Functions menu, press the **Escape** key.

## Backing Up Program Folders

To create a backup copy of the currently selected program folder, press **Backup (F4)** from the Program Folder Functions menu.



The backup folder is located in a subdirectory under the program folder. If no backup folder exists for the current program folder, one is created automatically.

The **BackupDestination** field allows you to specify where you want to put the backup archive. The default destination is the backup directory of the current folder. To change the destination, move the cursor to this field and enter the new destination.

### Note

When backing up to a floppy diskette, you must back up to a subdirectory. For example, **A:\** cannot be used as the backup destination. If **LESSON** is the folder name, **A:\LESSON** should be used. If only **A:\** is specified, you cannot restore from the floppy diskette.

Use the cursor keys to move from one option field to another. To back up the entire contents of the current folder, enter **Y** (Yes) in the **Entire Folder** field. To back up only selected options, enter **Y** (Yes) in the corresponding fields:

Field	Description
ProgramLogic	The ladder logic program.
Configuration	The current configuration.
Reference Tables	The reference tables for the program.

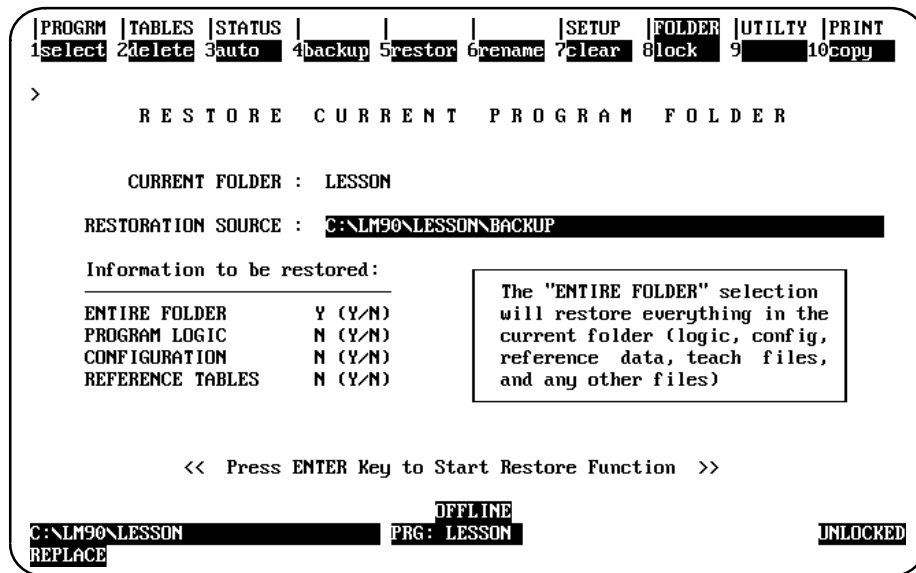
Then press the **Enter** key. If you back up the entire folder, the software will prompt you for confirmation and then automatically delete the previous backup. A new backup archive will be created to reflect the current contents of the program folder. If you back up only program logic or any other options besides the entire folder, the previous backup is left intact and the selected options are added to the backup. If the selected options were already stored to the backup, they will first be deleted.

**Once started, the backup operation cannot be aborted.** To return to the Program Folder Functions menu, press the **Escape** key.

## Restoring Program Folders

To replace the contents of the current program folder with the backup copy, press **Restore (F5)** from the Program Folder Functions menu. To restore a program folder, a backup copy must exist.

If the program folder is locked (locked status is shown in the lower right corner of your screen), you must change the status to unlocked before a restore can occur. Please refer to the information on unlocking program folders provided later in this chapter.



The **RestorationSource** field allows you to specify where you want to retrieve the backup archive from. The default source is the backup directory of the current folder. To change the source, move the cursor to this field and enter the new source.

### Note

The backup archive must be restored to a folder with the same name as the folder from which the backup originally was made. If not, the message “No backup found to restore” is displayed.

Use the cursor keys to move from one option field to another. To restore the entire contents of the current folder, enter **Y** (Yes) in the **Entire Folder** field. To restore only selected options, enter **Y** (Yes) in the corresponding fields:

Field	Description
ProgramLogic	The ladder logic program.
Configuration	The current configuration.
Reference Tables	The reference tables for the program.

Then press the **Enter** key. If any information exists in the current program folder that would be written over by the backup copy, the software will prompt you for confirmation. **Once started, the restore operation cannot be aborted.**

If the programmer is in **ONLINE** or **MONITOR** mode, an auto-verification is performed to determine whether the restored program or configuration is equal to the PLC.

To return to the Program Folder Functions menu, press the **Escape** key.

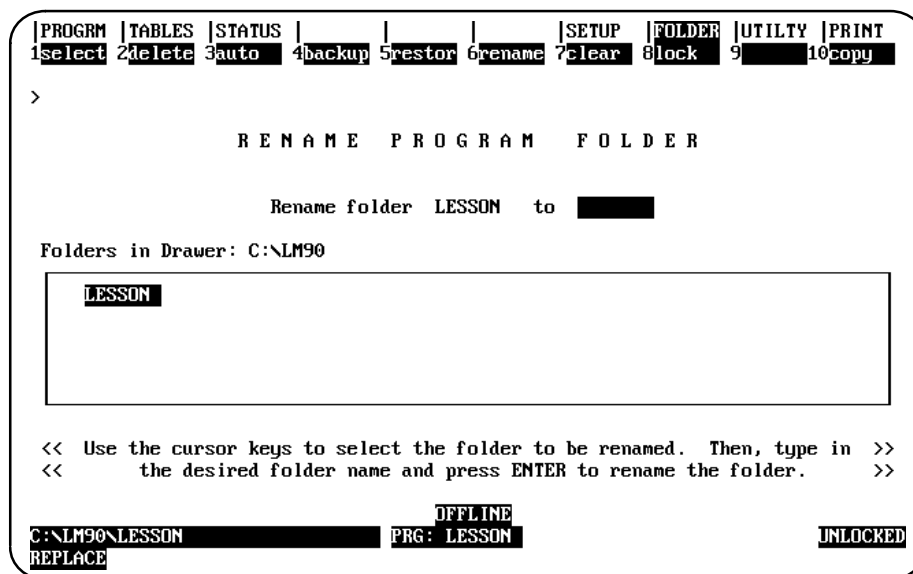
### Note

The Backup and Restore commands are meant to provide a quick snapshot of your folder to a hard disk. Up to 5 files are created in the BACKUP subdirectory under your current folder. The files are not compatible with PKUNZIP and may not be accessible with other versions of Logicmaster. It also does not automatically create the proper folder name when you do a Backup to a floppy, so it will be difficult to restore the folder.

If you are moving a folder to or from a floppy for transfer to another computer, use the Copy command (page 7-15).

## Renaming Program Folders

This feature of Logicmaster 90-30/20/Micro software enables you to rename any folder in the current drawer to another folder name that is not being used. To rename a program folder, press **Rename (F6)** from the Program Folder Functions menu or from another Program Folder screen.



Enter the new folder name into the **Rename Folder to** field. If you press an alphanumeric key, it is automatically displayed in the **Rename Folder to** field. To clear this field, press **ALT-C** or use the **Delete** key to delete the contents of the field.

Press the **Enter** key to start the renaming operation. The software will prompt you for confirmation to continue the operation with the message “Current folder (<file\_name>) will be renamed to <file\_name>. Continue? (Y/N).” **Once started, the rename operation cannot be aborted.**

Once the rename operation is complete, the message “Folder renamed successfully” is displayed and the **Rename Folder to** field is cleared.

If the rename operation is aborted, the **Rename Folder to** field is not automatically cleared since the rename never completed. You must manually clear the field by pressing **ALT-C** or use the **Delete** key to delete the contents of the field.

## Clearing Program Folders

To delete the contents of the current program folder while keeping the folder itself for future use, press **C**lear (F7) from the Program Folder Functions menu.

If a program folder is locked (look at the lower right corner of your screen), you cannot clear it. Please refer to the information on unlocking program folders provided later in this chapter.

```

PROGRAM | TABLES | STATUS | _____ | _____ | SETUP | FOLDER | UTILITY | PRINT
1select 2delete 3auto 4backup 5restor 6rename 7clear 8lock 9 10copy
>

      CLEAR CURRENT PROGRAM FOLDER

Current Folder : LESSON

Information to be cleared:
-----
ENTIRE FOLDER      Y (Y/N)
PROGRAM LOGIC     N (Y/N)
CONFIGURATION      N (Y/N)
REFERENCE TABLES N (Y/N)
PRINT XREF FILES  N (Y/N)

The "ENTIRE FOLDER" selection
will clear everything in the
current folder (logic, config,
reference data, cross reference
files, and any other files)

<< Press ENTER Key to Start Clear Function >>

C:\LM90\LESSON  OFFLINE  PRG: LESSON  UNLOCKED
REPLACE

```

Use the cursor keys to move from one option field to another. To clear the entire contents of the current folder, enter **Y** (Yes) in the **Entire Folder** field. To clear only selected options, enter **Y** (Yes) in the corresponding fields:

Field	Description
ProgramLogic	The ladder logic program.
Configuration	The current configuration.
Reference Tables	The reference tables for the program.
Print Cross Reference Files	The PRINT.XOV and all .XRF files for the program.

The Tab key may also be used to toggle the selection of each option. The default selection for the **Entire Folder** field is **Y** (Yes); all other fields default to **N** (No). Once the **Entire Folder** field is set to **Y**, the remaining fields are automatically set to **N**. Once any of the other fields is set to **Y**, the **Entire Folder** field is automatically set to **N**.

### Note

Program annotation files only exist in the folder and not in the PLC. Clearing the folder and then loading the program from the PLC results in lost annotation.

Press the **Enter** key to begin the clear operation. If there is any information in the folder, the software will prompt you for confirmation to continue the clear operation. **Once started, the clear operation cannot be aborted.**

If you answer the continuation prompt by pressing **Y** (Yes), you will have a chance to back up the current program folder. To create a backup copy, press **Y** (Yes) at the prompt. Press **N** (No) if you do not want to back up the information first.

If the programmer is in **ONLINE** or **MONITOR** mode, the status is automatically set to **NOT EQUAL**.

To return to the Program Folder Functions menu, press the **Escape** key.

### Locking/Unlocking Program Folders

Locking a program folder protects its files from being accidentally changed. To change the locked status of the current program folder, press **Lock (F8)** from the Program Folder Functions menu.

```

PROGRAM | TABLES | STATUS |      |      |      | SETUP | FOLDER | UTILITY | PRINT
1select 2delete 3auto  4backup 5restor 6rename 7clear 8lock  9      10copy
>
  LOCK / UNLOCK CURRENT PROGRAM FOLDER

Current Folder: LESSON

Selection: UNLOCKED (Unlocked, Locked)

<< Press ENTER Key to Change Protection Status of Current Program Folder >>

C:\M90\LESSON OFFLINE
REPLACE      PRG: LESSON UNLOCKED

```

Use the **Tab** key to change the access description. Then press the **Enter** key. The new access description will appear in the lower right corner of the screen.

### Note

Program folders on write-protected floppy diskettes are automatically locked. Remove the write-protect tab and unlock the folder using this function.



## Copying Program Folders

### Copying to Another Folder on the Hard Disk

Use the copy function to copy from another program folder into the current folder. The source program folder is copied into the "Current" program folder. If there are any files in the current folder, they are destroyed by the copy process.

#### Note

The Copy feature functions somewhat differently than you might expect. Notice that you are copying another program folder *into* the current folder. That is, if you have been using a folder called "LESSON," then LESSON is the name of your current folder (as displayed in the Status line toward the bottom of your screen). If you copy the folder called "ACCTRL," then the folder called LESSON becomes an exact copy of ACCTRL.

#### Note

The copy function is only available in the programming software; it is not available in the configuration software.

To make a copy of a program folder, press **Copy (F10)** from the Program Folder Functions menu.

```

PROGRAM | TABLES | STATUS | | | | SETUP | FOLDER | UTILITY | PRINT
1select 2delete 3auto 4backup 5restor 6rename 7clear 8lock 9 10copy
>
COPY PROGRAM FOLDER TO CURRENT FOLDER

Source Folder : ████████████████████████████████████████████████████████████
Current Folder : LESSON

Current drawer is C:\LM90

Information to be copied:
ENTIRE FOLDER      Y (Y/N)
PROGRAM LOGIC     N (Y/N)
CONFIGURATION      N (Y/N)
REFERENCE TABLES N (Y/N)

The "ENTIRE FOLDER" selection
will copy everything from the
source folder (logic, config,
reference data, teach files,
and any other files) to the
current folder.

<< Press ENTER Key to Start Copy Function >>

C:\LM90\LESSON OFFLINE UNLOCKED
REPLACE PRG: LESSON

```

Enter the name of the program folder whose contents you want to copy into the currently selected program folder. Then, use the cursor keys to move from one option field to another. To copy the entire contents of the source folder, enter **Y** (Yes) in the

**Entire Folder** field. To copy only selected options, enter **Y** (Yes) in the corresponding fields:

Field	Description
ProgramLogic	The ladder logic program.
Configuration	The current configuration.
Reference Tables	The reference tables for the program.

The Tab key may also be used to toggle the selection of each option. The default selection for the **Entire Folder** field is **Y** (Yes); all other fields default to **N** (No). Once the **Entire Folder** field is set to **Y**, the remaining fields are automatically set to **N**. Once any of the other fields is set to **Y**, the **Entire Folder** field is automatically set to **N**.

Press the **Enter** key to begin the copy operation. If there is any information in the current folder, the software will prompt you for a confirmation to continue the copy operation since everything in the current folder will be overwritten. **Once started, the copy operation cannot be aborted.**

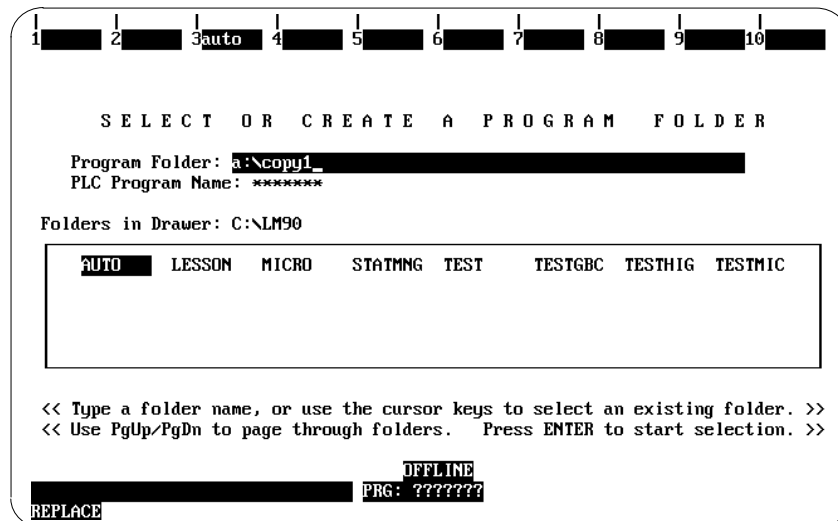
If the programmer is in **ONLINE** or **MONITOR** mode, an auto-verification is performed to determine whether the programmer/configurator is equal to the PLC.

To return to the Program Folder Functions menu, press the **Escape** key.

## Copying to a Diskette

To copy to a diskette, you will need to follow these steps:

1. Place a diskette in your disk drive (usually an A or B drive). There should be plenty of room on the diskette.
2. Press **F8** from the main menu to select the **Folder** functions.
3. Press **select (F1)** "Select/Create a Program Folder" (see page 7-3 for screen sample if needed).
4. In the Program Folder blank, type in the name you wish to use for the copy you are about to make. Make sure you specify the disk drive letter, e.g., **a:\**as shown below:



5. Press **Enter** to select this new folder on your A or B drive.

The following prompt will appear at the top of your screen: "Program folder does not exist; create new folder? (Y/N)"

6. Type **Y** to confirm that you want to create a new folder on your diskette.

### Note

Notice that you are only creating an empty folder. In the next steps you will copy the contents of your source folder into that empty folder.

7. After Logicmaster has finished creating a new folder, press the **Copy** softkey (**F10**).

```

|PROGRAM |TABLES |STATUS |      |      |      |SETUP |FOLDER |UTILITY |PRINT
|1select |2delete |3auto  |4backup|5restor|6rename|7clear|8lock  |9      |10copy
-----
>
C O P Y   P R O G R A M   F O L D E R   T O   C U R R E N T   F O L D E R

Source Folder : c:\lm90\lesson_
Current Folder : COPY1

Current drawer is A:\

Information to be copied:
-----
ENTIRE FOLDER      Y (Y/N)
PROGRAM LOGIC     N (Y/N)
CONFIGURATION     N (Y/N)
REFERENCE TABLES N (Y/N)

The "ENTIRE FOLDER" selection
will copy everything from the
source folder (logic, config,
reference data, teach files,
and any other files) to the
current folder.

<< Press ENTER Key to Start Copy Function >>

A:\COPY1          OFFLINE
REPLACE          PRG: COPY1          UNLOCKED

```

8. Type in the entire path and folder name, e.g., **c:\lm90\lessons** shown above. Be sure to include the DOS drive and directory which is usually **c:\lm90**.
9. Press **Enter** to start the Copy function.

When the Copy is complete, the words "Selected folder items have been copied to current folder" will appear towards the top of your screen.

10. Press the **Escape** key to exit the Copy function which takes you to the Program Folder Functions. Press the **Escape** key again to return to the Program Folder Functions menu. Press the **Escape** key again if you wish to exit the Programming software.

(An alternate way of exiting the software is to press **Ctrl-Break** once. A prompt will appear asking you if you wish to exit. Type **Y** (for Yes) to return to the Logicmaster main menu where you can press either **F10** or the **Escape** key to exit. When exiting through **Ctrl-Break**, you do not need to press the **Escape** key multiple times.)

You can now remove your diskette which has a copy of the source folder on it.

# Chapter 8

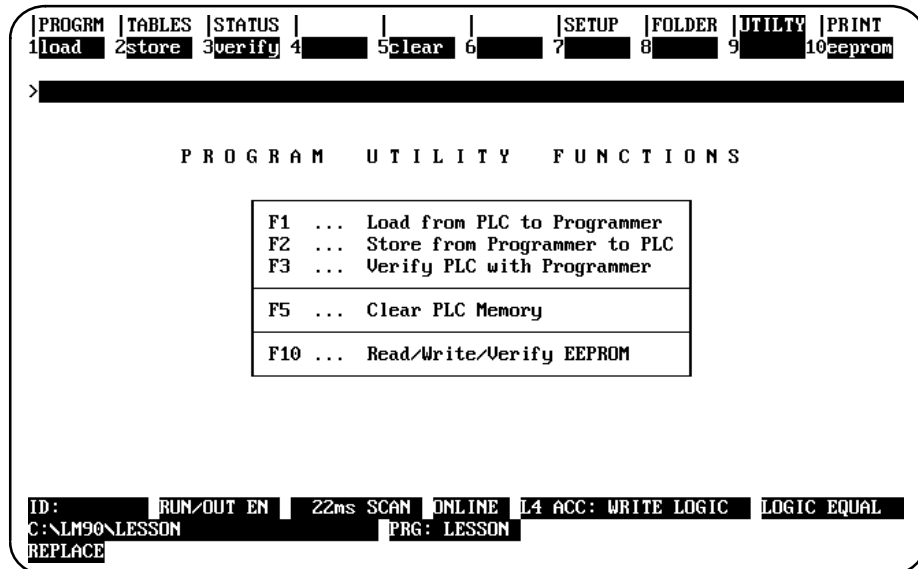
## Program Utilities

The program utility functions are used to transfer programs, configuration data, and reference tables between the programmer and the PLC. They are also used to compare the program, configuration data, and reference tables in the programmer with the program, configuration data, and reference tables in the PLC, to clear PLC memory, and to read/write/verify EEPROM.

### Note

In the configuration software, only the configuration may be loaded, stored, verified, or cleared. No operations on program logic or tables may be performed.

To access the program utility functions, press **Utility (F9)** from the main menu, or **Shift-F9** from any main menu function screen.



<b>Function Key</b>	<b>Function</b>	<b>Description</b>	<b>Page</b>
F1	Load	Copy program logic, configuration data, and/or reference tables from the PLC to the programmer.	8-3
F2	Store	Copy program logic, configuration data, and/or reference tables from the programmer to the PLC.	8-5
F3	Verify	Compare program logic, configuration data, and/or reference data in the programmer with the program logic, configuration data, and/or reference tables in the PLC.	8-8
F5	Clear	Delete program, configuration data, and/or reference tables from PLC memory.	8-10
F10	EEPROM	Read the EEPROM contents into PLC memory, write the entire contents of PLC memory to EEPROM, or verify the EEPROM contents with the PLC memory.	8-12

## Loading from PLC to Programmer

Use the load function to transfer program logic, configuration data, and/or reference tables from a PLC to the programmer. The load function transfers the program, which remains unchanged in the PLC. This function may be password protected in the PLC. If so, you must know the password in order to use the function.

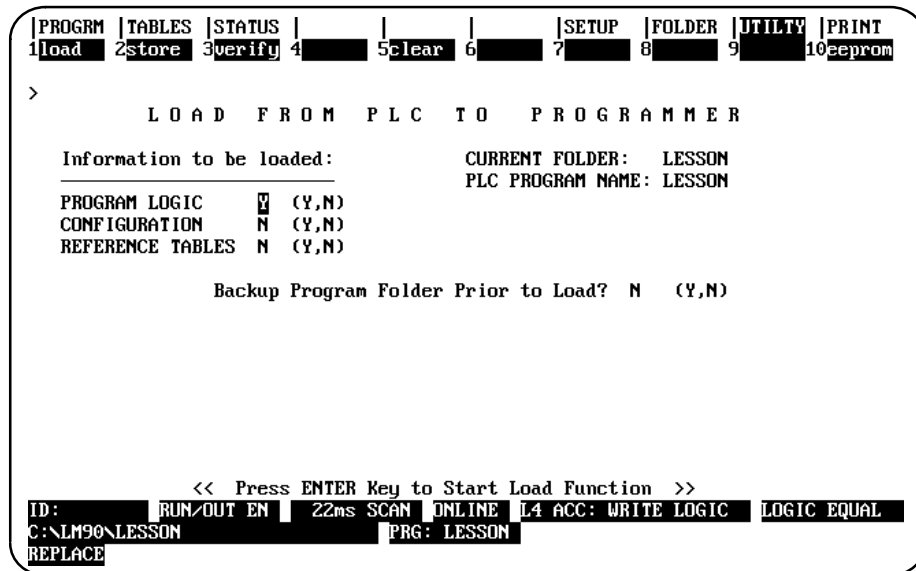
### Caution

**You cannot load a Release 4 or above Program with a release earlier than 3.50 of of Logicmaster 90-30/20 software.**

### Note

In the configuration software, only the configuration may be loaded. No operations on program logic or tables may be performed.

To use the load function, the programmer must be in **ONLINE** or **MONITOR** mode. Press **Load (F1)** from the Program Utility Functions menu or from another Program Utilities screen. The Load Program screen appears:



### Note

Annotation files (nicknames, reference descriptions, and comment text) remain in the folder and are not stored to the PLC. Therefore, you must load from the PLC to the original folder containing the annotation in order to retain its annotation.

## Note

If comment text has been edited since the program was stored to the PLC, do not load the program to the folder to restore equality. The program with the updated text should be stored to the PLC instead. This also applies to identifier descriptions for subroutine blocks which have been edited off-line. The subroutine block containing the edited identifier descriptions should be stored to the PLC in order to restore equality.

The Load Program screen shows the currently selected program folder and the program name in the PLC; these cannot be changed on this screen. A program must be loaded to a folder whose name matches the PLC program name or a TEMP folder (e.g., a folder whose name is TEMP).

Three types of data can be loaded from the PLC to the programmer: program logic, configuration data, or reference tables. When this screen first appears, only the program logic is set to **Y** (Yes). To load all of the data, change the selections for reference tables and configuration to **Y** (Yes). To load only part of the data, select **N** (No) for any of the following:

Field	Description
ProgramLogic	The ladder logic program.
Configuration	The current configuration.
Reference Tables	The reference tables for the program.

Also, select whether or not you want to create a backup copy of the current program folder before loading the data. Use the cursor keys to select items, and type in new selections as appropriate. To restore the original selections while editing this screen, press **ALT-A**.

To begin loading the data, press the **Enter** key.

The name of the program in the PLC is checked against the name of the current program folder. If the names are the same, the load continues. If the names are not the same, you cannot start the load unless the folder name is TEMP.

After a successful transfer of data, the software displays the message "Load Complete". If a communication error occurs during the load process (indicated by a message on the screen), the selected items are cleared from the current folder. Correct the error and repeat the load function.

To stop a program transfer in progress, press **ALT-A**.

To return to the Program Utility Functions menu, press the **Escape** key.

### Loading Reference Tables from the PLC to Save in the Computer Folder

Reference Table data is not automatically uploaded from the PLC unless you select it. You may want to do this periodically to capture a snapshot of all PLC register values and override conditions to save in your folder. When you make Online changes to tables or overrides, the information is **NOT** saved in the folder. If you clear PLC memory or replace the CPU, the reference table information in the PLC is lost. Loading tables from the PLC into the folder allows you to store the same information in a replacement PLC.

### Storing to PLC from Programmer

Use the store function to copy program logic, configuration data, and/or reference tables from the programmer to the PLC. The store function copies the program, which remains unchanged in the programmer. If the PLC program name is not the same as the folder name, the store function clears both the program and configuration data from the PLC. The data is then stored from the new program folder.

The store function may be password protected in the PLC. If so, you must know the password in order to use the function.

#### Note

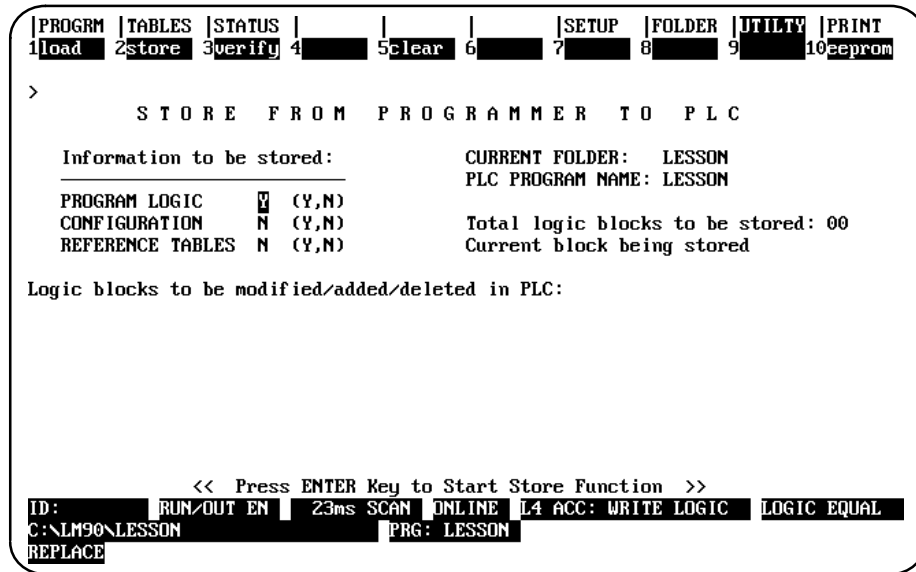
In the configuration software, only the configuration may be stored. No operations on program logic or tables may be performed.

To use the store function, the programmer must be in **ONLINE** mode. The PLC can be in either **STOP** or **RUN** mode. (For more information, see “Run Mode Store Function” on page 8-6.)

Logic Size (Bytes)	2400 Baud	19.2K Baud
1K	12seconds	2 seconds
2K	17seconds	3 seconds
4K	26seconds	4 seconds
8K	45seconds	8 seconds
16K	84 seconds	13seconds



Press **store (F2)** from the Program Utility Functions menu. The Store Program screen appears:



The screen shows the currently selected program folder, which cannot be changed.

Three types of data can be stored from the PLC to the programmer: program logic, configuration data, or reference tables. When this screen first appears, only the program logic is set to **Y** (Yes). To store all of the data, change the selection for reference tables and configuration to **Y** (Yes). To store only part of the data, select **N** (No) for any of the following:

Field	Description
ProgramLogic	The ladder logic program.
Configuration	The current configuration.
Reference Tables	The reference tables for the program.

### Note

Annotation files (nicknames, reference descriptions, and comment text) remain in the folder and are not stored to the PLC.

A program containing subroutine blocks and/or subroutine CALL instructions cannot be stored to a Pre-Release 3 PLC.

Use the cursor keys to select items, and type in new selections as appropriate. To restore the original selections while editing this screen, press **ALT-A**.

The information to be transferred must fit within the configured boundaries of the PLC (for example, its register memory size).

To begin storing the data, press the **Enter** key. The program must be complete, and must not contain errors in syntax or any instructions which are not supported by the attached PLC. If there are errors, the store operation is not attempted.

After a successful transfer of data, the software displays the message “Store Complete”. If a communication or disk error occurs during the store process (indicated by a message on the screen), the selected items are cleared from the current folder. Correct the error and repeat the store function. (Refer to the Caution note below for further information.)

To stop a program transfer in progress, press **ALT-A** if the PLC is in **STOP** mode. If the PLC is in **RUN** mode when the store begins, you cannot abort the transfer of program logic.

To return to the Program Utility Functions menu, press the **Escape** key.

### Note

If you store a program that was written using a CPU hardware configuration other than model 351 to a 351 CPU, you will be prompted to convert the folder to one that uses the code written for 351 CPUs when you start the storing process. Refer to page 7-4 for other considerations when using previously created folders with a 351 CPU.

### Caution

**A Store failure, such as a “Program too large” fault or any other failure during a Store function, clears (resets to the default value) everything that had been selected to be stored (program logic, configuration, or reference tables) regardless of whether or not they were successfully stored before the point at which the Store failed. The only exception to this clearing is that passwords are not re-enabled as part of the configuration in the failure.**

**If one of the selected items is the configuration, remember that this includes SNP ID (especially important in a multidrop communications link) and has to be reset manually. Communications parameters such as Baud rate/parity would have to be reset as well. In addition, on a 351 or 352 CPU, all serial port configuration parameters would be lost. The connection to the PLC remains usable until you disconnect. The next attach, however, would be affected by this unless you correct the situation before disconnecting.**

**To correct these problems caused a Store failure, correct the problem area in the program or configuration, then try again to store the program or configuration or reference tables to the PLC. If you had selected configuration when the Store failed, you will also have to set the SNP ID as a separate step.**

## Run Mode Store Function

The **RUN MODE STORE** function enables you to transfer program logic and reference table data to a running PLC system. To do this, the Logicmaster 90-30/20/Micro software automatically switches the PLC into **STOP/IOSCAN** mode during the transfer and then switches it back to **RUN** mode after the transfer is complete. The logic sweep is stopped for approximately 1 to 10 seconds, depending on the size of the program and the baud rate. (A very large program combined with a very low baud rate could result in more than 10 seconds.)

### Caution

**With Release 3.50 or earlier of Logicmaster 90-30/20 software and a CPU Release 3.x or earlier, non-retentive outputs are maintained during the store operation, but they are not maintained after the store operation is complete, as when the system toggles from STOP/IOSCAN mode back to RUN mode. All non-retentive outputs are turned OFF, regardless of their prior state, during this STOP-TO-RUN transition.**

**Retentive outputs are not affected by this operation. They are maintained at all times during and after the RUN MODE STORE.**

Beginning with Release 4.01 of Logicmaster 90-30/20 software and Release 4.02 CPUs, a PLC mode called "Pause" will prevent non-retentive outputs from being cleared. Pause mode will cause the CPU sweep to be suspended and all I/O states frozen until the entire program has been transferred. The PLC will then resume operation where it paused, without the **STOP-TO-RUN** transition that clears non-retentive outputs in CPUs prior to Release 4.02.

### Note

You can store a changed block from Block Edit mode without stopping or pausing the PLC. For more information about this feature, refer to page 3-104 in this manual.

## Verifying a Program with the PLC

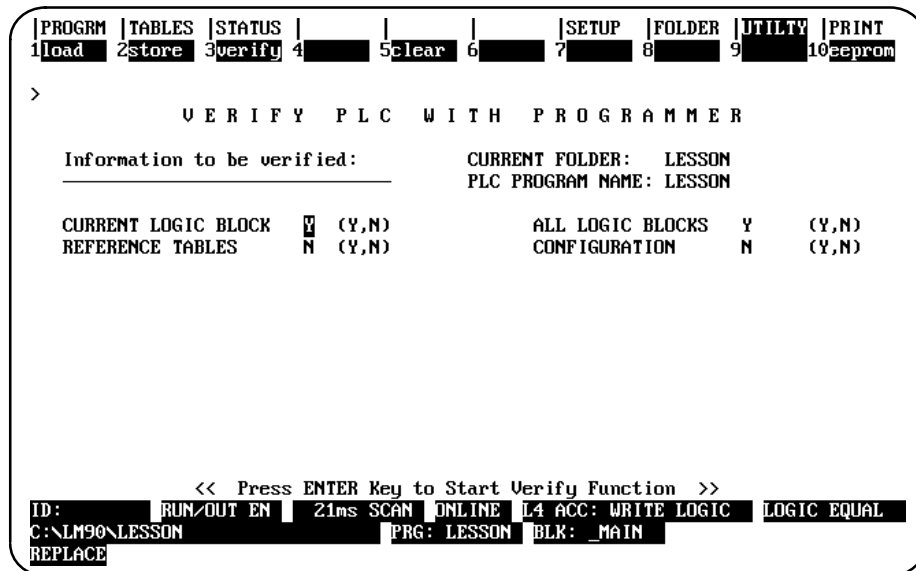
Use the verify function to compare program logic, configuration data, and/or reference tables in the programmer with the program logic, configuration data, and/or reference tables in the PLC. This function may be password protected. If so, you must know the password in order to use the function.

### Note

You cannot verify a view locked block with an unlocked block. For more information on viewing locked blocks, refer to chapter 3, section 8, "SubroutineBlocks."

In the configuration software, only the configuration may be verified. No operations on program logic or tables may be performed.

To use the verify function, the programmer must be in **ONLINE** or **MONITOR** mode. Press **Verify (F3)** from the Program Utility Functions menu or from another Program Utilities screen.



The screen shows the currently selected program folder, which cannot be changed.

Four types of data can be compared: program logic, configuration data, reference tables, or all logic blocks. When this screen first appears, the selections for program logic and all logic blocks are set to **Y** (Yes). To compare all of the data, you must change the selection for configuration data and reference tables to **Y** (Yes). To compare only part of the data, select **N** (No) for any of the following:

Field	Description
Current Logic Block	The block currently selected in the program display/edit function.
All Logic Blocks	The main program block and all of the subroutine blocks (all blocks in the program).
Reference Tables	The reference tables for the program.
Configuration	The current configuration.

Press the **Enter** key to begin the verify function. While the verify is occurring, you can press **ALT-A** to abort.

### Note

A running PLC with timers, counter and input data will have many changes in reference table data compared to the programmer folder. For that reason, including Reference Tables when verifying a running PLC will generate many mismatches that are meaningless.

If the information being compared is the same, the screen prompts: "Verify complete; no mismatches detected" and the program logic equality state is set to EQUAL. If the program is not successfully verified, the logic equality state is set to NOT EQUAL.

If mismatches are found, they are listed on the screen. For example:

```

PROGRAM | TABLES | STATUS | | | | SETUP | FOLDER | UTILITY | PRINT
1load 2store 3verify 4 5clear 6 7 8 9 10eprom
Verify complete, mismatches were detected: Press SPACE BAR key to continue ...
>
      V E R I F Y   P L C   W I T H   P R O G R A M M E R

Information to be verified:          CURRENT FOLDER:  LESSON
                                   PLC PROGRAM NAME:  LESSON

CURRENT LOGIC BLOCK  Y (Y,N)          ALL LOGIC BLOCKS  Y (Y,N)
REFERENCE TABLES   N (Y,N)          CONFIGURATION     N (Y,N)

      BLOCK NAME          MEMORY TYPES FAILING TO VERIFY
      -----          -----
           _MAIN          LOGIC
           MAKE_IT        LOGIC

ID:  STOP/NO IO          ONLINE  L4 ACC: WRITE LOGIC  LOGIC NOT EQ
C:\LM90\LESSON          PRG: LESSON  BLK: _MAIN
REPLACE
    
```

Each mismatch displays the name of the logic block and the memory type where the mismatch occurs. Pressing the keyboard space bar will restore the screen to its original form or bring up the next page of mismatch information.

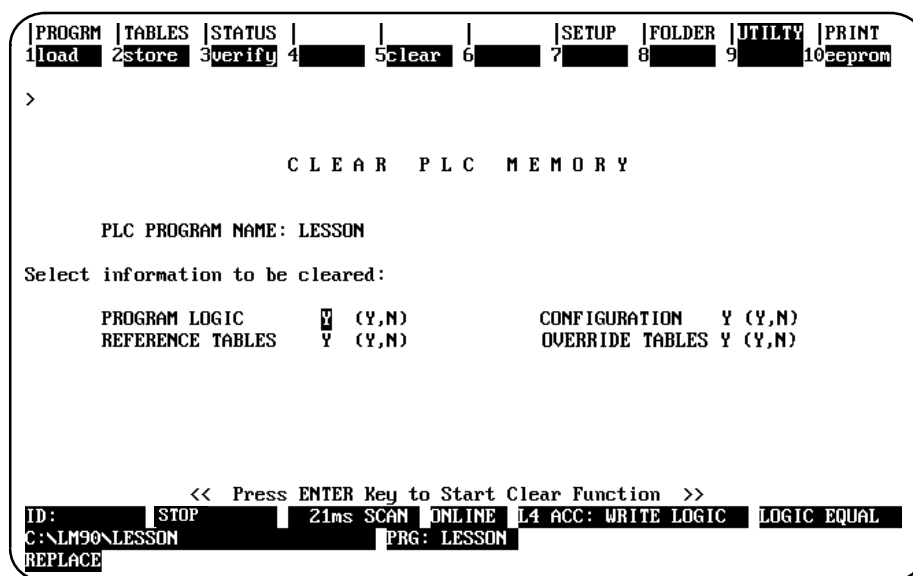
## Clearing PLC Memory

The Clear PLC Memory function gives you the ability to selectively clear the program logic, configuration, reference tables, and/or reference override tables in the attached PLC.

### Note

Only Model 331 and higher CPUs support reference overrides; therefore, this is nonapplicable to Models 211, 311, 313, 321, 323, and the Micro.

This function may be password protected in the PLC. If so, you must know the password to use the function. To use the clear function, the programmer must be in **ONLINE** mode. Press **Clear (F5)** from the Program Utility Functions menu or from another Program Utilities screen to reach the screen shown below:



Four types of data can be cleared: program logic, configuration data, reference tables, and reference override tables. When this screen first appears, the selection for each type is set to **Y** (Yes). To clear all of the data, do not change any of these selections. When all options are set to **Y**, the %S fault bits and overrides are cleared. When only the reference tables option is set to **Y**, %S bits and overrides are **not** cleared. Select **N** (No) for those you do not want to clear.

Field	Description
ProgramLogic	The ladder logic program
Configuration	The current configuration
Reference Tables	The reference tables for the program
Override Tables	The reference override tables for the program

Press the **Enter** key to begin the clear function. **Once the clear operation has begun, it cannot be aborted.** When the clear operation is complete, the software displays the “Clear Complete” message.

## Clear Reference Override Tables

The Clear PLC Memory function gives you the ability to selectively clear the program logic, configuration, and/or reference tables in the attached PLC. Beginning with Release 4.5, you can clear Reference Override Tables as well.

### Special Considerations for Clearing Reference Overrides

The override tables in PLC memory are as follows: Input (%I), Output (%Q), User Internal (%M), and Global (%G). When you select "OVERRIDE TABLES" on the CLEAR PLC MEMORY screen, all reference override tables will be cleared in PLC memory.

### Note

Do not confuse this with clearing the reference tables. To clear reference tables, set the "REFERENCE TABLES" to Y (Yes) as discussed on the previous page.

### Restrictions

To clear the reference override tables, the CPU must be in STOP mode and online with Logicmaster. If you are using passwords with associated privileges, you must have a privilege level of 3 or above to use this feature.

## Read/Write/Verify EEPROM/Flash

The Series 90-30 PLC supports an EEPROM (Electronic Erasable PROM) for storing PLC logic, configuration data, register data, passwords, and the OEM key. This EEPROM is located on the Series 90-30 PLC.

### Note

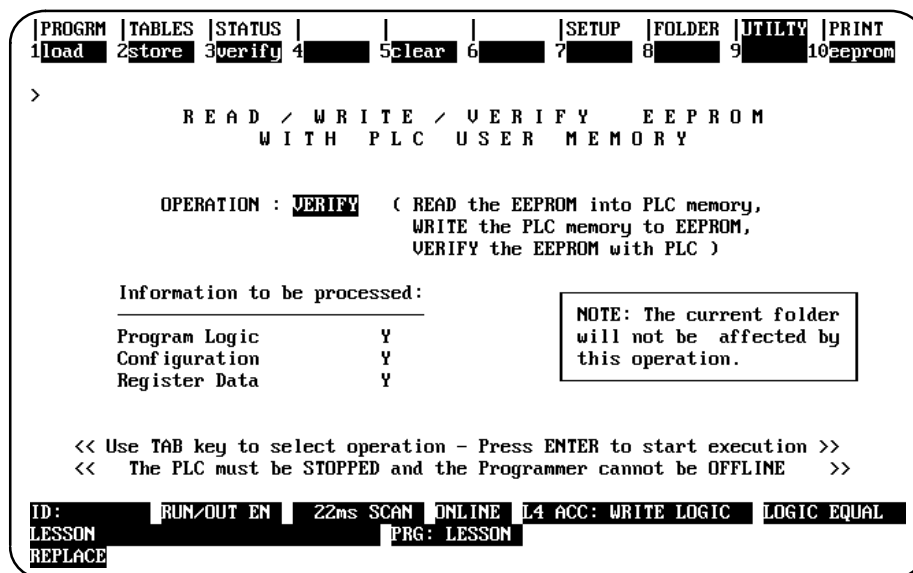
CPUs for the Micro as well as Model 340 and higher 90-30 CPUs have Flash memory instead of EEPROM; however, the Read/Write/Verify process is the same for Flash as for EEPROM. That is, all of the instructions listed below apply to all models having Flash memory, as well as all models having EEPROM.

The read/write/verify EEPROM function of Logicmaster 90-30/20/Micro software enables you to selectively read the EEPROM contents into PLC memory, write the entire contents of PLC memory to the EEPROM, and selectively verify the EEPROM contents with the PLC memory.

### Note

The read/write EEPROM operations require the programmer to be online and the PLC to be stopped. The verify EEPROM operation can be performed while the programmer is in either **ONLINE** or **MONITOR** mode. In addition, these operations cannot be performed if the PLC is in OEM protection mode.

To use this function, press the **EEPROM (F10)** softkey from the Program Utility Functions menu or from another Program Utilities screen.



When the Read/Write/Verify EEPROM screen is first displayed, the **Operation** field defaults to VERIFY. Use the Tab key to toggle through the valid operations, or enter the desired operation directly into the field. Then, select the information to be processed or just press the **Enter** key to start execution.



Operation	Description
Read EEPROM	<p>Selectively read the EEPROM contents into PLC memory. Once the read operation is selected, you may choose the information to be read or use the default options provided:</p> <p><b>Program Logic:</b> The ladder logic program.</p> <p><b>Configuration</b> The current configuration</p> <p><b>Data:</b> %R (%R only) data for 3.52 and earlier CPU versions %R, %M, %Q, %AI, %AQ, and %I for version 4.21 CPUs %R, %M, %Q, %AI, %AQ, %I, and %G for versions 4.40 and later (see Note below).</p> <p>When this screen first appears, all three types of data default to <b>Y</b> (Yes). Enter <b>N</b> (No) for those types of data you do not wish to read. After all the options are selected, press the <b>Enter</b> key (with the PLC connected and stopped). The software will prompt you to confirm this operation. <b><u>Once the read operation has begun, it cannot be aborted.</u></b> When the read operation is complete, the software displays the message "Read from EEPROM completed" to indicate that the read was successful.</p>
Write EEPROM	<p>Write the entire contents of PLC memory into EEPROM. Once the write operation is selected, all of the selective option fields default to <b>Y</b> (Yes); these fields can be changed for the write operation.</p> <p>Press the <b>Enter</b> key (with the PLC connected and stopped) to begin the write operation. <b><u>Once the write operation has begun, it cannot be aborted.</u></b> When the write operation is complete, the software displays the message "Write to EEPROM completed" to indicate that the write was successful.</p>
Verify EEPROM	<p>Selectively verify the EEPROM contents with PLC memory. Once the verify operation is selected, you may choose the information to be read or use the default options provided:</p> <p><b>Program Logic:</b> The ladder logic program.</p> <p><b>Configuration</b> The current configuration</p> <p><b>Data:</b> %R (%R only) data for 3.52 and earlier CPU versions %R, %M, %Q, %AI, %AQ, and %I for version 4.21 CPUs %R, %M, %Q, %AI, %AQ, %I, and %G for versions 4.40 and later (see Note below).</p> <p>When this screen first appears, all three types of data default to <b>Y</b> (Yes). Enter <b>N</b> (No) for those types of data you do not wish to read. After all the options are selected, press the <b>Enter</b> key (with the PLC connected and stopped).</p> <p>If no mismatches are found, the message "Verify EEPROM complete; no mismatches found" is displayed on the message line. If mismatches are found, a corresponding error message is displayed to indicate which option(s) mismatched.</p>

### Note

Prior to version 3.52 of the 90-30, only Register memory (%R) could be read and verified through this utility. Beginning with version 4.40, you can read, and verify %R, %M, %Q, %AI, %AQ, %I, and %G (version 4.21 could handle all but %G memory).

**Caution**

**A read EEPROM/Flash failure clears (resets to the default value) everything that had been selected to be read (program logic, configuration, or register data) regardless of whether or not they were successfully read before the point at which the read failed. This includes SNP ID which has to be set manually. The only exception to this clearing is that passwords are not re-enabled as part of the configuration in the failure.**

**If one of the selected items is the configuration, remember that this includes SNP ID (especially important in a multidrop communications link) and has to be reset manually. Communications parameters such as Baud rate/parity would have to be reset as well. In addition, on a 351 or 352 CPU, all serial port configuration parameters would be lost. The connection to the PLC remains usable until you disconnect. The next attach, however, would be affected by this unless you correct the situation before disconnecting.**

**To correct this problem, try again to store the program or configuration or register data to the PLC. If you had selected configuration when the Store failed, you will also have to set the SNP ID as a separate step.**

# Chapter 9

## Print Functions

---

---

The print function is used to:

- Determine a printing device or file destination for Logicmaster 90-30/20/Micro screen print text.
- Enter printer parameters.
- Print copies of programs, configurations, and reference tables to files or a printer.

### Note

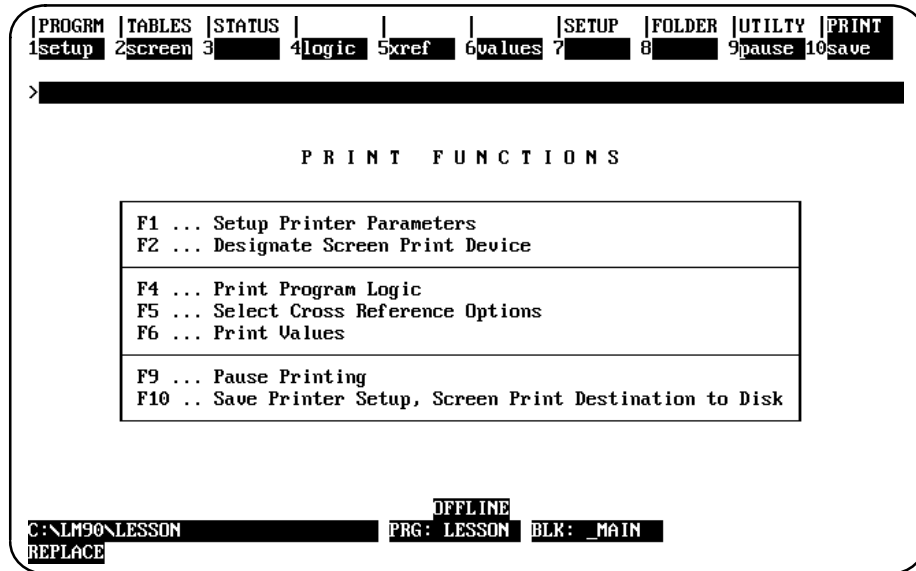
Print may be used to print the values in the program folder only. To print the values from the PLC, the reference values must first be loaded to the folder by using the load utility function (see chapter 8).

Chapter 9 contains the following sections:

Section	Title	Description	Page
1	Printer Parameters	Explains how to check or to change the currently selected printer parameters.	9-3
2	Selecting a Screen Print Device	Explains how to designate a printer or file to receive screen prints.	9-5
3	Print Program	Explains how to print program logic, cross references, variables, and program annotation to a printer or to a file.	9-7
4	Print Reference Tables	Explains how to print reference tables to a printer or to a file.	9-14
5	Print Coil References	Explains how to print coil references to a printer or to a file.	9-16
6	Print Configuration	Explains how to print software configuration data to a printer or a file.	9-18
7	Print Function Examples	Illustrates various hard copy listings which can be produced using the print function.	9-21

## Print Functions Menu

To display the Print Functions menu in the programming software, press **Print (F10)** from the main menu, or **Shift-F10** from any main menu function screen.

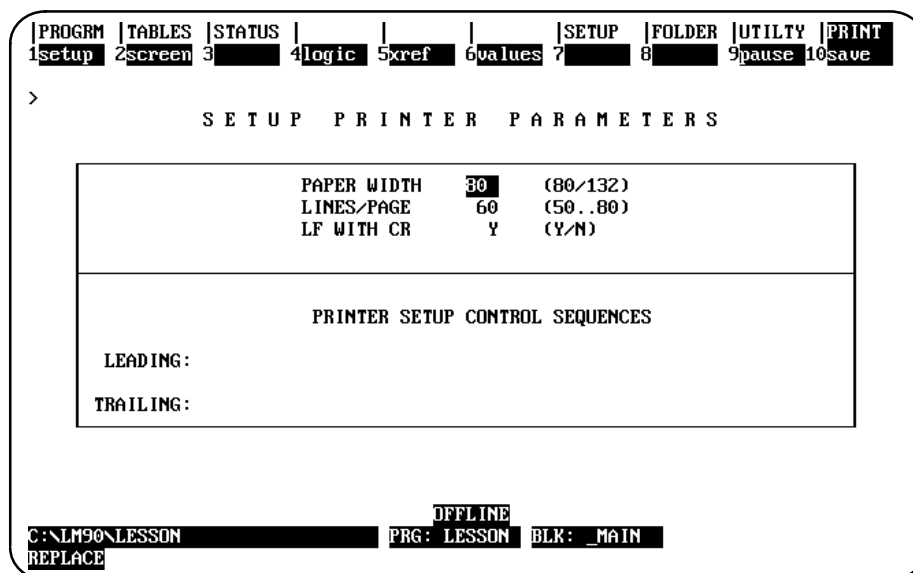


### Caution

**Do not use the same file extensions for printing as are used for the program folder files. Logicmaster 90-30/20/Micro software does not prevent the printing of a listing to a file with the same name as one of the program folder files. If this is done, part of the folder may be lost. Refer to appendix G for a list of file extensions used by the software.**

## Section 1: Printer Parameters

The printer setup parameters used by the programmer must match those of your printer. Default values are supplied. These can be changed and stored in a file for continued use. To change your printer setup or check its contents, press **Setup (F1)** from the Print Functions menu or from another Print Functions screen.



The setup screen shows the value and possible selections for each parameter.

### Changing Setup Printer Parameters

To change entries on the Printer Parameters screen:

1. Move the cursor to the item you want to change.
2. Type (or use the **Tab** key) to enter new values as needed.
3. When you are finished, decide whether your entries or the default values shown above should be used when the system is started.
  - A. To save the printer parameters only until the next power-up (and then return to the default parameters), do not press **Save (F10)**.
  - B. To save the printer parameters, press **Save (F10)**. This causes the printer parameters to be written to the PRINT.SET file. This file is written to the home directory. When the programmer is powered up, these entries will replace the standard default values.

## Printer Parameters

Refer to the following descriptions when changing the entries on the Printer Parameters screen.

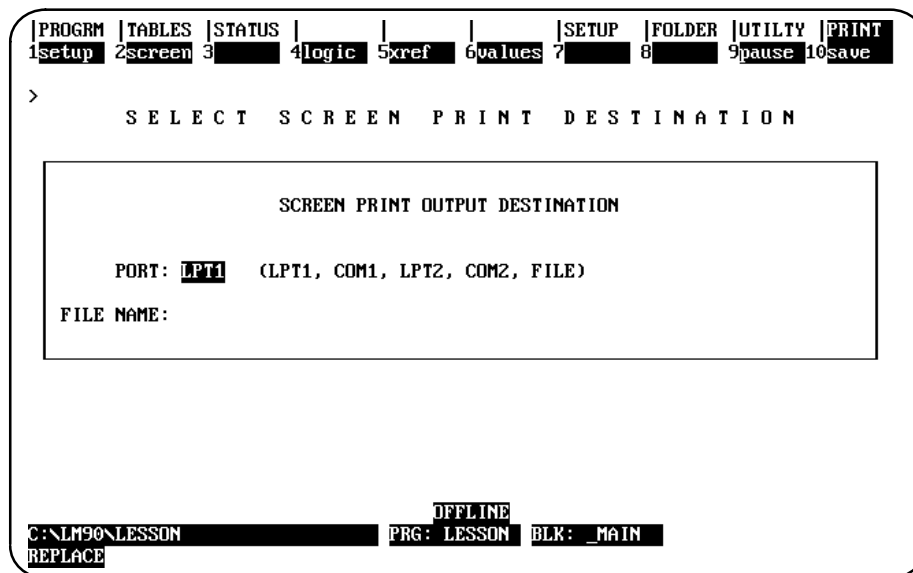
Table 9-1. Printer Parameters

Parameter	Description
Paper Width	<p>The number of characters printed on a line. If the printer is set up for standard 8-1/2 inch wide paper, select 80 characters for the paper width. If the printer uses 11-inch (or wider) paper, you may select either 80 or 132 characters.</p> <p>In 80-character mode, rung numbers and cross references are printed above the rungs. In 132-character mode, the format of the ladder logic is the same. However, rung numbers and cross references are printed to the right of the rungs.</p>
Number of Lines per Page	<p>The number of lines (from 50 to 120) that can be printed on a page (default= 60). If an associated group of lines will not fit on one page, the system will command the printer to advance to the next page after the number of lines specified by this entry.</p> <p><b>Note:</b> Some printers automatically insert a formfeed after printing a certain number of lines (typically, 66). If the printer has this feature, specify a shorter page length to prevent an automatic page eject.</p>
Line Feed with a Carriage Return	<p>The line feed character advances the paper to the next line for printing. This item determines whether the system automatically inserts a line feed (&lt;CR&gt;&lt;LF&gt;) each time the printer head should return to the left page margin.</p> <p><b>Note:</b> Some printers can be set up to automatically advance to the next line by entering only &lt;CR&gt;. Refer to the instructions for your printer to determine how it works.</p> <p>If you do not want the programmer to insert a line feed character after each carriage return character, enter <b>N</b> (No). To have the programmer insert the &lt;LF&gt; character after each &lt;CR&gt; character, enter <b>Y</b> (Yes).</p>
Printer Setup Sequences	<p>If the printer uses leading or trailing control characters, enter the characters here. If you move the cursor to either of these entries, a field will appear where you can enter the appropriate characters.</p> <p>Printer control sequences are issued immediately before and after a listing. This feature can be used to put the printer into a particular mode (for example, compressed output), returning to the original mode after the listing is finished.</p> <p>From 0 to 60 characters can be specified. To enter non-printing characters, use one backslash followed by the three-digit decimal equivalent of the ASCII representative. For example, to identify the Escape character &lt;ESC&gt;, you would enter \027. A leading zero is required. If you need to enter the backslash character itself, enter two backslashes: \\. </p>

## Section 2: Selecting a Screen Print Device

The print screen function can be used to print a copy of any screen in the programmer. To execute a screen print, go to the screen you wish to print and press **ALT-P**. The printing of the screen is sent to the selected device or file. To abort the printing before it completes, press any key.

Screen prints can be sent to an output port or directed to a file. To specify a screen print destination, press **screen (F2)** from the Print Functions menu or from another Print Functions screen.



### Sending Screen Prints to a Printer

If screen prints should be sent to a printer, the selection for the port should match the current printer port. If the port is a serial port (COM1 or COM2), be sure to set the baud rate using the program setup function. To use COM2, the PC must have two serial ports physically present. (This is an MS-DOS restriction.)

To save the port designation, press **save (F10)**. When you exit the screen, the programmer will create a file named SCRPRINT.SET, located in the home directory. This file is read by the programmer at startup. To use the port designation temporarily, exit the screen without pressing **save**. The port selected is used until the programmer is powered down or until changed on this screen.

## Sending Screens to a File

If screen prints should be sent to a file, then the port entry should be "file". You can enter any valid file name for the screen print file. This file is used to store all future screen prints, which are added to the end of the file.

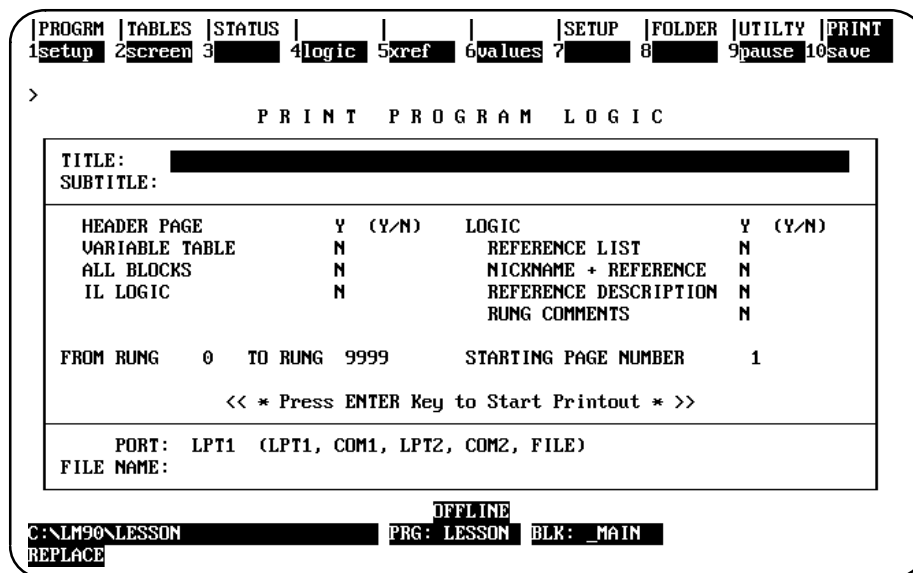
When specifying the screen print device destination to disk, use the full path name. If only a file name is specified, the screen print file is written to the folder that was active when the screen print device was specified; this may or may not be the current folder. Refer to appendix G for a listing of reserved file extensions.

If printing is paused while printing to a floppy disk, the disk swapped for a second disk, and then printing is resumed, neither disk will end up with a usable listing file. In addition, the second disk must be formatted before further use.



## Section 3: Print Program

The print program function is used to print program logic, cross references, variables, and program annotation. To use the print program function, press **Logic (F4)** from the Print Functions menu or from another Print Functions screen.



Specify the content for the printout on this screen as instructed on the following pages. If you do not change a selection, its default is used. If the printout should also include program cross references (which show other program uses of selected references), press **F5** to select the cross references to be included.

To save the content of this screen to a file called PSCREENS.SET, press **Save (F10)**. This file is located in the directory of the active program folder and is read when the print function is entered. If you do not save the content of this screen, the default values will be used the next time you print program logic, cross references, variables, or program annotation.

After completing the selections, press the **Enter** key to start printing. Printing will continue unless paused or aborted. When the printing is complete, the software displays the message "Listing complete".

To temporarily suspend printing after it begins, press **Pause (F9)**. To restart printing, press **F9** again.

An attempt to print to a printer that is out of paper results in the message, "Printing device is offline; press pause to resume," being displayed.

To abort printing while in progress, press **ALT-A** and respond to the prompt. Printing will also abort if you press **CTRL-Break**, which will exit the programming software. When a listing is being printed to a file and the listing is aborted, the file containing the listing up to that point is closed and is not deleted. It contains valid data up to the point where it was stopped.

## Print Program Parameters

Refer to the following definitions when changing the entries on the Print Program page. Examples are provided in section 6 of this chapter.

Table 9-2. Print Program Parameters

Parameter	Description
Title	An optional title of up to 62 characters centered at the top of each page. The title may change if specified in a rung comment.
Subtitle	An optional line of up to 62 characters printed below the title on each page. The subtitle may change if specified in a rung comment.
Header Page	The header page for a program shows the program name, CPU reference sizes, highest references used in the program, and the size of the program, in bytes. To print header pages, enter <b>Y</b> (Yes).
Variable Table	To print tables of variables and identifiers, enter <b>Y</b> (Yes).
AllBlocks	<p>To print all blocks (except those that have been view locked) in the selected program, enter <b>Y</b> (Yes). During listing or printing, the name of the block being printed appears on the status line at the bottom of the screen.</p> <p>Logic for blocks which have been view locked cannot be printed, even if <b>Y</b> (Yes) is selected for <b>AllBlocks</b>. A list of the view locked blocks that could not be printed will be included at the end of the printout. For more information on locked blocks, refer to chapter 3, section 8, "Subroutine Blocks."</p> <p>If <b>AllBlocks</b> is set to <b>N</b> (No), only the current block is printed. The name of the block is displayed on the status line.</p> <p>Whether one block or all blocks are printed, the printout will include the IL Logic and Logic items specified below.</p>
ILLogic	To print Instruction List logic, enter <b>Y</b> (Yes). IL Logic adds Hand Held Programmer key strokes under each rung which is very useful for maintaining the program with the HHP.
Logic	<p>To print ladder diagram logic, enter <b>Y</b> (Yes). Then, specify the content of the logic printout:</p> <p><b>Reference List:</b> Enter <b>Y</b> (Yes) to print a table at the bottom of a page listing all the program references on that page. For each reference, the table lists the machine address, nickname, and reference description.</p> <p><b>Nickname and Reference:</b> Enter <b>Y</b> (Yes) to print both machine addresses and their nicknames with the ladder logic. If <b>N</b> (No) is entered, only nicknames are printed in the logic. Machine addresses are printed for any references that do not have nicknames assigned.</p> <p><b>Reference Description:</b> Enter <b>Y</b> (Yes) to print reference descriptions above the reference addresses at the next print logic request.</p> <p>When the print reference description feature is selected, the first 28 characters of the reference description for each reference address in the logic are printed above the reference address. The reference description is divided into four lines of seven characters each. If the nickname and reference option is also selected, the reference description is printed above the nickname, for a total of six lines above the instruction. If the nickname and reference option is not selected, the reference description is printed directly above the reference address or nickname, for a total of five lines above the instruction. Refer to chapter 3, section 7, "Changing the Display Mode," and chapter 6, "Programmer Setup," for more information on selecting the view modes.</p> <p><b>Rung Comments:</b> Enter <b>N</b> (No) to print comments in instruction form only. If <b>Y</b> (Yes) is entered, rung explanation text is printed in place of the comment instruction.</p> <p>Additional text for rung comments may be created as separate files, as described in chapter 3, section 4, "Program Annotation." If this has been done, be sure the files are present on the specified drive when printing begins.</p>

**Table 9-2. Print Program Parameters (Continued)**

Parameter	Description
From/To Rung	To print a range of logic rungs within the program, enter the first and last rung number of the group to be printed. If the <b>AllBlocks</b> option is set to Y (Yes), these fields will be ignored.
Starting Page Number	To begin numbering the printout with a page number other than 1, enter the number here.
Port	Specify the destination of the listing.
File Name	To send the printout to a file instead of to a printer, enter a name for the file here. A printout file cannot be written to a locked folder. If a file already exists with the selected name, the new file will overwrite it. Refer to appendix G for a listing of the extensions used by Logicmaster 90-30/20/Microsoft software. You do not want to overwrite one of these files with print text.

### Shortcuts for Printing Program Logic

The following suggestions can help you shorten a program logic printout:

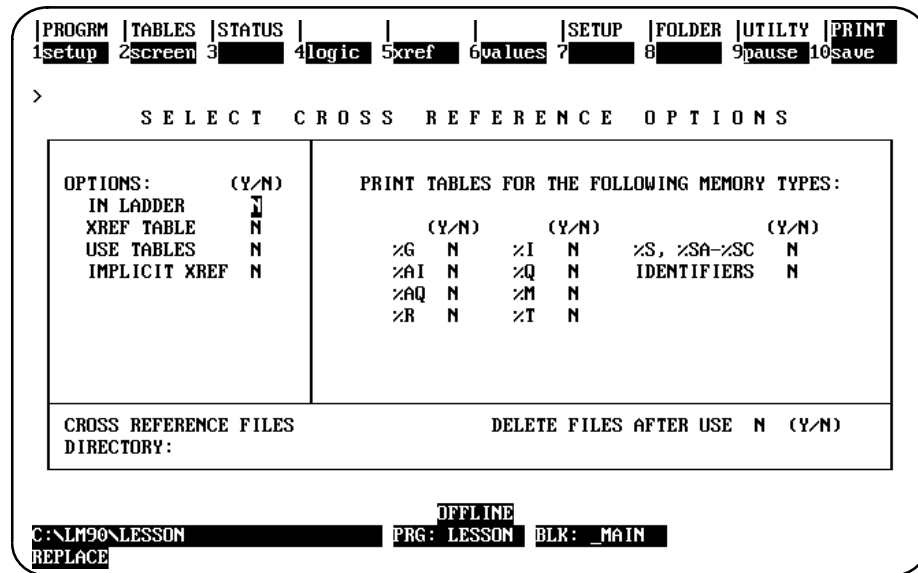
1. Do not select both the reference list and reference description options.
2. If the program contains many reference descriptions, use the reference list option instead of the reference description option.
3. If the program contains few reference descriptions, use the reference description option instead of the reference list option.
4. Use the 132-column page width instead of the 80-column page width. This is very helpful when printing in-ladder cross references with logic.

## Cross References

Cross references in a printout show the use of references in the program. Cross references can be included as part of the ladder logic text and/or listed as separate tables.

If the **AllBlocks** parameter on the Print Program Logic screen is set to **Y** (Yes), the cross reference tables for each block are printed following the logic for that block.

To include cross references, press **F5** from the Print Program screen or from another Print Functions screen.



Specify the content and format of the cross reference printing on this screen. If you do not change a selection, its default is used.

To save the content of this screen to a file called PSCREENS.SET, press **Save (F10)**. This file is located in the directory of the active program folder and is read when the print function is entered. If you do not save the content of this screen, the default values will be used the next time you print program logic, cross references, variables, or program annotation.

Press **Logic (F4)** from this screen to return to the Print Program Logic screen.

The following key can be used for interpreting cross reference instruction symbols:

Symbol	Description
###	Explicitreference.
(###)	Implicitreference.
FBIO	Function block direct reference.
JMP	Jumpcontrol.
LABEL	Label.
MCR	Master controlrelay.
EMCR	End master control relay.
CALL	Block CALLinstruction.

Refer to the following definitions when changing the entries on the Cross Reference screen.

**Table 9-3. Cross Reference Parameters**

Parameter	Description
InLadder	To include cross references as part of the ladder logic, enter <b>Y</b> (Yes). Cross references for each reference address used on the coils in a rung are printed with that rung. In addition, each time the reference address is used on a contact, the most recent rung on which the reference address was used on a coil (referred to as the controlling rung) is listed under the contact.
CrossReference Table	<p>Specify whether to print separate tables of cross references for each block and/or a single set of cross reference tables for the program. The printout contains a table for each reference type which has at least one address reference used anywhere in a program. Within each table, cross reference information is printed in ascending order of reference address offset. For each reference address offset, the cross reference data is grouped by block and, then within each block, by the instructions on which the reference is used. For each instruction, rung numbers are printed in ascending order. For references used more than once on a given rung in the same instruction, the rung number is listed only once.</p> <p><b>PerBlock:</b> If <b>Y</b> (Yes) is selected for <b>PerBlock</b>, sets of unique cross reference tables for each block are printed. A per block cross reference table contains only the cross reference information for references used within the scope of a single block. The default for this option is <b>N</b> (No).</p> <p><b>Global:</b> If <b>Y</b> (Yes) is selected for <b>Global</b>, one set of cross reference tables for the program is printed. A global cross reference table for a reference type will contain all the cross reference information for references of that type used anywhere within the scope of the program. The information for a reference address will consist of each block, rung number, and instruction on which the address was used. The information will also include the unique nickname and reference description assigned to the reference address within each block. The default for this option is <b>N</b> (No).</p> <p>The nickname and reference description defined for a reference address within the scope of a block is printed when the global cross reference data for that reference address and that block is printed.</p> <p>Implicit cross reference data may be included or excluded from either type of cross reference table listing by entering <b>Y</b> or <b>N</b> in the <b>ImplicitXref</b> field. If included, the implicit usage of a reference address by an instruction is printed with parentheses around the rung number of the rung containing that instruction.</p> <p>An identifier name global cross reference table will be printed when the <b>Xref TablesGlobal</b> option is selected. Since the only identifier table names known and used globally in a program are the block names, the global identifier name cross reference table will consist only of the usage information for block names. For each block name, the global cross reference identifier table will list the blocks in which it is called, and the rung numbers within each block of the rungs containing the CALL instruction.</p> <p>Global cross reference tables will be printed after all other logic listing text that you selected has been printed and before the global use tables.</p>

Table 9-3. Cross Reference Parameters (Continued)

Parameter	Description
Use Tables	<p>Specify whether to print separate reference use tables for each block and/or a single set of reference use tables for the program.</p> <p>The symbol <b>-</b> means a reference is not used. The symbol <b>*</b> means used explicitly on an instruction operand. The symbol <b>+</b> means used implicitly by an instruction. When both <b>*</b> and <b>+</b> apply, the symbol <b>#</b> is used. (Refer to the entry for implicit cross references in this table for more information.)</p> <p><b>PerBlock:</b> If <b>Y</b> (Yes) is selected for <b>PerBlock</b>, sets of unique reference use tables for each block are printed. A per block reference use table contains only the reference use information for references used within the scope of a single block. The default for this option is <b>N</b> (No).</p> <p><b>Global:</b> If <b>Y</b> (Yes) is selected for <b>Global</b>, one set of reference use tables for the program is printed. A global reference use table for a reference type will contain reference use information for the entire program. The default for this option is <b>N</b> (No).</p> <p>Implicit reference usage data may be included or excluded from either type of reference use table listing by entering <b>Y</b> or <b>N</b> in the <b>Implicit Xref</b> field.</p>
ImplicitCross Reference	<p>Implicit references are references that are used, but which do not appear directly in the program. For example, if the MOVE_INT instruction has an input %AI001, an output %AQ001, and a length of 5, %AI001 and %AQ001 are explicitly used. Due to the length, there are really 5 words of input/output. The next 4 words are implicitly used.</p> <p>To include both implicit and explicit references in cross reference tables, enter <b>Y</b> (Yes). If <b>N</b> (No) is entered, only explicit references are printed.</p>
Memory Types	<p>Enter <b>Y</b> to select the reference types for which cross reference tables and/or reference use tables will be printed. The tables will begin at reference address offset 1 and end with the highest used reference address offset of each reference address type in the program or block. Enter <b>N</b> for those references whose cross reference tables and/or reference use tables will not be printed.</p> <p>To include cross references of block names, and MCR, ENDMCR, JUMP, and LABEL names in a cross reference table printout, enter <b>Y</b> (Yes) in the <b>Identifiers</b> field.</p>

Table 9-3. Cross Reference Parameters (Continued)

Parameter	Description
Directory	<p>Specify a directory to store cross reference data files.</p> <p>The specified location for the .XRF, PRINT.XOV, and GLOBAL.PTX cross reference data files must be either a hard disk, an expanded memory device, or a RAM Disk. The cross reference files cannot be sent to a floppy disk or a locked folder. In either case, an error message is displayed. If no entry is made here, the files are stored in the current folder. Refer to appendix G for extensions used by Logicmaster 90-30/20/Microsoft software. Do not overwrite one of the Logicmaster 90 files with print text.</p> <p><b>Note:</b> Cross reference data files are not automatically deleted when exiting the print function. The .XRF and PRINT.XOV files remain in the designated directory unless you enter <b>Y</b> (Yes) in the <b>Delete Files After Use</b> field. This prevents having to cross reference the same blocks each time a listing is started. When subsequent listings from the same folder are generated, only the cross reference data files for the blocks which have changed since the previous listing are deleted and recreated. The default selection for the <b>Delete Files After Use</b> field is <b>N</b> (No).</p> <p>In addition to the .XRF and PRINT.XOV files, a temporary file named GLOBAL.PTX is created. The file will exist during a listing that includes global cross reference text; it will be deleted when the global cross reference listing terminates successfully, terminates with an error, or aborts.</p>
Delete Files After Use	<p>To delete the cross reference files after the listing completes, enter <b>Y</b> (Yes). If you want the cross reference files to remain on disk, enter the default <b>N</b> (No).</p> <p>To delete cross reference files from a folder without printing a listing, use the selected clear folder function.</p>

## Section 4: Print Reference Tables

To print reference table values (excluding coil references) for the program, press **Values** (F6) from the Print Functions menu or from another Print Functions screen. (See the next section for printing coil references.)

### Note

Print may be used to print the values in the program folder only. To print the values from the PLC, the reference values must first be loaded to the folder by using the load utility function (see chapter 8).

```

|PROGRAM |TABLES |STATUS | | | |SETUP |FOLDER |UTILITY |PRINT
|1setup  |2screen  |3      | | | |7      |8      |9pause  |10save
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----
>
                P R I N T   V A L U E S
-----
TITLE: ████████████████████████████████████████████████████████████████████████████
SUBTITLE: ████████████████████████████████████████████████████████████████████████████
-----
OPTIONS:      (Y/N)
HEADER PAGE   Y
VALUE TABLES Y
OVERRIDES    N
-----
STARTING PAGE # 1
-----
                << * Press ENTER Key to Start Printout * >>
-----
                PORT: LPT1 (LPT1, COM1, LPT2, COM2, FILE)
                FILE NAME:
-----
                OFFLINE
C:\LM90\LESSON ██████████ PRG: LESSON BLK: _MAIN
REPLACE
  
```

The listing can be directed to a printer or to a file.

These tables correspond to the tables that can be displayed and formatted using the reference tables function. For example:

```

                ***** I N P U T   V A L U E   T A B L E *****
REF.   |64   |56   |48   |40   |32   |24   |16   |8   |1
ADDRESS +-----+-----+-----+-----+-----+-----+-----+-----+
%I0064 01101001 00001000 00010101 01011011 00010000 11100100 11010100 01001110
%I0128 00001101 11101111 01100100 01111011 01000011 11001010 00010110 00011000
%I0192 01011111 11110010 00111000 11100011 00010001 10101011 00000000 00000000
%I0256 00000000 00000101 00010010 00011100 00000000 00101101 01011011 10011100
  
```

Specify the content of the printout on the Print Values screen.



**Table 9-4. Value Table Parameters**

Field	Description
Title	An optional title of up to 62 characters centered at the top of each page. The title may change if specified in a rung comment.
Subtitle	An optional line of up to 62 characters printed below the title on each page. The subtitle may change if specified in a rung comment.
Header Page	The header page for a program shows the program name, CPU reference sizes, highest references used in the program, and the size of the program, in bytes. To print header pages, enter <b>Y</b> (Yes).
Value Tables	To print value tables (input, output, etc.), enter <b>Y</b> (Yes). Value tables are printed in groups of three lines at a time.
Overrides	To print the override value tables associated with selected %I, %Q, and %M references, enter <b>Y</b> (Yes).
Starting Page	To begin numbering the printout with a page number other than 1, enter the number here.
AddressRange	Specify the range of values to be printed. Enter values only for types of references to be printed.  If any references should not be printed, enter zeros in both the <b>FROM</b> and <b>TO</b> fields. If the configured limits change (i.e., by changing folders, clearing a folder, or downloading from the PLC), the address ranges will be re-initialized.
Port	Specify the destination of the listing.
File Name	To send the printout to a file instead of to a printer, enter a name for the file here. A printout file cannot be written to a locked folder. If a file already exists with the selected name, the new file will overwrite it. Refer to appendix G for a listing of the extensions used by Logicmaster90-30/20/Microsoftware. You do not want to overwrite one of these files with print text.

To save the content of this screen to a file called PSCREENS.SET, press **save (F10)**. This file is located in the directory of the active program folder and is read when the print function is entered. If you do not save the content of this screen, the default values will be used the next time you print program logic, cross references, variables, or program annotation.

After completing the selections, press the **Enter** key to start printing. Printing will continue unless paused or aborted. When the printing is complete, the software displays the message "Listing is complete".

To temporarily suspend printing after it begins, press **Pause (F9)**. To restart printing, press **F9** again.

During printing, press **ALT-A** to abort the printout. Printing will also abort if you press **CTRL-Break** and confirm your action. This will exit the programming software.

### Note

When a listing is being printed to a file and the listing is aborted, the file containing the listing up to that point is closed and is not deleted. It contains valid data up to the point where it was stopped.

## Section 5: Print Coil References

You can print reference descriptions for coils through the new parameter “COIL REF DESCRIPTION” as shown below:

```

|PROGRAM |TABLES |STATUS | | | |SETUP |FOLDER |UTILITY |PRINT
1|setup 2|screen 3| 4|logic 5|pref 6|values 7| 8| 9|pause 10|save
>
          PRINT PROGRAM LOGIC
-----
| TITLE: Lesson on Coil Reference
| SUBTITLE:
|-----
| HEADER PAGE          Y (Y/N)  LOGIC          Y (Y/N)
| VARIABLE TABLE     N          REFERENCE LIST  N
| ALL BLOCKS          N          NICKNAME + REFERENCE N
| IL LOGIC             N          REFERENCE DESCRIPTION Y
|                                     COIL REF DESCRIPTION Y
|                                     RUNG COMMENTS      N
| FROM RUNG 0 TO RUNG 9999  STARTING PAGE NUMBER 1
|-----
| << * Press ENTER Key to Start Printout * >>
|-----
| PORT: LPT1 (LPT1, COM1, LPT2, COM2, FILE)
| FILE NAME:
|-----
|                                     OFFLINE
| C:\LM90\LESSON  PRG: LESSON  BLK: MAIN
| REPLACE
  
```

The default for this print parameter is No. The “REFERENCEDescription” option controls the printing of reference descriptions for all other references. The sample section from a 132-column print (shown below) resulted from the settings shown above. See Note below for information about the differences between 80- and 132-column print.

```

|AUTOPB LISUP %Q0038 %Q0012 %I0013 EMERGST %Q0006 %Q0075 %Q0005 << RUNG 4 STEP #0001 >>
+--] [-----] [-----] [-----] [---+---]/[-----]/[-----] [-----] [-----] ( )--
|
| %Q0005
+--]/[-----]-----+
|
|EMSTOP %I0100 CLAMPED %I0073 EMERGST << RUNG 5 STEP #0011 >>
+--] [---+---] [---+---] [---+---] [-----] ( )-- (* Emergency Return *)
|
| EMERGST|UNISUP |
+--]/[-----]/[---+
|
| %Q0005 %Q0012 %I0021 %I0022 %M0001 << RUNG 6 STEP #0020 >>
+--]/[-----]/[-----] [-----] [-----] ( )--
  
```

### Note

For 132-column listings, the coil reference descriptions print in a single line to the right of each coil’s reference address as shown above. When set for an 80-column listing, the coil reference descriptions print in the standard LM90 reference description form of four lines above each reference address. See the table on the next page for additional considerations.

The following table is provided to explain the options you now have for printing references.

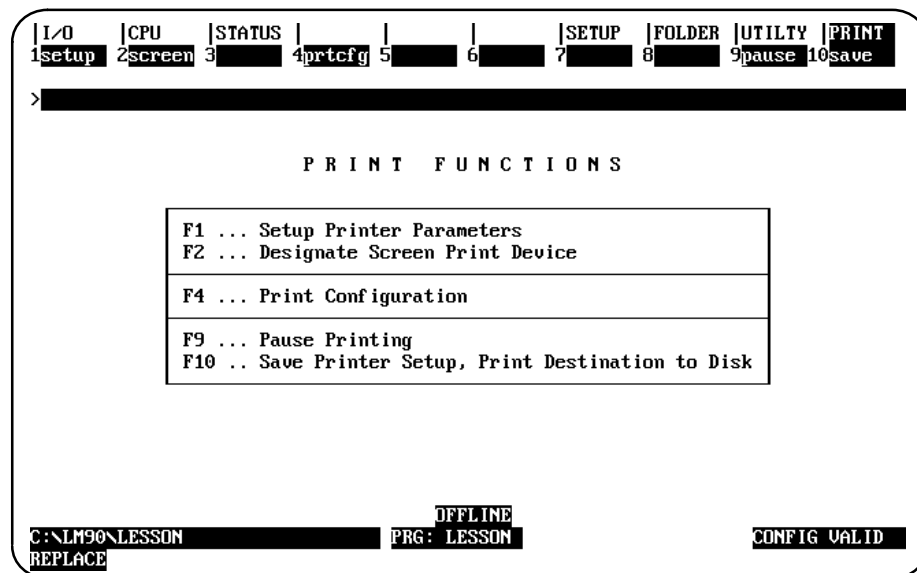
<b>If you set the REFERENCE DESCRIPTION print option to:</b>	<b>And the COIL REF DESCRIPTION print option to:</b>	<b>The results are:</b>
Y	N	Prints reference descriptions above contacts and instruction operands, but it does not print coil reference descriptions.
N	Y	Prints reference descriptions for coils only.
Y	Y	Prints reference descriptions for all references.

## Section 6: Print Configuration

In the Logicmaster 90-30/20/Micro configuration software package, the print function enables you to obtain a listing of all I/O and CPU configuration data. The listing can be directed to a port for immediate printout, or to a file for later printing. The following functions are provided:

- View/change printer parameters.
- Select the screen print destination.
- Generate a rack hardware configuration listing.
- Generate a configured reference address listing (Ref View).
- Generate a CPU configuration listing.

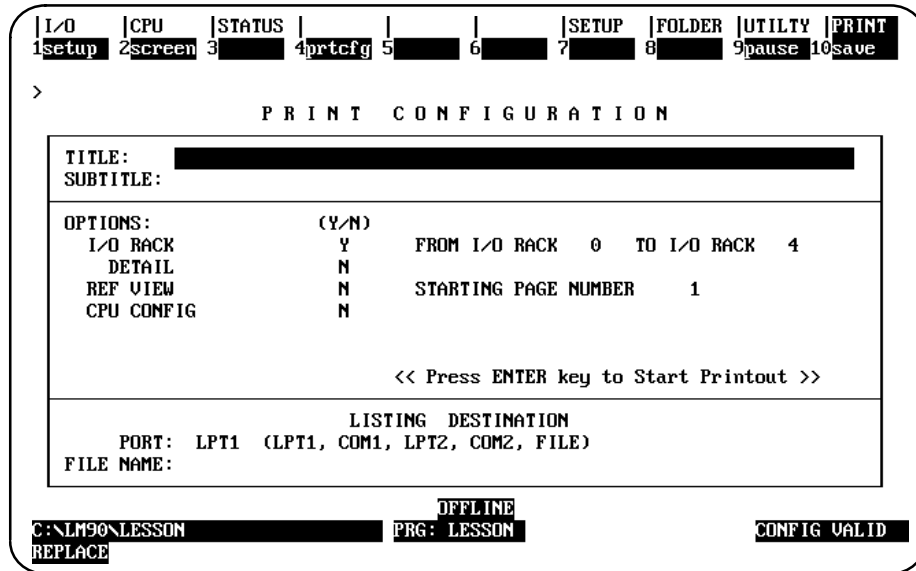
To display the Print Functions menu in the configuration software, press **Print (F10)** from the main menu, or **Shift-F10** from any main menu function screen.



The Setup (F1) and Screen (F2) keys perform the same functions as their corresponding keys in the programming software. Refer to section 1 of this chapter for information on setting up the printer parameters, and to section 2 for information on selecting the screen print output destination.

## Accessing the Print Configuration Screen

To use this function, press **Print Configuration (F4)** from the Print Configuration menu. The following screen will be displayed:



The Print Configuration menu defines which parts of the rack configuration should be printed out and the destination of the listing.

To save the content of this screen to a file called PSCREENS.SET, press **save (F10)**. This file is located in the directory of the active program folder and is read when the print function is entered. If you do not save the content of this screen, the default values will be used the next time you print program logic, cross references, variables, or program annotation.

Refer to the following definitions when changing the entries on the Print Configuration screen.

**Table 9-5. Print Configuration Parameters**

<b>Field</b>	<b>Description</b>
Title	An optional title of up to 62 characters centered at the top of each page.
Subtitle	An optional line of up to 62 characters printed below the title on each page.
I/O Rack	To print rack information, enter <b>Y</b> (Yes).
Detail	To print the detailed configuration screens, enter <b>Y</b> (Yes). This field is only available when the selection for the <b>I/O Rack</b> field is <b>Y</b> .
Reference View	To print the reference view tables, enter <b>Y</b> (Yes).
CPU Configuration	To print CPU configuration data, enter <b>Y</b> (Yes).
Starting Page Number	To begin numbering the printout with a page number other than 1, enter the number here. The value may range from 1 to +32,767.
From I/O Rack	Enter the number of the first rack whose configuration data is to be printed. The value of this field may not exceed the value of the <b>To I/O Rack</b> field.
To I/O Rack	Enter the number of the last rack whose configuration data is to be printed. The value of this field may not be less than the value of the <b>From I/O Rack</b> field. In order to print a single rack of information, both the <b>From I/O Rack</b> and <b>To I/O Rack</b> fields must contain the same value.
Port	Specify the destination of the listing.
File Name	If "file" is selected for the destination in the <b>Port</b> field, enter the name of the file to which the printout is to be directed. Any valid file specification may be entered in this field. A new file is always created, overwriting any existing file of the same name.

## Pagination Guidelines

Only one type of screen can be printed on a given page, but multiple detail screens may be printed on a single page if they will fit on that page following these guidelines:

**Table 9-6. Pagination Guidelines**

<b>Field</b>	<b>Description</b>
Rack	The printing of a given rack cannot be broken across pages. Only one rack per page is permitted.
Detail	The printing of a given detail screen cannot be broken across pages. Because the length of a listing page may be changed and there are a variable number of detail screens for a given module, all the screens of a particular module may not be displayed on the same page.

---

## *Section 7: Print Function Examples*

The following examples illustrate various hard copy listings which can be produced with the print function.

Certain information is presented on every page, regardless of what specific item is currently being printed. The first line of each page of the listing will contain the date, time, software version number, and page number. The user-supplied title and subtitle are centered on the second and third lines of the page, respectively. The program name appears in the bottom left corner of the page. The current program folder specification appears in the bottom center of the page.

The header page is printed when **Y** (Yes) is selected for the header page option on the Print Program Logic screen. The header page for the program includes the program name, the configured reference sizes supported by the host PLC, the highest reference address used in the program, and the size of the program in bytes. This is an example program header page:

```

01-15-93 12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00)      Page 1

      GGGG EEEEE      FFFFF AAA N N U U CCCC
      G E             F A A NN N U U C
      G GGG EEEEE     FFF AAAAA N N N U U C
      G G E           F A A N NN U U C
      GGG EEEEE      F A A N N UUU CCCC

      AAA U U TTTT   OOO M M AAA TTTT IIIII OOO N N
      A A U U T O O MM MM A A T I O O NN N
      AAAAA U U T O O M M M AAAAA T I O O NN N
      A A U U T O O M M A A T I O O NN N
      A A UUU T OOO M M A A T IIIII OOO N N

(*****
(*)
(*) Program: NEW (*)
(*)
(*) PLC PROGRAM ENVIRONMENT HIGHEST REFERENCE USED (*)
(*) ----- (*)
(*) INPUT (%I): 512 INPUT: %I0100 (*)
(*) OUTPUT (%Q): 512 OUTPUT: %Q0061 (*)
(*) INTERNAL (%M): 1024 INTERNAL: %M0006 (*)
(*) GLOBAL DATA (%G): 1280 GLOBAL DATA: NONE (*)
(*) TEMPORARY (%T): 256 TEMPORARY: NONE (*)
(*) REGISTER (%R): 2048 REGISTER: %R0003 (*)
(*) ANALOG INPUT (%AI): 128 ANALOG INPUT: NONE (*)
(*) ANALOG OUTPUT (%AQ): 64 ANALOG OUTPUT: NONE (*)
(*)
(*) PROGRAM SIZE (BYTES): 208 (*)
(*)
(*)
(*****

Program: NEW D:\ACME\CONVEYOR\NEW

```



This example printout was generated by selecting only **Y** (Yes) for the IL Logic option.

```
01-15-93 12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00)      Page 1

<< START OF BLOCK _MAIN >>

<< RUNG 4  STEP #0001 >>
#0001 01  NOOP

<< RUNG 5  STEP #0002 >>
#0002 LD      %I0004
#0003 AND     %Q0002
#0004 AND     %Q0044
#0005 AND     %Q0012
#0006 OR      NOT %Q0005
#0007 AND     NOT %I0003
#0008 AND     NOT %Q0025
#0009 AND     NOT %Q0006
#0010 AND     %Q0061
#0011 OUT     %Q0005

<< RUNG 6  STEP #0012 >>
#0012 LD      %I0002
#0013 LD      %I0100
#0014 OR      NOT %Q0025
#0015 AND     BLK
#0016 LD      %Q0020
#0017 OR      NOT %Q0004
#0018 AND     BLK
#0019 AND     %I0073
#0020 OUT     %Q0025

<< RUNG 7  STEP #0021 >>
#0021 LD      NOT %Q0005
#0022 AND     NOT %Q0012
#0023 LD      NOT %M0005
#0024 AND     %M0006
#0025 OR      BLK
#0026 AND     %I0021
#0027 AND     %I0022
#0028 OUT     %M0001

<< RUNG 8  STEP #0029 >>
#0029 LD      NOT %Q0044
#0030 AND     %M0002
#0031 FUNC   60  ADD
                P1: %R0001
                P2: %R0002
                P3: %R0003
#0032 OUT     %Q0006
#0033 END OF PROGRAM

Program: NEW                      D:\ACME\CONVEYOR\NEW                      Block: _MAIN
```

In this example printout, Logic is set to **Y** (Yes), and all Logic options (Reference List, Nickname and Reference, Reference Description, and Rung Comments) are set to **N** (No).

```

01-15-93  12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00)      Page    1
                TITLE APPEARS HERE
                SUBTITLE APPEARS HERE

[ START OF LD PROGRAM SAMPLE ]      (* This is a sample program      *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]

<< RUNG 4 STEP #0001 >>
I140_00 I141_07 SMP_PMP                                     %T0086
+----] [----+] [----+] [----+]----- (S)-----
          |
          | B152_00
          +----] [----+----- B152_00
                                     (S)-----

<< RUNG 5 STEP #0008 >>
I140_00 I141_07                                     SMP_PMP
+----] [----+] [----+]----- ( )-----
          |
          | B152_00
          +----] [----+-----
                                     |
                                     +-----+
                                     | ONDTR |
                                     | 0.10s |
                                     +-----+
                                     |
                                     +-----+
                                     | R      |
                                     +-----+
I140_01
+----] [----+]----- CONST -+PV
                                     +00100
                                     |
                                     +-----+
                                     | %R0004 |
                                     +-----+

(* COMMENT *)

<< RUNG 7 STEP #0016 >>
I141_01
+----] [----+]----- B152_00
                                     (R)-----

<< RUNG 8 STEP #0018 >>
PB_SUM
+----] [----+]----- %T0075
                                     ( )-----
BAD_RAM
+----] [----+]----- %T0107
                                     ( )-----

[ END OF PROGRAM LOGIC ]

Program: SAMPLE                      D:\30FOLDERS\SAMPLE                      Block: _MAIN
  
```

In this example printout, the Nickname and Reference option is set to Y (Yes) and the Reference Description option is set to N (No). Both a nickname, if defined, and a reference address are printed on an instruction operand.

```

01-15-93 12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00)      Page 1
                               TITLE APPEARS HERE
                               SUBTITLE APPEARS HERE

[ START OF LD PROGRAM SAMPLE ]          (* This is a sample program *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]

<< RUNG 4 STEP #0001 >>

I140_00 I141_07 SMP_PMP
%I0501 %I0505 %M0997                                     %T0086
+-----] [-----] [-----] [-----] [-----] (S)-----

      B152_00
      %M0627
+-----] [-----] [-----] [-----] [-----] (S)-----

<< RUNG 5 STEP #0008 >>

I140_00 I141_07 SMP_PMP
%I0501 %I0505                                     +-----+
+-----] [-----] [-----] [-----] [-----] +ONDTR+ SMP_PMP
      B152_00                                     0.10s %M0997
      %M0627                                     +-----+
+-----] [-----] [-----] [-----] [-----] +R ( )-----

140_01
%I0502
+-----] [-----] [-----] [-----] [-----] +CONST+ +PV
      +00100                                     +-----+
                                               +-----+
                                               %R0004

(* COMMENT *)

<< RUNG 7 STEP #0016 >>

I141_01
%I0504                                     B152_00
+-----] [-----] [-----] [-----] [-----] (R)-----

<< RUNG 8 STEP #0018 >>

PB_SUM
%SA001                                     %T0075
+-----] [-----] [-----] [-----] [-----] ( )-----

BAD_RAM
%SB010                                     %T0107
+-----] [-----] [-----] [-----] [-----] ( )-----

Program: SAMPLE          D:\30FOLDERS\SAMPLE          Block: _MAIN

```

In this example printout, the Nickname and Reference option is set to **N** (No) and the Reference Description option is set to **Y** (Yes). With the Reference Description option enabled, a four-line reference description, if defined, is printed above each instruction operand. Since the Nickname and Reference option is not selected, only a single line of either a nickname or a reference address is printed on an instruction operand.

```

01-15-93 12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00) Page 1
          TITLE APPEARS HERE
          SUBTITLE APPEARS HERE

[ START OF LD PROGRAM SAMPLE ] (* This is a sample program *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]

<< RUNG 4 STEP #0001 >>

      Intake Sump
      Valve Pump 1
      Control Density
      Switch meter
I140_00 I141_07 SMP_PMP
+---] [---] [---] [---]
      Bulb Bulb
      B152 B152
      Circ 00 Circ 00
      Switch Switch
      B152_00 B152_00
+---] [---]
      (S)

<< RUNG 5 STEP #0008 >>

      Intake Sump
      Valve Pump 1
      Control Density
      Switch meter
I140_00 I141_07
+---] [---] [---]
      Bulb Bulb
      B152 B152
      Circ 00 Circ 00
      Switch Switch
      B152_00 B152_00
+---] [---]
      Air Intake Sump
      Valve Pump 1
      Switch meter
      I140_01
+---] [---]
      CONST +PV
      +00100
      Seconds
      to chk
      complet
      ion
      %R0004

Program: SAMPLE D:\30FOLDERS\SAMPLE Block: _MAIN

```

This example printout was generated by selecting Y (Yes) for the Cross Reference Table and Implicit Cross Reference options, and for %M on the Select Cross Reference Options screen.

```

01-15-93 12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00)      Page 1

          ***** I N T E R N A L *****
          ***** C R O S S   R E F E R E N C E   T A B L E *****

REFERENCE  NICKNAME  REFERENCE DESCRIPTION / CROSS REFERENCES
-----
%M0247    : MSTP_14  ::STEP 14 MAIN BIT SEQ
              -] [- 287
              FBIO (296)

%M0248    : MSTP_15  ::STEP 15 MAIN BIT SEQ
              -] [- 287, 294, 295, 512
              FBIO (296)

%M0249    : MSTP_16  ::STEP 16 MAIN BIT SEQ
              -] [- 287, 293, 480, 481, 482, 483, 508, 513, 514, 516,
              519, 520, 521, 523, 526, 527, 528, 530, 533, 534,
              535, 537, 540, 553
              FBIO (296)

%M0250    : MSTP_17  ::STEP 17 MAIN BIT SEQ
              -] [- 227, 228, 229, 288, 293, 591, 592
              FBIO (296)

%M0251    : MSTP_18  ::STEP 18 MAIN BIT SEQ
              -] [- 262, 263, 288, 291
              FBIO (296)

%M0252    : MSTP_19  ::STEP 19 MAIN BIT SEQ
              -] [- 232, 265, 288, 487, 641
              FBIO (296)

%M0253    : MSTP_20  ::STEP 20 MAIN BIT SEQ
              -] [- 275, 289, 487
              FBIO (296)

%M0254    : MSTP_21  ::STEP 21 MAIN BIT SEQ
              -] [- 289, 588
              FBIO (296)

%M0255    : MSTP_22  ::STEP 22 MAIN BIT SEQ
              -] [- 267, 269, 271, 273, 289, 590
              -]/[- 268, 270, 272, 274
              FBIO (296)

%M0256    : MSTP_23  ::STEP 23 MAIN BIT SEQ
              -] [- 156, 289
              FBIO (296)

%M0257    : MSTP_24  ::STEP 24 MAIN BIT SEQ
              -] [- 154, 289
              FBIO (296)

Program: SAMPLE                      D:\30FOLDERS\SAMPLE                      Block: _MAIN

```



This example printout was generated by selecting Y (Yes) for the Value Tables option and setting the %M reference address range to 1 to 1024 on the Print Values screen.

```

01-15-93 12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00)      Page 1

          ***** INTERNAL VALUE TABLE *****

REF.    |64    |56    |48    |40    |32    |24    |16    |8    |1
ADDRESS +-----+-----+-----+-----+-----+-----+-----+-----+
%M0064  00000000 00000100 00000000 00000000 00000000 00011000 00000000 00010010

          ***** INTERNAL VALUE TABLE *****

REF.    |64    |56    |48    |40    |32    |24    |16    |8    |1
ADDRESS +-----+-----+-----+-----+-----+-----+-----+-----+
%M0064  00000000 00000100 00000000 00000000 00000000 00011000 00000000 00010010
%M0128  00000000 00000011 10010100 00000000 00000000 00000000 00001100 00000000
%M0192  00000000 00000100 00000110 00000000 00000000                0008 00000000

%M0256  00000000 00000100 00000000 00000000 00000000 00100101 01110011 00001111
%M0320                +00724 00000001 10000000 01000111 01010000 01000000 10000000
%M0384  00000000 00000100 00000000 01000000 00000001 00010000 01000001 10010000

%M0448  00000000 00000000 00000000                6179 00000001 11011100 10000100
%M0512  00000001 00000100 00000000 00000000 00000001 01010010 00000000 00000100
%M0576  00000000 00000000 00001111 00000000 00000000 00000010 00010001 00000000

%M0640  00000000 00000000 00000000 00000000 00000000                +09232 00000100
%M0704  00000000 00000000 00000000 00000000 00000000 00000100 10101100 10011000
%M0768  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

There are no non-zero values in the range from %M0769 to %M1024.

Program: SAMPLE                      D:\30FOLDERS\SAMPLE

```

This example printout was generated by selecting **Y** (Yes) for the Overrides option and setting the %M reference address range to 1 to 1024 on the Print Values screen.

```

01-15-93 12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00)      Page 1

***** INTERNAL OVERRIDE TABLE *****

REF. ADDRESS |64 |56 |48 |40 |32 |24 |16 |8 |1
+-----+-----+-----+-----+-----+-----+-----+-----+
%M0064 -----
%M0128 -----0-
%M0192 -----01--
%M0256 -----0-1--
%M0320 -----1--
%M0384 -----00 0-----0-----1-----
%M0448 -----0-----
%M0512 -----00000000 000--0-- --0--- -10-0---
%M0576 -----0-----
%M0640 -----
%M0704 -----0-----
%M0768 -----0--0-
%M0832 -----
%M0896 -----
%M0960 -----
%M1024 -----

- = not overridden    0 = overridden in OFF state    1 = overridden in ON state

Program: SAMPLE                      D:\30FOLDERS\SAMPLE
  
```



This example printout of an 80-column listing of a short program was generated by selecting Y (Yes) for the IL Logic, Logic, and In Ladder Cross Reference options. IL Logic adds Hand Held Programmer instructions under each rung, required to maintain the program with an HHP.

```

01-15-93 12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00)      Page 1
                                TITLE APPEARS HERE
                                SUBTITLE APPEARS HERE

[ START OF LD PROGRAM NEW ] (* *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]

(* COMMENT *)
#0001 01 NOOP

<< RUNG 5 STEP #0002 >> Cross reference for %Q0005
-)/[- 5, 7

%I0004 %Q0002 %Q0044 %Q0012 %I0003 %Q0025 %Q0006 %Q0061 %Q0005
+---] [-----] [-----] [-----] [-----] / [-----] / [-----] [-----] ( )---
|
%Q0005
+---] / [-----] +
|
0005

#0002 LD %I0004
#0003 AND %Q0002
#0004 AND %Q0044
#0005 AND %Q0012
#0006 OR NOT %Q0005
#0007 AND NOT %I0003
#0008 AND NOT %Q0025
#0009 AND NOT %Q0006
#0010 AND %Q0061
#0011 OUT %Q0005

<< RUNG 6 STEP #0012 >> Cross reference for %Q0025
-)/[- 5, 6

%I0002 %I0100 %Q0020 %I0073 %Q0025
+---] [-----] [-----] [-----] [-----] ( )---
|
| %Q0025 | %Q0004 |
+---] / [-----] / [-----] +
|
0006

#0012 LD %I0002
#0013 LD %I0100
#0014 OR NOT %Q0025
#0015 AND BLK
#0016 LD %Q0020
#0017 OR NOT %Q0004
#0018 AND BLK
#0019 AND %I0073
#0020 OUT %Q0025

Program: NEW D:\ACME\CONVEYOR\NEW Block: _MAIN

```

```

01-15-93 12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00)      Page 2
                               TITLE APPEARS HERE
                               SUBTITLE APPEARS HERE

<< RUNG 7 STEP #0021 >>      Cross reference for %M0001
                               NONE

%Q0005 %Q0012 %I0021 %I0022                                     %M0001
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0005 |                                     |                                     |
| %M0005 %M0006 |                             |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

#0021 LD NOT %Q0005
#0022 AND NOT %Q0012
#0023 LD NOT %M0005
#0024 AND %M0006
#0025 OR BLK
#0026 AND %I0021
#0027 AND %I0022
#0028 OUT %M0001

<< RUNG 8 STEP #0029 >>      Cross reference for %Q0006
                               -]/[- 5

%Q0044 %M0002 +-----+                                     %Q0006
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | ADD |                                     |
|                                     | INT |                                     |
| %R0001 --+I1 Q+--%R0003 |         |                                     |
| %R0002 --+I2 |         |                                     |
|                                     |-----+                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

#0029 LD NOT %Q0044
#0030 AND %M0002
#0031 FUNC 60 ADD
           P1: %R0001
           P2: %R0002
           P3: %R0003
#0032 OUT %Q0006

[      END OF PROGRAM LOGIC      ]

#0033 END OF PROGRAM

```

Program: NEW

D:\ACME\CONVEYOR\NEW

Block: \_MAIN

This example printout of an 80-column listing was generated by selecting **Y** (Yes) for the IL Logic, Logic, Reference List, Rung Comments, and In Ladder Cross Reference options.

```

01-15-93 12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00)      Page 1
                                TITLE APPEARS HERE
                                SUBTITLE APPEARS HERE

[ START OF LD PROGRAM NEWPROG ]      ( *                               * )
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]

(*****
* The following logic rung enables automatic mode when all enabling condi- *
* tions are met. See operator's manual, page 6-12, for more information. *
*****)
#0001 01 NOOP

<< RUNG 5 STEP #0002 >>      Cross reference for AUTO
                                -)/[- 5, 7

AUTOPB LISUP UNISUP %Q0012 RSTAHPB EMERGST %Q0006 COOLANT AUTO
+---] [-----] [-----] [-----] [---+---] / [-----] / [-----] / [-----] [-----] ( ) ---
                                0006 0008
AUTO
+---] / [-----] +
0005

#0002 LD %I0004
#0003 AND %Q0002
#0004 AND %Q0044
#0005 AND %Q0012
#0006 OR NOT %Q0005
#0007 AND NOT %I0003
#0008 AND NOT %Q0025
#0009 AND NOT %Q0006
#0010 AND %Q0061
#0011 OUT %Q0005

REFERENCE NICKNAME REFERENCE DESCRIPTION
%Q0006 Hand Light
%Q0012 Table is out
%Q0005 AUTO Auto Light
%I0004 AUTOPB Auto Push Button
%Q0061 COOLANT Coolant Valve
%Q0025 EMERGST Emergency Return
%Q0002 LISUP Loader is up
%I0003 RSTAHPB Reset Auto / Hand PB
%Q0044 UNISUP Unloader is up

Program: NEWPROG D:\ACME\CONVEYOR\NEWPROG Block: _MAIN

```

```

01-15-93 12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v3.00)      Page 2
                                TITLE APPEARS HERE
                                SUBTITLE APPEARS HERE

<< RUNG 6 STEP #0012 >>      Cross reference for EMERGST
                                -]/[- 5, 6

EMSTOP  START  CLAMPED %I0073      EMERGST
+---] [---+---] [---+---]-[---+---] [-----] ( )---
|
| EMERGST | SPN_MTR |
| +---] / [---+---] / [---+ |
| 0006 |
|
#0012 LD      %I0002
#0013 LD      %I0100
#0014 OR NOT  %Q0025
#0015 AND BLK
#0016 LD      %Q0020
#0017 OR NOT  %Q0004
#0018 AND BLK
#0019 AND      %I0073
#0020 OUT      %Q0025

<< RUNG 7 STEP #0021 >>      Cross reference for %M0001
                                NONE

AUTO  %Q0012      %M0001
+---] / [-----] / [-----] ( )---
0005

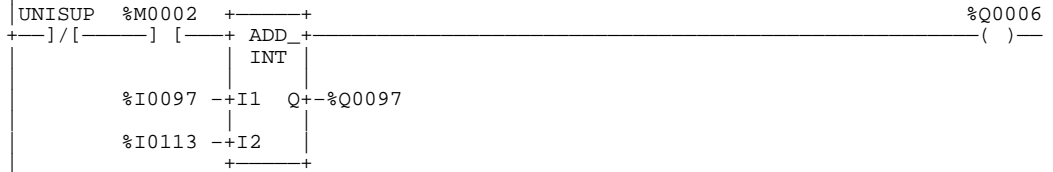
#0021 LD NOT  %Q0005
#0022 AND NOT %Q0012
#0023 OUT      %M0001

REFERENCE NICKNAME      REFERENCE DESCRIPTION
%I0073
%Q0012      Table is out
%M0001
%Q0005 AUTO      Auto Light
%Q0020 CLAMPED
%Q0025 EMERGST   Emergency Return
%I0002 EMSTOP    Emergency Stop PB
%Q0004 SPN_MTR   Start Spindle Motor
%I0100 START

Program: NEWPROG      D:\ACME\CONVEYOR\NEWPROG      Block: _MAIN

```

<< RUNG 8 STEP #0024 >> Cross reference for %Q0006  
-]/[- 5



```
#0024 LD NOT %Q0044
#0025 AND %M0002
#0026 FUNC 60 ADD
      P1: %I0097
      P2: %I0113
      P3: %Q0097
#0027 OUT %Q0006
```

```
[ END OF PROGRAM LOGIC ]
#0028 END OF PROGRAM
```

REFERENCE NICKNAME	REFERENCE DESCRIPTION
%I0097	
%I0113	
%Q0006	Hand Light
%Q0097	
%M0002	
%Q0044 UNISUP	Unloader is up

Program: NEWPROG D:\ACME\CONVEYOR\NEWPROG Block: \_MAIN



01-15-92 15:38 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00) page 2  
TITLE APPEARS HERE  
SUBTITLE APPEARS HERE

```
UNISUP
%Q0044 %M0002 +-----+ %Q0006
-]/[-----] [---+ ADD ---+
                  INT
%I0097 --+I1 Q+--%Q0097
%I0113 --+I2
                  +-----+

[ END OF PROGRAM LOGIC ]
```

REFERENCE NICKNAME	REFERENCE DESCRIPTION	REFERENCE NICKNAME	REFERENCE DESCRIPTION
%I0097		%Q0097	
%I0113		%M0002	
%Q0006	Hand Light	%Q0044 UNISUP	Unloader is up

Program: NEWPROG C:\ACME\CONVEYOR\NEWPROG

The print function must be able to print large rungs which have been truncated by the editor. Truncated rungs will be shown in the listing as they appear on the editor screen. If the number of allowed columns is exceeded, the special truncation character (+) is printed in the rightmost column. If the number of allowed rows is exceeded, the special truncation character (+) is printed below the seventh rung line (bottom displayable row).

The following examples show truncated rungs which exceed both the displayable width and height.

```

01-15-93      12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00)      Page  1
              TITLE APPEARS HERE
              SUBTITLE APPEARS HERE

[ START OF LD PROGRAM TEMP      ]      (*                          *)
[   VARIABLE DECLARATIONS      ]
[   BLOCK DECLARATIONS         ]
[ START OF PROGRAM LOGIC       ]

(*****
(* This is a sample program to help illustrate the print feature.          *)
(* The following rung is truncated on the right and on the bottom.        *)
(*****

<< RUNG 5 STEP #0002 >>

%I0001 %M0001 %M0002 %M0003 %M0004 %M0005 %M0006 %M0007 %M0008
+----] [---+] [-----] [-----] [-----] [-----] [-----] [-----] [-----] [-----]
%I0002 |
+----] [---+ |
%I0003 |
+----] [---+ |
%I0004 |
+----] [---+ |
%I0005 |
+----] [---+ |
%I0006 |
+----] [---+ |
%I0007 |
+----] [---+ |
+      | +      +      +      +      +      +      +      +      +

Program: TEMP                      A:\TEMP

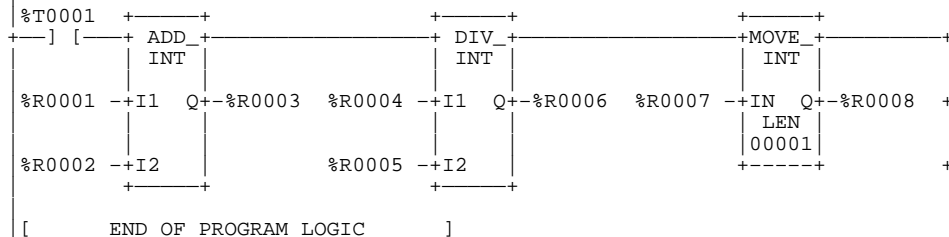
```



01-15-93 12:02 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00) Page 2  
TITLE APPEARS HERE  
SUBTITLE APPEARS HERE

(\*\*\*\*\*  
\* The following rung is truncated on the right only. \*  
\*\*\*\*\*)

<< RUNG 7 STEP #0023 >>



Program: TEMP

A:\TEMP

The following pages are sample pages from a program named DEMO30, which calls several subroutines. One of these subroutines, named LIGHTS, is also included in this sample printout.

```

01-15-93 16:23 GE FANUC SERIES 90-30/90-20 DOCUMENTATION (v4.00)      Page 1

      GGGG EEEEE      FFFFF AAA N N U U CCCC
      G E      F A A NN N U U C
      G GGG EEEE      FFF AAAAA N N N U U C
      G G E      F A A N NN U U C
      GGG EEEEE      F A A N N UUU CCCC

      AAA U U TTTT OOO M M AAA TTTT IIIII OOO N N
      A A U U T O O MM MM A A T I O O NN N
      AAAAA U U T O O M M M AAAAA T I O O N N N
      A A U U T O O M M A A T I O O N NN
      A A UUU T OOO M M A A T I IIII OOO N N

(*****
(*)
(*) Program: DEMO30 (*)
(*)
(*) PLC PROGRAM ENVIRONMENT HIGHEST REFERENCE USED (*)
(*) ----- (*)
(*) INPUT (%I): 512 INPUT: %I0200 (*)
(*) OUTPUT (%Q): 512 OUTPUT: %Q0010 (*)
(*) INTERNAL (%M): 1024 INTERNAL: %M0001 (*)
(*) GLOBAL DATA (%G): 1280 GLOBAL DATA: NONE (*)
(*) TEMPORARY (%T): 256 TEMPORARY: NONE (*)
(*) REGISTER (%R): 2048 REGISTER: %R0002 (*)
(*) ANALOG INPUT (%AI): 128 ANALOG INPUT: NONE (*)
(*) ANALOG OUTPUT (%AQ): 64 ANALOG OUTPUT: NONE (*)
(*)
(*) PROGRAM SIZE (BYTES): 848 (*)
(*)
(*)
(*****)

Program: DEMO30 D:\ACME\CONVEYOR\DEMO30

```

```

(*****
(*)
(*)          BLOCK:  _MAIN          (*)
(*)
(*)          BLOCK SIZE (BYTES):  170 (*)
(*)          DECLARATIONS (ENTRIES): 9 (*)
(*)
(*)          HIGHEST REFERENCE USED (*)
(*)          ----- (*)
(*)          INPUT (%I):  %I0002      (*)
(*)          OUTPUT (%Q): %Q0001      (*)
(*)          INTERNAL (%M):  NONE      (*)
(*)          GLOBAL DATA (%G): NONE  (*)
(*)          TEMPORARY (%T):  NONE      (*)
(*)          REGISTER (%R):  NONE      (*)
(*)          ANALOG INPUT (%AI): NONE  (*)
(*)          ANALOG OUTPUT (%AQ): NONE (*)
(*****

```

Program: DEMO30

D:\ACME\CONVEYOR\DEMO30

Block: \_MAIN

```
[ START OF LD PROGRAM DEMO30 ] (* *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]

SUBR 1 |LIGHTS| LANG: LD (* Shift lights controlled by timer *)
SUBR 2 |DISPLAY| LANG: LD (* Move data from %R2 to lamps *)
SUBR 3 | LOG | LANG: LD (* Count shifts of the lights *)
SUBR 4 |MANUAL| LANG: LD (* Manual setup and control *)
SUBR 5 |GENIUS| LANG: LD (* Control genius demo case *)

#0001 SUBR 01
#0002 SUBR 02
#0003 SUBR 03
#0004 SUBR 04
#0005 SUBR 05

[ START OF PROGRAM LOGIC ]
(* COMMENT *)
#0001 01 NOOP

<< RUNG 5 STEP #0002 >>

LS1
+ ] [-----] (S)
#0002 LD %I001
#0003 SET %Q001
```

```

<< RUNG 6 STEP #0004 >>
METAL
+-----+
+ ] [-----+-----+
FST_SCN
+-----+
#0004 LD %I0002
#0005 OR %S0001
#0006 RST %Q0001

<< RUNG 7 STEP #0007 >>
AUTO +-----+
+ ] [-----+ CALL LIGHTS +-----+ CALL LOG +
+ | (SUBROUTINE) | | (SUBROUTINE) |
+-----+
#0007 LD %Q0001
#0008 FUNC 90 CALLSUB
P1: 00001
#0009 FUNC 90 CALLSUB
P1: 00003

<< RUNG 8 STEP #0010 >>
AUTO +-----+
+ ]/[-----+ CALL MANUAL +
+ | (SUBROUTINE) |
+-----+
#0010 LD NOT %Q0001
#0011 FUNC 90 CALLSUB
P1: 00004

<< RUNG 9 STEP #0012 >>
AUTO +-----+
+ ] [-----+ CALL GENIUS +
+ | (SUBROUTINE) |
+-----+
#0012 LD %Q0001
#0013 FUNC 90 CALLSUB
P1: 00005

[ END OF PROGRAM LOGIC ]
#0014 END OF PROGRAM

```



```

(*****
(*)
(*)          BLOCK:  LIGHTS          (*)
(*)
(*)          BLOCK SIZE (BYTES):  151 (*)
(*)          DECLARATIONS (ENTRIES):  2 (*)
(*)
(*)          HIGHEST REFERENCE USED (*)
(*)          ----- (*)
(*)          INPUT (%I):  %I0200      (*)
(*)          OUTPUT (%Q):  %Q0010     (*)
(*)          INTERNAL (%M):  %M0001   (*)
(*)          GLOBAL DATA (%G):  NONE  (*)
(*)          TEMPORARY (%T):  NONE    (*)
(*)          REGISTER (%R):  %R0002   (*)
(*)          ANALOG INPUT (%AI):  NONE (*)
(*)          ANALOG OUTPUT (%AQ):  NONE (*)
(*)
(*****

```

```
[ START LD SUBROUTINE LIGHTS ]
[ VARIABLE DECLARATIONS ]
[ START OF SUBROUTINE LOGIC ]
(* COMMENT *)
<< RUNG 4 STEP #0002 >>
%M0001 SFT_RT +-----+
+---] [-----]/[-----+ SHL
                             WORD
                             |
                             +---IN B2+---
                             |
                             LEN
                             00001
                             |
                             CONST -+N Q+-%R0002
                             00001
ALW_ON
+---] [-----+B1
                             |
                             +-----+

<< RUNG 5 STEP #0006 >>
LASTBIT
+---] [-----SFT_RT
                             (SM)-
[ END OF SUBROUTINE LOGIC ]
```



# Chapter 10

## I/O Configuration

The I/O configuration function is used to specify the mapping of logical references used by the user program to the physical I/O modules. This chapter contains the following sections:

Section	Title	Description	Page
1	Default Configuration	Explains how to use the default I/O configuration provided by the PLC.	10-2
2	Displaying the I/O Configuration Rack Screen	Explains how to display the rack screen, copy a configuration from one slot to another, change or delete the configuration of a slot, move a configuration to another slot, and save the configuration to disk.	10-4
3	Configuring the CPU	Explains how to configure the CPU module.	10-9
4	Selecting the Base Rack	Explains how to select the base rack and expansion racks.	10-26
5	Configuring the Model 211 CPU	Explains I/O configuration for the Model 211 CPU are added to the Series 90-20 base.	10-31
6	Configuring a Micro PLC	Explains how to configure a Micro PLC.	10-36
7	Configuring 90-30 I/O Modules	Explains how to configure 90-30 I/O modules.	10-45
8	Configuring an HSC or Embedded HSC	Explains how to configure the HSC module to function as four 16-bit counters, two 32-bit counters, or as one 32-bit differential counter.	10-50
9	Configuring a PCM Module	Explains how to configure the PCM as one CCM port, two independent CCM ports, one CCM port and one BASIC application having one port, or one BASIC application using one or both serial ports.	10-57
10	Configuring a TCP/IP Ethernet Module	Explains how to configure a TCP/IP Ethernet Module.	10-63
11	Configuring a CMM Module	Explains how to configure a Communications Module in <b>RTU ONLY</b> , <b>RTU/CCM</b> , or <b>CCM/RTU</b> mode.	10-67
12	Configuring an APM Module	Explains how to configure both a one-axis and two-axis APM.	10-73
13	Configuring an ADC Module	Explains how to configure the ADC module.	10-82
14	Configuring a GCM or Enhanced GCM	Explains how to configure the module to transfer global data to and from the PLC.	10-83
15	Configuring a GBC	Explains how to configure the Genius Bus Controller.	10-88
16	Configuring a High Density Analog Output Module	Explains how to configure the High Density Analog Output module.	10-90
17	Configuring an Analog Combo Module	Explains how to configure the Analog Combo module.	10-94
18	Configuring a Third-Party Module	Explains how to configure third-party modules.	10-99
19	Configuration Reference View	Explains how to use the Reference View table to display a list of configured modules with the same user reference (%I, %Q, %AI, %AQ, %R, or %G).	10-101

## Section 1: Default Configuration

When the PLC is first powered up or configuration is cleared, a default I/O configuration is created. The default I/O configuration is based on the I/O modules installed in the system. If you are satisfied with the default configuration, no further configuration is required.

The PCM, CMM, and ADC modules are included in the default configuration. All other smart modules (HSC, APM, etc.) are *not* included and must be manually configured using the Logicmaster 90 I/O configuration function.

For those users who want their system configured differently from the default (additional I/O modules, different I/O references, etc.), system configuration can be changed using the Logicmaster 90 I/O configuration function, described in the following sections of this chapter. If you edit the default I/O configuration or create a new one and store it to the PLC, it will replace the default configuration; and the PLC will not automatically configure itself again until the configuration is cleared.

The PLC will automatically configure the system based on the I/O modules installed according to the following table. This table shows how I/O references are assigned to each slot in the PLC. The 5-slot Model 311 or 313 PLC will have I/O addresses assigned to every slot. The 10-slot Model 311 or 313 PLC will have discrete I/O addresses assigned to each slot, but slots 9 and 10 will not be assigned analog I/O addresses. The Model 331 and higher PLCs will have analog and discrete addresses assigned to 15 of its slots (rack 0, slot 2 to rack 1, slot 6).

Table 10-1. Default I/O Configuration

Rack	Slot	Discrete Input	Discrete Output	Analog Input	Analog Output	Notes
0	1	%I001-032	%Q001-032	%AI001-008	%AQ001-004	This slot not configured in Model 331 or 341.
0	2	%I033-064	%Q033-064	%AI009-016	%AQ005-008	
0	3	%I065-096	%Q065-096	%AI017-024	%AQ009-012	
0	4	%I097-128	%Q097-128	%AI025-032	%AQ013-016	
0	5	%I129-160	%Q129-160	%AI033-040	%AQ017-020	Last slot in 5-slot Model 311 or 313.
0	6	%I161-192	%Q161-192	%AI041-048	%AQ021-024	
0	7	%I193-224	%Q193-224	%AI049-056	%AQ025-028	
0	8	%I225-256	%Q225-256	%AI057-064	%AQ029-032	Last slot to receive analog configuration in the 10-slot Model 323.
0	9	%I257-288	%Q257-288	%AI065-072	%AQ033-036	
0	10	%I289-320	%Q289-320	%AI073-080	%AQ037-040	Last slot in 10-slot Model 323.
1	1	%I321-352	%Q321-352	%AI081-088	%AQ041-044	
1	2	%I353-384	%Q353-384	%AI089-096	%AQ045-048	
1	3	%I385-416	%Q385-416	%AI097-104	%AQ049-052	
1	4	%I417-448	%Q417-448	%AI105-112	%AQ053-056	
1	5	%I449-480	%Q449-480	%AI113-120	%AQ057-060	
1	6	%I481-512	%Q481-512	%AI121-128	%AQ061-064	Last slot in Model 331 or higher.
1	7	-	-	-	-	
1	8	-	-	-	-	
1	9	-	-	-	-	
1	10	-	-	-	-	

You may view the current configuration by loading the configuration from the PLC to the programmer. (Refer to Chapter 8, "Program Utilities," for more information about loading configuration from the PLC to the programmer.) The configuration can then be viewed from the I/O configuration function and can be printed using the configuration print function.

The PLC remains in default configuration mode until a configuration that is entered or modified by the Logicmaster 90-30/20 Micro configuration software is stored to the PLC. Only the CPU parameters (baud rate, parity, etc.) may be changed without taking the PLC out of default configuration mode. To put the PLC back into default configuration mode so that it will automatically configure the system, follow these steps:

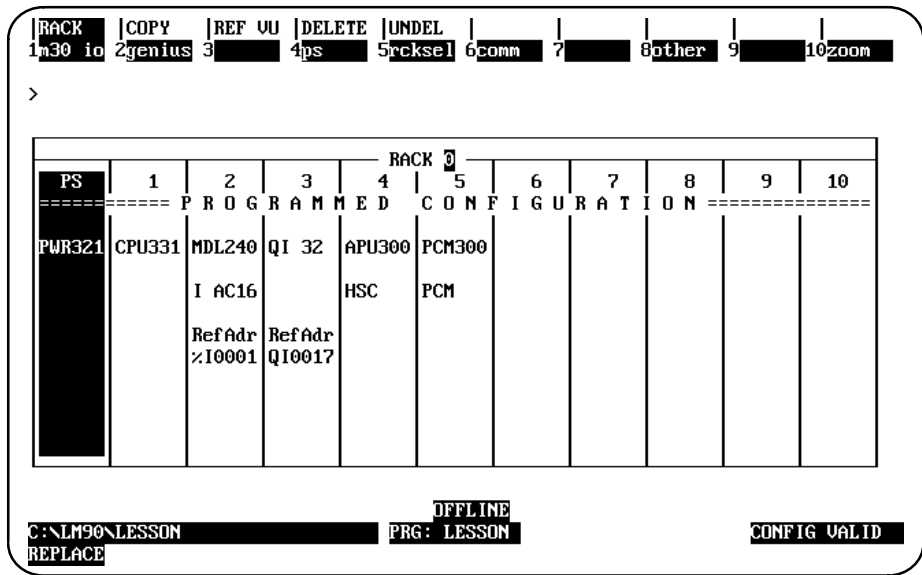
1. Press **Clear (F5)** from the Program Utility Functions menu to clear the configuration.
2. Verify that the selection for configuration is set to **Y (Yes)**.
3. If you do not want to clear the ladder logic program or reference tables for that program, enter **N (No)** for program logic or reference tables.
4. Press the **Enter** key to begin the clear function. **Once the clear operation has begun, it cannot be aborted.** When the operation is complete, the software displays the message "Clear complete."

The PLC will now automatically configure the system using the default I/O configuration shown in table 10-1.

5. Press the **Load** function key (**F1**) to load the configuration.

## Section 2: Displaying the I/O Configuration Rack Screen

When I/O (F1) is selected from the Configuration Software main menu, the I/O Configuration Rack screen, similar to the one shown below, is displayed. The following example represents a sample Series 90-30 PLC Model 331 system.



The rack shown above is divided into 11 slots. The first slot on the left is labeled “PS” at the top and always contains the power supply. The remaining slots are numbered 1 through 10. The CPU module always occupies slot 1 of rack 0. The remaining slots may contain any Series 90-30 I/O module, Genius Communications Module, Axis Positioning Module, or Programmable Coprocessor Module.

Model 331 and higher CPUs can have expansion racks. Up to four subsystem racks are allowed (up to seven subsystem racks in a 351 CPU). The main CPU rack is rack 0, and expansion racks are numbered 1 through 7. (All five racks default to be 10-slot racks.) The size of each rack can be changed on an individual rack basis by pressing **Rack Selection** (F5) and selecting the desired rack. This allows a mixture of 5-slot and 10-slot racks in a Series 90-30 PLC configuration. The **Up/Down** cursor movement keys and **Page Up/Down** keys are used to move between racks in ascending or descending order, respectively.

Both the 5-slot and 10-slot racks of the Model 311 PLC and the Model 313 PLC are one-rack systems. Moving from rack to rack is not allowed. The following example represents a sample 10-slot rack of the Model 311 PLC. Notice that on Models 311, 321, 323, and 313, Slot 1 *can* be used for I/O.

```

[RACK] | [COPY] | [REF UU] | [DELETE] | [UNDEL] | | | | | |
1m30 io 2genius 3 4ps 5rcksel 6comm 7 8other 9 10zoom
>
----- RACK 3 -----
PS/CPU | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10
===== PROGRAMMED CONFIGURATION =====
PWR321 | CMM301 | MDL240 | MDL640 | | | | | | |
CPU 30 | GENCOM | I AC16 | I DC16 | | | | | | |
        |         | RefAdr | RefAdr | | | | | | |
        |         | ∕I0001 | ∕I0017 | | | | | | |
-----
OFFLINE
C:\LM90\LESSON PRG: LESSON CONFIG VALID
REPLACE
  
```

The following example represents a sample 5-slot rack of the Model 311 PLC.

```

[RACK] | [COPY] | [REF UU] | [DELETE] | [UNDEL] | | | | | |
1m30 io 2genius 3 4ps 5rcksel 6comm 7 8other 9 10zoom
>
----- RACK 3 -----
PS/CPU | 1 | 2 | 3 | 4 | 5
===== PROGRAMMED CONFIGURATION =====
PWR321 | CMM301 | MDL240 | MDL640 | |
CPU 30 | GENCOM | I AC16 | I DC16 | |
        |         | RefAdr | RefAdr | |
        |         | ∕I0001 | ∕I0017 | |
-----
OFFLINE
C:\LM90\LESSON PRG: LESSON CONFIG VALID
REPLACE
  
```

The cursor position on the I/O Configuration Rack screen is indicated by having the slot highlighted in reverse video. Upon entering the Rack screen, the Power Supply slot is shown in reverse video. Use the Left or Right cursor keys to move the cursor from one slot to the next. To display another rack, use the Next and Previous page keys or the Up and Down cursor keys.

The rack screen presents an overview perspective of the Series 90-30 PLC system.

Function Key	Function	Description
F1	M30I/O	Add Model 30I/O modules.
F2	Genius	Add a Genius Communications Module.
F4	Power Supply	Add the power supply. If the power supply already exists, press <b>Zoom (F10)</b> to configure it.
F5	Rack Selection	Change the rack selection.
F6	Communications	Add a Communications Module.
F8	Other	Add other modules (e.g., PCM, ADC).
F10	Zoom	Display the current configuration of a slot.

## Configuration Validation

CONFIG VALID is displayed in the lower right corner of each display screen after the configuration is successfully validated. When CONFIG INVALID is indicated, the file may not be stored to the PLC.

The most common cause of the CONFIG INVALID status is fatal overlaps among %I or %AI references.

A warning occurs when non-fatal overlaps occur; however, the configuration is still valid. Non-fatal overlaps are most likely to occur between references other than those described for fatal overlaps.

## Copying Configuration from Slot to Slot

To copy configuration from slot to slot:

1. Position the cursor on the first module and press **Zoom (F10)** to display the slot configuration screen.
2. Configure the module.
3. Press **Rack (Shift-F1)** or the **Escape** key to return to the rack screen shown above.
4. With the cursor on that slot, press **Copy (Shift-F2)**.
5. Move the cursor to another slot and press the **Enter** key. You can repeat this as many times as needed. As each module is entered, its reference address is set to the next highest available address.

If the limit for the reference address has been reached, the copy function will adjust the module's reference address to the maximum allowed. Each copy after the maximum reference address has been reached will result in an address overlap.

6. When finished copying, exit copy mode by pressing the **Escape** key. The software displays the message "Copy Mode Ended".
7. For each copy, press **Zoom (F10)** to display the detail screen. Edit the reference address (if required) and other characteristics as needed.

---

## Changing the Configuration of a Slot

### Replacing the Module with Another of the Same Type

To change the configuration of a slot by replacing the module with another of the **same** type:

1. Place the cursor at the slot to be changed.
2. Press **Zoom (F10)** to display the module's detail screen.
3. Use one of the function keys to select a new module type for the slot. With the list of available modules displayed, move the cursor to the correct module and press the Enter key. Then, enter **Y** (Yes) after the prompt "REPLACE displayed module ? (Y/N)".
4. Complete the configuration; then press **Rack (Shift-F1)** or the **Escape** key to return to the rack display.

### Replacing the Module with One of a Different Type

To change the configuration of a slot by replacing the module with one of a **different** type:

1. Place the cursor at the slot to be changed.
2. Press the function key that represents the type of module you want to place in the slot.
3. After confirming the deletion of the existing module, the module selection screen is displayed.
4. Complete the configuration; then, press **Rack (Shift-F1)** or the **Escape** key to return to the rack display.

## Deleting the Configuration of a Module

To delete the configuration of a module:

1. Place the cursor on the module and press **Delete (Shift-F4)**. Enter **Y** (Yes) after prompt "DELETE displayed module from slot ? (Y/N)". The configuration of the slot will be deleted.
2. To return the configuration to the slot, press **Undelete (Shift-F5)**.

## Moving a Module to Another Slot

To move the configuration of a module to another slot:

1. With the cursor at the configuration to be moved, press **Delete** (**Shift-F4**). Enter **Y** (Yes) after prompt "DELETE displayed module from slot? (Y/N)". The configuration of the slot will be deleted.
2. Move the cursor to the new location; it may be in another rack.
3. Press **Undelete** (**Shift-F5**). The deleted configuration will appear in the new location. The reference address will not be affected.

## Saving the Configuration to Disk

The configuration is automatically saved to a file on disk during the configuration process:

- When module data has been changed within a rack, and that new data is moved from that rack to another rack or when you cursor to a new rack.
- When module data has been changed within a rack, and then the **Escape** key is pressed to return to the Configuration Software main menu.
- When module data has been changed and **Reference View** (**Shift-F3**) is pressed.

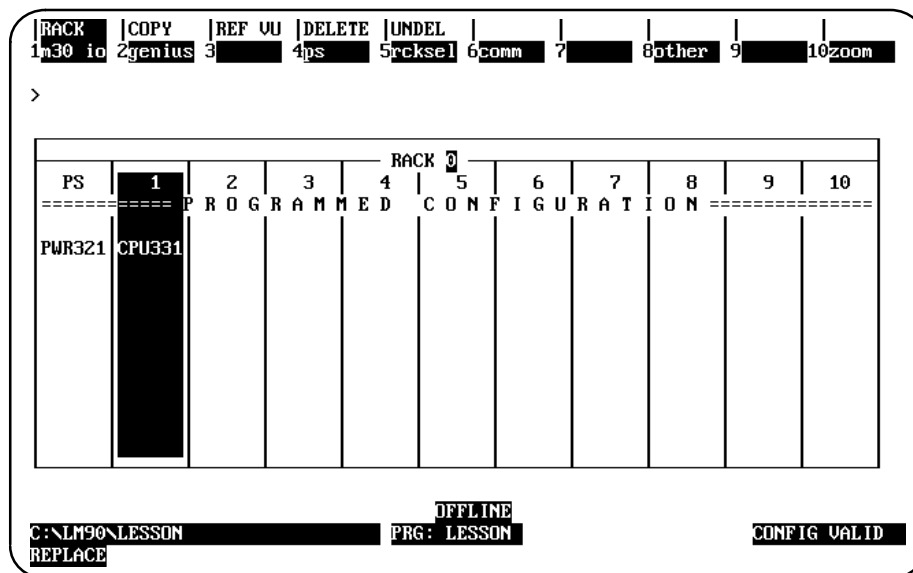
The configuration may also be written to disk at any time by pressing **ALT-U** to update the disk.



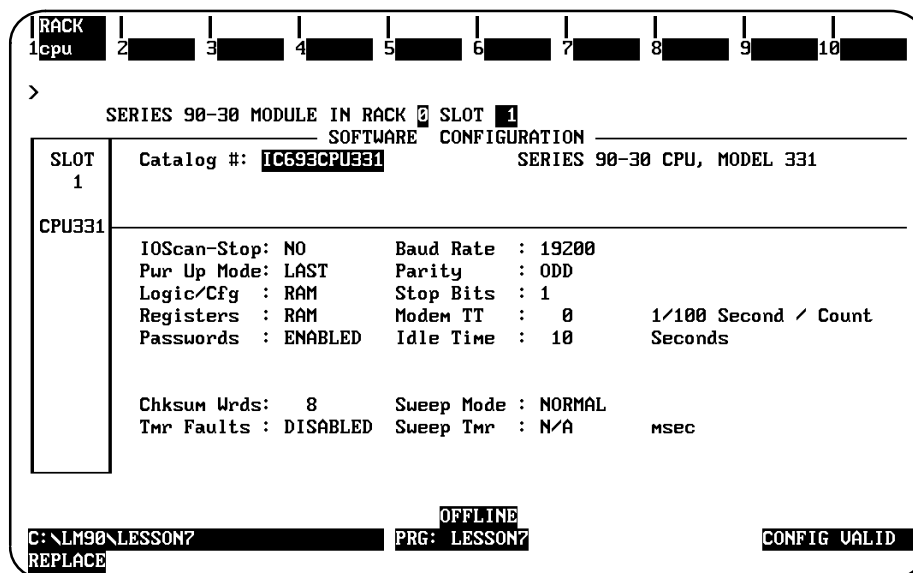
## Section 3: Configuring the CPU

To configure the CPU module:

1. Model 331 or higher CPUs must be located in slot 1 of rack 0. Move the cursor to this slot by pressing the cursor movement keys. (Note that the Model 331 CPU is used in the following screens.)



2. Press **Zoom (F10)** to display the CPU detail screen:



Parameter	Description
I/O Scan-Stop	Indicates whether the I/O is to be scanned while the PLC is in <b>STOP</b> mode. Choices are <b>YES</b> or <b>NO*</b> .
Power-Up Mode	Indicates which state the PLC is to be powered up in. Choices are <b>RUN</b> , <b>STOP</b> or <b>LAST*</b> . <b>LAST</b> indicates that the PLC will power up in the same mode it powered down in. Generally, <b>RUN</b> is more suitable for a production system.
Logic From	Indicates whether the program logic is stored in <b>RAM*</b> , or in an additional, optional <b>PROM</b> (select <b>PROM</b> for an EEPROM you have installed).
Registers	Indicates whether the source of registers is <b>RAM*</b> or <b>PROM</b> .
Passwords	Indicates whether passwords are <b>ENABLED*</b> or <b>DISABLED</b> .
Baud Rate	Transmission rate in bits per second. Choices are <b>300</b> , <b>600</b> , <b>1200</b> , <b>2400</b> , <b>4800</b> , <b>9600</b> , or <b>19200*</b> .
Parity	Type of parity bits added to each word. Choices are <b>ODD*</b> , <b>EVEN</b> , or <b>NONE</b> . Generally, use <b>ODD</b> for direct or multi-drop or None for modem connection.
Stop Bits	Communication uses at least one stop bit. Slower communication uses two stop bits. Choices are <b>1*</b> or <b>2</b> .
Modem Turnaround Time	Modem turnaround delay time counts. 1 count = 1/100 second. Choices are <b>0</b> to <b>255</b> (default = <b>0</b> ). Use zero ( <b>0</b> ) for direct connection, or <b>2</b> to <b>10</b> for modems (larger for radio modem).
Idle Time	Maximum communication idle time from <b>1</b> to <b>60</b> seconds (default = <b>10</b> seconds).
Checksum Words	The number of words of user program to be applied to the checksum function per sweep. Choices are <b>0</b> through <b>32</b> , inclusive.
Sweep Mode	<p>Sweep mode can be either <b>NORMAL SWEEP</b> mode (<b>NORMAL*</b>) or <b>CONSTANT SWEEP</b> mode (<b>CNST SWP</b>). To change the settings, tap the Down Arrow key till the cursor rests in the "Sweep Mode:" parameter. Then press the <b>Tab</b> key to change the selection from "NORMAL" to "CNST SWP" (if necessary—if you are just changing the time of the Constant Sweep, it may already say "CNST SWP"). Then tap the <b>Down Arrow</b> key once to move the cursor to the "Sweep Tmr" parameter and enter the number of milliseconds (from <b>5</b> to <b>200</b>—<b>5</b> to <b>500</b> for 351 CPUs) that you want. To store the changes to the folder, press the <b>Escape</b> key twice. Then store the configuration to the PLC and switch the PLC into <b>RUN</b> mode for this change to take effect. This Sweep Mode adjustment is different from the Active Constant Sweep which can be edited only in <b>RUN</b> mode. Refer to the "Standard Sweep Mode Variations" section of chapter 2 in the <i>Series90-30/20/MicroReferenceManual</i> (GFK-0467) and page 5-19 and following of this manual for a discussion of these two types of Constant Sweep Mode.</p> <p>In <b>NORMAL SWEEP</b> mode, the PLC sweep executes as fast as possible. The overall PLC sweep time depends on the logic program plus the time required to compute the checksum.</p> <p>In <b>CONSTANT SWEEP</b> mode, the overall PLC sweep time is fixed.</p>
Sweep Timer	<p>A byte value which can be configured, in 1 millisecond increments, to be any value from <b>5</b> to <b>200</b> ms. —<b>5</b> to <b>500</b> for 351 CPUs. Default = <b>100</b>.</p> <p><b>Note:</b> The sweep timer is not configurable when the sweep mode is <b>NORMAL</b>. In this mode, the timer is set to N/A (Not Applicable), and you cannot cursor to the <b>SweepTimer</b> field.</p>
Tmr Faults	(For use with SFC programming) Timer faults indicate whether or not a fault will be logged in the PLC Fault Table when a step's time is either outside its maximum or minimum limit. Default = <b>Disabled</b> .

\* Default selection.

For configuration of the Run/Stop switch on 351 and 352 CPUs, refer to the "Run/StopSwitchConfiguration (351 and 352 CPS)" section on page 10-17. For configuration of Port 1 and Port 2 on 351/2 CPUs, refer to the "Port Configuration for 351 and 352 CPUs" section on page 10-18.

**Caution**

In order to re-enable passwords once they have been disabled, PLC memory may be cleared with an HHP, or for certain models (see below) there are “MEM CLR” pads on the board. The HHP needs to be connected. Then power off the PLC. Then hold both the <CLR> and <M/T> keys down while powering the PLC back up. If you do not have an HHP, call the GE Fanuc Technical Service Hotline (1-800-828-5747) for assistance.

MEM CLR pad exist on the following models:

CPU331Z and CPU331-AA and later

All 350 AND 352 CPUs

CPU351-EH and later

All 360 and higher CPUs

If your CPU has MEM CLR pads, follow these steps:

- REMOVE THE CPU FROM THE BACKPLANE.
- REMOVE ANY AUXILARY BATTERIES.
- Place a small metal screwdriver across the pads for at least 15 seconds to clear all PLC memory.

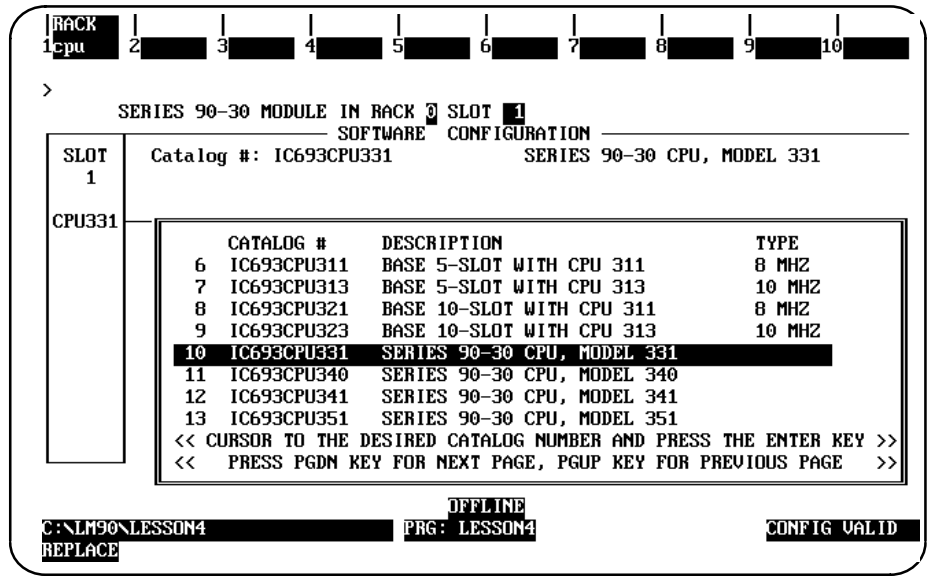
**NOTE:** To prevent physical damage to the PLC, you **MUST** remove the CPU from the backplane and remove any attached batteries.

**Note**

The MEM CLR pads are located on the back of 331, 351 and 352 CPUs. Once you remove the CPU from the backplane, you will be able to see the pads (labeled “MEM CLR”). For 350 CPUs and all 360 series CPUs, the MEM CLR pads are located on the front, about one inch below the key switch. To remove the front shroud, insert a small screwdriver into one of the two slots on either side of the unit above the key switch and gently push in. This will release the top, and you can use the same technique to release the bottom of the shroud.

3. Use the **Tab** key and **Back Tab** key (**Shift-Tab**) to scroll through the selections for each parameter displayed on this screen. For help selecting the parameters, press **ALT-H**.

- 4. When a value has been selected for each parameter, press **CPU (F1)** to display a list of catalog numbers and modules.



- 5. Position the cursor on the catalog number for the Model 331 CPU (IC693CPU331), and press the **Enter** key.
- 6. Press **Rack (Shift-F1)** or the **Escape** key to return to the rack display.

## Selecting a Different CPU Module

### Note

If you configure a newly-released CPU using Release 4 or later of Logicmaster 90-30/20 software, you cannot use this configuration with an earlier release of the programming software. That is to say, if you configure a CPU with a recent release of the software, use the same release (or later) for programming.

In addition, a port configuration error can occur if try to configure the ports of a pre-Release 7 351 CPU with a Release 7 or later version of Logicmaster (see page 10-18).

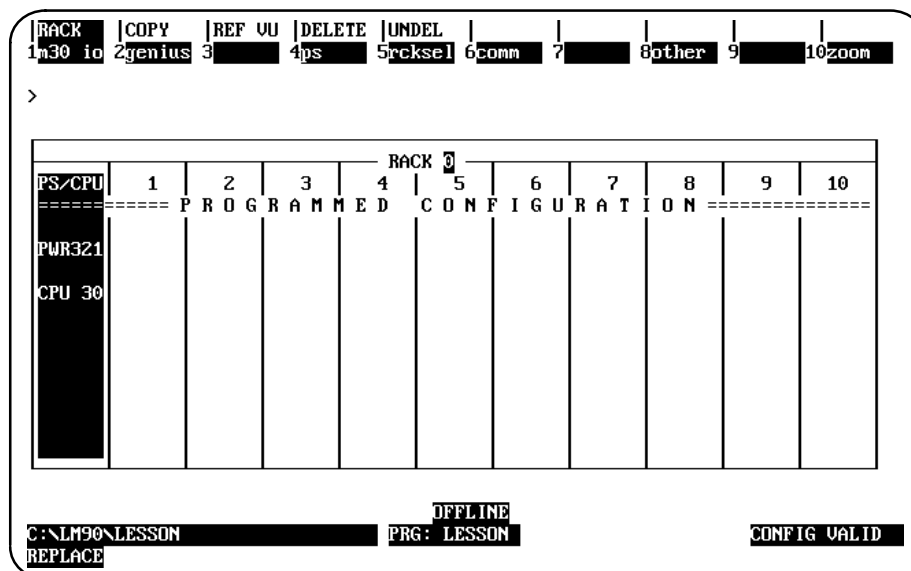
1. To select a different CPU module, move the cursor to the catalog number for the desired CPU type and press the **Enter** key. Then, enter **Y** (Yes) after the prompt "REPLACE displayed module?(Y/N)".

For example, to configure a 10-slot rack for the Model 311 CPU, move the cursor to the entry for that module (IC693CPU321) and press the **Enter** key.

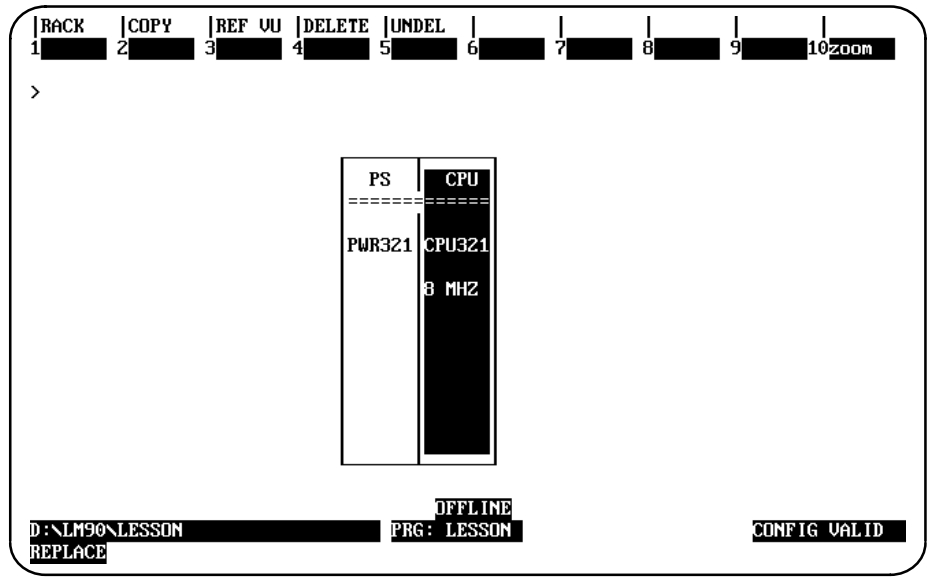
2. If any modules (e.g., intelligent modules) that are not supported by the 10-slot rack have been configured, the following error message is displayed:

Selected CPU does not support all currently configured modules

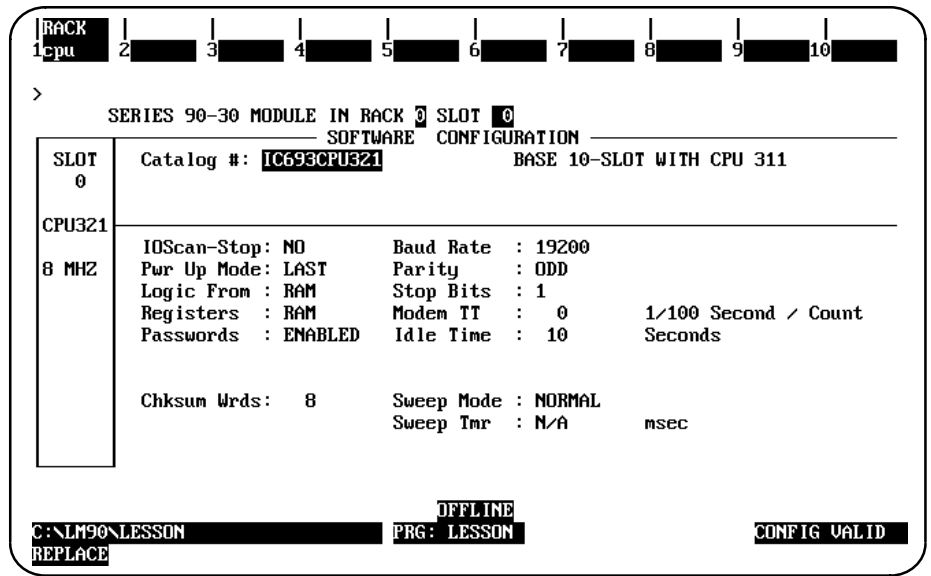
3. Press the **Escape** key twice, once to zoom out of the list of catalog numbers and a second time to zoom out of the CPU detail screen back to rack 0.
4. Use the delete function to delete modules which are not supported. Then, press **Zoom (F10)** to return to the list of catalog numbers.
5. Position the cursor once more on the catalog number for the 10-slot rack for the Model 311 CPU (IC693CPU321), and press the **Enter** key. Then, press the **Escape** key twice to return to the Rack Configuration screen shown below.



6. Press **Zoom (F10)** to display the PowerSupply/CPU screen.

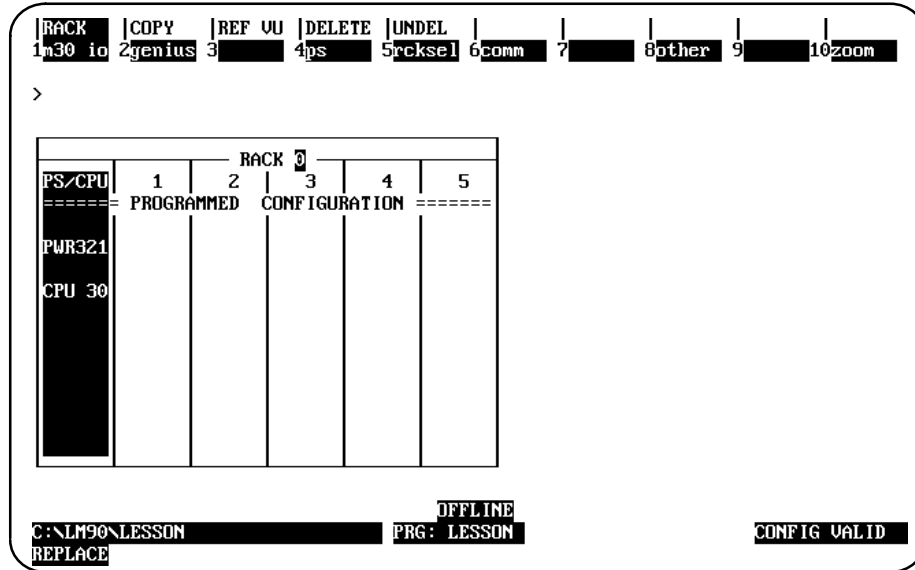


7. Then, press **Zoom (F10)** again to display the CPU detail screen.



- 8. Press the **Escape** key once to return to the detail screen. Press the **Escape** key again to save the module configuration and return to the rack display.
- 9. Or, to configure the 5-slot rack for the Model 311 CPU, press **CPU (F1)** to display a list of catalog numbers and modules.

10. Move the cursor to the entry for the 5-slot rack (IC693CPU311), and press the **Enter** key. Then, enter **Y** (Yes) after the prompt "REPLACE displayed module? (Y/N)".

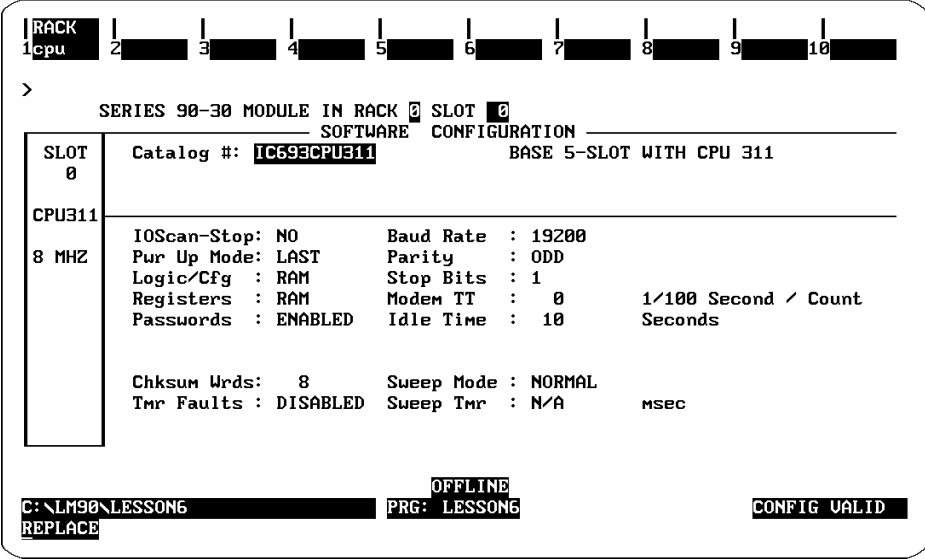


Since no boards had been configured in slots 6 through 10, the change from a 10-slot rack to a 5-slot rack is allowed. However, if boards had been configured in any of slots 6 through 10, the following error message would be displayed:

Slots greater than 5 will be lost, Continue REPLACE? (Y/N)

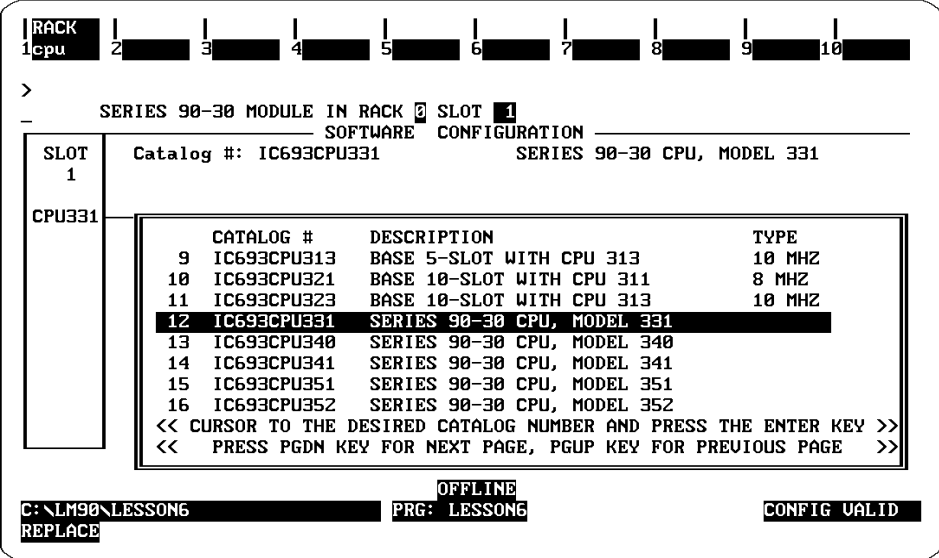
If **Y** (Yes) is entered, the system will automatically delete slots 6 through 10 before changing the CPU module.

- Press **Zoom (F10)** and then **CPU (F1)** to display the CPU detail screen for the 5-slot rack for the Model 311 CPU.



Use the **Tab** key and **Back Tab** key (**Shift-Tab**) to select a value for each parameter displayed on this screen.

- Then, press the **Escape** key twice to display the Rack Configuration screen.
- Press **Zoom (F10)** twice and then press **CPU (F1)** to display the list of catalog numbers and modules.



- Position the cursor on the catalog number for the Model 311 CPU (IC693CPU311), and press the **Enter** key.
- Press the **Escape** key to return to the rack display.

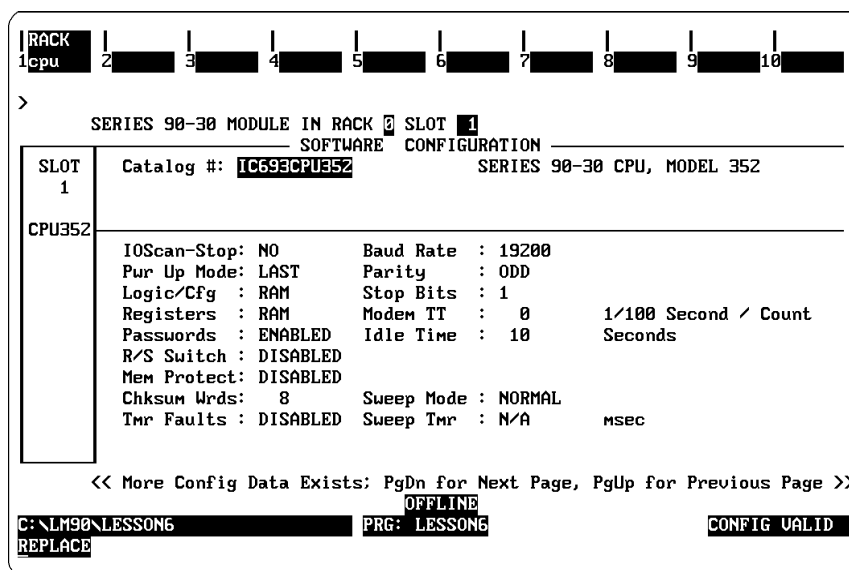


### Note

If you install and configure a 350 and higher CPU and use folders you created previously, you will be prompted to convert the folder to one that uses the code written for the higher CPUs.

## Run/Stop Switch Configuration (350 and higher CPUs)

The configuration screens for CPU models 350 and higher have two extra parameters, one of them being **R/S Switch**, on the first “page” of their configuration screens as shown below:



The **R/S Switch** parameter enables or disables the RUN/STOP Key Switch functionality available in 351 and 352 CPUs. With this field enabled, you can switch the PLC into Stop mode or from Stop mode into Run mode (Release 7 or later), and clear faults (Release 7 or later). Whether or not the **R/S Switch** parameter is enabled, the Key Switch always functions as a Flash memory protection switch.

For more information about the operation of the Key Switch, refer to the “Key Switch on 350 and 360 CPUs: Change Mode and Flash Protect” section of Chapter 2, “System Operation,” in the *Series 90™ -30/20/Micro Programmable Controllers Reference Manual* (GFK-0467).

The Key Switch can also be used to protect configuration and user program memory and overrides in Release 8 and later CPUs. Refer to the next page for information about this feature.

## Memory Protection Configuration for 350 and Higher CPUs

The configuration screens for 350 and higher CPUs have two extra parameters, one of them being **Mem Protect**, on the first “page” of their configuration screens as on the previous page.

If you change the **Mem Protect** field from its default state to **ENABLED**, then the Key Switch can provide memory protection as well. Once **ENABLED**, when the Key Switch is ON (i.e., in vertical position), two types of memory protection occur:

- User program and configuration cannot be modified.
- Force and override of point data is not allowed.

To change the **Mem Protect** field to **ENABLED**, move the cursor down to the **Mem Protect** field and press Tab.

### Note

This enhanced memory protection feature is **available only on Release 8 or later 350 and higher CPUs**.

Also, please note that you can change the Time-of-Day clock while protection is on using Logicmaster (refer to page 11-3), but not with the Hand-Held Programmer.

## Port Configuration for 351, 352, and 363 CPUs

The Release 7 and later 351, 352, and 363 CPUs have configurable communication ports. After you have configured the standard information and the Run/StopKey Switch discussed previously, press the **PgDn** key to display the following configuration screen:

```

RACK 1cpu 2 3 4 5 6 7 8 9 10
>
SERIES 90-30 MODULE IN RACK 0 SLOT 1
SOFTWARE CONFIGURATION
SLOT 1 Catalog #: IC693CPU352 SERIES 90-30 CPU, MODEL 352
CPU352
Port 1 Mode: SNP      Port 2 Mode: SNP
SNP Mode : SLAVE     SNP Mode : SLAVE
Data Rate : 19200    Data Rate : 19200
Flow Contrl: NONE    Flow Contrl: NONE
Parity : ODD         Parity : ODD
Stop Bits : 1        Stop Bits : 1
Timeout : LONG       Timeout : LONG
TurnA Delay: 0       TurnA Delay: 0
SNP ID :             SNP ID :

<< More Config Data Exists; PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\LESSON7 PRG: LESSON7 CONFIG UALID
REPLACE

```

Within each field except SNP ID, you can press the **Tab** key to toggle through the valid choices.

### Note

**The port configuration requires a Release 7 or later 351 or 352 CPU, or a Release 9 or later 363 CPU. You will receive an error if you attempt to store this configuration to a pre-Release 7 CPU. This screen did not exist in earlier Logicmaster releases; therefore, you have to have Release 7 or later of Logicmaster to fully utilize this functionality.**

Parameter	Field Definitions and Values	Default
Port 1 Mode	The mode of communication being used for Port 1: SNP, CCM, RTU, CUSTOM, DISABLED <b>NOTE:</b> The CUSTOM setting provides a way of configuring communications for a generic text output to facilitate communications to third-party modules. Refer to the <i>Series 90™ -30 Programmable Controller Installation Manual</i> (GFK-0356) for additional information.	SNP
Port 2 Mode	The mode of communication being used for Port 1: SNP, CCM, RTU, CUSTOM, DISABLED See <b>NOTE</b> for Port 1 Mode.	SNP
SNP Mode	Displays the type of access for SNP access: SLAVE, MASTER	SLAVE
Data Rate	Defines the data rate for serial communication: 300, 600, 1200, 2400, 4800, 9600, 19200	19200
Flow Control	Type of flow control to be used on the port: HARDWARE, NONE	NONE
Parity	Specifies the parity used in serial communication. The default parity is Odd. Parity selection can be changed to Even or None if required for communication over modems or with a different SNP master device. Valid choices are NONE, ODD, EVEN.	ODD
Stop Bits	Most serial communication uses at least one stop bit. Slower devices use two stop bits. Valid choices are 1 stop bit or 2 stop bits	1 stop bit
Timeout	Length of timeouts used on the port: LONG, MEDIUM, SHORT, NONE	LONG
TurnA (Turn-around) Delay	Turnaround delay time to be used on the port: 0 to 250 ms	NONE
SNP ID (Device Identifier) (7 bytes in length) *	This is an identifier that distinguishes this device from others on the same network.  <i>Note that for devices that use an SNP ID of 1, enter 49 decimal (31 hexadecimal).</i>	NULL

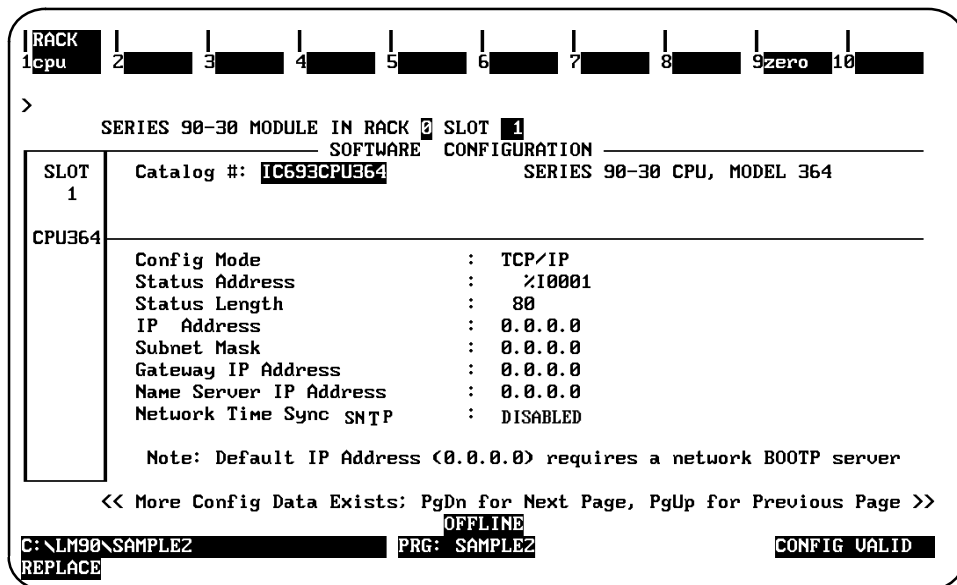
\* For Release 6.5 and 6.6, only the first six bytes are used. For Release 7 and later, the first seven bytes are used.

### Note

For additional information about configuration of 351, 352, and 363 CPUs, refer to the *Series 90™ -30 Programmable Controller Installation Manual*, GFK-0356.

## TCP/IP Ethernet Configuration on 364 CPUs

The configuration screens for the 364 CPU have two extra “pages” used for configuring the TCP/IP Ethernet communications for this CPU. After you select the 364 CPU, the first page you see is the same as that for the 350 and higher CPUs as shown in the discussion of configuration of the Run-Stop switch on page 10-17. When you press the PgDn key for the additional configurations, you will see a screen that looks like this:



### Ethernet Parameters for the 364 CPU

**Configuration Mode:** This is currently fixed as TCP/IP.

**Status Address:** The Status Reference Type is the location of the LAN Interface Status (LIS) bits (16 bits) and the Channel Status bits (64 bits). The Channel Status bits are always located immediately following the LAN Interface Status bits. The Status address can be assigned to %I, %Q, %R, %AI or %AQ memory. The default value is the next available %I address.

**IP Address, Subnet Mask, Gateway IP Address, and Name Server IP Address:** These values should be assigned by the person in charge of your network (the network administrator). TCP/IP network administrators are familiar with these parameters. It is important that these parameters are correct; otherwise, the Ethernet Interface may be unable to communicate on the network and/or network operation may be corrupted. It is especially important that each node on the network is assigned a *unique* IP address.

However, if you have no network administrator and are using a simple *isolated network* with no gateways, you can use the following range of values for the assignment of local IP addresses:

- 10.0.0.1 First PLC
- 10.0.0.2 Second PLC
- 10.0.0.3 Third PLC
- 10.0.0.255 PLC Programmer TCP or host

Also, in this case, set the subnet mask, gateway IP address, and name server IP address to 0.0.0.0.

#### Note

***If the isolated network is ever connected to another network, the IP addresses 10.0.0.1 through 10.0.0.255 must not be used and the subnet mask, gateway IP address, and name server IP address must be assigned by the network administrator.*** The IP addresses must be assigned so that they are compatible with the connected network. Refer to Chapter 5, “Network Administration Support”, for more information on addressing.

See also the section “Determining If an IP Address Has Already Been Used” in Procedure 4.

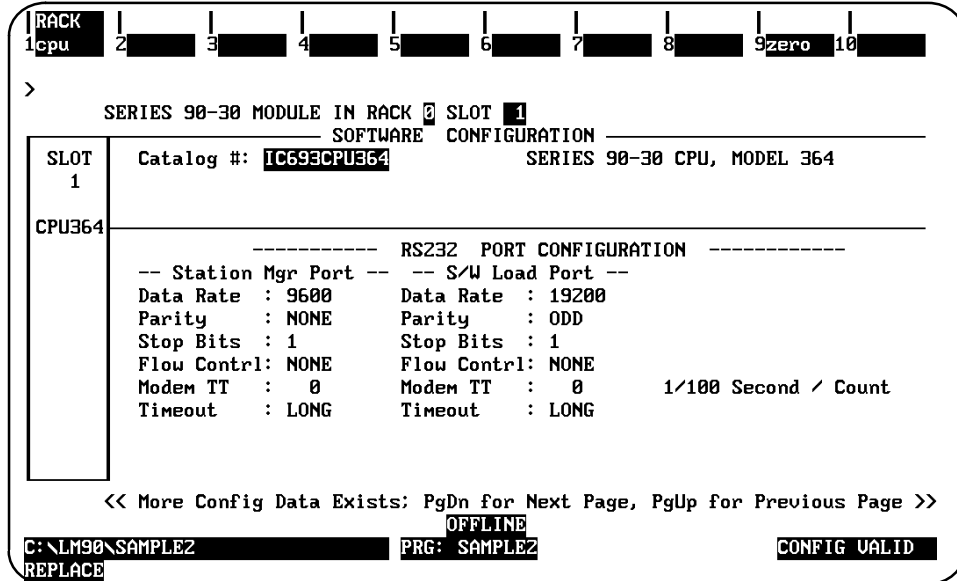
#### Note

Do not use the 80 bits assigned to the LIS bits and Channel Status bits for other purposes or your data will be overwritten.

**Status Length:** This is the sum of the LIS bits and the Channel Status bits. This value is automatically set to either 80 bits (for %I and %Q Status address locations) or 5 words (for %R, %AI, and %AQ Status address locations).

**Network Time Sync SNTP:** Selection of the method used to synchronize the Ethernet clocks over the network. Press the Tab key to choose from possible choices. Currently the choices are DISABLED (for no network time synchronization) and ENABLED (for synchronization to broadcast/multicast mode (mode 5) SNTP servers on the network).

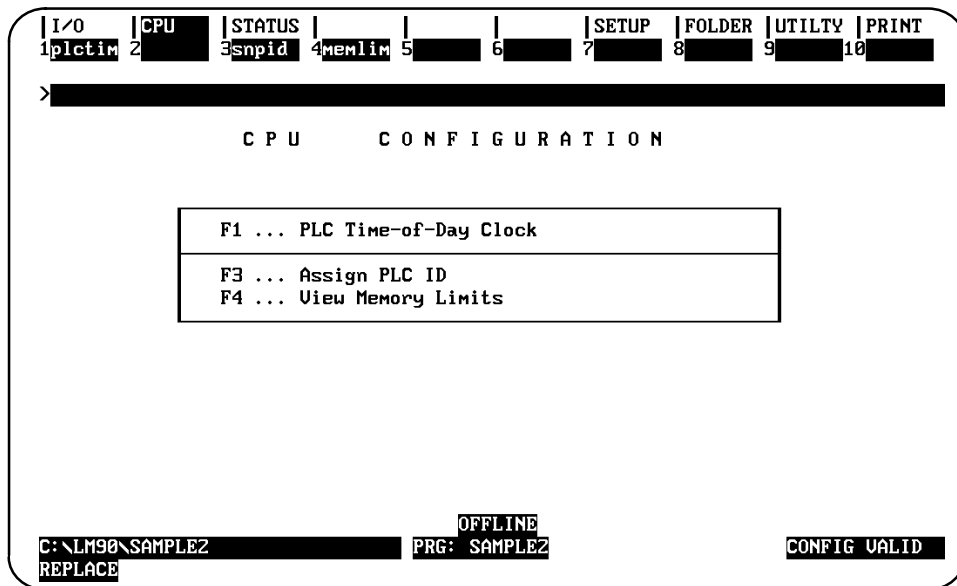
After you have configured the IP address, press PgDn to configure the parameters on the following screen:



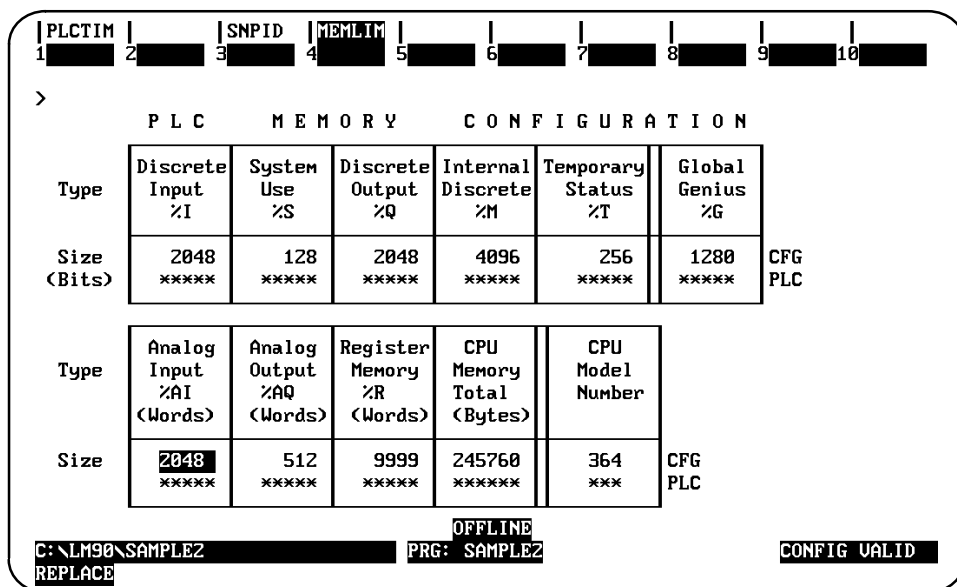
Refer to the *TCP/IP Ethernet Communications for Series 90™ PLCs* manual (GFK-1541) for information about the parameters shown on this screen. All TCP/IP Ethernet configuration information for this CPU is discussed in that manual.

## User-Configurable Memory Limits for 351 and Higher CPUs

After you have done the standard configuration on a 351 and higher CPU, you can change the default memory limit for %AI, %AQ, and %R references. To change these defaults, from the CPU part of I/O configuration, press the **Escape** key (usually twice) till you see "CPU" as the F2 selection. Then press **F2**. Your screen should look similar to this:



Then press **F4 (memlim)** to go to the PLC Memory Configuration screen:



NOTE: Asterisks (\*\*\*\*\*) only appear when the programmer is offline.

For CPU models that support this feature, the three Size fields under %AI, %AQ, and %R can be edited. Just type the value in each field. By default, the cursor will be on the



---

%AI field. Press the arrow keys to go to the %AQ and %R fields and type the value you want in each, then press the **Escape** key to save your changes. The following guidelines and restrictions apply:

1. The user-configurable memory is supported **only** for the 351, 352, 360, 363, and 364 CPUs.
2. Only the %AI, %AQ, and %R references are user-configurable.
3. Adjustments must be in increments of 128 words except for the default of %R (9999 words).
4. The default settings are %AI (2048 words) ; %AQ (512 words) ; %R (9999 words).
5. The following maximum and minimum limitations apply:
  - Maximum of 8192 for %AI and %AQ for Release 9.02 Logicmaster
  - Maximum of 16384 for %R for Release 9.02 Logicmaster
  - Minimum of 128 for each editable reference type (%AI, %AQ, and %R)

### Note

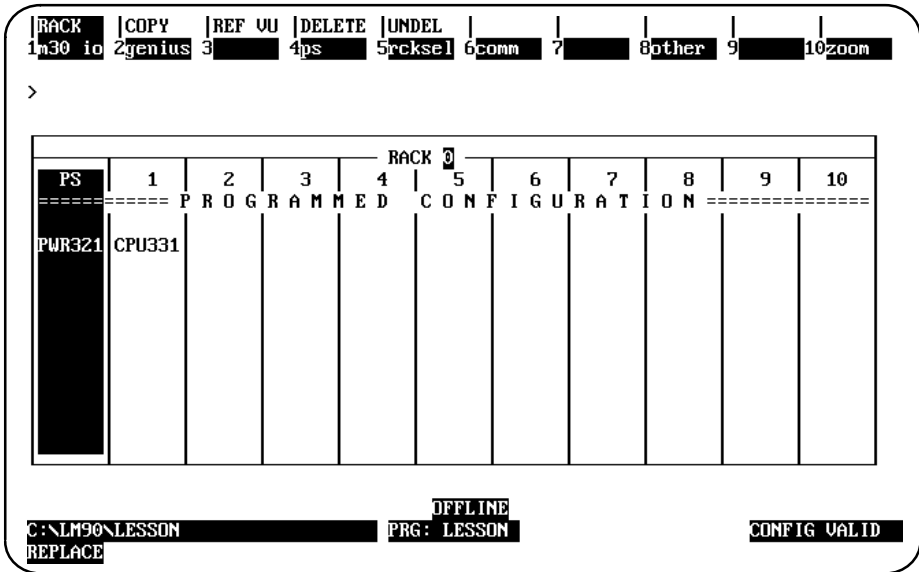
**If you are installing a version of Logicmaster later than Release 9.02, check the IPI for increased maximum limits.**

## Section 4: Selecting the Base Rack

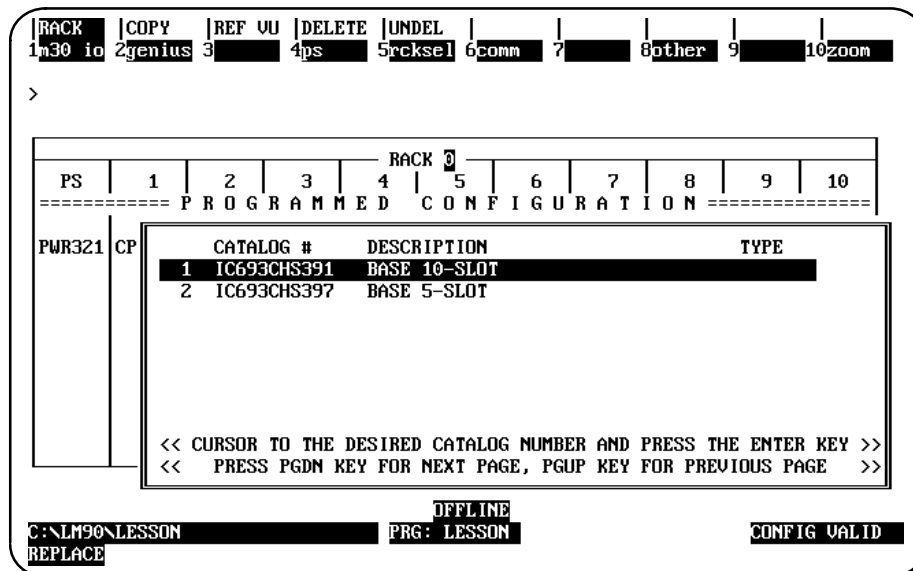
The Model 331 and higher CPUs are the non-backplane-mounted CPUs for the Series 90-30 PLC. These model CPUs are plug-in modules, which can be removed from the baseplate.

These model CPUs can have expansion racks. Up to five subsystem racks are allowed. The main CPU rack is rack 0, and expansion racks are numbered 1 through 4 (8 in a 351 system). (All racks default to be 10-slot racks.) The size of each rack can be changed on an individual rack basis, allowing a mixture of 5-slot and 10-slot racks in a Series 90-30 PLC configuration. The Up/Down cursor keys and Page Up/Down keys are used to move between racks in ascending or descending order, respectively.

- 1. Initially, a 10-slot rack is displayed, as shown below for the Model 331 PLC.



- 2. When **Rack Selection (F5)** is pressed from any slot of rack 0 of a 5-slot 10-slot Model 331 CPU, a list of catalog numbers and racks is displayed:



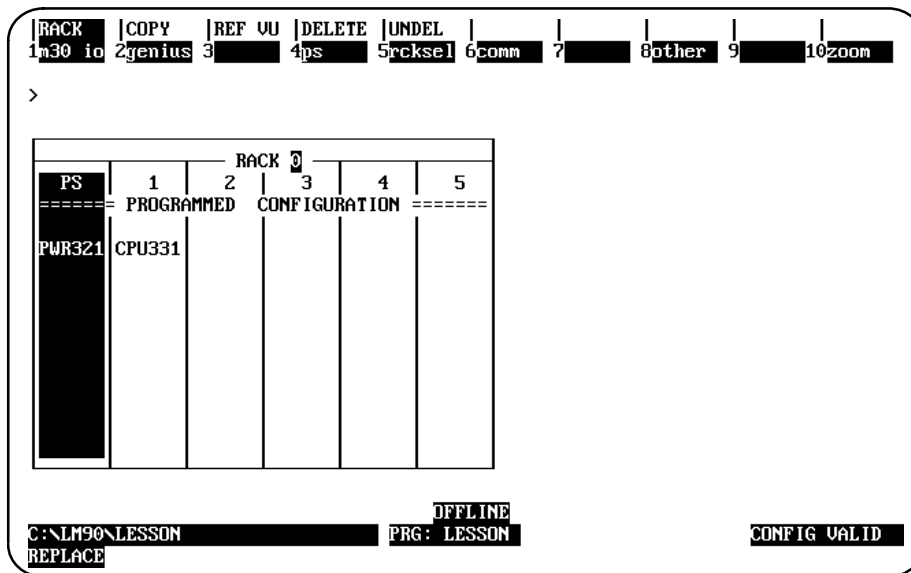
If the **Escape** key or **ALT-A** (Abort) is pressed while this list is displayed, the rack screen is redisplayed.

- 3. Position the cursor on the catalog number of the desired rack, and press the **Enter** key.

- When switching from a 10-slot rack to a 5-slot rack, a validation check is performed to see if there are any configured modules in slots 6 through 10. If there are, the following message is displayed:

Slots greater than 5 will be lost, continue REPLACE ? (Y/N)

If **N** (No) is entered, the current rack will remain displayed. If **Y** (Yes) is entered, the system will automatically delete slots 6 through 10 before changing the rack size. The 5-slot rack appears as shown below:



When changing the rack selection, note these rules:

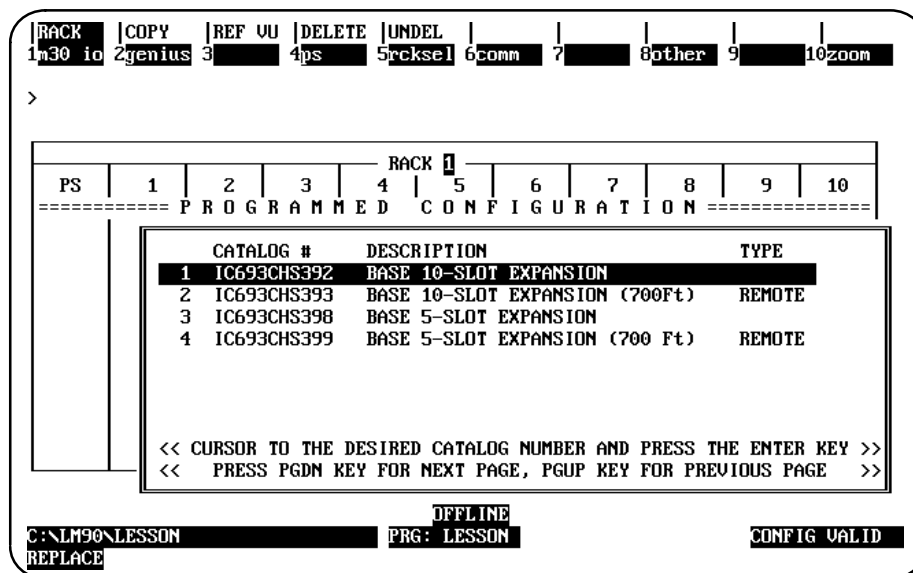
- The rack cursor will always be displayed on the Power Supply module after a successful rack change.
- Selecting the catalog number of the same size rack currently configured does not change the display.
- When a rack switch is made, the modules will have the same slot location they had in the previous rack.
- Rack 0, as well as any expansion rack, can be configured to be a 5-slot rack.
- When switching from a 5-slot rack to a 10-slot rack, no validation check is performed.
- When a previously configured 10-slot rack is changed to a 5-slot rack and then back to a 10-slot rack, the modules initially configured for the 10-slot rack are not restored.

## Base Rack Power Supply

The base rack power supply is automatically configured when the Model 331 or 341 CPU is selected. The Power Supply (F4) function cannot be used to change the base rack (rack 0) power supply. You must first press **Zoom (F10)** to zoom into the power supply, and then press **Power Supply (F1)** to change the base rack (rack 0) power supply.

## Selecting an Expansion Rack

Expansion rack catalog numbers are different from the base rack catalog numbers for a Model 331 or higher CPU subsystem. When **Rack Selection (F5)** is pressed from any slot of rack 1 of a subsystem rack, a list of catalog numbers and expansion racks is displayed.

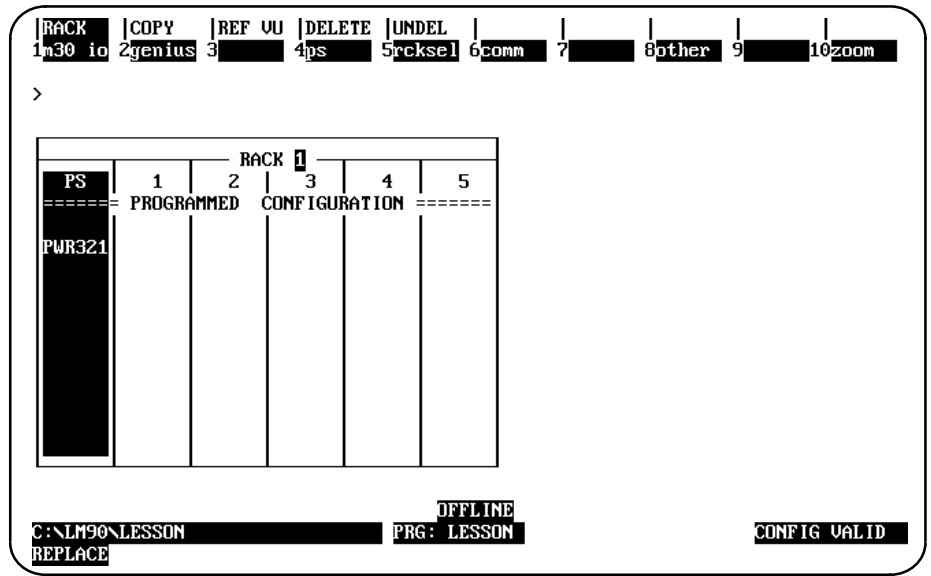


### Note

The I/O bus of the base rack is extended to the I/O bus of expansion racks via an I/O bus expansion cable.

To select an expansion rack (for example, the 5-slot expansion rack), position the cursor on the catalog number for that rack (for this example, IC693CHS398), and press the **Enter** key.

The following rack is displayed if a 10-slot expansion rack with no modules is switched to a 5-slot expansion rack.



### Note

No power supply is configured until an I/O module is configured or the Power Supply (F4) key is used to select one.

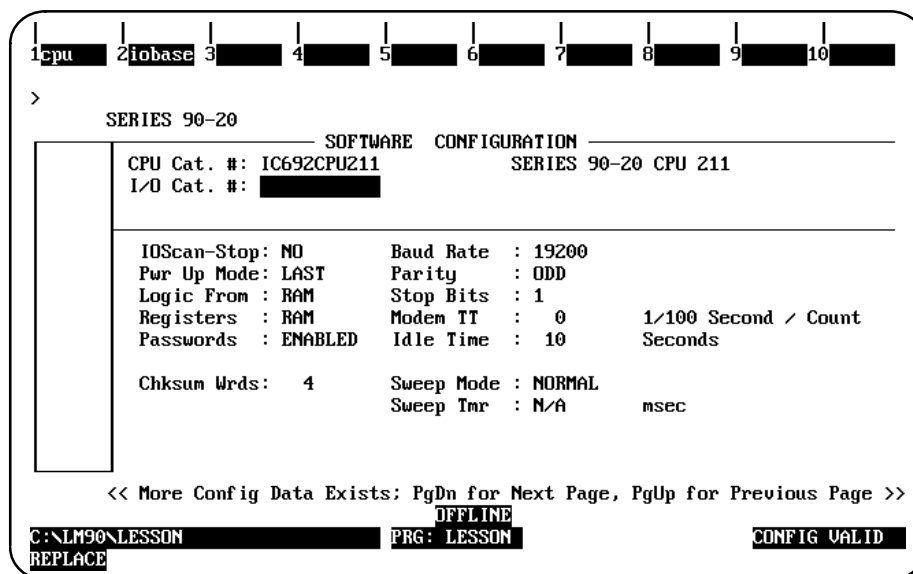
## Section 5: Configuring the Model 211 CPU

The I/O configuration of the CPU 211 is similar to the configuration of a 5-slot CPU 311, which has been configured with a power supply, CPU, generic discrete input, generic discrete output, and High Speed Counter modules. The reference address assignments for the CPU 211 I/O are fixed.

No rack screen is displayed for the 211 PLC.

1. Press **CPU (F1)** to display a list of catalog numbers and modules. Cursor to the desired CPU catalog number, and press the **Enter** key to select a new CPU.

If a Series 90-30 CPU rack is currently displayed, cursor to the CPU slot and press **Zoom (F10)**.



When the CPU 211 detail screen is first entered, the active field is the **I/O Catalog Number** field. This field is used for the catalog number of the I/O base. (Refer to the information below for configuring the I/O base.)

- A. If a different CPU is selected, the screen will display the detail for that CPU. If the new CPU is a CPU 211, the detail screen will be displayed and the active field will be the **I/O Catalog Number** field.
- B. If the active field is the **CPU Catalog Number** field, you may enter a new catalog number or edit the current catalog number. You cannot, however, leave this field unless it has a valid entry.

2. The *CPU Parameter* fields may be edited, as described in the following tables.

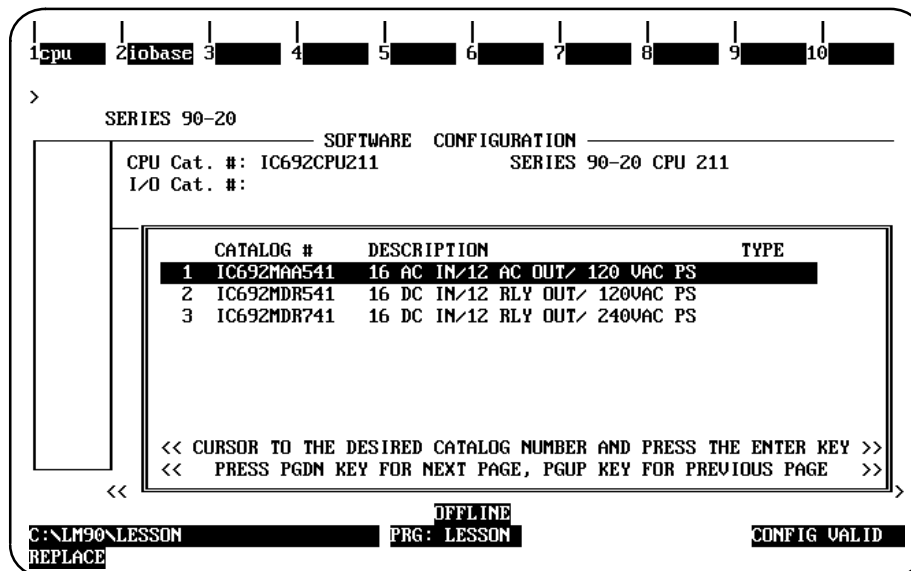
Parameter	Description
I/Oscan-Stop	Indicates whether the I/O is to be scanned while the PLC is in <b>STOP</b> mode. Choices are <b>YES</b> or <b>NO*</b> .
Power-Up Mode	Indicates which state the PLC is to be powered up in. Choices are <b>RUN</b> , <b>STOP</b> or <b>LAST*</b> . <b>LAST</b> indicates that the PLC will power up in the same mode it powered down in.
LogicFrom	Indicates whether the program logic is stored in <b>RAM*</b> , or in an additional, optional <b>EEPROM</b> .
Registers	Indicates whether the source of registers is <b>RAM*</b> or <b>EEPROM</b> .
Passwords	Indicates whether passwords are <b>ENABLED*</b> or <b>DISABLED</b> .
Baud Rate	Transmission rate in bits per second. Choices are <b>300</b> , <b>600</b> , <b>1200</b> , <b>2400</b> , <b>4800</b> , <b>9600</b> , or <b>19200*</b> .
Parity	Type of parity bits added to each word. Choices are <b>ODD*</b> , <b>EVEN</b> , or <b>NONE</b> .
Stop Bits	Communication uses at least one stop bit. Slower communication uses two stop bits. Choices are <b>1*</b> or <b>2</b> .
Modem Turnaround Time	Modem turnaround delay time counts. 1 count = 1/100 second. Choices are <b>0</b> to <b>255</b> (default = <b>0</b> ).
Idle Time	Maximum communication idle time from <b>1</b> to <b>60</b> seconds. (Default = <b>10</b> seconds)
Checksum Words	The number of words of user program to be checksummed per sweep. Choices are <b>0</b> through <b>32</b> , inclusive.
Sweep Mode	<p>Sweep mode can be either <b>NORMAL SWEEP</b> mode (<b>NORMAL*</b>) or <b>CONSTANT SWEEP</b> mode (<b>CNST SWP</b>). To change the settings, tap the <b>Down Cursor Movement</b> key (or <b>Down Arrow</b> key) till the cursor rests in the "Sweep Mode:" parameter. Then press the <b>Tab</b> key to change the selection from "NORMAL" to "CNST SWP" (if necessary—if you are just changing the time of the Constant Sweep, it may already say "CNST SWP"). Then tap the <b>Down Arrow</b> key once to move the cursor to the "Sweep Tmr" parameter and enter the number of milliseconds (from <b>5</b> to <b>200</b>) that you want. To store the changes to the folder, press the <b>Escape</b> key twice. Then store the configuration to the PLC and switch the PLC into <b>RUN</b> mode for this change to take effect.</p> <p>In <b>NORMAL SWEEP</b> mode, the PLC sweep executes as fast as possible. The overall PLC sweep time depends on the logic program plus the time required to compute the checksum.</p> <p>In <b>CONSTANT SWEEP</b> mode, the overall PLC sweep time is fixed.</p>
Sweep Timer	<p>A byte value which can be configured (in 1 millisecond increments) to be any value from <b>5</b> to <b>200</b> ms. (Default = <b>100</b>)</p> <p><b>Note:</b> The sweep timer is not configurable when the sweep mode is <b>NORMAL</b>. In this mode, the timer is set to <b>N/A</b> (Not Applicable), and you cannot cursor to the <b>SweepTimer</b> field.</p>

\* Default selection.

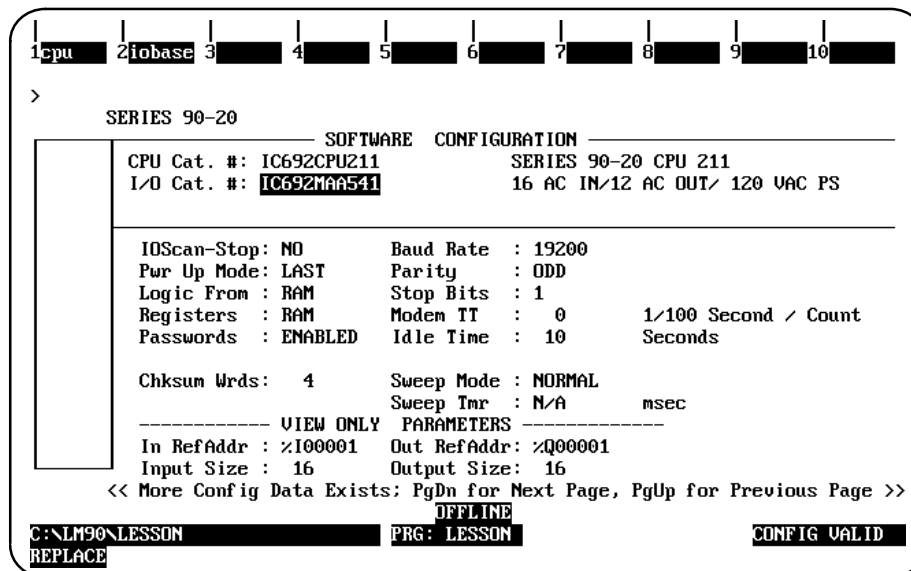


## I/O Base Selection for the CPU 211

1. The I/O Base (F2) softkey is used to select a new I/O base for the CPU 211. This key is only displayed after the CPU 211 has been selected.



2. Currently, there are three base modules available (IC692MAA541, IC692MDR541, and IC692MDR741). With the list of catalog numbers and modules displayed, cursor to the desired base and press the **Enter** key to select a new I/O base. The I/O base must be selected before a configuration can be saved or the *I/O Catalog #* field can be left.



### Note

“View Only Parameters” are not editable; therefore, you cannot cursor to them.

3. When a program is loaded from the PLC, if the module ID for each base configured in the system is within the range of 0 to 31 but does not match one of the three I/O bases displayed in the previous screen, the GENERICBASE shown below is displayed. The module ID for GENERICBASE (GENERIC 16 IN/12 OUT POWER SUPPLY) is 31.

```

1|cpu 2|iobase 3| 4| 5| 6| 7| 8| 9| 10|
>
SERIES 90-20
----- SOFTWARE CONFIGURATION -----
CPU Cat. #: IC692CPU211          SERIES 90-20 CPU 211
I/O Cat. #: GENERICBASE          GENERIC 16 IN/12 OUT PS
-----
IOScan-Stop: NO          Baud Rate : 19200
Pwr Up Mode: LAST       Parity      : ODD
Logic From : RAM         Stop Bits : 1
Registers  : RAM         Modem TT  : 0          1/100 Second / Count
Passwords  : ENABLED     Idle Time  : 10          Seconds
-----
Chksum Wrds: 4          Sweep Mode: NORMAL
                          Sweep Tmr  : N/A          msec
----- VIEW ONLY PARAMETERS -----
In RefAddr : %I00001     Out RefAddr: %Q00001
Input Size  : 16         Output Size: 16
<< More Config Data Exists; PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\LESSON          PRG: LESSON          CONFIG VALID
REPLACE

```

```

1|cpu 2|iobase 3| 4| 5| 6| 7| 8| 9| 10|
>
SERIES 90-20
----- SOFTWARE CONFIGURATION -----
CPU Cat. #: IC692CPU211          SERIES 90-20 CPU 211
I/O Cat. #: GENERICBASE          GENERIC 16 IN/12 OUT PS
-----
----- HIGH SPEED COUNTER -----
Count En   : DISABLED   Time Base  : 1000
Count Dir  : UP         Hi Limit   : +32767
Count Mode : CONTINU    Lo Limit   : +00000
Pld/strobe : PRELOAD    Pld Value  : +00000
Pld Filter : LOWFREQ    On Preset  : +32767
Cnt Filter : LOWFREQ    Off Preset  : +00000
Failure Mde: NORMAL
----- VIEW ONLY PARAMETERS -----
Ctrl/Status: %QI0033     HSC Data   : %AI001
%QI SIZE   : 16         %AI SIZE   : 15
<< More Config Data Exists; PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\LESSON          PRG: LESSON          CONFIG VALID
REPLACE

```

Refer to section 8, "Configuring an HSC or Embedded HSC," on page 10-50 for an explanation of the parameters and values displayed on this screen.

- 4. The module ID for UNKNOWNBASE (UNKNOWN/UNSUPPORTED I/O BASE) is 32. If the module ID received from the PLC for the I/O base module is greater than 31, the UNKNOWNBASE base is displayed.

```
1|cpu 2|iobase 3| 4| 5| 6| 7| 8| 9| 10|
>
SERIES 90-20
SOFTWARE CONFIGURATION
CPU Cat. #: IC692CPU211      SERIES 90-20 CPU 211
I/O Cat. #: UNKNOWNBASE     UNKNOWN/UNSUPPORTED IO BASE
-----
IOScan-Stop: NO           Baud Rate : 19200
Pwr Up Mode: LAST        Parity : ODD
Logic From : RAM          Stop Bits : 1
Registers : RAM           Modem TT : 0           1/100 Second / Count
Passwords : ENABLED      Idle Time : 10         Seconds
-----
Chksum Wrds: 4           Sweep Mode : NORMAL
                          Sweep Tmr : N/A           msec
-----
VIEW ONLY PARAMETERS
In RefAddr : %I00001     Out RefAddr: %Q00001
Input Size : 16          Output Size: 16
<< More Config Data Exists; PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\LESSON          PRG: LESSON          CONFIG VALID
REPLACE
```

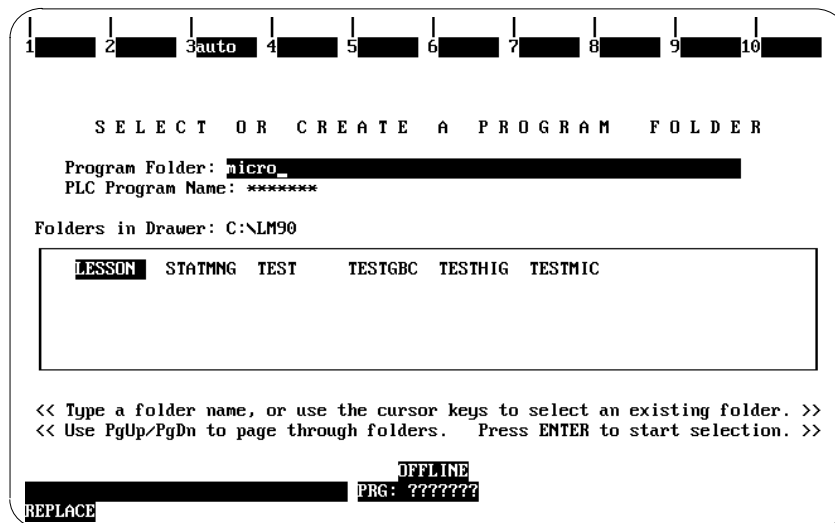
```
1|cpu 2|iobase 3| 4| 5| 6| 7| 8| 9| 10|
>
SERIES 90-20
SOFTWARE CONFIGURATION
CPU Cat. #: IC692CPU211      SERIES 90-20 CPU 211
I/O Cat. #: UNKNOWNBASE     UNKNOWN/UNSUPPORTED IO BASE
-----
HIGH SPEED COUNTER
Count En : DISABLED      Time Base : 1000
Count Dir : UP           Hi Limit : +32767
Count Mode : CONTINU     Lo Limit : +00000
Pld/strobe : PRELOAD     Pld Value : +00000
Pld Filter : LOWFREQ     On Preset : +32767
Cnt Filter : LOWFREQ     Off Preset : +00000
Failure Mde: NORMAL
-----
VIEW ONLY PARAMETERS
Ctrl/Status: %QI0033     HSC Data : %AI001
%QI SIZE : 16            %AI SIZE : 15
<< More Config Data Exists; PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\LESSON          PRG: LESSON          CONFIG VALID
REPLACE
```

## Section 6: Configuring a Micro PLC

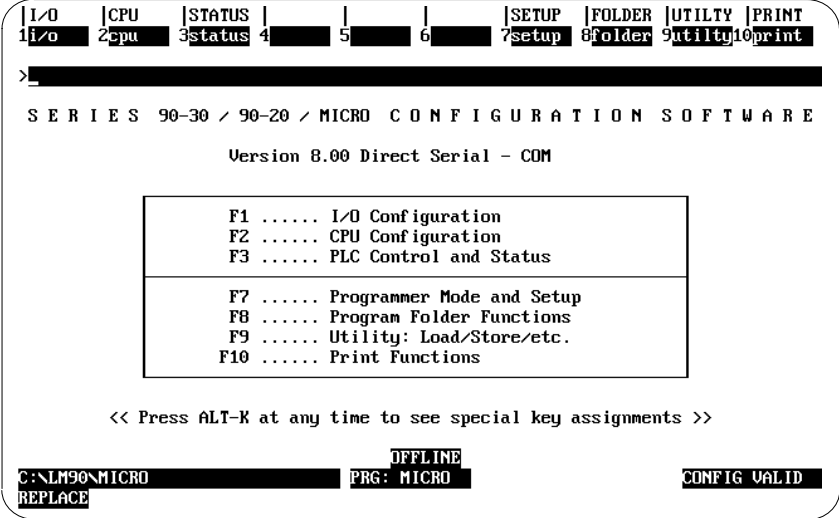
A Micro PLC has the CPU, power supply, a high speed counter (except for the ACIN/AC OUT models), inputs and outputs all built into one small device. You can configure the Micro with Logicmaster by following the steps shown below, but also refer to the *Series 90™ Micro Programmable Controller User's Manual* (GFK-1065) and the *Series 90™ Micro Programmable Logic Controller Self-Teach Manual* (GFK-1104A or later) for additional information.

To configure a Micro PLC, follow these steps:

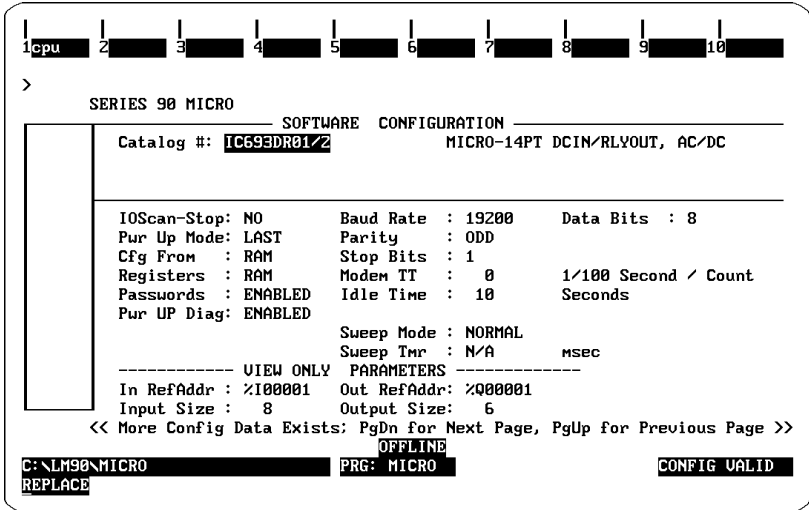
1. Press the **Micro** softkey (**Shift-F1**) from the Logicmaster 90-30 Main Menu (i.e., the first screen you see after entering Logicmaster). This highlights the word *Micro* at the top of the Logicmaster display showing you that it is selected.
2. Press the **Config** softkey (**F2**). After a short pause (during which the Logicmaster copyright screen appears), your screen display will change to one similar to the one shown below.



3. Type the name you want to call your new folder. (In the example shown above, “micro” was used.) Then press **Enter**. Logicmaster will then ask you if you want to “create new folder?” Type **Y** for **Yes**. Your screen will look like the one shown below.



4. Press the I/O softkey (F1). The following screen will appear.



5. If the Micro you are installing *has* the same Catalog # as the default selection, *then skip to step 7* discussed on the next page. If the Micro you are installing *does not* have the same Catalog # as the default selection, then press the CPU softkey (F1). The following screen will appear.

```

1cpu 2 3 4 5 6 7 8 9 10
>
SERIES 90 MICRO
SOFTWARE CONFIGURATION
Catalog #: IC693DR01/2 MICRO-14PT DCIN/RLYOUT, AC/DC

CATALOG # DESCRIPTION TYPE
4 IC693DR01/2 MICRO-14PT DCIN/RLYOUT, AC/DC
5 IC693UAA003 MICRO-14PT ACIN/ACOUT, AC PS
6 IC693UAL006 MICRO-23PT ANALOG, DCIN/OUT
7 IC693UAA007 MICRO-28PT ACIN/ACOUT, AC PS
8 IC693CPU311 BASE 5-SLOT WITH CPU 311 8 MHZ
9 IC693CPU313 BASE 5-SLOT WITH CPU 313 10 MHZ
10 IC693CPU321 BASE 10-SLOT WITH CPU 311 8 MHZ
11 IC693CPU323 BASE 10-SLOT WITH CPU 313 10 MHZ
<< CURSOR TO THE DESIRED CATALOG NUMBER AND PRESS THE ENTER KEY >>
<< PRESS PGDM KEY FOR NEXT PAGE, PGUP KEY FOR PREVIOUS PAGE >>
<<
OFFLINE
C:\LM90\MICRO PRG: MICRO CONFIG UALID
REPLACE

```

### Note

Step 6 is **only necessary** if the Micro you are installing **does not** have the same Catalog # as the default selection.

6. Move the cursor up or down using the **Cursor Movement** (or **Arrow**) keys to the correct selection. (When scrolling through the list, you may notice that other options include switching to a 90-20 or a 90-30.) When you have your cursor on the correct Catalog #, press **Enter** which will take you back to the screen shown on the top of this page.

```

1cpu 2 3 4 5 6 7 8 9 10
>
SERIES 90 MICRO
SOFTWARE CONFIGURATION
Catalog #: IC693DR01/2 MICRO-14PT DCIN/RLYOUT, AC/DC

IOScan-Stop: NO Baud Rate : 19200 Data Bits : 8
Pur Up Mode: LAST Parity : ODD
Cfg From : RAM Stop Bits : 1
Registers : RAM Modem TT : 0 1/100 Second / Count
Passwords : ENABLED Idle Time : 10 Seconds
Pur UP Diag: ENABLED

Sweep Mode : NORMAL
Sweep Trm : N/A msec
----- UIEW ONLY PARAMETERS -----
In RefAddr : %I00001 Out RefAddr: %Q00001
Input Size : 8 Output Size: 6
<< More Config Data Exists: PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\MICRO PRG: MICRO CONFIG UALID
REPLACE

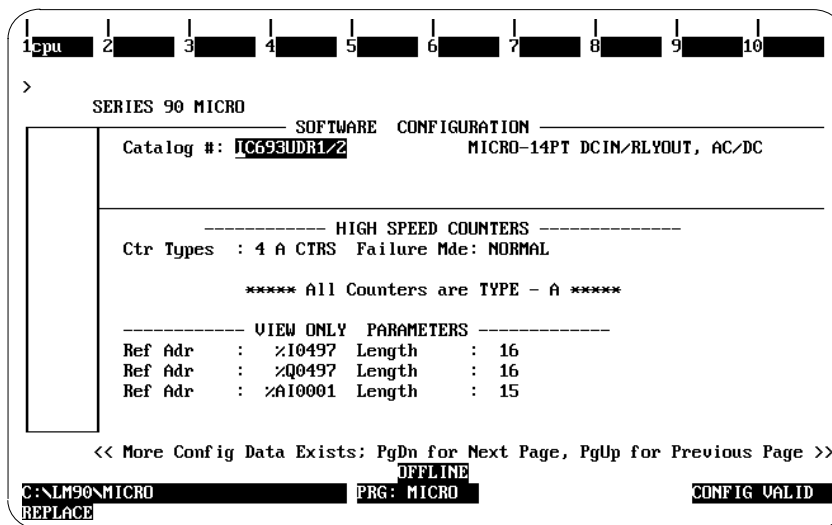
```

### Note

Refer to Chapter 4, "Configuration," in the *Series 90™ Micro Programmable Controller User's Manual* (GFK-1065) for parameter descriptions for each of the following screen.

7. On the screen shown above, you can move your cursor from field to field by pressing the **Cursor Movement** (or **Arrow**) keys. When you are in the field you want to modify, you can either type in your choice or press the **Tab** key to scroll through the

available selections (or **Shift-Tab** to reverse the direction of the scrolling). From this screen, press the **Page Down** key to take you to the screen shown below.



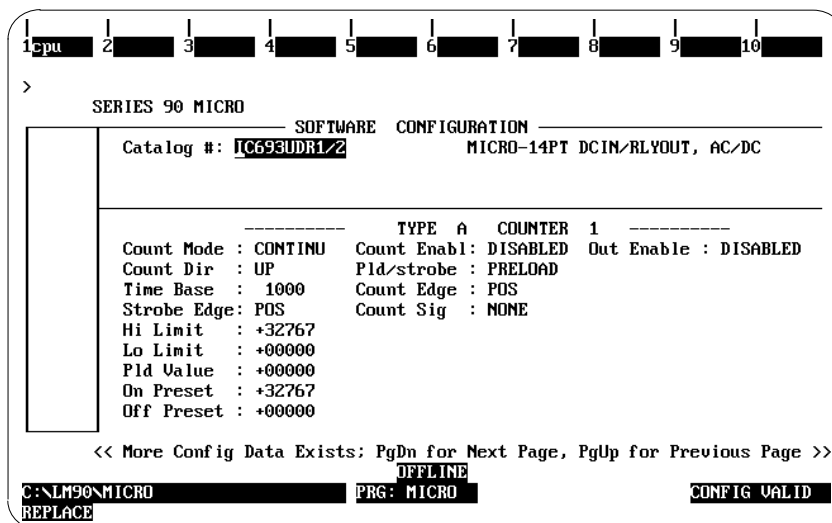
**Notes:**

- A. %AQ reference addresses will display for DC OUT models only.
- B. The High Speed Counter screens will not appear for ACIN / AC OUT models (since they do not have High Speed Counters).

**Note**

*If you are configuring the HSC (high speed counter) mode B1- 3, A4, move the cursor to the **Ctr Types** field, then press the **Tab** key once to toggle the default selection to B1- 3, A4. Counters B1- 3 are then configured on one screen, A4 on another, pressing the **Page Down** key to advance screens.*

- 8. You can move your cursor from field to field by pressing the **Cursor Movement (or Arrow)** keys. When you are in the field you want to modify, you can either type in your choice or press the **Tab** key to scroll through the available selections (or **Shift-Tab** to reverse the direction of the scrolling). From this screen, press the **Page Down** key to take you to the screen shown below.



## Note

Additional fields, **PWM Out %Q1** and **Pu1 Out %Q2**, will display on DC OUT models. **This applies to all of the following screens.**

- You can move your cursor from field to field by pressing the **Cursor Movement** (or **Arrow**) keys. When you are in the field you want to modify, you can either type in your choice or press the **Tab** key to scroll through the available selections (or **Shift-Tab** to reverse the direction of the scrolling). From this screen, press the **Page Down** key to take you to the screen shown below.

```

1 2 3 4 5 6 7 8 9 10
CPU
>
SERIES 90 MICRO      SOFTWARE CONFIGURATION
Catalog #: IC693UDR1Z2      MICRO-14PT DCIN/RLYOUT, AC/DC
-----
Count Mode : CONTINU      TYPE A      COUNTER 2      -----
Count Dir  : UP           Count Enabl: DISABLED  Out Enable : DISABLED
Time Base  : 1000        Pld/strobe : PRELOAD
Strobe Edge: POS         Count Edge  : POS
Hi Limit   : +32767      Count Sig   : NONE
Lo Limit   : +00000
Pld Value  : +00000
On Preset  : +32767
Off Preset : +00000

<< More Config Data Exists: PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\MICRO      PRG: MICRO      CONFIG VALID
REPLACE

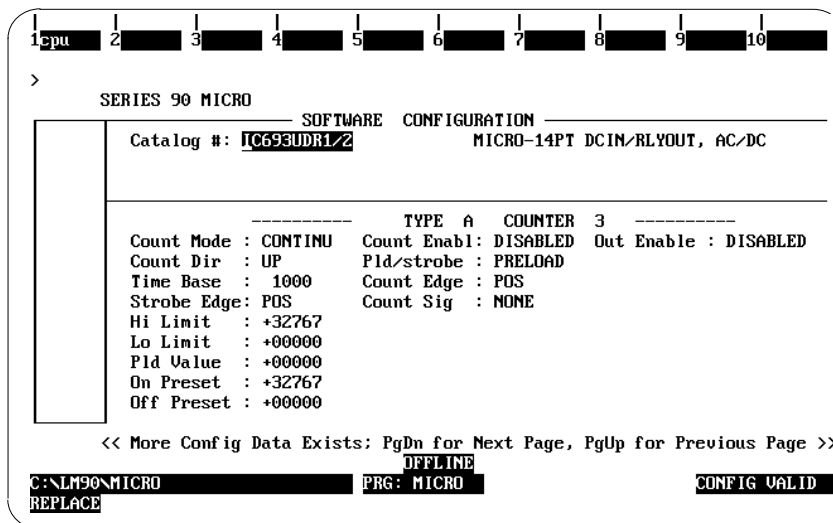
```

## Note

For parameter and other detailed configuration information about the High Speed Counter, refer to Chapter 6, “High Speed Counters,” in the *Series 90™ Micro Programmable Controller User’s Manual* (GFK-1065)—for newer models refer to GFK-1065D or later.



- 10. You can move your cursor from field to field by pressing the **Cursor Movement** (or **Arrow**) keys. When you are in the field you want to modify, you can either type in your choice or press the **Tab** key to scroll through the available selections (or **Shift-Tab** to reverse the direction of the scrolling). From this screen, press the **Page Down** key to take you to the screen shown below.



### Note

For detailed parameter descriptions, refer to Chapter 4, “Configuration,” in the *Series 90™ Micro Programmable Controller User’s Manual* (GFK-1065)—for newer models refer to GFK-1065D or later. For parameter and other detailed configuration information about the High Speed Counter, refer to Chapter 6, “High Speed Counters,” in the *Series 90™ Micro Programmable Controller User’s Manual* (GFK-1065)—for newer models refer to GFK-1065D or later.

- Use the same techniques to set up this screen as the previous ones; i.e., you can move your cursor from field to field by pressing the **Cursor Movement** (or **Arrow**) keys. When you are in the field you want to modify, you can either type in your choice or press the **Tab** key to scroll through the available selections (or **Shift-Tab** to reverse the direction of the scrolling). From this screen, press the **Page Down** key to take you to the screen shown below.

```

1 |cpu| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
>
SERIES 90 MICRO
SOFTWARE CONFIGURATION
Catalog #: IC693UDR1Z2 MICRO-14PT DCIN/RLYOUT, AC/DC
-----
Count Mode : CONTINU      TYPE A COUNTER 4
Count Dir  : UP           Count Enabl: DISABLED Out Enable : DISABLED
Time Base  : 1000         Pld/strobe : PRELOAD
Strobe Edge: POS         Count Edge  : POS
Hi Limit   : +32767       Count Sig   : NONE
Lo Limit   : +00000
Pld Value  : +00000
On Preset  : +32767
Off Preset : +00000
<< More Config Data Exists: PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\MICRO PRG: MICRO CONFIG VALID
REPLACE

```

- After you have set up the final counter, press the **Escape** key to save your configuration. This will also return you to the main menu of the Configurator package.

### Expansion Units for the Micro PLC

For 28-point Micro PLCs, you can configure expansion units (up to 4 expansion units per Micro PLC). Refer to Chapter 4, “Configuration,” in the *Series 90™ Micro Programmable Controller User’s Manual* (GFK-1065)—for newer models refer to GFK-1065D or later. For parameter and other detailed configuration information about the High Speed Counter, refer to Chapter 6, “High Speed Counters,” in the *Series 90™ Micro Programmable Controller User’s Manual* (GFK-1065)—for newer models refer to GFK-1065D or later.

### Micro PLC Memory Configuration Limits

Memory Type	Memory Limit
%R	256 words
%AI	128 words
%AQ	128 words
%I	512 bits
%Q	512 bits
%G	1280 bits
%M	1024 bits
%T	256 bits
%S	128 bits

#### Note

These limits are specific to Micro PLCs.

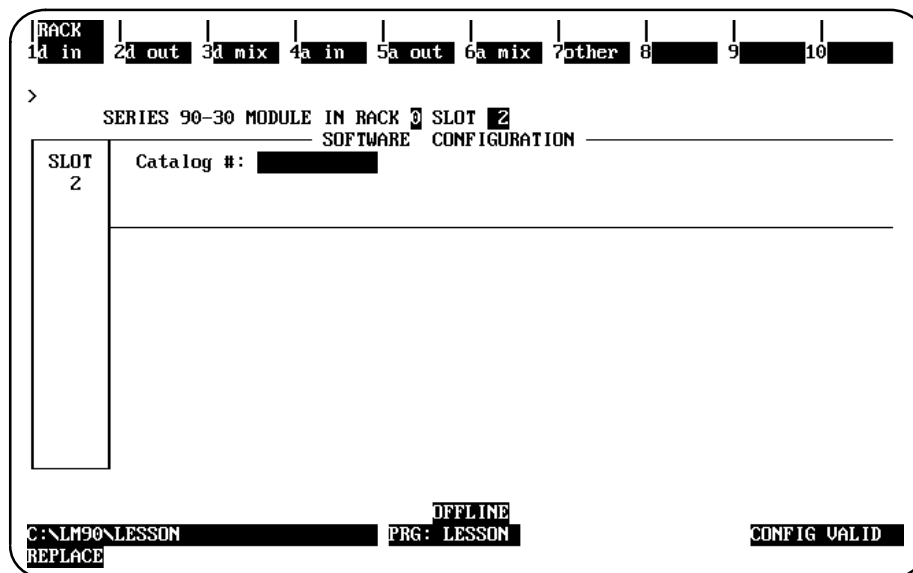
## Configuration Rules and Miscellaneous Information

1. When replacing the CPU (not a CPU211) of a non-default configuration with a SERIES 90 Micro, the additional prompt, “Existing I/O Configuration will be lost, Continue REPLACE ? (Y/N)” will be displayed. Confirming the prompt will cause the replacement to occur; otherwise, the replacement will not occur.
2. When replacing a CPU211 with a SERIES 90 Micro, no additional prompts will be displayed.
3. When replacing a SERIES 90 Micro with another CPU (not a CPU211), the resulting configuration will consist of the following: (1) a PS in rack 0, slot 0; (2) the new CPU in rack 0, slot 1; (3) an 8-pt. Discrete Input module in rack 0, slot 2; (4) an 8-pt Discrete Output module in rack 0, slot 3, (5) a 9030 HSC in rack 0, slot 4, and (6) all other racks unconfigured.
4. If you select the **MICRO** softkey from the Logicmaster 90 main menu and then select an existing folder that contained either a 9030 or the 9020 configuration, then that existing configuration will be displayed when the **I/O** softkey is pressed from the Configurator main menu.
5. If you select either the **90-20** or **90-30** softkey from the Logicmaster 90 main menu and then select an existing folder that contained a SERIES 90 Micro configuration, then that existing configuration will be displayed when the **I/O** softkey is pressed from the Configurator main menu.

## Section 7: Configuring 90-30 I/O Modules

To configure a 90-30 I/O module:

1. Move the cursor to the slot where the module will be located and press Module 30 I/O(F1).



2. Select the module type by pressing the corresponding function key listed below.

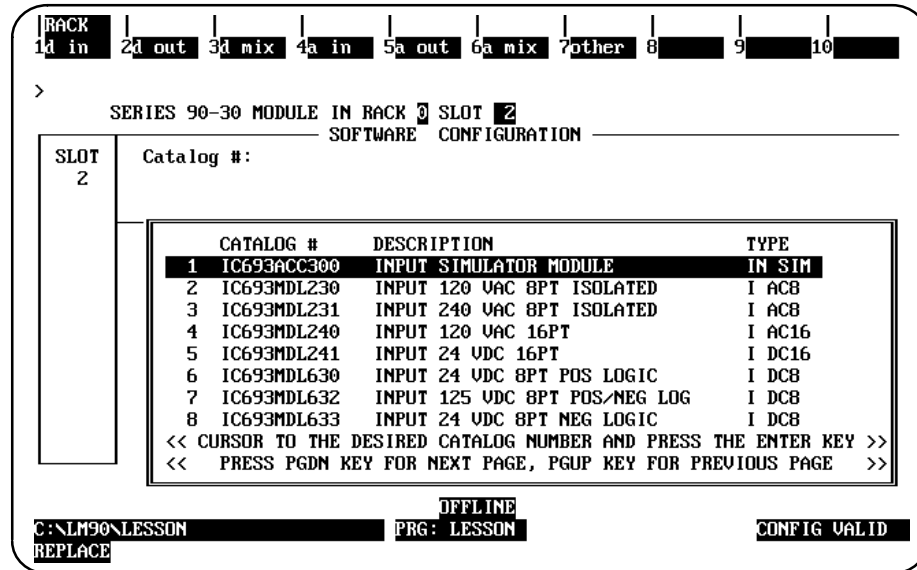
Function Key	Module Type	Description
F1	d in	Discrete input module.
F2	d out	Discrete output module.
F3	d mix	Discrete mixed module.
F4	a in	Analog input module.
F5	a out	Analog output module.
F6	a mix	Analog mixed module
F7	other	Other modules if available (see Note below)

### Note

Other module types are currently unavailable. This key is reserved for possible future use.

Press **ALT-H** to display help information for further explanations.

3. The next screen lists the catalog numbers and available modules of the type you selected. For example, when **F1** is pressed, the following screen is displayed.



4. Select a module by moving the cursor down to any entry. Additional entries may be displayed by pressing the **Page Down** key. Then, press the **Enter** key.
5. Move the cursor to the module's reference address. The next highest available reference address is automatically configured.

If you want to use a different reference address, enter it here. For a discrete module, the reference address must begin on a byte boundary. (A byte boundary is a reference address which is a multiple of 8 + 1; e.g., 1, 9, 17, or 25.) The software will automatically adjust an incorrect entry to begin on a byte boundary. A message is displayed when the highest available address has been assigned.

6. Press **Rack** (**Shift-F1**) or the **Escape** key to return to the rack display.

### Note

12-point modules do not use consecutive address bits in PLC memory, but they use bits 1 through 6 and 9 through 14. For example, if %Q0001 is assigned to a 12-point output module, %Q0001 through %Q0006 and %Q0009 through %Q0014 will be associated with the output point.

## Configuring Generic I/O Modules

If a specific catalog number does not exist for a particular module, the configuration can assign a generic reference address to that module.

The following table lists the catalog numbers to be used to configure modules generically and the function key that would be pressed in order to select a particular generic module.

**Table 10-2. Generic I/O Module Configuration**

Generic Catalog Number	Generic Description	Function Key
INPUT8 INPUT16 INPUT32 INPUT64	Generic Input 8 PT Generic Input 16 PT Generic Input 32 PT Generic Input 64 PT	F1 (d in)
OUTPUT5/8 OUTPUT12/26 OUTPUT32 OUTPUT64	GenericOutput5/8PT GenericOutput12/16PT Generic Output 32 PT Generic Output 64 PT	F2 (d out)
ALGIN4	Generic Input Analog 4 CH	F4 (a in)
ALGOUT2 ALGOUT4	Generic Output Analog 2 CH Generic Output Analog 4 CH	F5 (a out)
INOUT8 INOUT16 INOUT32 INOUT64	Generic Input/Output 8 PT GenericInput/Output16PT GenericInput/Output32PT GenericInput/Output64PT	F3 (d mix)

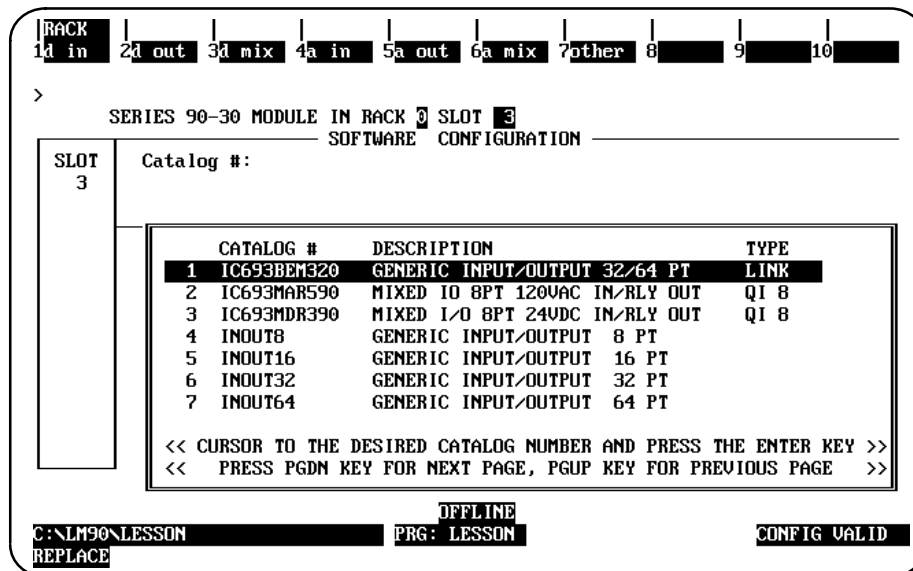
### Note

Do not configure Horner Electric analog or special modules as Generic. Configure them as Foreign (third party) module so that the required soft switch configuration data can be entered. For information about configuring a third party module, refer to page 10-99.

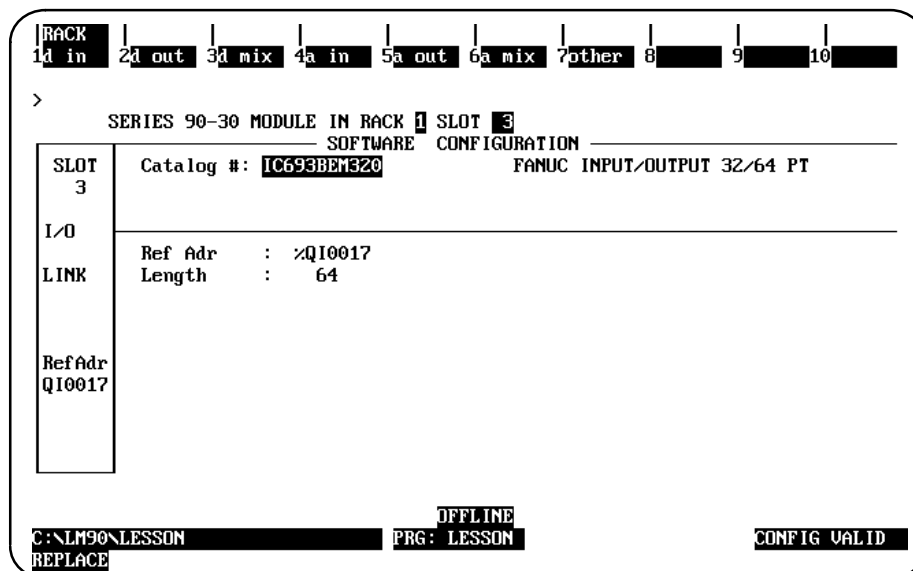
## Configuring an I/O Link Interface Module

The Fanuc I/O link modules are grouped under the discrete mixed modules. To configure a Fanuc I/O link module on the I/O Configuration Rack screen:

1. Move the cursor to the desired rack and slot location. The slot may be either unconfigured or previously configured.
2. Press Module 30 I/O (F1) and then Discrete Mixed (F3) from the I/O Configuration Rack screen to display a list of catalog numbers and discrete mixed modules.



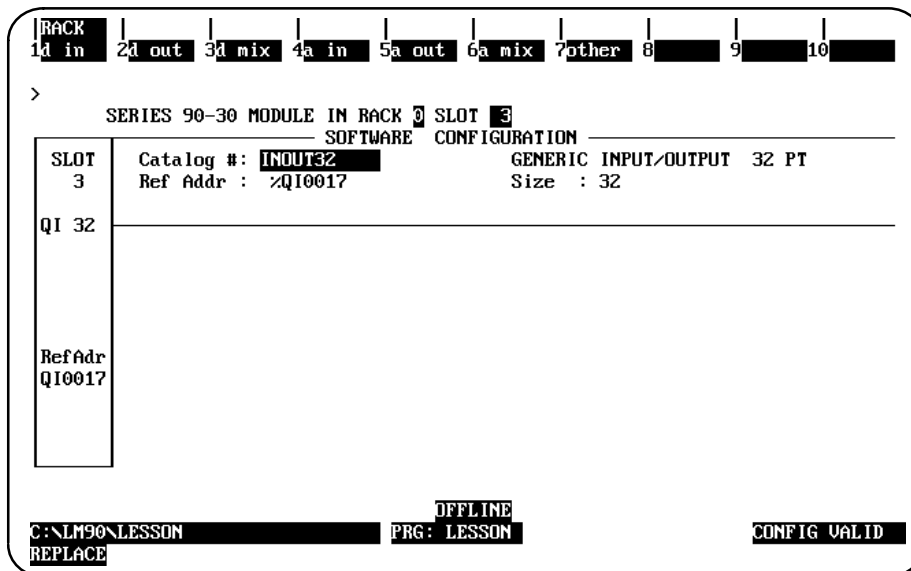
3. Position the cursor on the catalog number, IC693BEM320, and press the Enter key. The following detail screen is displayed.



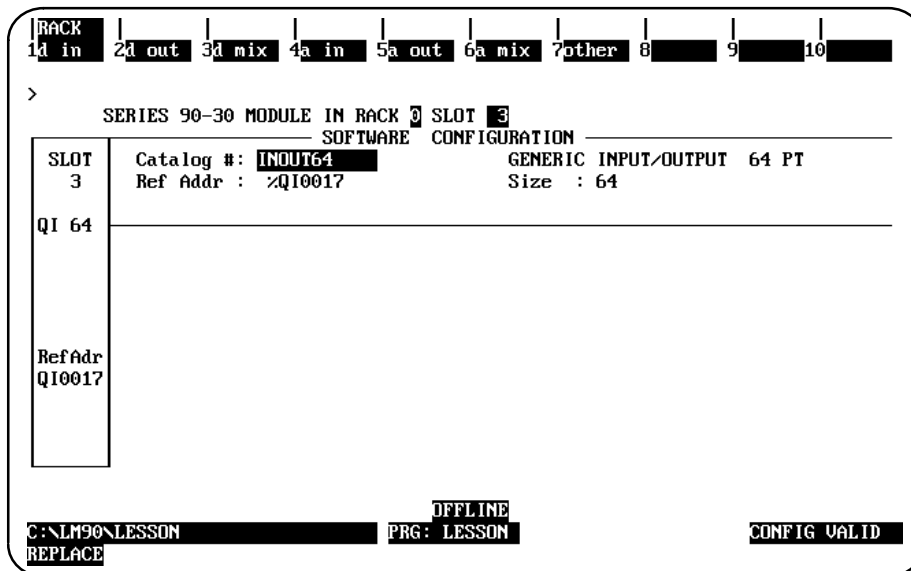
On this screen, the choices for the I/O address size are 32 and 64. The default value is 64.



- 4. If you select 32 as the reference size, the module will occupy 32 discrete inputs and 32 discrete output points. This is the same as if you select the catalog number INOUT32 for the Generic Input/Output 32-Point Module.



- 5. If you select 64 as the reference size, the module will occupy 64 discrete inputs and 64 discrete output points. This is the same as if you select the catalog number INOUT64 for the Generic Input/Output 64-Point Module.



### Configuring an I/O Link Master Module

Complete configuration instructions along with parameter descriptions for this module are in Chapter 3, "Configuration," of the *Series 90-30 I/O Link Master Module* manual (GFK-0823A or later).

## Section 8: Configuring an HSC or Embedded HSC

The Series 90-30 High Speed Counter module may be configured to count either up or down, to count both up and down, or to count the difference between two changing values. The module is configured for the application to function in one of three ways:

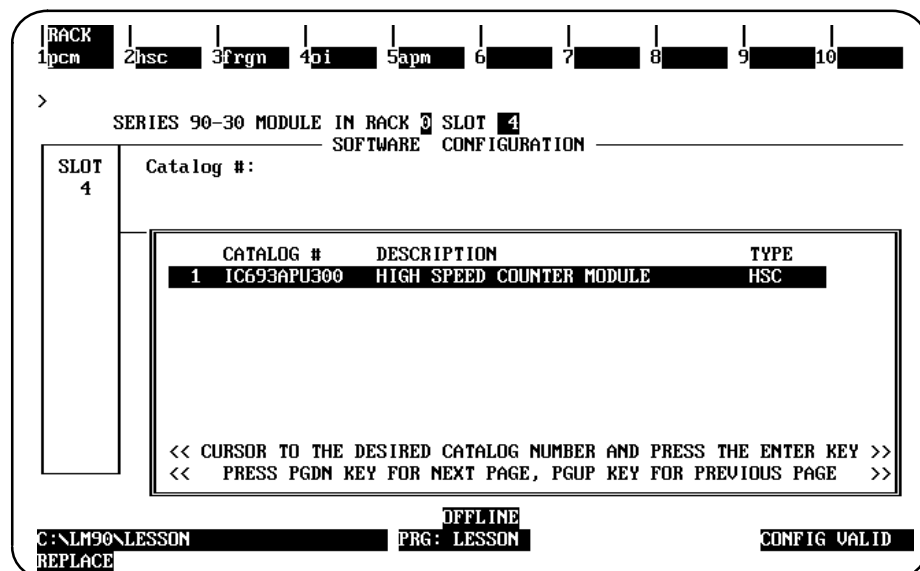
- As four 16-bit counters. Each of the four counters may independently count either up or down. This module configuration is referred to as “Type A.” Each Type A counter has 3 inputs and 1 output.
- As two 32-bit counters. Each may independently operate in **UP/DOWN**, **PULSE/DIRECTION**, or **A QUAD B** mode. This configuration is referred to as “Type B.” Each Type B counter has 6 inputs and 2 outputs.
- As one 32-bit differential counter, which can operate in **UP/DOWN**, **PULSE/DIRECTION**, or **A QUAD B** mode. This configuration is suitable for applications requiring motion control, differential counting, or homing capability. This counter uses all 12 of the module’s inputs and all 4 outputs.

For more information on the High Speed Counter module, refer to GFK-0293.

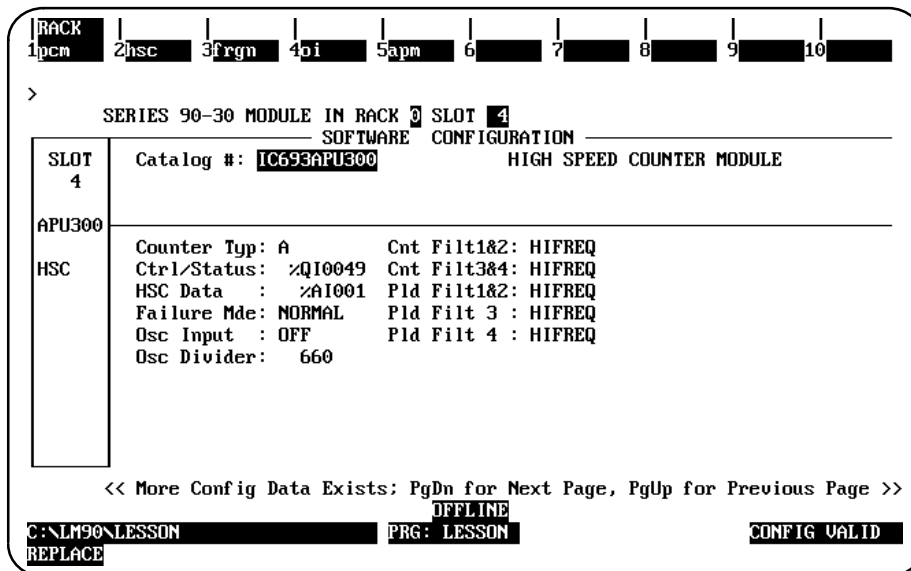
The High Speed Counter module accepts 12 configurable input signals, and provides 4 output signals for counter operations. Many additional features can be changed in the configuration software for the application program.

To configure the High Speed Counter module:

1. Move the cursor to the slot where the module will be located.
2. Press **Other (F8)** and then **High Speed Counter (F2)** to locate a High Speed Counter Module in the slot.



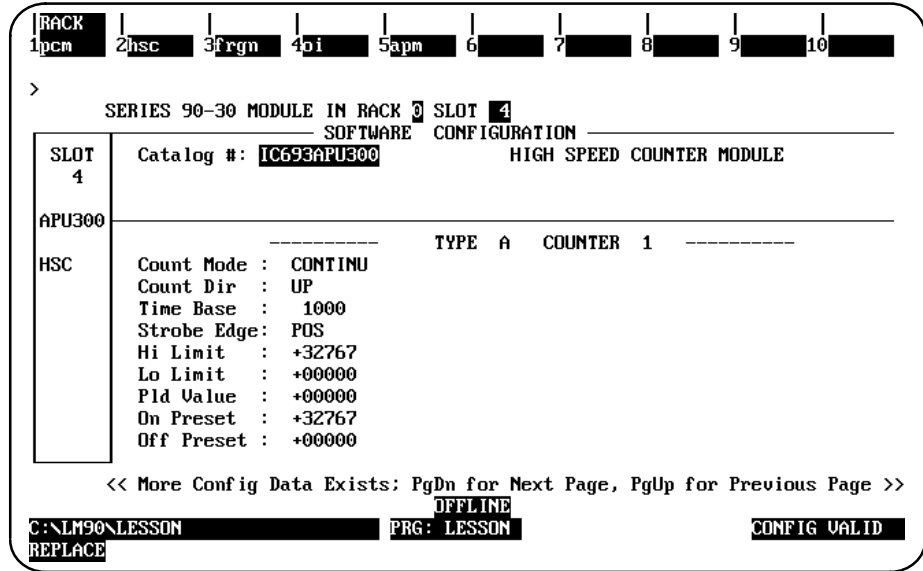
- Press the **Enter** key to enter the catalog number shown in reverse video. After pressing the **Enter** key, the following screen is displayed. This screen displays the module configuration parameters for a Type A counter.



Parameter	Description
Counter Type	Select the desired counter type by entering <b>A*</b> , <b>B</b> , or <b>C</b> ; or, press the <b>Tab</b> key.
Control/Status	Select the %QI reference location for discrete input data that the High Speed Counter Module sends to the PLC, and discrete output data that the PLC sends to the HSC Module. The length of the discrete data is fixed at 16 bits (2 bytes).  Discrete input data consists of status information, such as output status, strobe input status, preload input status, home input status (for type C), and module ready status.  Discrete output data consists of control information, such as output enable/disable, strobe enable/disable, counter enable/disable, preload enable/disable, clear error, and home command (for type C).
HSC Data	Select the %AI reference location for data sent to the PLC by the HSC module. The length of the data is fixed at 15 words. HSC Data consists of accumulated counts, strobe register contents, and the counter per timebase value.
Failure Mode	Enter a value to select an output failure mode. Choices are <b>NORMAL*</b> , <b>FORCEOFF</b> and <b>HOLDLAST</b>
Oscillator Input	This field allows the use of an internal oscillator reference in place of one count input. The base value is 660 Khz. The reference is divisible by a 16-bit value (refer to the Oscillator Divider explained below). Choices are <b>ON</b> or <b>OFF*</b> . <b>ON</b> selects the input to be used as an oscillator reference; <b>OFF</b> selects the input to be used as a counter input.
Oscillator Divider	Select the 16-bit value that is divided into 660 Khz in order to achieve the final reference frequency. This value is only used if the <i>Oscillator Input</i> field above is set to ON.
Count Filter	This field identifies the Counter Input Filter.
Preload Filter	This field identifies the Counter Preload Filter.

\* Default selection.

- Press the **Page Down** key to display the detail screen for Type A Counter 1. Repeatedly press the **Page Down** key to scroll through the detail screen for each counter and then return to the main detail screen for a Type A counter. Press the **Page Up** key to display the detail screen for Type A Counter 4 and then scroll through the counters in reverse order.

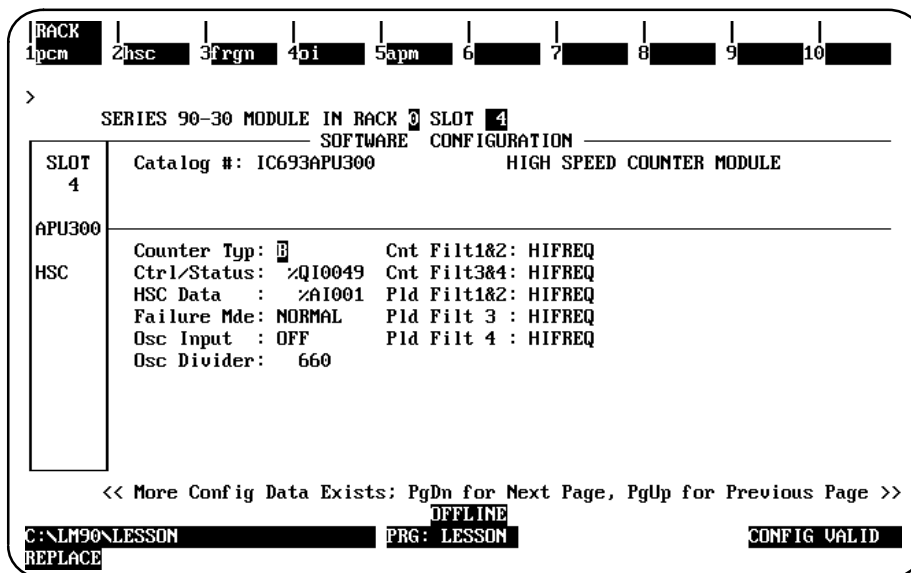


Parameter	Description
Count Mode	Select continuous (CONTINU*) or single-shot (SINGSHOT) counting.
Count Direction	Select the count direction for the type A counter. Choices are UP* or DOWN.
Time Base	Select the counts per time base (0 to +65,535). (Default = 1000)
Strobe Edge	Specify the positive (POS*) or negative (NEG) edge active for the strobe input.
HighLimit	Select the highest value for the count accumulator (-32,768 to +32,767). (Default = 32,767)
LowLimit	Select the lowest value for the count accumulator (-32,768 to +32,767). (Default = 0)
Preload Value	Select the accumulator preload value (-32,768 to +32,767). (Default = 0)
On Preset	Select the output on accumulator value (-32,768 to +32,767*).
Off Preset	Select the output off accumulator value (-32,768 to +32,767). (Default = 0)

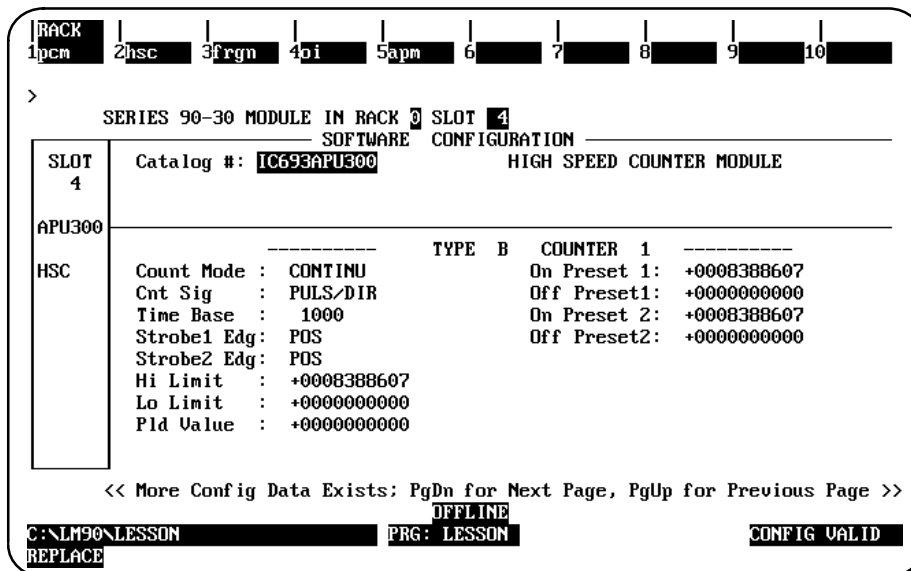
\* Default selection.

- To change from a Type A to Type B counter, press the **Tab** key with the cursor positioned on the **Counter Type** field on the main detail screen for the Type A counter. When B is displayed in this field, press the **Enter** key.

This example screen lists the module configuration parameters for a Type B counter.



- Press the **Page Down** key to display the detail screen for Type B Counter 1. Repeatedly press the **Page Down** or **Page Up** key to scroll through the detail screen for each counter and then return to the main detail screen for the Type B counter.



### Note

Because there are two 32-bit counters, there will be two screens similar to these example screens.

Parameter	Description
Count Mode	Select continuous ( <b>CONTINU*</b> ) or single-shot ( <b>SINGSHOT</b> ) counting.
CountSignal	For Type B or C counter, select how each counter's signals will be used. Choices are <b>UP/DN</b> , <b>PULS/DIR*</b> , or <b>AQUADB</b> .
Time Base	Select the counts per time base (0 to +65,535). (Default = 1000)
Strobe Edge	Specify the positive or negative edge active for the strobe input. Choices are <b>POS*</b> or <b>NEG</b> .
HighLimit	Select the highest value for the count accumulator (-2,147,483,648 to +2,147,483,647). (Default = +8,388,607)
LowLimit	Select the lowest value for the count accumulator (-2,147,483,648 to +2,147,483,647). (Default = 0)
Preload Value	Select the accumulator preload value (-2,147,483,648 to +2,147,483,647). (Default = 0)
On Preset	Select the output on accumulator value (-2,147,483,648 to +2,147,483,647). (Default = +8,388,607)
Off Preset	Select the output off accumulator value (-2,147,483,648 to +2,147,483,647). (Default = 0)

\* Default selection.

## Note

The Low Limit and High Limit parameters must be validated so that low limit is less than or equal to the high limit.

- To change from a Type B to Type C counter, press the **Tab** key with the cursor positioned on the **CounterType** field on the main detail screen for the Type B counter. When C is displayed in this field, press the **Enter** key.

This example screen lists the module configuration parameters for a Type C counter.

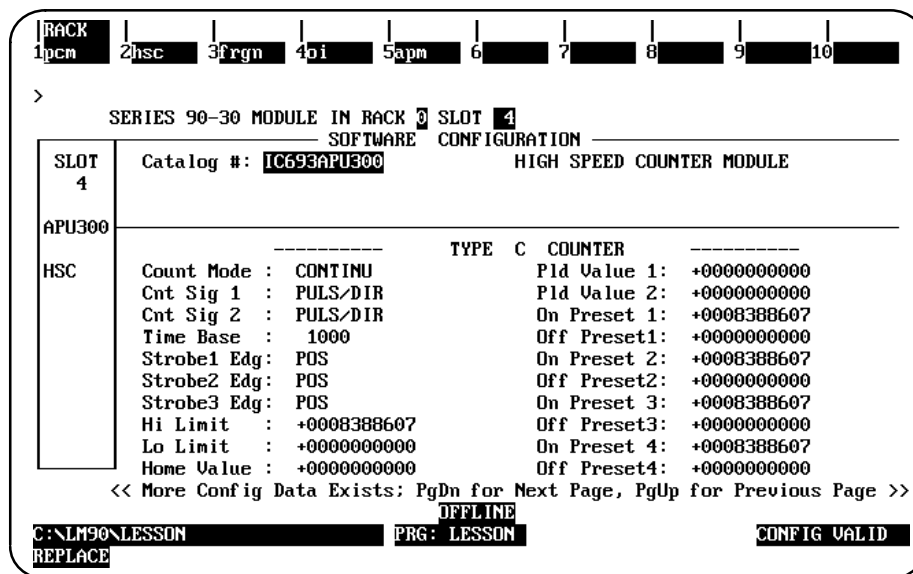
```

RACK 1pcn 2hsc 3frgn 4bi 5apn 6 7 8 9 10
>
SERIES 90-30 MODULE IN RACK 2 SLOT 4
SOFTWARE CONFIGURATION
SLOT 4 Catalog #: IC693APU300 HIGH SPEED COUNTER MODULE
APU300
HSC Counter Typ: C Cnt Filt 1 : HIFREQ
Ctrl/Status: %QI0049 Cnt Filt 2 : HIFREQ
HSC Data : %AI001 Pld Filt1&2: HIFREQ
Failure Mde: NORMAL Disbl Filt1: HIFREQ
Osc Input : OFF Disbl Filt2: HIFREQ
Osc Divider: 660

<< More Config Data Exists; PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C: NLM90NLESSON PRG: LESSON CONFIG VALID
REPLACE

```

- 8. Press the **Page Down** key to display the detail screen for a Type C counter. Press the **Page Down** key to return to the main detail screen for the module.



The configurable parameters on this screen are as described previously for the Type B counter.

Use these detail screens to configure the features of the High Speed Counter module.

## Configuring the Embedded HSC for a CPU 211

The High Speed Counter module for a CPU 211 can only be a Type A counter.

```

1|cpu 2|iobase 3| 4| 5| 6| 7| 8| 9| 10|
>
SERIES 90-20
SOFTWARE CONFIGURATION
CPU Cat. #: IC692CPU211      SERIES 90-20 CPU 211
I/O Cat. #: IC692MDR541     16 DC IN/12 RLY OUT/ 120VAC PS
----- HIGH SPEED COUNTER -----
Count En  : DISABLED  Time Base : 1000
Count Dir : UP        Hi Limit  : +32767
Count Mode: CONTINU  Lo Limit  : +00000
Pld/strobe: PRELOAD  Pld Value : +00000
Pld Filter: LOWFREQ  On Preset : +32767
Cnt Filter: LOWFREQ  Off Preset : +00000
Failure Mde: NORMAL
----- VIEW ONLY PARAMETERS -----
Ctrl/Status: %QI0033 HSC Data : %AI001
%QI SIZE : 16      %AI SIZE : 15
<< More Config Data Exists: PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\LESSON PRG: LESSON CONFIG VALID
REPLACE

```

The HSC module may be configured after paging down from the CPU 211 detail screen. Additional HSC parameters may be configured, as described in the following table.

Parameter	Description
Count Enable	Enable or disable the counter. Choices are <b>ENABLE</b> or <b>DISABLE*</b> .
Count Direction	Select the count direction for the module. Choices are <b>UP*</b> or <b>DOWN</b> .
Count Mode	Select continuous ( <b>CONTINU*</b> ) or single-shot ( <b>SINGSHOT</b> ) counting.
Preload/Strobe	Select <b>PRELOAD*</b> or <b>STROBE SELECT</b> .
Preload Filter	Select <b>HIFREQ</b> or <b>LOFREQ*</b> .
Count Filter	Select <b>HIFREQ</b> or <b>LOFREQ*</b> .
Failure Mode	Choices are <b>HOLDLAST NORMAL*</b> , or <b>FORCEOFF</b>
Time Base	Select the counts per time base (0 to +65,535). (Default = 1000)
High Limit	Select the highest value for the count accumulator (-32,768 to +32,767). (Default = +32,767)
Low Limit	Select the lowest value for the count accumulator (-32,768 to +32,767). (Default = 0)
Preload Value	Select the accumulator preload value (-32,768 to +32,767). (Default = 0)
On Preset	Select the output on accumulator value (-32,768 to +32,767). (Default = +32,767)
Off Preset	Select the output off accumulator value (-32,768 to +32,767). (Default = 0)

\* Default selection.



## Section 9: Configuring a PCM Module

The Programmable Coprocessor Module (PCM) combines the function of a Communications Module and the ASCII/BASIC Module into a single module. It may be configured to behave as one CCM port, two independent CCM ports, one CCM port and one BASIC application having one port, or one BASIC application using one or both serial ports.

The following modes of configuration are available:

Mode	Description
PCM CFG	All configuration data for the PCM is located in the User Configuration Data File (UCDF), created and loaded to the PCM using PCOP.
PROG PRT	Select Programmer Port mode.
CCM ONLY	Select CCM on both ports 1 and 2.
PROG/CCM	Select port 1 for programmer connection and port 2 for CCM.
CCM/PROG	Select port 1 for CCM and port 2 for the programmer.
BASIC	Select both ports 1 and 2 for BASIC.
BAS/CCM	Select port 1 for BASIC configuration and port 2 for CCM.

Logicmaster 90-30/20/Micro programming software provides configuration support of the following modules:

- PCM with 160 KB memory module.
- PCM (multiple mode) with 192 KB memory module.
- PCM (multiple mode) with 640 KB memory module.

Available modes for each module are listed below:

Mode	Module Type		
	160K PCM	192K PCM	640K PCM
PCM CFG	Yes	Yes	Yes
PROG PRT	Yes	Yes	Yes
CCM ONLY	Yes *	Yes	Yes
PROG/CCM	Yes *	Yes	Yes
CCM/PROG	Yes *	Yes	Yes
BASIC	Yes	Yes	Yes
BAS/CCM	Yes	Yes	Yes

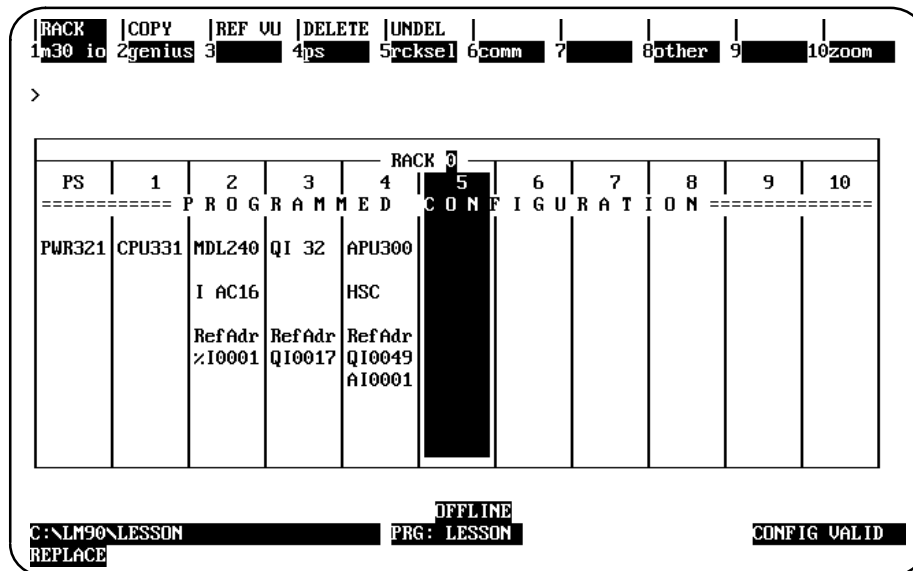
\* The interface for port 2 is fixed at RS-485 for the 160K modules.

For more information on the PCM, refer to the *Series 90 Programmable Coprocessor Module and Support Software User's Manual*, GFK-0255.

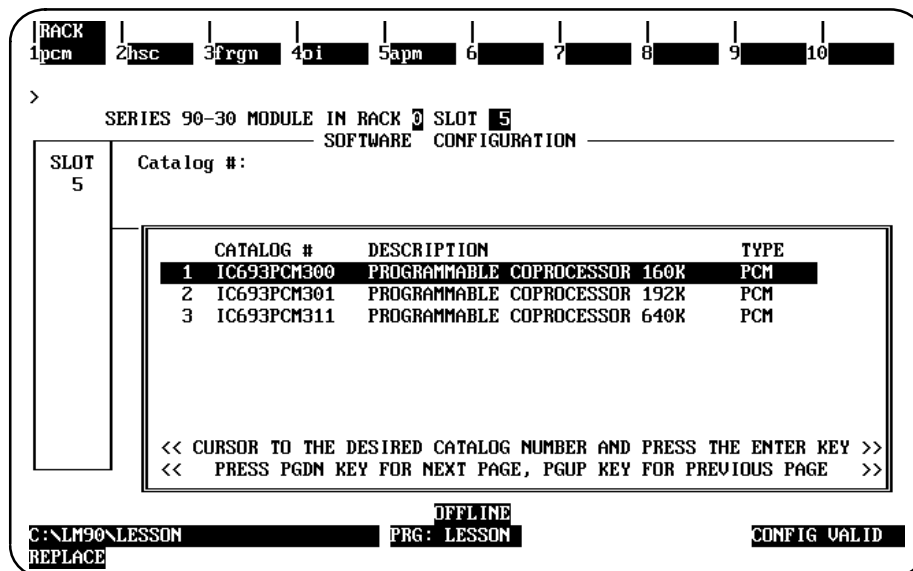
## Configuring a PCM

To configure a PCM on the I/O Configuration Rack screen:

1. Move the cursor to the desired rack and slot location. The slot may be either unconfigured or previously configured.



2. Press **Other (F8)** and then **PCM (F1)** from the I/O Configuration Rack screen to display a list of catalog numbers and modules.



3. Position the cursor on the catalog number for the module you wish to select, and press the **Enter** key. The module detail screen for the module you selected will be displayed.

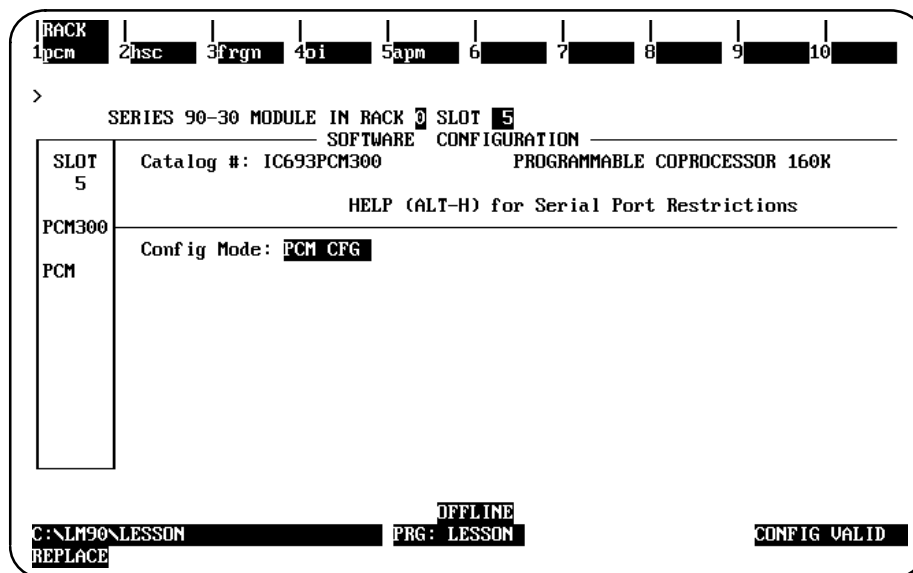
## Selecting the Configuration Mode

The configuration mode is selected in the **Configuration Mode** field on the module detail screen. (The module detail screen is displayed by positioning the cursor on the catalog number for the module you wish to select and pressing the **Enter** key.)

1. To select a different configuration mode, move the cursor to the **Configuration Mode** field and repeatedly press the **Tab** key until the desired mode is displayed on the screen. Then, press the **Enter** key.
2. Complete the detail screen for the desired configuration mode, and press the **Enter** key. For help selecting parameters, press **ALT-H**.

## PCM CFG Mode

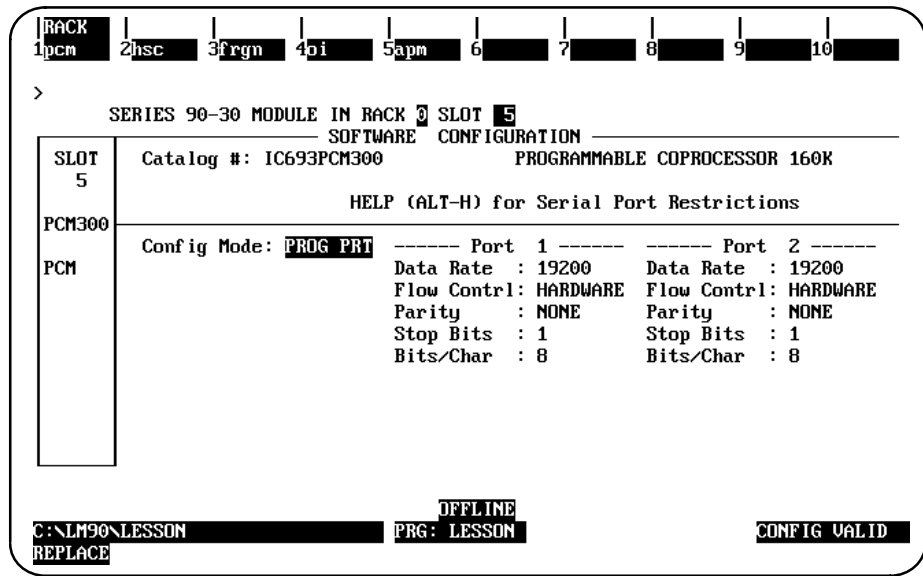
When **PCM CFG** mode is selected, the following PCM detail screen is displayed.



In this screen, the configuration mode is set to **PCM CFG** for the PCM Configuration Data File mode. This mode uses the User Configuration Data (UCDF) that was loaded into a user module named UCDF in the PCMs battery-backed RAM memory. (The UCDF is created using PCOP software.) The configuration data is used for building the configuration of the PCM during the power-up sequence, initializing available hardware on the PCM and specifying the user or system tasks to be started. For more information, refer to the *Series 90 Programmable Coprocessor Module and Support Software User's Manual*, GFK-0255.

## PROG PRT Mode

When **PROG PRT** (Programmer Port) mode is selected, the following PCM detail screen is displayed.

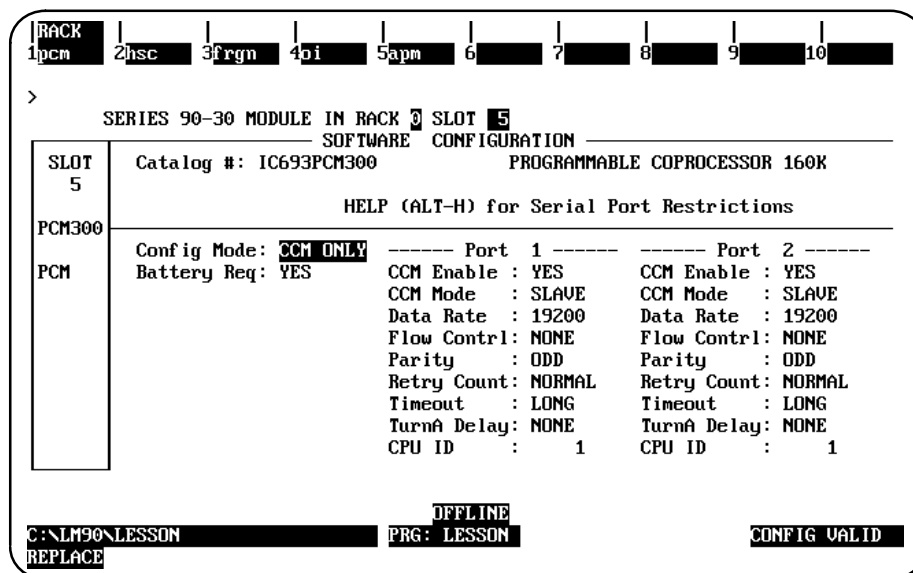


Parameter	Description
ConfigurationMode	The configuration mode is set to <b>PROG PRT</b>
Interface	The interface parameter for port 1 can only be <b>RS-232</b> ; port 2 of the 192K and 640K modules may have an interface value of either <b>RS-232*</b> or <b>RS-485</b> . The interface parameter is not used or displayed for the 160K module.
Data Rate	Data rate (bits per second or bps) for the port. Choices are 300, 600, 1200, 2400, 4800, 9600, or 19200*.
Flow Control	Type of flow control to be used on the port. Choices are <b>HARDWARE*</b> , <b>SOFTWARE</b> , or <b>NONE</b> .
Parity	Type of parity to be used on the port. Choices are <b>NONE*</b> , <b>EVEN</b> , or <b>ODD</b> .
Stop Bits	Number of stop bits for the port. Choices are <b>1*</b> or <b>2</b> .
Bits per Character	Number of bits per character for data transfer on the port. Choices are <b>7</b> or <b>8*</b> .

\* Default selection.

## CCM ONLY Mode

When **CCM ONLY** mode is selected, the following PCM detail screen is displayed.



Parameter	Description
ConfigurationMode	The configuration mode is set to <b>CCM ONLY</b> .
BatteryRequired	Specify whether a battery is required. Choices are <b>YES*</b> or <b>NO</b> .
CCMEnable	Specify whether the port is to be configured for use as a CCM port. Choices are <b>YES*</b> or <b>NO</b> .
CCMMode	This parameter displays the availability of ports for CCM access. Choices are <b>SLAVE*</b> , <b>PEER</b> , or <b>MASTER</b> .
Interface	The interface parameter for port 1 can only be <b>RS-232</b> ; port 2 of the 192K and 640K modules may have an interface value of either <b>RS-232*</b> or <b>RS-485</b> . The interface parameter is not used or displayed for the 160K module.
Data Rate	Data rate (bits per second or bps) for the port. Choices are 300, 600, 1200, 2400, 4800, 9600, or 19200*.
Flow Control	Type of flow control to be used on the port. Choices are <b>HARDWARE</b> or <b>NONE*</b> .
Parity	Type of parity to be used on the port. Choices are <b>NONE</b> , <b>EVEN</b> , or <b>ODD*</b> .
Retry Count	Retry counts for CCM mode. Choices are <b>NORMAL*</b> or <b>SHORT</b> .
Timeout	Length of timeouts used for CCM on the port. Choices are <b>LONG*</b> , <b>MEDIUM</b> , <b>SHORT</b> , or <b>NONE</b> .
TurnaroundDelay	Turnaround delay time to be used for CCM on the port. Choices are <b>NONE*</b> , 10 ms, 100 ms, or 500 ms.
CPUID	Address of the port on a multi-drop network. This value is used to calculate the backoff delay upon an inquiry collision in <b>PEER</b> mode. The range of values allowed in this field is 1* to 254.

\* Default selection.

## PROG/CCM Mode

**PROG/CCM** mode selects port 1 as the programmer port and port 2 for CCM. The parameters for port 1 are the same as for **PROG PRT** mode; and for port 2, the parameters are the same as for **CCM ONLY** mode.

## CCM/PROG Mode

**CCM/PROG** mode selects port 1 for CCM and port 2 as the programmer port. The parameters for port 1 are the same as for **CCM ONLY** mode; and for port 2, the parameters are the same as for **PROG PRT** mode.

## BASIC Mode

**BASIC** mode selects BASIC configuration for both ports 1 and 2. The parameters for ports 1 and 2 are the same as for **PROG PRT** mode.

## BAS/CCM Mode

**BAS/CCM** mode selects BASIC configuration for port 1 and CCM for port 2. The parameters for port 1 are the same as for **PROG PRT** mode; and for port 2, the parameters are the same as for **CCM ONLY** mode.

## Section 10: Configuring a TCP/IP Ethernet Module

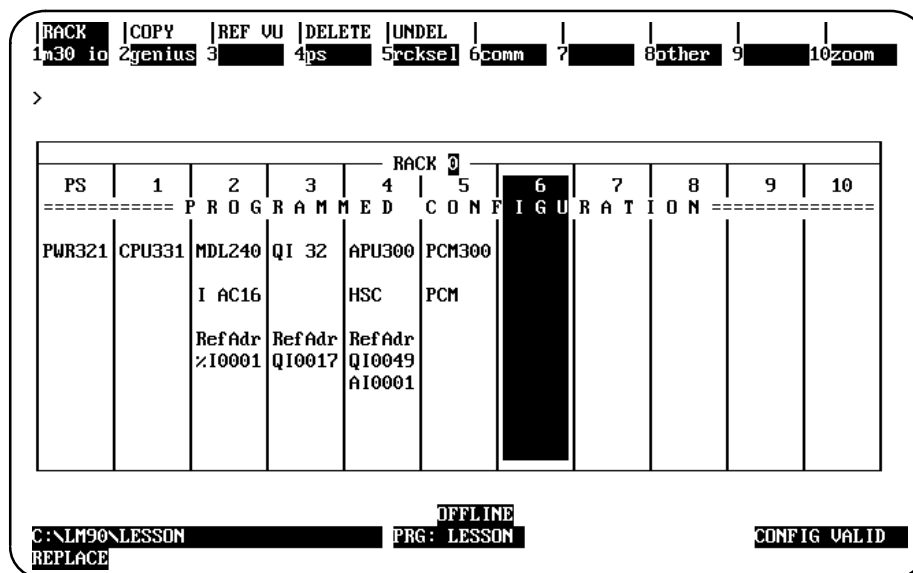
Logicmaster 90-30/20/Micro programming software provides configuration support of the TCP/IP Ethernet Module. For details about the TCP/IP Ethernet Module, refer to the *TCP/IP Ethernet Communications for the Series 90™ -30 PLC* manual (GFK-1084).

Use the information in this section in conjunction with the “Procedure 2: Configuring the Ethernet Interface with Logicmaster 90-30 Configuration Software” section of chapter 2 of the *TCP/IP Ethernet Communications for the Series 90™ -30 PLC* manual (GFK-1084).

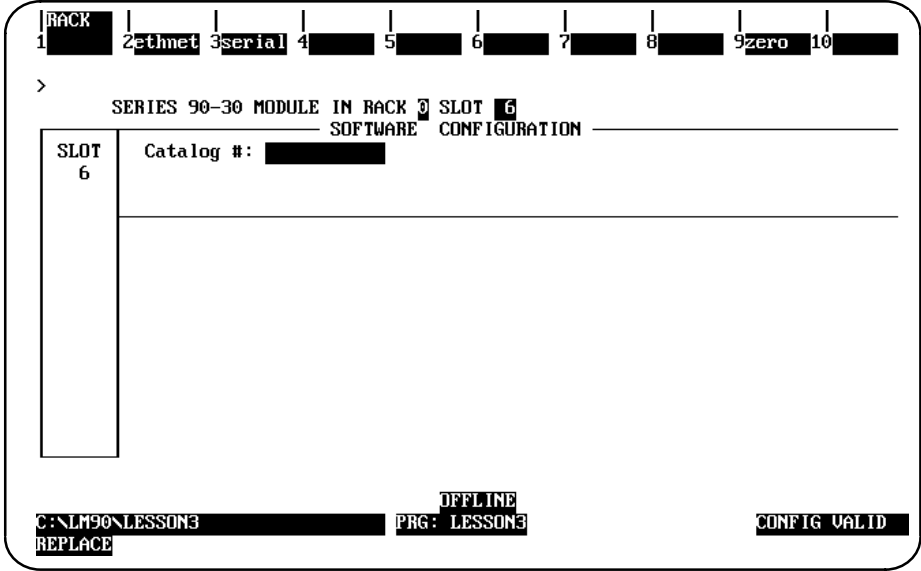
### Configuring a TCP/IP Ethernet Module

To configure a TCP/IP Ethernet Module on the I/O Configuration rack screen, do the following:

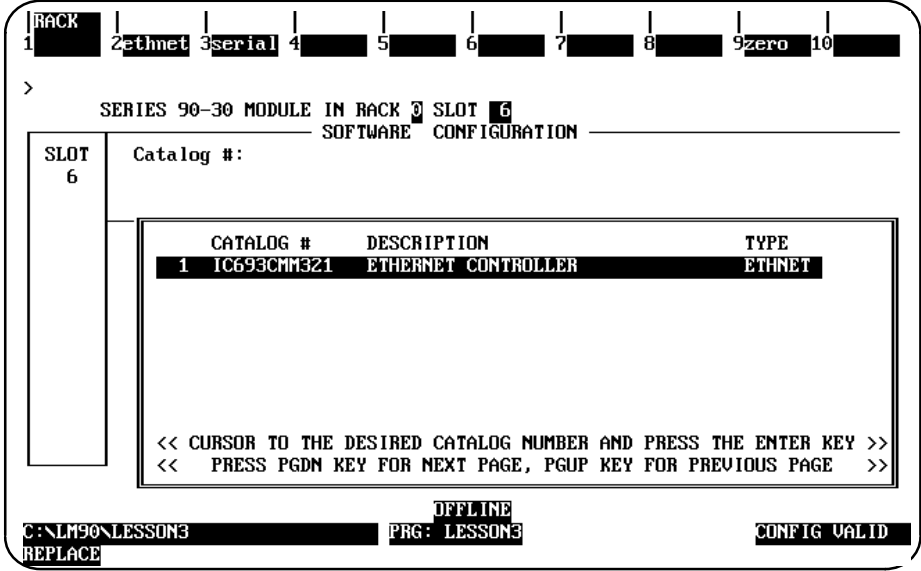
1. Move the cursor to the desired rack and slot location. The slot may be either unconfigured or previously configured.



2. Press the **Communications** softkey, i.e., **Comm (F6)**. Your screen display will change to the one shown on the following page.

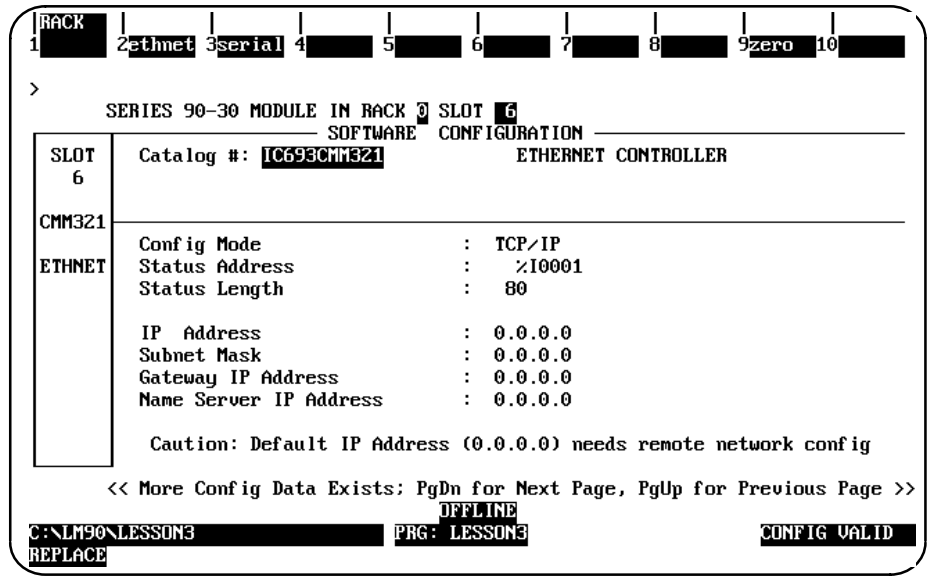


3. Press **ethnet (F2)**. Your screen display will change to the one shown below.



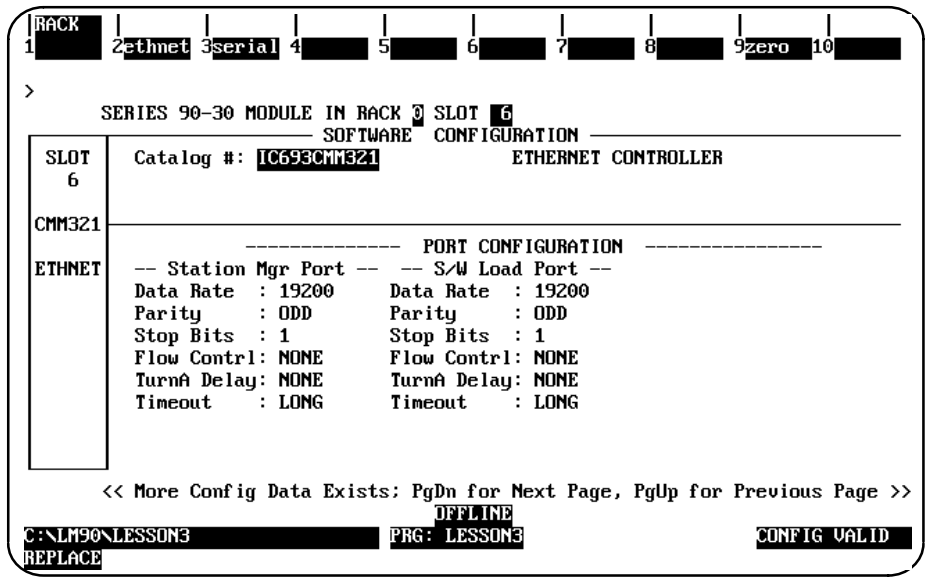
4. Press **Enter** to select the Ethernet Controller. You will then see the screen shown on the following page.





For detailed information about this screen, refer to the first two pages of the “Procedure 2: Configuring the Ethernet Interface with Logicmaster 90-30 Configuration Software” section of chapter 2 of the *TCP/IP Ethernet Communications for the Series 90™ -30 PLC* manual (GFK-1084).

- After you have assigned the IP address, etc., press **Page Down** to display the following screen.



Parameter	Description
Data Rate	Data rate (bits per second or bps) for the port. Choices are 300, 600, 1200, 2400, 4800, 9600, or 19200*.
Parity	Type of parity to be used on the port. Choices are NONE, EVEN, or ODD*.
Stop Bits	Enter the number of stop bits. Choices are 1* or 2.
Flow Control	Type of flow control to be used on the port. Choices are HARDWARE or NONE*.
Turnaround Delay	Turnaround delay time to be used for CCM on the port. Choices are NONE*, 10 ms, 100 ms, or 500 ms.
Timeout	Length of timeouts used for CCM on the port. Choices are LONG*, MEDIUM, SHORT, or NONE.

\* Default selection.

- Press the **Escape** key to return to the rack display. Press **Escape** again to save the configuration.

## Section 11: Configuring a CMM Module

Logicmaster 90-30/20/Micro programming software provides configuration support of the CMM Communications Module (multiple mode) with 32 KB. The following modes of configuration are available:

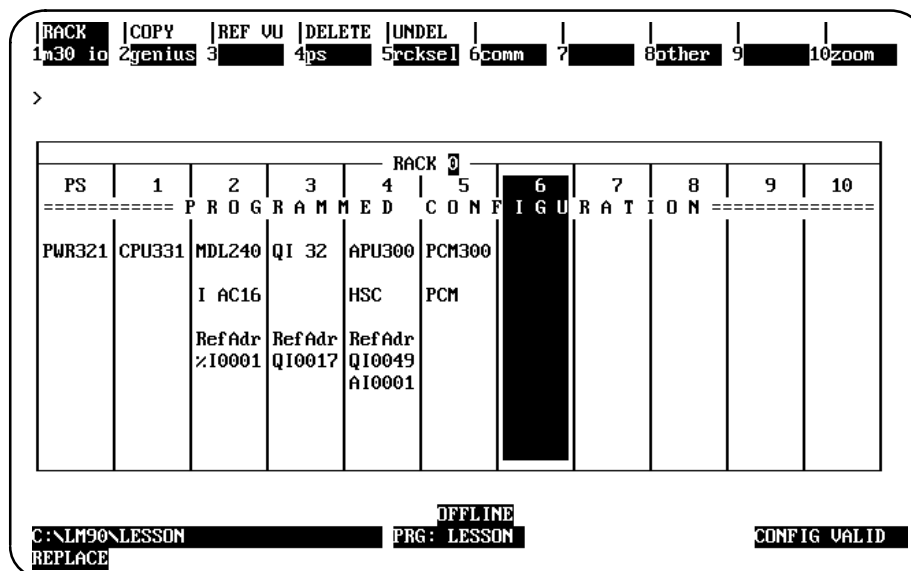
Mode	Description
CCM ONLY	Select CCM on both ports 1 and 2.
CCM/RTU	Select port 1 for CCM and port 2 for RTU.
RTU/CCM	Select port 1 for RTU and port 2 for CCM.
RTU ONLY	Select RTU on both ports 1 and 2.
SNP ONLY	Select SNP on both ports 1 and 2.
SNP/CCM	Select port 1 for SNP and port 2 for CCM.
CCM/SNP	Select port 1 for CCM and port 2 for SNP.
SNP/RTU	Select port 1 for SNP and port 2 for RTU.
RTU/SNP	Select port 1 for RTU and port 2 for SNP.

For more information on the CMM, refer to the *Series 90 Programmable Coprocessor Module and Support Software User's Manual* (GFK-0255) and the *Series 90 PLC Serial Communications User's Manual* (GFK-0582).

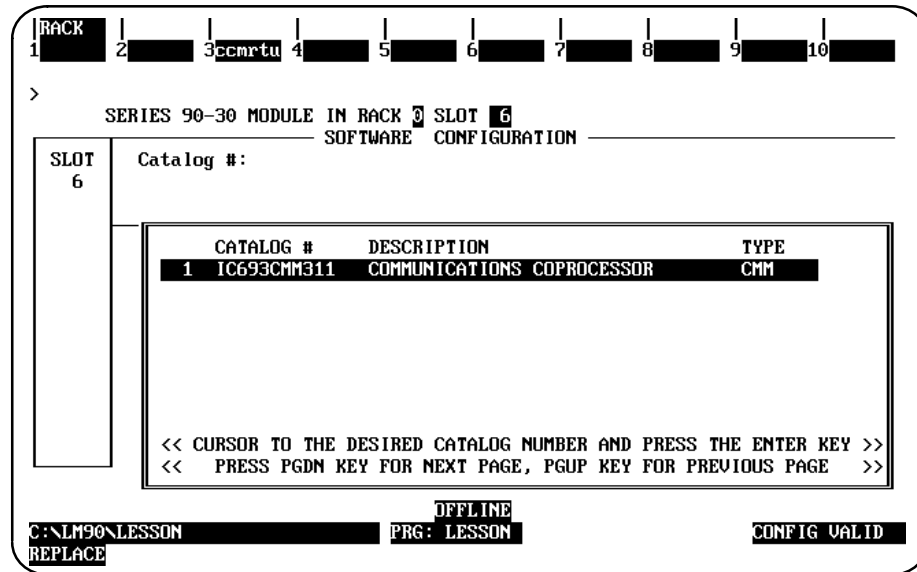
### Configuring a CMM

To configure a CMM on the I/O Configuration Rack screen:

1. Move the cursor to the desired rack and slot location. The slot may be either unconfigured or previously configured.



2. Press **Communications (F6)** and then **Serial (F3)** from the I/O Configuration Rack screen to display the catalog number of the CCM/RTU module.



3. Position the cursor on the catalog number for the module you wish to select, and press the **Enter** key. The module detail screen for the module you selected will be displayed.

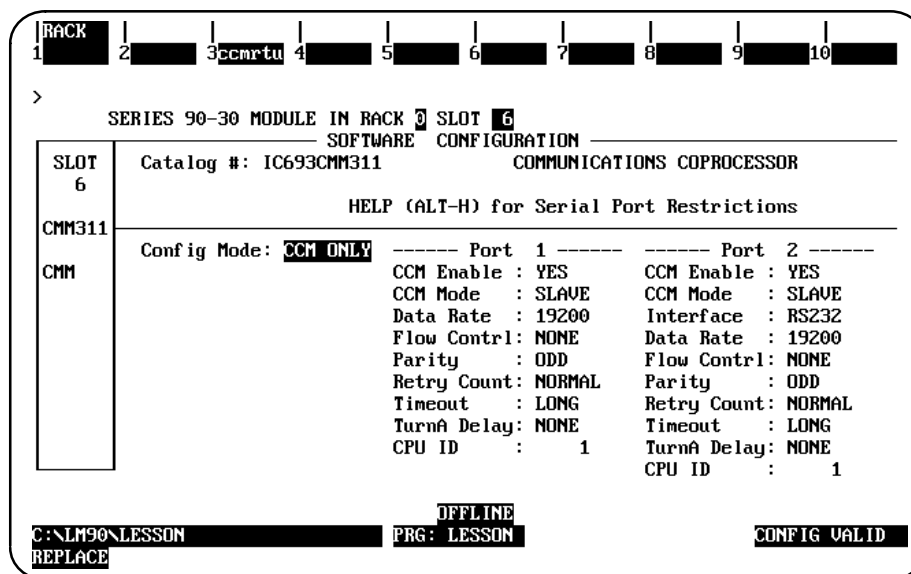
## Selecting the Configuration Mode

The configuration mode is selected in the **Configuration Mode** field on the module detail screen. (The module detail screen is displayed by positioning the cursor on the catalog number for the module you wish to select and pressing the **Enter** key.)

1. To select a different configuration mode, move the cursor to the **Configuration Mode** field and repeatedly press the **Tab** key until the desired mode is displayed on the screen. Then, press the **Enter** key.
2. Complete the detail screen for the desired configuration mode, and press the **Enter** key. For help selecting parameters, press **ALT-H**.

## CCM ONLY Mode

When **CCM ONLY** mode is selected, the following detail screen is displayed.

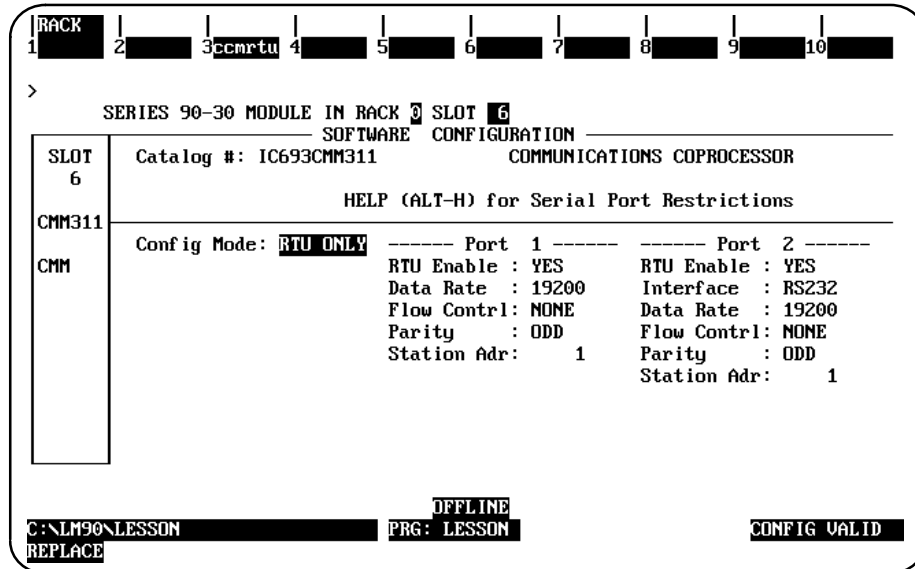


Parameter	Description
ConfigurationMode	The configuration mode is set to <b>CCM ONLY</b> .
BatteryRequired	Specify whether a battery is required. Choices are <b>YES*</b> or <b>NO</b> .
CCMEnable	Specify whether the port is to be configured for use as a CCM port. Choices are <b>YES*</b> or <b>NO</b> .
CCMMode	This parameter displays the availability of ports for CCM access. Choices are <b>SLAVE*</b> , <b>PEER</b> , or <b>MASTER</b> .
Interface	The interface parameter for port 1 can only be <b>RS-232</b> ; port 2 of the 192K and 640K modules may have an interface value of either <b>RS-232*</b> or <b>RS-485</b> . The interface parameter is not used or displayed for the 160K module.
Data Rate	Data rate (bits per second or bps) for the port. Choices are <b>300</b> , <b>600</b> , <b>1200</b> , <b>2400</b> , <b>4800</b> , <b>9600</b> , or <b>19200*</b> .
Flow Control	Type of flow control to be used on the port. Choices are <b>HARDWARE</b> or <b>NONE*</b> .
Parity	Type of parity to be used on the port. Choices are <b>NONE</b> , <b>EVEN</b> , or <b>ODD*</b> .
Retry Count	Retry counts for CCM mode. Choices are <b>NORMAL*</b> or <b>SHORT</b> .
Timeout	Length of timeouts used for CCM on the port. Choices are <b>LONG*</b> , <b>MEDIUM</b> , <b>SHORT</b> , or <b>NONE</b> .
TurnaroundDelay	Turnaround delay time to be used for CCM on the port. Choices are <b>NONE*</b> , <b>10 ms</b> , <b>100 ms</b> , or <b>500 ms</b> .
CPUID	Address of the port on a multi-drop network. This value is used to calculate the backoff delay upon an inquiry collision in <b>PEER</b> mode. The range of values allowed in this field is <b>1*</b> to <b>254</b> .

\* Default selection.

## RTU ONLY Mode

When **RTU ONLY** mode is selected, the following detail screen is displayed.



Parameter	Description
ConfigurationMode	The configuration mode is set to <b>CCM ONLY</b> .
RTUEnable	Specify whether the port is to be configured for use as an RTU port. Choices are <b>YES*</b> or <b>NO</b> .
Interface	The interface parameter for port 1 can only be <b>RS-232</b> ; port 2 of the 192K and 640K modules may have an interface value of either <b>RS-232*</b> or <b>RS-485</b> . The interface parameter is not used or displayed for the 160K module.
Data Rate	Data rate (bits per second or bps) for the port. Choices are 300, 600, 1200, 2400, 4800, 9600, or 19200*.
Flow Control	Type of flow control to be used on the port. Choices are <b>HARDWARE</b> or <b>NONE*</b> .
Parity	Type of parity to be used on the port. Choices are <b>NONE</b> , <b>ODD*</b> , or <b>EVEN</b> .
StationAddress	Enter a value from 1* to 247.

\* Defaultselection.

## CCM/RTU Mode

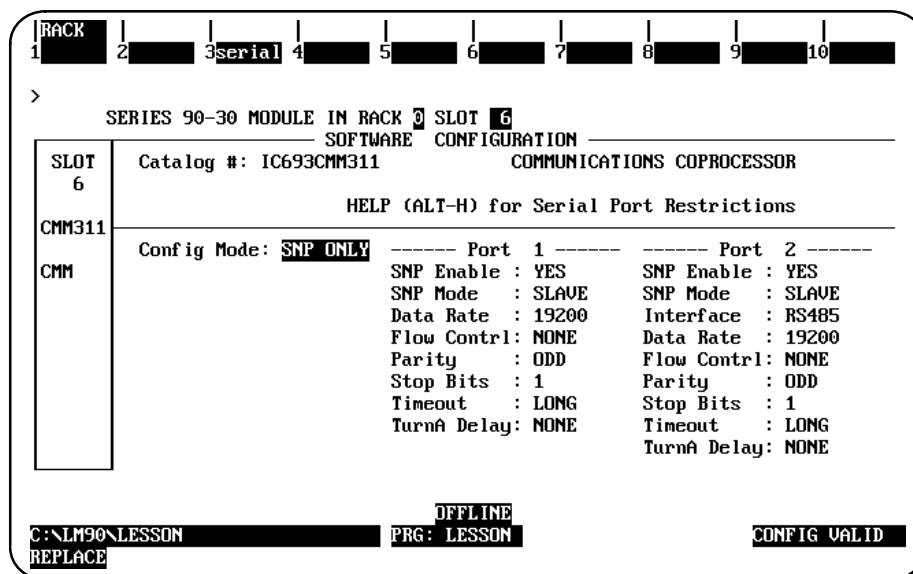
**CCM/RTU** mode selects port 1 for CCM and port for RTU. The parameters for port 1 are the same as for **CCM ONLY** mode; and for port 2, the parameters are the same as for **RTU ONLY** mode.

## RTU/CCM Mode

**RTU/CCM** mode selects port 1 for RTU and port 2 for CCM. The parameters for port 1 are the same as for **RTU ONLY** mode; and for port 2, the parameters are the same as for **CCM ONLY** mode.

## SNP ONLY Mode

When **SNP ONLY** mode is selected, the following detail screen is displayed.



Parameter	Description
ConfigurationMode	The configuration mode is set to <b>SNP ONLY</b> .
SNP Enable	Specify whether the port is to be configured for use as an SNP port. Choices are <b>YES*</b> or <b>NO</b> .
SNP Mode	This parameter displays the availability of ports for SNP access. Choices are <b>SLAVE*</b> or <b>MASTER</b> .
Interface	The interface parameter can be <b>RS-232*</b> or <b>RS-485</b> .
Data Rate	Data rate (bits per second or bps) for the port. Choices are <b>300</b> , <b>600</b> , <b>1200</b> , <b>2400</b> , <b>4800</b> , <b>9600</b> , or <b>19200*</b> .
Parity	Type of parity to be used on the port. Choices are <b>NONE</b> , <b>ODD*</b> , or <b>EVEN</b> .
Stop Bits	Most serial communication uses at least one stop bit. Slower devices use two stop bits. Choices are <b>1*</b> or <b>2</b> .
Flow Control	Type of flow control to be used on the port. Choices are <b>NONE*</b> or <b>HARDWARE</b> .
TurnaroundDelay	Turnaround delay time to be used for CCM on the port. Choices are <b>NONE*</b> , <b>10 ms</b> , <b>100 ms</b> , or <b>500 ms</b> .
Timeout	Length of timeouts used for SNP on the port. Choices are <b>LONG*</b> , <b>MEDIUM</b> , <b>SHORT</b> , or <b>NONE</b> .

\* Default selection.

## **SNP/CCM Mode**

**SNP/CCM** mode selects port 1 for SNP and port 2 for CCM. The parameters for port 1 are the same as for **SNP ONLY** mode; and for port 2, the parameters are the same as for **CCM ONLY** mode.

## **CCM/SNP Mode**

**CCM/SNP** mode selects port 1 for CCM and port 2 for SNP. The parameters for port 1 are the same as for **CCM ONLY** mode; and for port 2, the parameters are the same as for **SNP ONLY** mode.

## **SNP/RTU Mode**

**SNP/RTU** mode selects port 1 for SNP and port 2 for RTU. The parameters for port 1 are the same as for **SNP ONLY** mode; and for port 2, the parameters are the same as for **RTU ONLY** mode.

## **RTU/SNP Mode**

**RTU/SNP** mode selects port 1 for RTU and port 2 for SNP. The parameters for port 1 are the same as for **RTU ONLY** mode; and for port 2, the parameters are the same as for **SNP ONLY** mode.

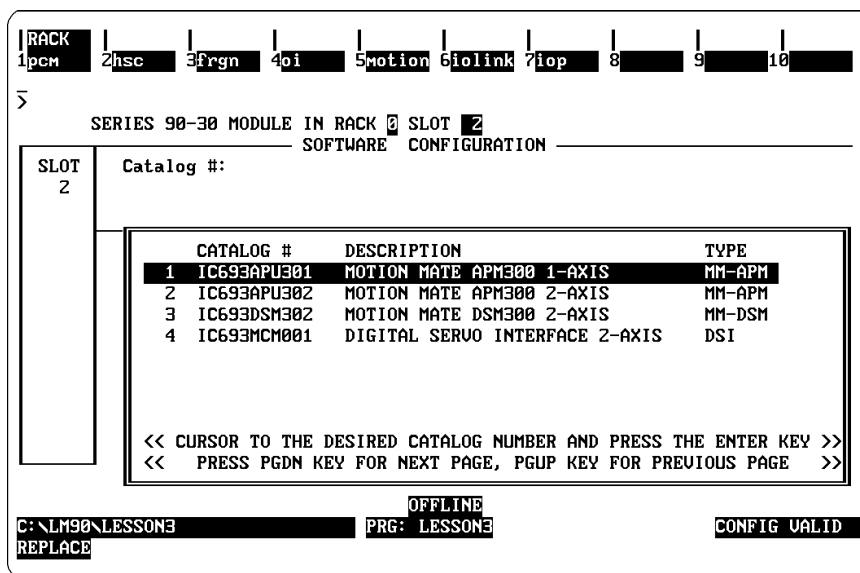


## Section 12: Configuring a Motion Product (Formerly Configuring an APM)

The motion control products are intelligent, fully-programmable, one or two-axis positioning controller integrated into the Series 90-30 PLC system. The Motion Mate APM is an intelligent, fully-programmable, one or two-axis positioning controller. You must have a Release 3.52 or later CPU module in order to configure a slot in the I/O rack for a Motion Mate APM300. The Digital Servo Interface module also requires a 3.52 or later CPU. The Motion Mate DSM302 requires a 6.50 or later CPU. For more information on this family of products, refer to GFK-0707 and GFK-0664.

To configure a single axis Motion Mate APM on the I/O Configuration Rack screen:

1. Move the cursor to the desired rack and slot location. The slot may be either configured or previously unconfigured.
2. Press **Other (F8)** and then **motion (F5)** from the I/O Configuration Rack screen to display the catalog number of the APM.



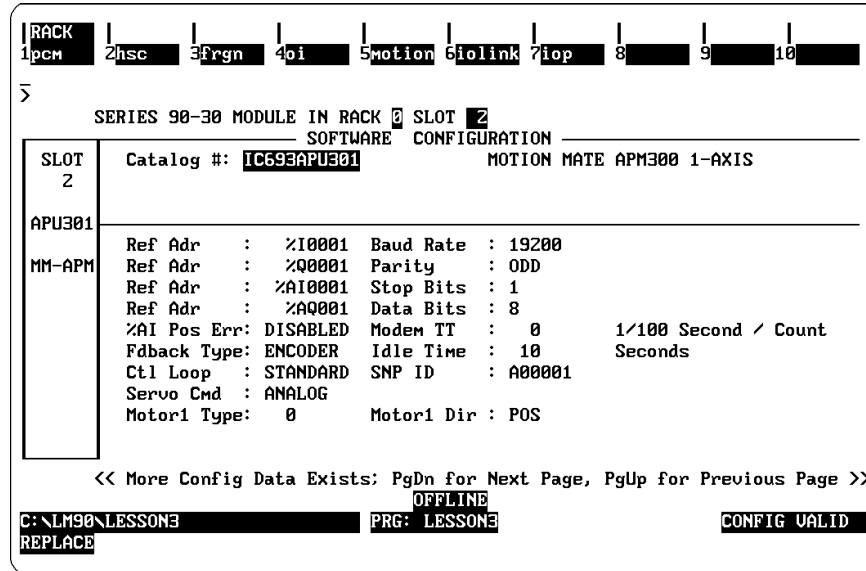
### Note

The instructions on the following pages describe how to configure a slot in the I/O rack for a single axis Motion Mate APM. These instructions also apply to the two-axis Motion Mate APM as well. The only difference is in the catalog number you select (IC693APU302 for the two-axis Motion Mate APM instead of IC693APU301 for the single axis Motion Mate APM).

## Configuring a Single Axis Motion Mate APM

To configure a single axis Motion Mate APM:

1. Position the cursor on the catalog number for the single axis Motion Mate APM (IC693APU301), and press the **Enter** key. The following detail screen is displayed.



Parameter	Description
Reference Address	The starting address for the %I user reference. Default= %I0001 or the next highest available address. For both the single axis Motion Mate APM and the two-axis Motion Mate APM, the reference length is fixed at 32 bits.
Reference Address	The starting address for the %Q user reference. Default= %Q0001 or the next highest available address. For both the single axis Motion Mate APM and the two-axis Motion Mate APM, the reference length is fixed at 32 bits.
Reference Address	The starting address for the %AI user reference. Default = %AI001 or the next highest available address. For the single axis Motion Mate APM, the reference length is fixed at 15 words. For the two-axis Motion Mate APM, the reference length is fixed at 28 words.
Reference Address	The starting address for the %AQ user reference. Default = %AQ001 or the next highest available address. For both the single axis Motion Mate APM, and the two-axis Motion Mate APM, the reference length is fixed at 6 words.
%AI Position Error	If <b>DISABLED*</b> , the command position is displayed in the %AI table. If <b>ENABLED</b> , the position error is displayed in the %AI table.

Parameter	Description
Feedback Type	<b>ENCODER*</b> selects A quad B (x4) incremental encoder input mode. <b>LINEAR</b> selects Temposonics linear transducer (absolute feedback) input mode. <b>RESOLVER</b> selects single or multiple resolver (absolute feedback) input mode. <b>CUSTOM1</b> and <b>CUSTOM2</b> configure the Motion Mate APM inputs for special applications.
Control Loop	<b>STANDARD*</b> selects the normal Motion Mate APM motion control loop. The <b>STANDARD</b> loop provides a velocity command output proportional to position error, with optional velocity feed forward and integrator gain terms. <b>CUSTOM</b> loops are individually designed for special applications. For the two-axis Motion Mate APM, <b>FOLLOWER</b> selects a control loop which allows ratio tracking of a master input with zero following error.
Baud Rate	Transmission rate (in bits per second or bps) of data through the SNP port. Choices are 300, 600, 1200, 2400, 4800, 9600, or 19200*.
Parity	Parity is indicated by an <b>ODD*</b> or <b>EVEN</b> number of bits, or whether no parity bit ( <b>NONE</b> ) is added to the word.
Stop Bits	Number of stop bits. Most serial communications use one stop bit. Slower devices may use two stop bits. Choices are 1* or 2.
Data Bits	Number of data bits. Specify whether the CPU recognizes 7 or 8* bitwords.
Modem Turnaround Time	The time required for the modem to start data transmission after receiving the transmit request. Values are 0* to 2550, in multiples of 10 milliseconds.
Idle Time	Maximum link idle time. The time the module should wait for the next message to be received from the communicating device before it believes that the program device has failed and proceeds to its base state. Values are 1 to 60 seconds. (Default = 10 seconds)
SNP ID	The identification number of the SNP port. Messages for the module are sent to this address. (Default = A00001)
Servo Cmd	Defines the Servo Interface Type (Analog or Digital), i.e., the type of command output—refer to GFK-0781 or GFK-0840 for details.
Motor Type	GE Fanuc Motor Type—refer to GFK-0781 or GFK-0840 for details.

\* Default selection.

2. Press the **Page Down** key to display the following detail screen:

```

RACK 1pcm 2hsc 3frgn 4oi 5motion 6iolink 7iop 8 9 10
>
SERIES 90-30 MODULE IN RACK 0 SLOT 2
SOFTWARE CONFIGURATION
SLOT 2 Catalog #: IC693APU301 MOTION MATE APM300 1-AXIS
APU301
MM-APM ----- AXIS 1 -----
User Units : 1 Pos Loop TC: 1000
Counts : 1 Vel at 10v : +0000004000
OT Limit Sw: ENABLED Vel FF % : 0
Pos EOT : +000838607 Intgr TC ms: 0
Neg EOT : -000838608 Intgr Mode : OFF
Pos Err Lim: +0000004096 Rev Comp : 0
In Pos Zone: 10 DisDly (ms): 100

<< More Config Data Exists: PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\LESSON3 PRG: LESSON3 CONFIG VALID
REPLACE
    
```

Parameter	Description
User Units	Scale factor which allows the APM to be programmed in units appropriate for the application. The ratio of user units to counts must be in the range 32:1 to 1:32. Values are 1* to +65,535.
Counts	Scale factor which allows the APM to be programmed in units appropriate for the application. The ratio of user units to counts must be in the range 32:1 to 1:32. Values are 1* to +65,535.
Overtravel Limit Switch	Specify whether the APM uses the hardware overtravel limit switch inputs (CTL05 and CTL06 for axis 1; CTL07 and CTL08 for axis 2). If <b>ENABLED</b> , then 10 - 30 VDC must be applied to the inputs in order for the APM to operate. Values are <b>ENABLED*</b> or <b>DISABLED</b> .
Positive Software End of TravelLimit	If the APM is commanded to go to a position greater than the positive EOT, an error will result and the APM will not allow axis motion. Values are -8,388,608 to +8,388,607* user units.
Negative Software End of TravelLimit	If the APM is commanded to go to a position less than the negative EOT, an error will result and the APM will not allow axis motion. Values are 0 to -8,388,608* user units.
Position Error Limit	The maximum position error (Commanded Position - Actual Position) allowed when the APM is controlling a servo. This parameter should normally be set to a value 10 to 20 percent higher than the highest position error encountered under normal servo operations. Range = 256* (user units / counts) <= POS ERR LIM <= 60,000* (user units / counts). (Default = 4096)
In Position Zone	When the servo position error is within this value and no motion is commanded, the <b>IN_ZONE</b> status bit is set. This parameter also determines the position error at which <b>PMOVES</b> are considered to be complete. Range = 0 to 2000. (Default = 10)
Position Loop Time Constant	The desired servo position loop time constant (in milliseconds). The lower the value, the faster the system's response. Values which are too low will cause system instability and oscillation. For accurate tracking of the commanded velocity profile, the position loop time constant should be 1/4 to 1/2 of the minimum system deceleration time. The time will not be accurate unless the velocity at 10V value is set correctly. Values are 20 to +32,767 ms. (Default = 1000)
Velocity at 10V	Actual servo velocity (in user units per second) for an APM velocity command output of 10V. This value must be configured correctly in order for the position loop time constant and the velocity feedforward gain percent factor to be accurate. The APMS <b>FORCE D/A OUTPUT %AQ</b> immediate command and the <b>ACTUAL VELOCITY %AI</b> return data can be used to determine the proper configuration value. Values are 100 to +8,388,607. (Default = 4000)
Velocity Feedforward Gain	The percentage of commanded velocity that is added to the APM velocity command output. Increasing this parameter causes the servo to operate with faster response and reduced position error. Values are 0* to 100 percent; optimum values are 80 to 90 percent. The velocity at 10V value must be set correctly for proper operation of the velocity feedforward gain factor.
Integrator Time Constant (in milliseconds)	Values should be 5 to 10 times the position loop time constant setting. Servo instability will result if the integrator time constant is set too low. When the integrator mode is set to <b>CONTINUOUS</b> , deceleration times must be 5 to 10 times longer than the integrator time constant setting or servo overshoot will occur. Setting the time constant to zero turns off the integrator. (Default = 0)
Integrator Mode	Operating mode for the position error integrator. <b>OFF*</b> means that the integrator is not used. <b>CONTINU</b> means the integrator runs continuously, even during servo motion. <b>IN_ZONE</b> means the integrator only runs when the servo is at rest.

Parameter	Description
Reversal Compensation	A compensation factor which allows the servo to reverse direction and still provide accurate positioning in systems containing backlash. Values are 0* to +255.
Servo Drive DisableDelay	The time delay (in milliseconds) from zero velocity command to the drive enable output switching off. Disable delay is effective when the <b>ENABLE DRIVE %I</b> bit is turned off or certain error conditions occur. This time delay should be longer than the deceleration time of the servo from maximum speed. Values are 0 to +32,767 ms. (Default = 100)

\* Defaultselection.

3. Press the **Page Down** key again to display this detail screen:

```

RACK 1 | 2hsc | 3frgn | 4oi | 5motion | 6iolink | 7iop | 8 | 9 | 10
>
SERIES 90-30 MODULE IN RACK 0 SLOT 2
SOFTWARE CONFIGURATION
SLOT 2 Catalog #: IC693APU301 MOTION MATE APM300 1-AXIS
APU301
-----
MM-APM AXIS 1
Jog Vel : +0000001000 Hi Limit : +0008388607
Jog Acc : +0000010000 Lo Limit : -0008388608
Jog Acc Mod: LINEAR Home Positn: +0000000000
InitData 1 : 0 Home Offset: +00000
InitData 2 : 0 Fnl Hm Vel : +0000000500
Find Hm Vel: +0000002000
Home Mode : HOMESW
<< More Config Data Exists: PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
C:\LM90\LESSON3 PRG: LESSON3 CONFIG UALID
REPLACE
  
```

Parameter	Description
Jog Velocity	The velocity (in user units per second) at which the servo moves during a jog operation. Values are 0 to +8,388,607. (Default = 1000)
Jog Acceleration Rate	The acceleration rate (in user units per second) used during jog, find home, move at velocity, and abort operations. Values are 0 to +8,388,607. (Default = 10000)
Jog Acceleration Mode	The acceleration mode for jog, find home, move at velocity, and abort operations. <b>LINEAR*</b> causes commanded velocity to change linearly with time. <b>SCURVE</b> causes commanded velocity to change more slowly than the linear mode at the beginning and end of acceleration intervals.
DataInit1	This field is reserved for future use.
DataInit2	This field is reserved for future use.
High Count Limit	When the axis is moving in the positive direction and this value is reached, actual position will roll over to the low count limit. (Default = 8,388,607)
Low Count Limit	When the axis is moving in the negative direction and this value is reached, actual position will roll over to the high count limit. (Default = -8,388,607)
Home Position	The value (in user units) assigned to actual position at the end of a find home cycle. Values are -8,388,608 to +8,388,607. (Default = 0)
Home Position Offset	The offset (in user units) of the servo final stopping point at the completion of a find home cycle. Home offset adjusts the final servo stopping point relative to the encoder marker. Values are -32,768 to +32,767. (Default = 0)
Final Home Velocity	The velocity (in user units per second) at which the servo seeks the final home switch transition and encoder marker pulse at the end of a find home cycle. Final home velocity must be slow enough to allow a 5 millisecond delay between the final home switch transition and the encoder marker pulse. Values are 0 to +8,388,607. (Default = 500)
Find Home Velocity	The velocity (in user units per second) at which the servo seeks the initial home switch transitions during the find home cycle. If desired, find home velocity can be set to a high value to allow the servo to quickly locate the home switch. Values are 0 to +8,388,607. (Default = 2000)
Home Mode	Select <b>HOMESW</b> if a home switch input is used as part of the find home cycle. Select <b>MOVE+</b> or <b>MOVE-</b> for a unidirectional home cycle which does not use the home switch input.

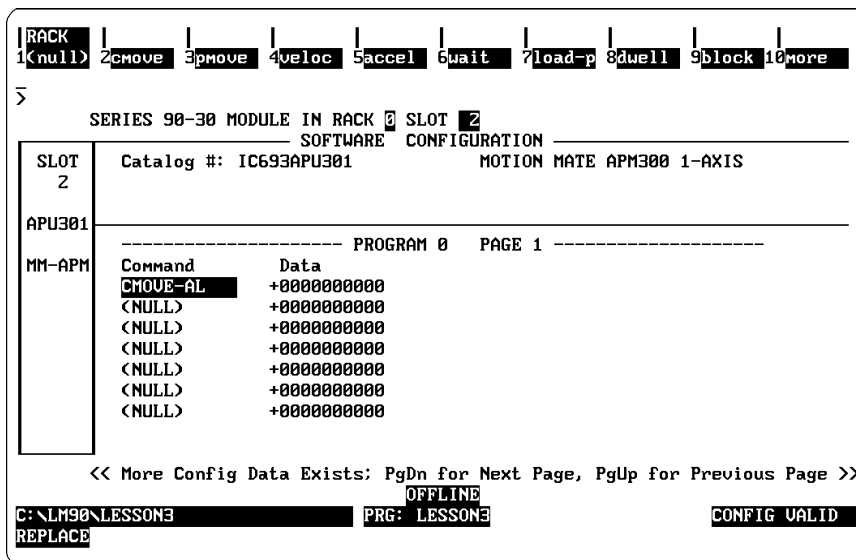
\* Defaultselection.



Field	Description
Data	This field contains either a 24-bit signed integer or the number of an APM register if the command has a <b>-P</b> extension.
Command	A legal APM command. For configuration, the legal commands are listed below.
	<b>(NULL):</b> Do nothing. <b>BLOCK:</b> Define block number. Values are 1 to +65,635. (Default = 1)
	The following commands have a value range from -8,388,608 to +8,388,607. When the <b>-P</b> suffix is used on the command, however, the maximum data value is 255. (Default = 1) (A -1 placed at the end of the command indicates that the command is for axis 1; a -2 indicates axis 2.)
	<b>CMOVE-AL:</b> Regular move, absolute, linear. <b>CMOVE-AS:</b> Regular move, absolute, s-curve. <b>CMOVE-AS-P:</b> Regular move, absolute, s-curve, use data in APM register. <b>CMOVE-IL:</b> Regular move, incremental, linear. <b>CMOVE-IL-P:</b> Regular move, incremental, linear, use data in APM register. <b>CMOVE-IS:</b> Regular move, incremental, s-curve. <b>CMOVE-IS-P:</b> Regular move, incremental, s-curve, use data in APM register. <b>PMOVE-AL:</b> Positioning move, absolute, linear. <b>PMOVE-AL-P:</b> Positioning move, absolute, linear, use data in APM register. <b>PMOVE-AS:</b> Positioning move, absolute, s-curve. <b>PMOVE-AS-P:</b> Positioning move, absolute, s-curve, use data in APM register. <b>PMOVE-IL:</b> Positioning move, incremental, linear. <b>PMOVE-IL-P:</b> Positioning move, incremental, linear, use data in APM register. <b>PMOVE-IS:</b> Positioning move, incremental, s-curve <b>PMOVE-IS-P:</b> Positioning move, incremental s-curve, use data in APM register.
	The following commands have a value range of 1 to 98,388,607. When the <b>-P</b> suffix is used on the command, however, the maximum data value is 255. (Axis 1 = -1; Axis 2 = -2)
<b>VELOC:</b> Set velocity. (Default = 2000) <b>VELOC-P:</b> Set velocity to data in APM register. (Default = 1)	
The following commands have a value range of 1 to 134,217,727. When the <b>-P</b> suffix is used on the command, however, the maximum data value is 255. (Axis 1 = -1; Axis 2 = -2)	
<b>ACCEL:</b> Set acceleration. (Default = 5000) <b>ACCEL-P:</b> Set acceleration to data in APM register. (Default = 1)	
<b>WAIT:</b> Wait for some bit to go high before moving. Values are 1, 2, 4, 16, etc. (Default = 1)	
The following commands have a value range from -8,388,608 to +8,388,607. (Default = 0)	
<b>LOAD-P01:</b> Load APM parameter register number 1. <b>LOAD-P02:</b> Load APM parameter register number 2. • • • <b>LOAD-P20:</b> Load APM parameter register number 20.	
The following commands have a value range from 0 to +65,535. When the <b>-P</b> suffix is used on the command, however, the maximum data value is 255. (Axis 1 = -1; Axis 2 = -2)	
<b>DWELL:</b> Wait X milliseconds. Default = 0 <b>DWELL-P:</b> Wait X milliseconds, where X is the value of the APM register. (Default = 1)	
	If an entry other than one of the legal commands listed above is made, Logicmaster 90-30/90-20 software will display the error message, "Incorrect data entry."
Axis Number	The axis number may be 1, 2, or N/A (Not Applicable).



- Using the list of commands above, enter the command in the **Command** field, or use the function softkeys to select a command. To use the softkeys, move the cursor to the appropriate command field and press the softkey for the desired command. In the following example, **CMOVE (F2)** was pressed.



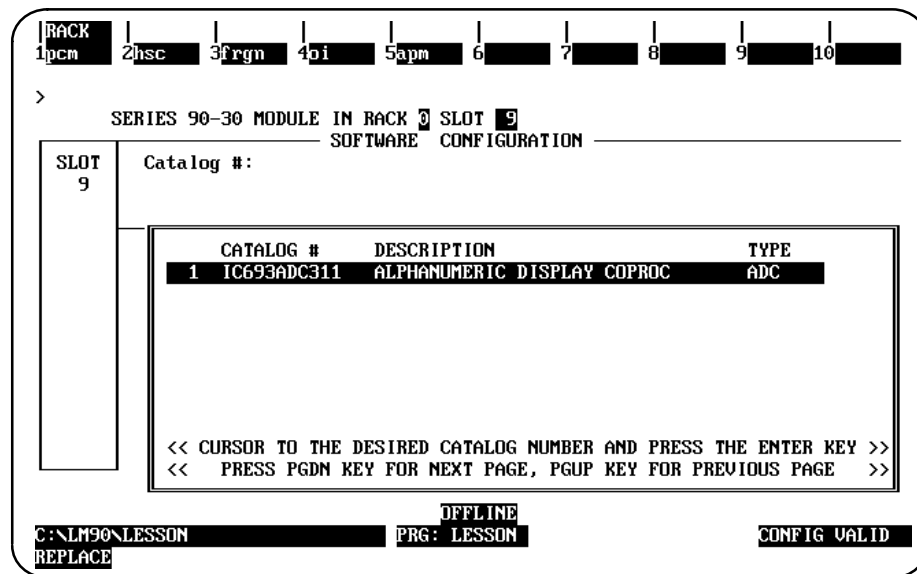
- If there is more than one type of command for a particular command (e.g., **CMOVE**), use the **Tab** key to move through the various selections. In the example, **CMOVE-AL** is the default for the **CMOVE (F2)** softkey. When **F2** is initially pressed, **CMOVE-AL** is displayed. To select another **CMOVE** command in this field, press the **Tab** key to display **CMOVE-AL-P**, **CMOVE-AS**, **CMOVE-AS-P**, **CMOVE-IL**, **CMOVE-IL-P**, **CMOVE-IS**, and **CMOVE-IS-P**, in order.
- When you press **Load (F7)** for the **LOAD** command, **LOAD-P01** is initially displayed. To change the **APM** register number, press the **Tab** key. The display will change from **LOAD-P01** to **LOAD-P02** ..... **LOAD-Pn** (where **n** is the parameter register number). This eliminates having to make an entry in the **Data** field for the **LOAD** command. If a value is entered into the data register for the **LOAD** command, it is simply ignored.
- The other commands do not support the **Tab** key. If the **Tab** key is pressed, an "Inactive Key" error message is displayed and the key is ignored. In addition, the **Tab** key will also generate an "Inactive Key" error message if it is pressed in the **Data** field.

## Section 13: Configuring an ADC Module

The Alphanumeric Display Coprocessor (ADC) Module is used with the CIMPPLICITY 90-ADS software for developing user screens and graphic displays. The CIMPPLICITY 90-ADS system is a character-based operator interface generator system for use with the Series 90-30 PLC. For more information on this module, refer to the *CIMPPLICITY 90-ADS User's Manual*, GFK-0499.

To configure this module on the I/O Configuration Rack screen:

1. Move the cursor to the desired rack and slot location. The slot may be either unconfigured or previously configured.
2. Press **Other (F8)** and then **Operator Interface (F4)** from the I/O Configuration Rack screen to display the catalog number for the module.

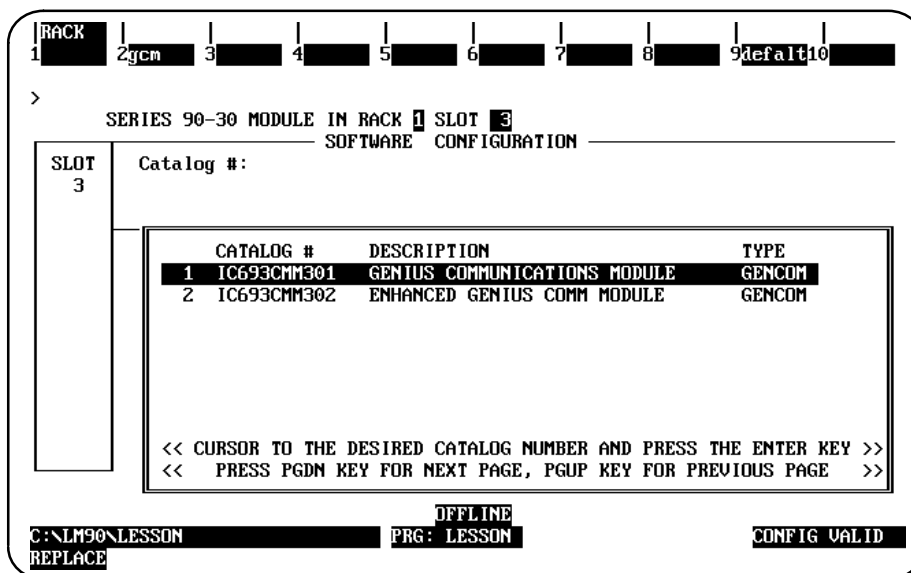


3. With the cursor positioned on IC693ADC311, press the **Enter** key.

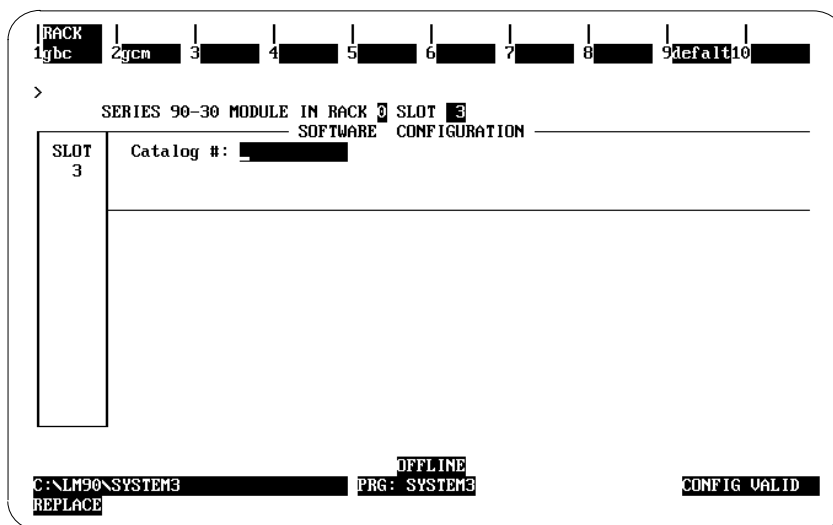
## Section 14: Configuring a GCM or Enhanced GCM

The Genius Communications Module provides the means for configuring global data to and from the Series 90-30 programmable controller. To configure the Genius Communications Module:

1. Move the cursor to the correct slot.
2. Press **Genius (F2)** and then **GCM (F2)** from the I/O Configuration Rack screen to display a list of catalog numbers and modules.

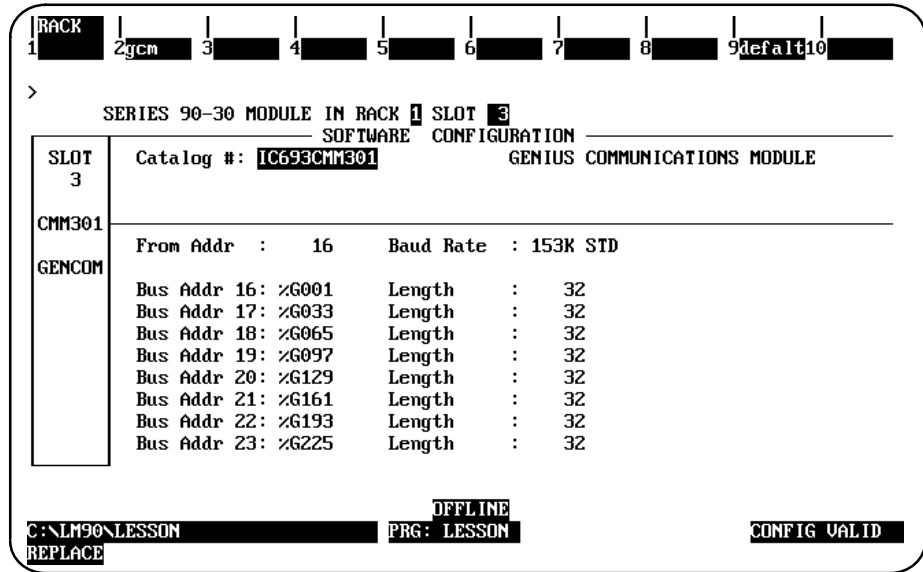


3. Then press **F2 (genius)**. The following screen will appear.



4. Press **F2 (gcm)** and then press the **Enter** key to select it. The following screen will appear.

- With the cursor positioned on IC693CMM301, press the **Enter** key to display the detail screen for the Genius Communications Module.



- Complete the detail screen, using the definitions provided in the following table, and press the **Enter** key again.

Field	Description
FromAddress	Specifies the serial bus address from where the outgoing global data is sent. The valid range for this field is 16 to 23, inclusive. By specifying one bus address as output, the other seven serial bus addresses, by default, receive incoming global data.
Baud Rate	Selections for the baud rate include 153K STD, 76.8K, 38.4K, and 153K EXT
Length	The global references are fixed fields, referring to the address associated with each of the eight serial bus addresses 16 through 23. Each Length field represents the amount of data transferred to or from the corresponding serial bus address, in units of bits. The length value, when added to the fixed starting address, must not exceed %G256, the maximum allowable global data reference.

The table below states the allowable values for each length, validated at the time of entry.

Bus Address	Starting Address	Valid Lengths
16	%G001	0 - 256
17	%G033	0 - 224
18	%G065	0 - 192
19	%G097	0 - 160
20	%G129	0 - 128
21	%G161	0 - 96
22	%G193	0 - 64
23	%G225	0 - 32

If one field is increased beyond the default of 32 bits, the next field(s) must be decreased to avoid overlapping addresses. For example, if 64 bits is specified for serial bus address 16, the data transmitted from this node maps to %G001 to %G064. Therefore, a length of 0 must be assigned to serial bus address 17. The **Length** fields will be automatically adjusted so that overlapping %G addresses are not possible.

## Enhanced Genius Communications Module

### Note

Beginning with Release 4.5, the I/O configuration is no longer limited to two Enhanced Genius Communications Modules. The number of Enhanced GCM modules that can be configured is now limited only by the following:

- The presence of a Genius Communications Module. If a GCM module is already configured, an Enhanced GCM module cannot be configured.
- The maximum configuration size of the PLC. If the references assigned to the Enhanced GCM Modules causes the I/O configuration size to exceed the maximum allowed, then the configuration cannot be stored to the PLC.

1. To configure an Enhanced Genius Communications Module, position the cursor on IC693CMM302 in the catalog list of modules, and press the **Enter** key. The following detail screen will be displayed.

```

RACK 1 2 3 4 5 6 7 8 9 default 10
  2jcm
>
SERIES 90-30 MODULE IN RACK 3 SLOT 3
SOFTWARE CONFIGURATION
SLOT 3 Catalog #: IC693CMM302 ENHANCED GENIUS COMM MODULE
CMM302 %I,%Q,%G lengths are bits, %AI,%AQ,%R are words
GENCOM
SBA of This DROP ID : 33
Module : 16 Baud Rate : 153K STD Data Default: OFF
S6 Ref : 0 Status : %I0049 Report Flts: NO

SBA Start Ref Reference Length Msg Buffer Byte Offset
24 %G0257 0 0
25 %G0385 0 0
26 %G0513 0 0
27 %G0641 0 0
28 %G0769 0 0

<< More Config Data Exists; PgDn for Next Page, PgUp for Previous Page >>
OFFLINE
D:\LM90\LESSON PRG: LESSON CONFIG VALID
REPLACE

```

2. Complete the detail screen, using the definitions provided in the following table. Then, press the **Enter** key again.

Parameter	Description
Serial Bus Address (SBA) of this Module	Indicates the Serial Bus Address of this module. This is the address from which global data will be sent. Values are 0 to 31. (Default = 16)
Baud Rate	Selections for the baud rate include 153K STD, 76.8K, 38.4K, and 153K EXT (Default = 153K STD)
DataDefault	If set to <b>OFF</b> , the Enhanced GCM module will either zero out data being supplied to the host PLC whenever the corresponding Genius device supplying the data ceases to communicate on the bus or zero out data being transmitted onto the bus when its host PLC ceases to scan its own I/O. If set to <b>HOLD</b> , the Enhanced GCM module will continue to pass on the last valid state sent to it in either of these situations. (Default = <b>OFF</b> )
Series Six Reference	Specifies the register location in a Series Six or Series Five CPU that should be reserved for the global data that will be transmitted to it by the Enhanced GCM module. A value of zero indicates that no register location should be reserved. Values are 0 - 16,383, inclusive. (Default = 0)
Status	Specifies the location in PLC memory used to hold the status specified by the Enhanced GCM module. The memory type for this field is restricted to %I. (Default = the next highest available %I value)
Report Faults	If set to <b>YES</b> , the Enhanced GCM module will accept fault mail from its host Series 90-30 PLC and transmit it onto the bus as a Genius datagram which can be interpreted by a Series 90-70 Genius Bus Controller. If <b>NO</b> is selected, the Enhanced GCM module will not pass on any fault information. (Default = <b>NO</b> )
Drop ID	This parameter is used by a Series 90-70 PLC to identify the location of a fault that has been reported by a Series 90-30 PLC via an Enhanced GCM module. The Report Faults option of the Enhanced GCM module must also be enabled. When locating two Enhanced GCM modules in one Series 90-30 PLC, it is recommended that you assign a <b>unique</b> Drop ID to each module in order to avoid conflicting identification information if both modules are set up to report faults. Valid range is 16 through 254. (Default = 33)

The following parameters apply for each configurable Serial Bus Address (SBAs 0 through 31):

Parameter	Description
Start Reference	The starting reference address from or to which global data is being transferred. The reference type can be %G, %R, %AI, %AQ, %I, or %Q.
Reference Length	The total length of the reference data, starting at the address specified by START REF. Values are 0 to 128 bytes; however, the units for this field will be bits if the configured reference type is %I, %Q, or %G, and words if the configured reference type is %AI, %AQ, or %R.
Message Buffer Byte Offset	<p>The offset, measured in bytes, from the start of an incoming message at which the Enhanced GCM module will start to extract data from the incoming message. The first byte extracted is placed in PLC memory at the start reference given in the configuration of the Serial Bus Address of the device transmitting the message. A value of zero indicates that the first byte of the incoming message is placed at the starting references defined for each Serial Bus Address.</p> <p>The offset for the Serial Bus Address of the Enhanced GCM module itself is always zero; it cannot be changed. (Default = 0)</p> <p>For each Serial Bus Address, the sum of the reference length and the message buffer byte offset cannot exceed 128 bytes.</p> <p>Since the Serial Bus Address of this module is used to transmit global data, its reference address may overlap with other configured reference addresses.</p>

- The Default (F9) can be used to set the reference addresses and reference lengths to a combined Scheme 1 and Scheme 2 Genius global data. Serial Bus Addresses 16 through 23 may be configured to use Genius Scheme 1, and Serial Bus Addresses 24 through 31 may be configured to use Genius Scheme 2. All other Serial Bus Addresses are set to their default value. In addition, all message buffer byte offsets are set to zero.

**For additional information on the configuration of devices and remote drops, see:**

*Genius Discrete and Analog Blocks User's Manual* (GEK-90486-2) – includes instructions for configuring most I/O blocks.

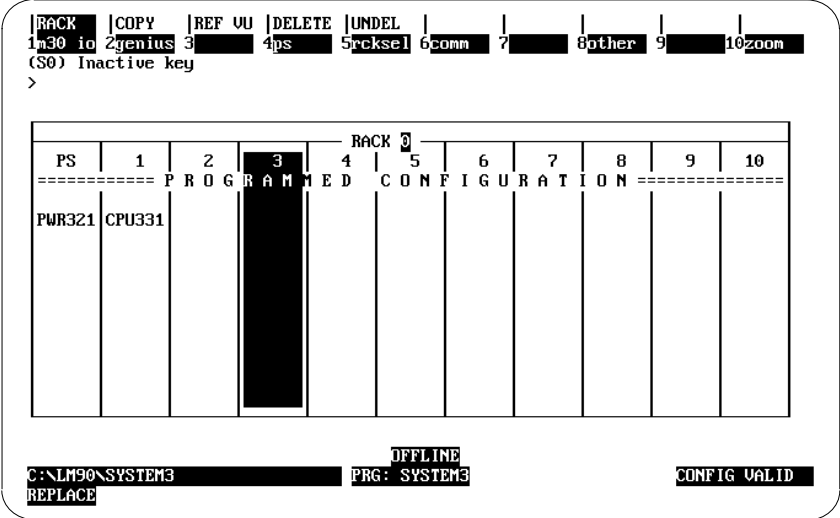
*Genius I/O System User's Manual* (GEK-90486-1) – details the data that can be transferred using Read Configuration and Write Configuration COMREQs.

# Section 15: Configuring a Genius Bus Controller

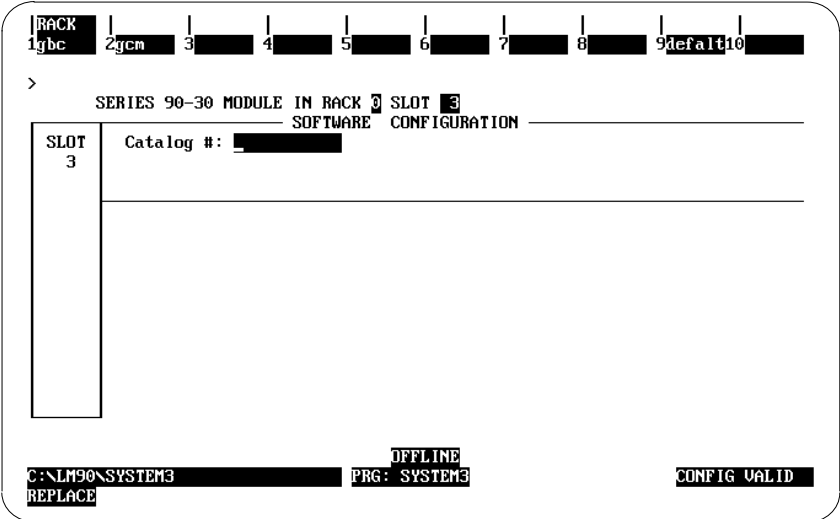
The Genius Bus Controller (GBC) must be configured as part of the Series 90-30 PLC system using the Logicmaster 90-30/20/Micro software (release 5 or later) or a Hand-held Programmer (HHP). This includes configuring parameters for the GBC module itself, and parameters specific to the devices on the GBC's bus.

The LM90 configurator software program (release 5 or later) can be used to configure the module in the offline mode. Once the complete set of configuration data has been entered, it must then be downloaded to the PLC (in the online mode) to become effective in the Genius Bus Controller.

1. The GBC is configured by completing setup screens in the Logicmaster 90-30 configuration software. The setup screens that are used for this module are shown and described below. In the I/O configuration screen, place the cursor at the slot representation corresponding to the GBC's installed location in the PLC rack.

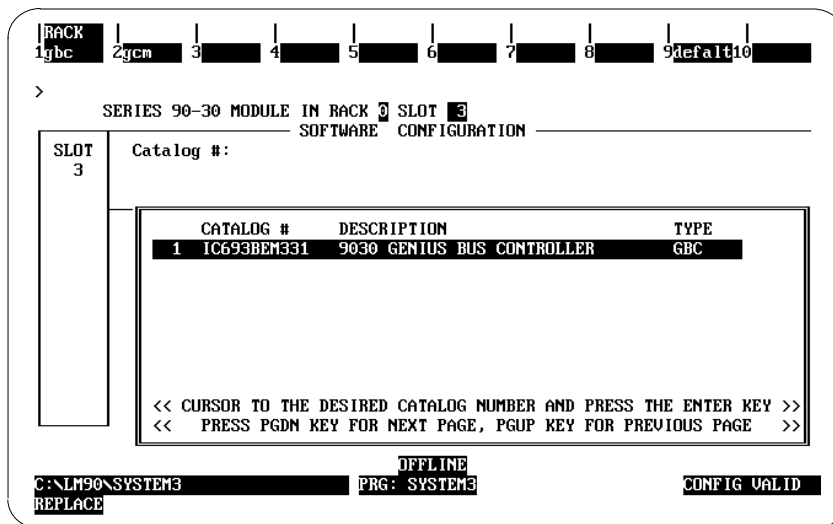


2. Then press F2 (genius). The following screen will appear.

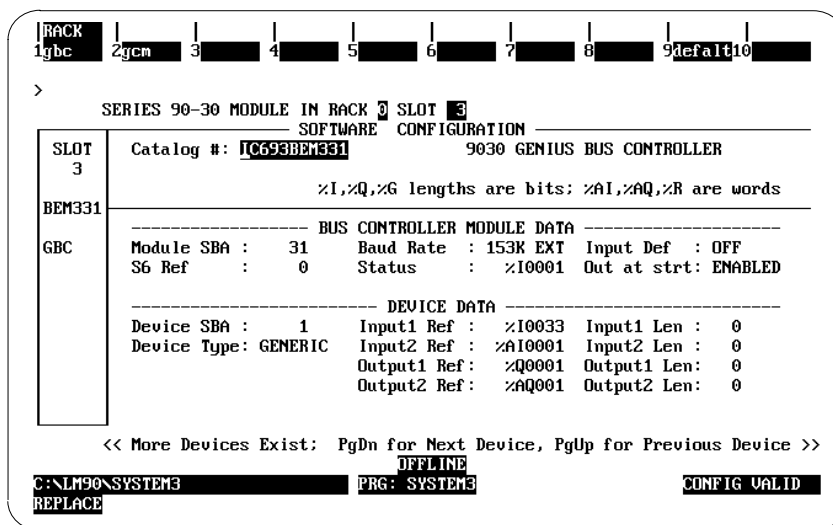




- Press **F1** (**gbc**) and then press the **Enter** key to select the GBC. The following screen will appear. (Note that the “default” softkey, **F9**, is inactive.)



- Press the **Enter** key to select the GBC. Complete the GBC configuration entries in the following screen:



Note that the configuration screen consists of two parts: module-specific data (BUS CONTROLLER MODULE DATA) and device-specific data (DEVICE DATA).

The default entries can be used as is, or changed. *Until a valid configuration is stored to the PLC CPU, the GBC will not operate on the bus, and its Channel OK LED will not light.*

**For additional information on the configuration of devices and remote drops, see:**

*Genius Discrete and Analog Blocks User's Manual* (GEK-90486-2) – includes instructions for configuring most I/O blocks.

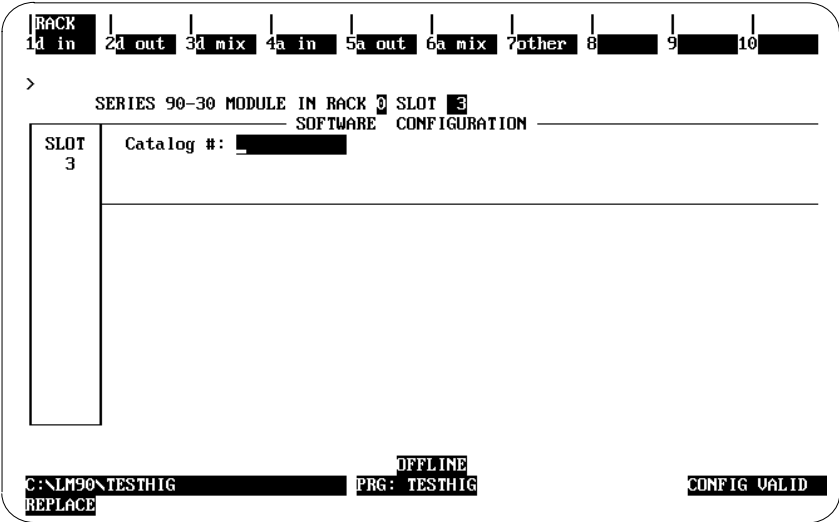
*Genius I/O System User's Manual* (GEK-90486-1) – details the data that can be transferred using Read Configuration and Write Configuration COMREQs.

*Series 90-30 Genius Bus Controller User's Manual* (GFK-1034) – includes configuration and additional details on relevant several topics including configuring for global data, as well as details on assigned configuration parameters.

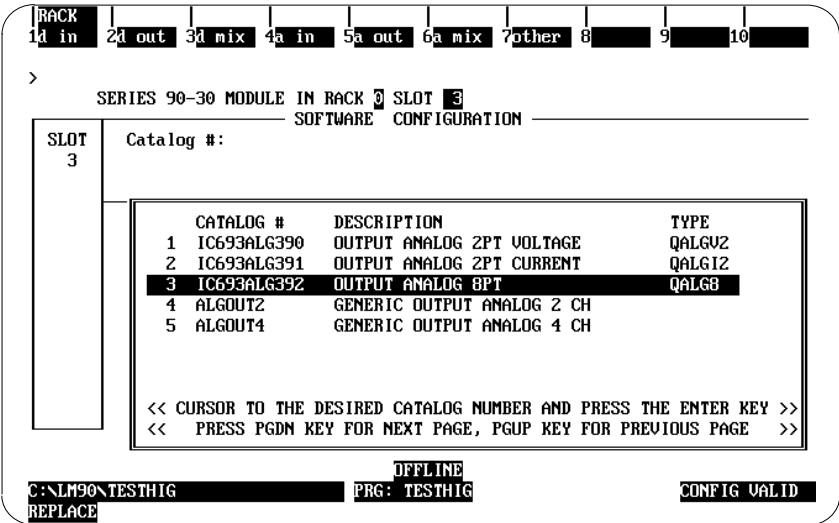
# Section 16: Configuring a High Density Analog Output Module

To configure a High Density Analog Output module, follow these steps:

- 1. Move the cursor to the desired rack and slot location. The slot may be either unconfigured or previously configured.
- 2. Press the **lm30 io** key (**F1**). You will then see a screen similar to the following:

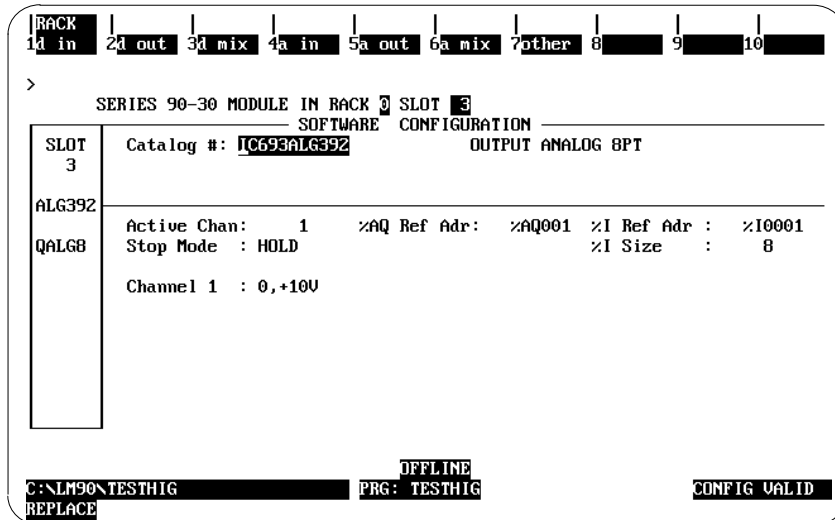


- 3. Press the **a out** key (**F5**). Your screen will now look like the one displayed below:



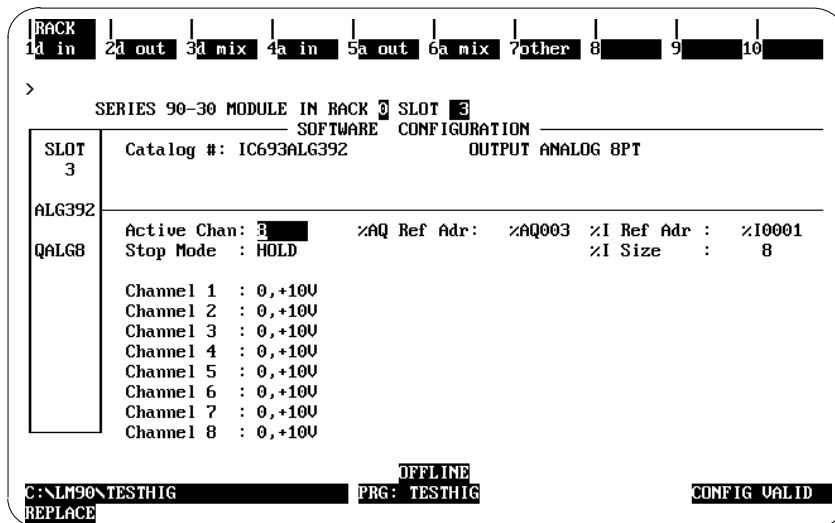
- 4. Move the cursor to the IC69ALG392 selection as shown above. Then press **Enter**.

The next screen that appears will look like the one displayed below:



- 5. Enter the remaining configuration parameters on this screen. You can move your cursor from field to field by pressing the **Cursor Movement** (or **Arrow**) keys. When you are in the field you want to modify, you can either type in your choice or press the **Tab** key to scroll through the available selections (or **Shift-Tab** to reverse the direction of the scrolling).

The default number of Active Channels (**Active Chan:**) is 1. You will not be able to configure additional channels until you change this field (by typing in the correct number (1 through 8) or by pressing the **Tab** key to increment the number). The screen displayed below shows the default selections after changing the **Active Chan:** field.



## Note

The entry in the **Stop Mode** field (**HOLD** or **DEFLOW**—which stands for default low) determines how the outputs will behave when the module is toggled from **RUN** to **STOP** mode. When this value is set to **HOLD** (the default), the outputs will retain their last state. When you change this value to **DEFLOW**, the output will go to zero.

## Other Configuration Considerations

Channels are scanned in sequential, contiguous order with channel 1 being the first to be scanned.

The entry in **%I Size** will only accept 8 or 16. This field denotes the number of bits returned to the user.

The only allowable entries for the **%AQ Ref Adr** are %AQ addresses. The only allowable entries for the **%I Ref Adr** are %I addresses.

The following tables show the configuration parameters applicable to this module.

Parameter	Parameter Description	Parameter Values	Parameter Defaults	Parameter Units
Stop Mode	Hold last state from RUN to STOP	HOLD or DEFLOW	HOLD	N/A
ACTIVEChannels	Number of channels converted	1 through 8	1	Channels
%AQ Ref Adr	Starting address for the %AQ reference type	Standard range	%AQ0001, or next highest available	N/A
%I Ref Adr	Starting address for the %I reference type	Standard range	%I0001, or next highest available	N/A
%I Size	Number of %I status locations	8 or 16	8	bits
Range of channel output (not identified as such on the screen)	Type of output range	0, +10V; -10, +10V; 4,20mA,0,20mA	0, +10V	Volts (Volt) mA (Curr)

### Notes:

1. The **%AQ Ref Adr** field is the reference address for the %AQ data and points to the start of the locations in the %AQ memory where the output data to the module begins. Each channel provides 16 bits of analog output data as an integer value from 0 to 32,767 or -32,768 to 32,767, depending on the range type selected. For detailed information of the data format, see the “CPU Interface to Analog Modules” section in chapter 3 of the *Series 90-30 Programmable Controller I/O Module Specifications* (GFK-0898).
2. The **%I Ref Adr** is the reference address for the %I data and points to the start of the locations in the %I memory (i.e., the Input Table) where status information from the module is reported. The user can select the number of %I status locations reported to the PLC by editing the value in the **%I Size** field. Values allowable in the %I Size field are 8 or 16, which refer to the number of %I locations reported to the PLC.

The **%I Ref Adr** field will only accept %I for %I Size values 8 or greater; the data brought back is in the format that follows:

The first eight %I locations (available for %I SIZE values 8, 16)

%I Locations	Description
%I	Module OK - '0' indicates NOT OK, '1' indicates module OK
%I+1	User Supply OK - Indicates when user supply is in specified limits; reads a '0' when User supply below specified limit, '1' when User supply OK
%I+2 - %I+7	Reserved for future modules. Not used in this module.

Second eight locations - (available for %I SIZE values 16)

%I Locations	Description
%I+8	Channel #1 BROKEN WIRE - 0 = OK, 1 = Wire broken (I modes only)
%I+9	Channel #2 BROKEN WIRE - 0 = OK, 1 = Wire broken (I modes only)
%I+10	Channel #3 BROKEN WIRE - 0 = OK, 1 = Wire broken (I modes only)
%I+11	Channel #4 BROKEN WIRE - 0 = OK, 1 = Wire broken (I modes only)
%I+12	Channel #5 BROKEN WIRE - 0 = OK, 1 = Wire broken (I modes only)
%I+13	Channel #6 BROKEN WIRE - 0 = OK, 1 = Wire broken (I modes only)
%I+14	Channel #7 BROKEN WIRE - 0 = OK, 1 = Wire broken (I modes only)
%I+15	Channel #8 BROKEN WIRE - 0 = OK, 1 = Wire broken (I modes only)

One of four output ranges can be selected. Two of which are voltage ranges. The default range is 0-10V, where output voltage values ranging from 0 to 10 volts correspond to 0 to 32000 integer values from the 90-30 CPU. The -10 to +10V range, when selected, corresponds from -32000 to 32000 from the CPU over an output voltage range of -10 to +10V. The two current ranges are 4 to 20 mA, and 0 to 20 mA. In each of the current ranges values between 0 and 32000 are sent to the module. Depending on which range is selected, will determine if the module is in Current or Voltage mode.

The following tables shows values sent from the CPU to the module:

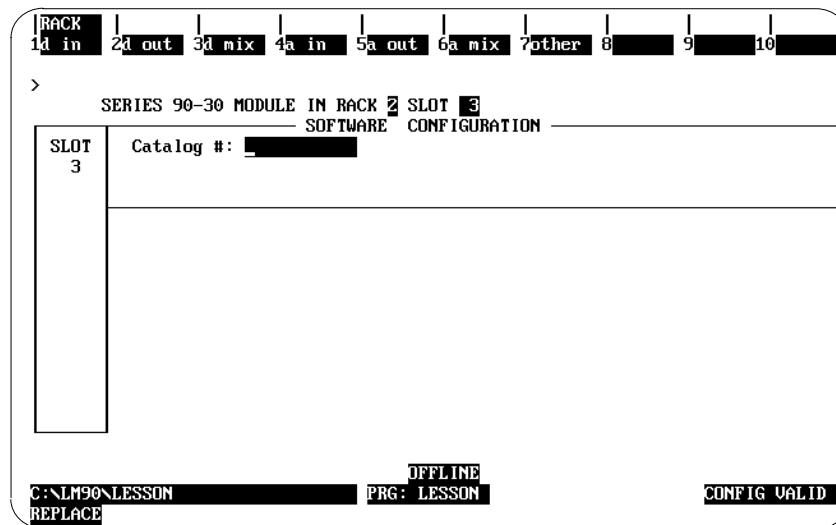
Range	Module Mode	*Allowed Values	Sent values from CPU
0 to 10 V	Voltage	0 to 32767	0 to 32767
-10 to 10 V	Voltage	- 32768 to 32767	-32768 to 32767
4 to 20 mA	Current	0 to 32000*	0 to 32767
0 to 20 mA	Current	0 to 32767	0 to 32767

\*The phrase, *Allowed Values*, refers to the values that are valid. If you send a value > 32000, the module will truncate that value to 32000 before sending it to the DAC.

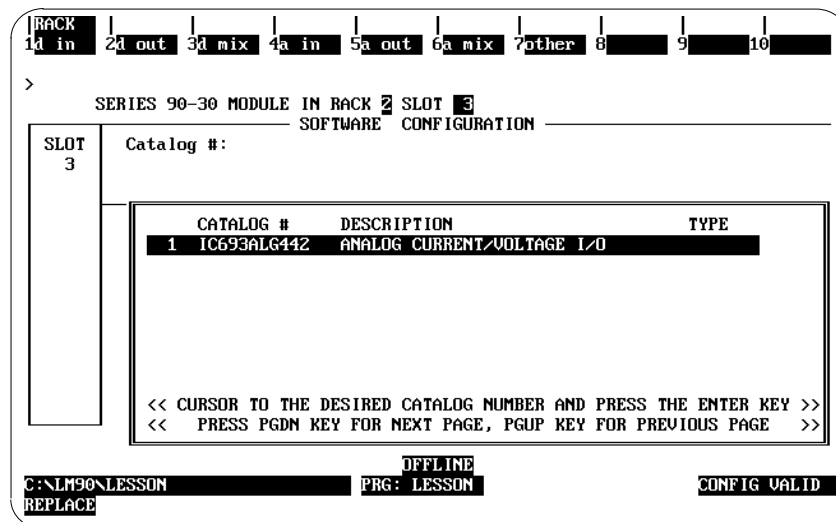
## Section 17: Configuring an Analog Combo Module

To configure an Analog Combo module, follow these steps:

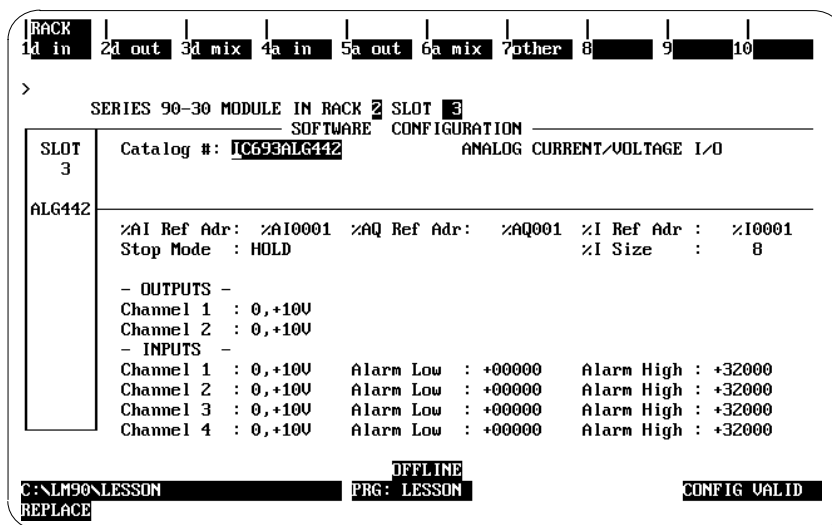
1. Move the cursor to the desired rack and slot location. The slot may be either unconfigured or previously configured.
2. Press the **lm30 io** key (**F1**). Your screen will change to one similar to the one shown below.



3. From this screen, press the **a mix** key (**F6**). Your screen will change to one similar to the one shown below.



4. Currently, there is only one selection. (If more than one selection appears, use your **Cursor Movement** (or **Arrow**) keys to move to Catalog # IC693ALG442.) Press **Enter** to accept this selection and to move to the screen shown on the top of the next page.



5. All the remaining configuration need to be done on this screen. You can move your cursor from field to field by pressing the **Cursor Movement** (or **Arrow**) keys. When you are in the field you want to modify, you can either type in your choice or press the **Tab** key to scroll through the available selections (or **Shift-Tab** to reverse the direction of the scrolling).

### Note

The entry in the **Stop Mode** field (**HOLD** or **DEFLOW**) determines how the outputs will behave when the module is toggled from **RUN** to **STOP** mode. When this value is set to **HOLD** (the default), the outputs will retain their last state. When you change this value to **DEFLOW** the output will go to zero.

### Other Configuration Considerations

The **Alarm Low** limit for each channel must be less than its corresponding **Alarm High** limit.

The entry in **%I Size** will only accept 8, 16 and 24. This field denotes the number of bits returned to the user.

The only allowable entries for the **%AQ Ref Adr** are %AQ addresses. The only allowable entries for the **%I Ref Adr** are %I addresses.

## Analog Combo Configuration Parameters

Parameter	Parameter Description	Parameter Values	Parameter Defaults	Parameter Units
STOP MODE	Hold last state from RUN to STOP	HOLD or DEFLOW	HOLD	N/A
%AIADR	Starting address for the %AI reference type	standardrange	%AI0001, or next highestavailable	N/A
%AQ ADR	Starting address for the %AQ reference type.	standardrange	%AQ0001, or next highestavailable	N/A
%I ADR	Starting address for the %I reference type	standardrange	%I00001, or next highestavailable	N/A
%I SIZE	Number of %I status locations	8, 16, 24	8	bits
RANGE	Type of input range and range	0,+10 V, -10,+10 V, 4,20 mA, 0, 20mA	0,+10 V	volts (Volt) mA (Curr)
ALARMLO	Low limit alarm value	-32768 – 32752	0	User counts
ALARM HIGH	High limit alarm value	-32760 – 32760	+32000	User counts

### Notes:

1. The **%AI Ref Adr** field is the reference address for the %AQ data and points to the start of the locations in the %AI memory where the output data to the module begins. Each channel provides 16 bits of analog output data as an integer value from 0 to 32,767 or -32,768 to 32,767. depending on the range type selected. For detailed information of the data format, see the “CPU Interface to Analog Modules” section in chapter 3 of the *Series 90-30 Programmable Controller I/O Module Specifications* (GFK-0898).
2. The **%AQ Ref Adr** field is the reference address for the %AQ data and points to the start of the locations in the %AQ memory where the output data to the module begins. Each channel provides 16 bits of analog output data as an integer value from 0 to 32,767 or -32,768 to 32,767. depending on the range type selected. For detailed information of the data format, see the “CPU Interface to Analog Modules” section in chapter 3 of the *Series 90-30 Programmable Controller I/O Module Specifications* (GFK-0898).
3. The **%I Ref Adr** is the reference address for the %I data and points to the start of the locations in the %I memory (i.e., the Input Table) where status information from the module is reported. You can select the number of %I status locations reported to the PLC by editing the value in the **%I Size** field. Values allowable in the %I Size field are 8 or 16, which refer to the number of %I locations reported to the PLC. For %I SIZE values 8 or greater, the data brought back is in the format described in the table on the next page.



First eight %I locations – (available for %I SIZE values 8, 16, 24)

%I Locations	Description
%I	Module OK – '0' indicates NOT OK, '1' indicates module OK
%I+1	User Supply OK – Indicates when user supply is in specified limits; reads a '0' when User supply below specified limit, '1' when User supply OK
%I+2 – %I+3	Reserved for future modules. Not used in this module.
%I + 4 – %I + 7	E2 COMMREQ Status Bits %I + 4 = 0 OK , if 1 = Invalid channel requested %I = 5 = 0 OK , if 1 = Invalid alarm setting Low > High or Low or High < 0 in unipolar mode

Second eight locations (available for %I SIZE values 16, 24)

%I Locations	Description
%I+8	Input: Ch #1 ALARM LO – '0' indicates value above limit, '1' below
%I+9	Input Ch #1 ALARM HI – '0' indicates value below limit, '1' above
%I+10	Input Ch #2 ALARM LO – '0' indicates value above limit, '1' below
%I+11	Input Ch #2 ALARM HI – '0' indicates value below limit, '1' above
%I+12	Input Ch #3 ALARM LO – '0' indicates value above limit, '1' below
%I+13	Input Ch #3 ALARM HI – '0' indicates value below limit, '1' above
%I+14	Input Ch #4 ALARM LO – '0' indicates value above limit, '1' below
%I+15	Input Ch #4 ALARM HI – '0' indicates value below limit, '1' above

The third eight locations (available for %I SIZE values 24)

%I Locations	Description
%I+16	Output Ch #1 BROKEN WIRE 0 = OK, 1 = Wire Broken (Imodes only)
%I+17	Output Ch #2 BROKEN WIRE 0 = OK, 1 = Wire Broken (Imodes only)
%I+18 .. %I+23	Reserved for future modules. Not used in this module

One of four input or output ranges can be selected. Two of which are voltage ranges. The default range is 0–10V, where input or output voltage values range from 0 to 10 volts. In input mode they report 0 to 32000 integer values to the 90-30 CPU and in output mode values between 0 and 32000 are sent to the module. The –10 to +10V range, values between –32000 to 32000 are sent or received from the CPU over an input voltage range of –10 to +10V. The two current ranges are 4 to 20 mA, and 0 to 20 mA. In each of the current ranges, values between 0 and 32000 are reported back from the module to sent to the module for the entire range.

The following tables shows values sent from the CPU to the module for the Output channels:

Range	Module Mode	*Allowed Values	Sent values from CPU
0 to 10 V	Voltage	0 to 32767	0 to 32767
-10 to 10 V	Voltage	- 32768 to 32767	-32768 to 32767
4 to 20 mA	Current	0 to 32000*	0 to 32767
0 to 20 mA	Current	0 to 32767	0 to 32767

\*The phrase, *Allowed Values*, refers to the values that are valid. If a user sends a value > 32000, the module will truncate that value to 32000 before sending it to the DAC.

The following table shows values sent from the Module back to the PLC for the Input channels

Range	Module Mode	Sent values to CPU
0 to 10 V	Voltage	0 - 32767
-10 to 10 V	Voltage	-32768 - 32767
4 to 20 mA	Current	0 - 32767
0 to 20 mA	Current	0 - 32767

The ALARM LO and ALARM HI data fields allow the user to enter values that cause 'alarm' indications to be passed to the PLC. Each channel has a low limit alarm value (ALARM LO) and a high limit alarm value (ALARM HI). These alarm values cause %I points to be set as indicated in the tables above. Values can be entered in all high and low limit fields, even though those channels may not be enabled. Values entered without a sign are assumed to be positive. Value checking should be done to determine if the ALARM LO and ALARM HI values are allowable for the appropriate RANGE. The allowable values are:

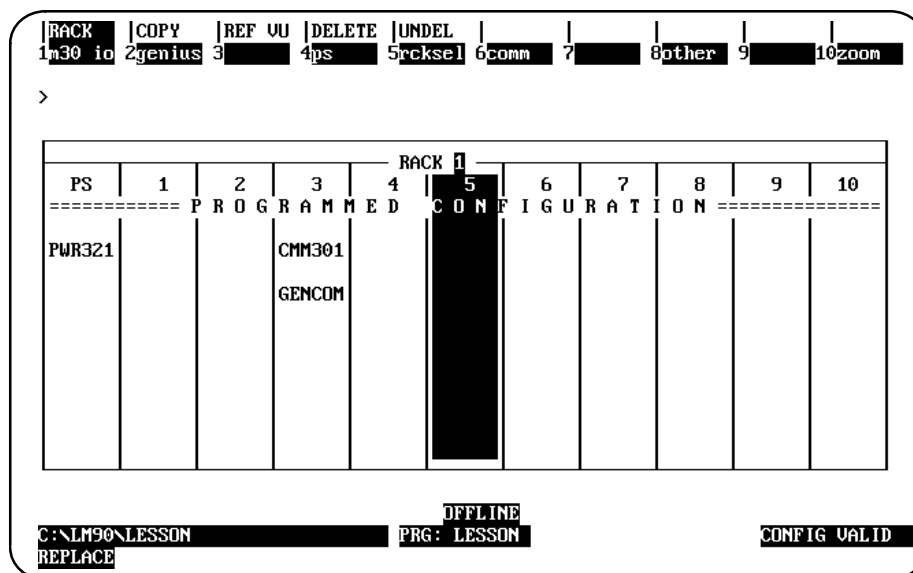
RANGE	Possible limit values
4-20 mA	0..32760
0-20 mA	0..32760
0-10V	0..32760
-10 - +10V	-32768..32760

## Section 18: Configuring a Third-Party Module

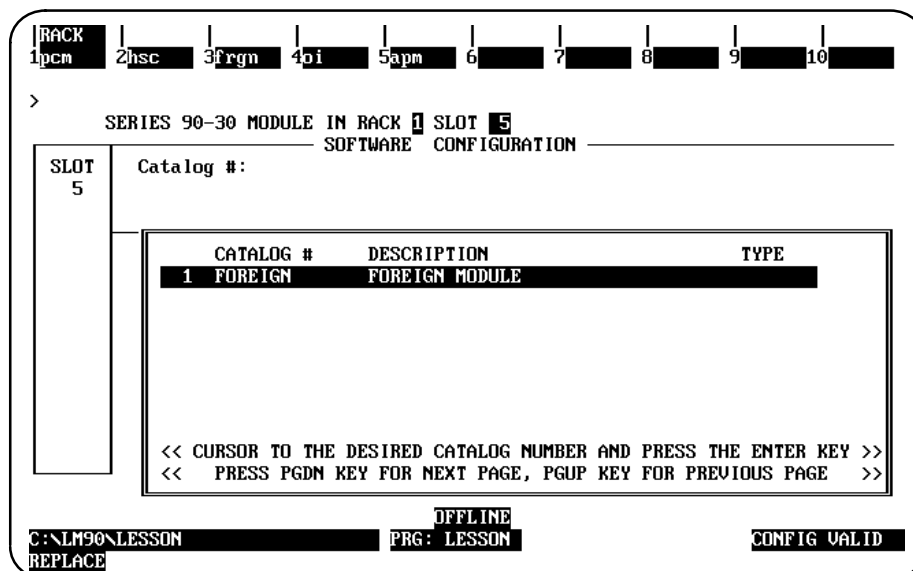
This section describes configuration of third-party modules. Reference addresses can be configured for %I, %Q, %AI, and %AQ, in addition to a module ID and 16 bytes of hexadecimal data.

To configure a third-party module on the I/O Configuration Rack screen:

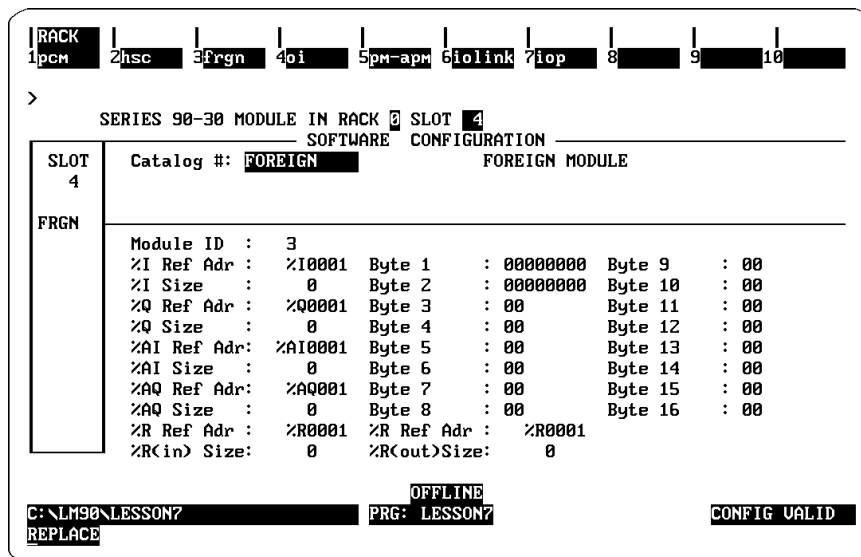
1. Move the cursor to the desired rack and slot location. The slot may be either unconfigured or previously configured.



2. Press **Other (F8)** and then **Foreign (F3)** from the I/O Configuration Rack screen to display the list of third-party modules.



3. Press the **Enter** key to display the detail screen.



Parameter	Description
Module ID	A 1, 2, or 3-digit signed integer value representing a vendor module designation. A range of valid vendor module IDs is used to allow validation.
Reference Data	Enter the starting address and length for each of the %I, %Q, %AI, %AQ, and %R references. The reference address parameters default to the next highest reference address. The size parameters default to zero and are validated to be within the configuration memory limits and the configured CPU for the respective reference type.
Soft Switch Data	Sixteen bytes of soft switch data can be configured. Bytes 1 and 2 are binary values (00000000 to 11111111). Bytes 3 through 16 are byte hexadecimal values (00 to FF).

### Note

Horner Electric modules must be configured as Foreign and the Module ID must remain at the default value of 3. Refer to the Horner data sheet for Soft Switch Data which is required to configure the modules. All data sheets are available on the GE Fanuc BBS (804) 978-5458 in the Horner Electric area under PLC files. Note that configuration data can be stored to a Stopped PLC, but configuration data will **not** be transferred to Horner modules like with GE Fanuc smart modules. You must power-cycle the PLC for Horner modules to receive updated configuration data from the CPU.

---

## Section 19: Configuration Reference View

The configuration reference view feature enables you to view tabular displays of configured modules with the same reference (%I, %Q, %AI, %AQ, %G, or %R). This feature can be helpful when assigning new reference addresses or resolving address conflicts, such as overlapping.

### Note

The configuration reference view feature is not available for the CPU 211.

The reference view table is sorted in ascending order by user reference, with the lowest address listed first. Mixed discrete inputs and outputs (%QI) are shown in both the %Q and %I tables. Mixed analog inputs and outputs (%AQI) are shown in both the %AQ and %AI tables.

Data contained in the reference view table may only be viewed; it cannot be edited. Editing may only be done on the detail screens. Modules configured from the rack screens have a reference address assigned to them. When a module is configured, it is automatically entered into the Reference View table for its reference type. When a module is deleted, it is automatically removed from the table.

When the reference address of a module is changed, the reference view table is updated automatically.

## Displaying the Reference View Table

The Reference View screen for the current module reference type may be displayed by pressing **Reference View (Shift-F3)** from any rack or detail screen in the I/O configuration rack function. From the detail screen, the cursor will be positioned at that same module on the Reference View screen.

To display a different Reference View screen, after pressing **Shift-F3** to select the reference view function, press the appropriate function key (F3 through F8) for the view screen you wish to display. For example, to display the %I Discrete Input Reference screen shown below, press **Reference View (Shift-F3)** from an input module detail screen, or **Shift-F3** and then **%I View (F3)** from the detail screen of another reference type.

REFERENCE START - END	PHYSICAL ADDRESS	ID TYPE	MODULE TYPE	DESCRIPTION
00001-00016	0.2	90-30	I AC16	INPUT 120 VAC 16PT
00017-00048	0.3	90-30		GENERIC INPUT/OUTPUT 32 PT
00049-00064	0.4	90-30	HSC	HIGH SPEED COUNTER MODULE
00065-00096	0.7	APM	APM	AXIS POSITIONING MODULE 1-AXIS
00097-00128	0.8	APM	APM	AXIS POSITIONING MODULE 2-AXIS

OFFLINE  
 C:\LM90\LESSON PRG: LESSON ADDR OVERLAPS: N CONFIG VALID  
 REPLACE ENTRY: 1 TOTAL ENTRIES: 5

The first line of the display area clearly identifies the reference type of this view. On the second line, **TotalUsed** on an analog reference view screen shows the total number of references configured. On a discrete reference view screen, **Total I+Q** shows the combined total. The highest reference number configured is also displayed on the second line.

Each entry in the reference view table contains the following information:

Field	Description
Reference Start - End	The starting and ending reference addresses for the module.
PhysicalAddress	The PLC hardware physical address for the module. Each entry in this column will contain the rack and slot information, each separated by a period (for example,rack.slot).
I/OType	The input/output type which controls this module.
Module Type	The type and size of the module.
Description	The description field.

If overlaps exist in the table, a **Y** (Yes) is displayed on the second status line at the bottom of the screen. If there are no overlaps, an **N** (No) is displayed.

The position of the selected entry is displayed in the **Entry** field, and the number of reference table entries is displayed in the **TotalEntries** field. For example, in the previous screen, the first entry is the selected entry, and the total number of entries is five.

### Moving the Cursor

Use the Up and Down cursor movement keys to scroll between rows of the Reference View table. If the cursor is at the beginning or end of the table, pressing these keys will scroll additional information not already displayed, one single row at a time. The cursor field will always include an entire module (e.g., a single row).

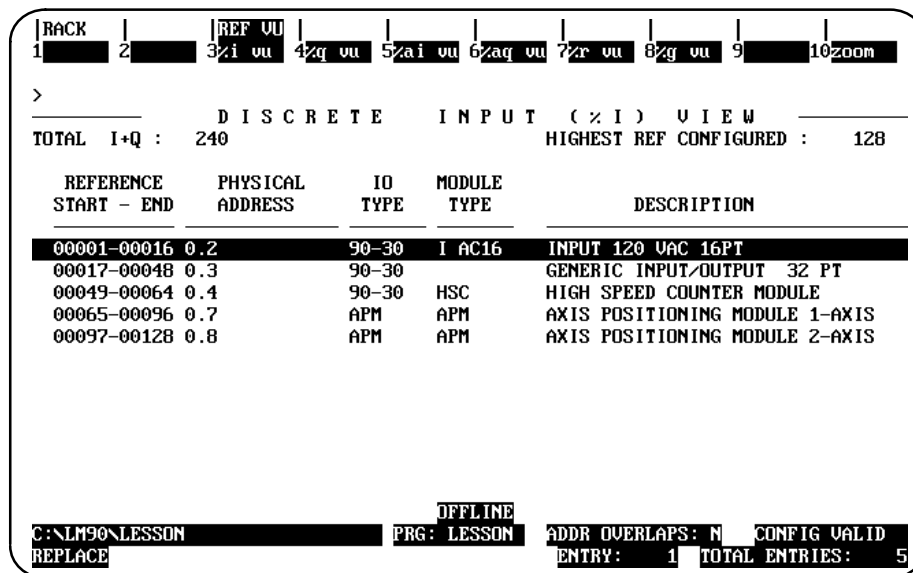
Press the **Home** key to position the cursor at the beginning of the table, or the **End** key to go to the end of the table.

If all the configured modules in the PLC cannot be displayed on a single screen, press the **Page Down** key to view additional entries, or the **Page Up** key to view previous entries.

### Displaying the Detail Screen

All of the Reference View screens have a **Zoom (F10)** function key displayed at the top of the screen. This key may be used to go directly to the detailed module screen for the module highlighted on the Reference View screen.

In the following example, the 16-point input module with a starting reference address at 00001 is shown in reverse video.



When the **Zoom (F10)** key is pressed, the detail screen for this module is displayed.

To return to the Reference View screen, press **Reference View (Shift-F3)**.

## Displaying the Rack Screen

All of the Reference View screens have a **Rack (Shift-F1)** function key displayed at the top of the screen. This key may be used to go directly to the Rack screen from the reference view function.

In the following example, the High Speed Counter module is shown in reverse video. This module is physically located in slot 4 of the main rack.

```

| RACK | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----|
>
DISCRETE INPUT (%I) VIEW
TOTAL I+Q : 240 HIGHEST REF CONFIGURED : 128

REFERENCE PHYSICAL IO MODULE
START - END ADDRESS TYPE TYPE DESCRIPTION
-----
00001-00016 0.2 90-30 I AC16 INPUT 120 VAC 16PT
00017-00048 0.3 90-30 GENERIC INPUT/OUTPUT 32 PT
00049-00064 0.4 90-30 HSC HIGH SPEED COUNTER MODULE
00065-00096 0.7 APM APM AXIS POSITIONING MODULE 1-AXIS
00097-00128 0.8 APM APM AXIS POSITIONING MODULE 2-AXIS

OFFLINE
C:\LM90\LESSON PRG: LESSON ADDR OVERLAPS: N CONFIG VALID
REPLACE ENTRY: 3 TOTAL ENTRIES: 5
  
```

When **Rack (Shift-F1)** is pressed from the Reference View screen, the screen will display the main rack and highlight the slot of the module selected on the Reference View screen (in this example, slot 4).

```

| RACK | COPY | REF UU | DELETE | UNDEL | 6 | 7 | 8 | 9 | 10 |
|-----|
>
RACK 4
=====
PS | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
=====
PWR321 CPU331 MDL240 QI 32 APU300 PCM300 CMM311 APU301 APU302 ADC311
I AC16 HSC PCM CMM APM APM ADC
RefAdr RefAdr
%I0001 %I0017
%Q0065 %Q0097
%AI016 %AI040
%AQ001 %AQ007

OFFLINE
C:\LM90\LESSON PRG: LESSON CONFIG VALID
REPLACE
  
```

To return to the Reference View screen, press **Reference View (Shift-F3)**.



## Overlapping References

Overlapping references may result in an invalid configuration. When the reference address assigned to a module overlaps with another module's reference address of the same reference type, an asterisk (\*) is displayed at the beginning of the row for both modules. If the overlap is fatal, resulting in CONFIG INVALID, double asterisks (\*\*) are displayed.

REFERENCE START - END	PHYSICAL ADDRESS	IO TYPE	MODULE TYPE	DESCRIPTION
* 00001-00016	0.2	90-30	Q AC12	OUTPUT 120 VAC 0.5A 12PT
* 00001-00005	2.6	90-30	Q AC5	OUTPUT 120/240 VAC 2A 5PT ISOL
00017-00032	1.4	90-30	QRLY16	OUTPUT RELAY 2A 16PT
00033-00040	2.3	90-30	Q ACB	OUTPUT 120/240 VAC 1A 8PT

OFFLINE  
 C:\LM90\LESSON PRG: LESSON ADDR OVERLAPS: Y CONFIG VALID  
 REPLACE ENTRY: 4 TOTAL ENTRIES: 4

To resolve the conflicting overlap, position the cursor on the reference address which must be changed and press **Zoom (F10)**. The detail module screen will be displayed, allowing the address to be adjusted.

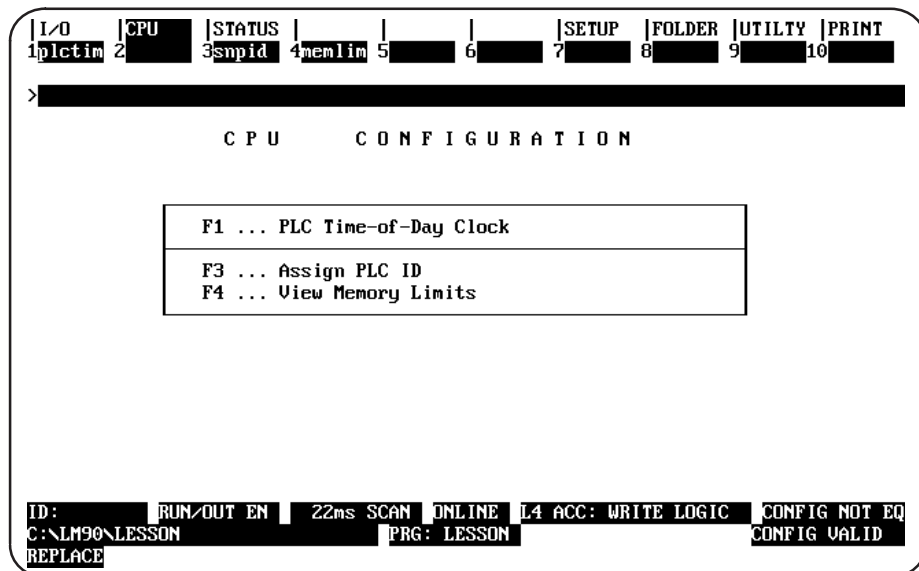
**Caution**

**When adjusting reference addresses, be careful not to cause another overlap.**

# Chapter 11

## CPU Configuration

Use the CPU configuration function to set the operating characteristics of the CPU. To access the CPU configuration functions, press **CPU (F2)** from the main menu. Then continue below.



Function Key	Function	Description	Page
F1	PLC Time	View the computer's time-of-day clock, or view and change the PLC time-of-day clock.	11-2
F3	SNP ID	Assign a new value to the Series Ninety Protocol (SNP) ID.	11-4
F4	Memory Limits	Display the CPU memory allocation.	11-5

These functions are described on the pages that follow.

### Storing the CPU Configuration to the PLC

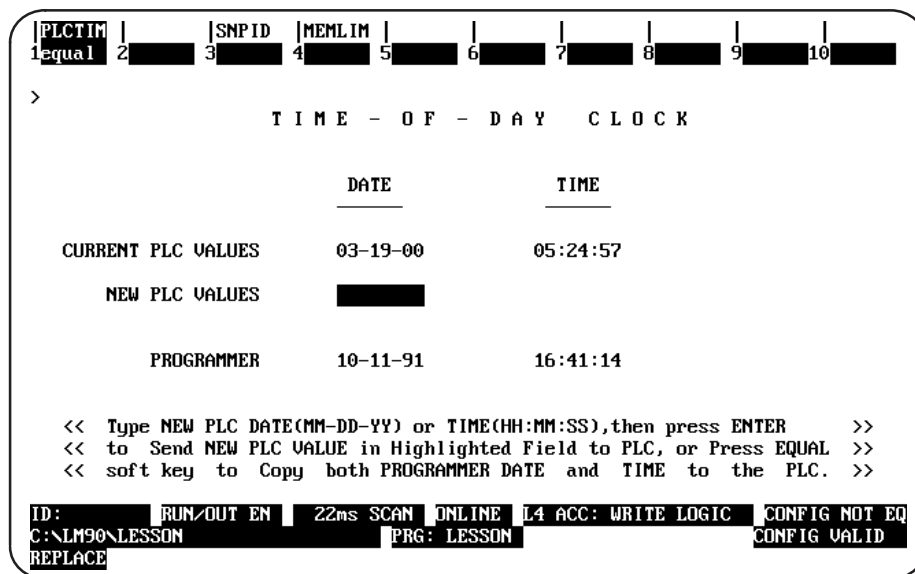
Use the program utility functions (in **ONLINE** mode) to store configuration data to the PLC. For instructions, refer to chapter 8, "Program Utilities."

## PLC Date and Time

To display the current date and time, press **PLC Time (F1)**.

### Note

The PLC time-of-day clock function is only available with the Model 331 and higher CPUs and the 28-point Micro PLCs. It is not available with the Model 311 or Model 313 CPU. If the **PLCTIM (Shift-F1)** key is pressed, the following screen is disabled and the software displays a message indicating that the function is not available.



### Note

When this screen is first displayed, the time-of-day clock is read from the PLC. Elapsed time is tracked by the Logicmaster 90-30/20/Micro software. Other devices, such as CIMPLICITY 90-ADS, may change the time-of-day clock, but the change will not be reflected on this screen until you exit and return to this screen.

The current date and time for the programmer are displayed in all operating modes. To also display the PLC date and time, place the computer in **ONLINE** or **MONITOR** mode. If the programmer is in **OFFLINE** mode or is not communicating with the CPU, asterisks are displayed in the *Current PLC Values* fields.

### Changing the PLC Date and Time

To change the date and/or time shown for the PLC, the computer must be in **ONLINE** mode and communicating with the PLC.

To make the PLC date and time the same as the values shown for the computer, press **Equal (F1)**. To use a different date and/or time:

1. Enter the new date, using dash (minus) characters between fields (MM-DD-YY).

Abbreviation	Description
MM	Month from 1 to 12.
DD	Day from 01 to 31.
YY	Year from 00 to 99.

2. Enter the new time, using colons between fields (HH:MM:SS).

Abbreviation	Description
HH	Hours from 00 to 23.
MM	Minutes from 00 to 59.
SS	Seconds from 00 to 59.

### Note

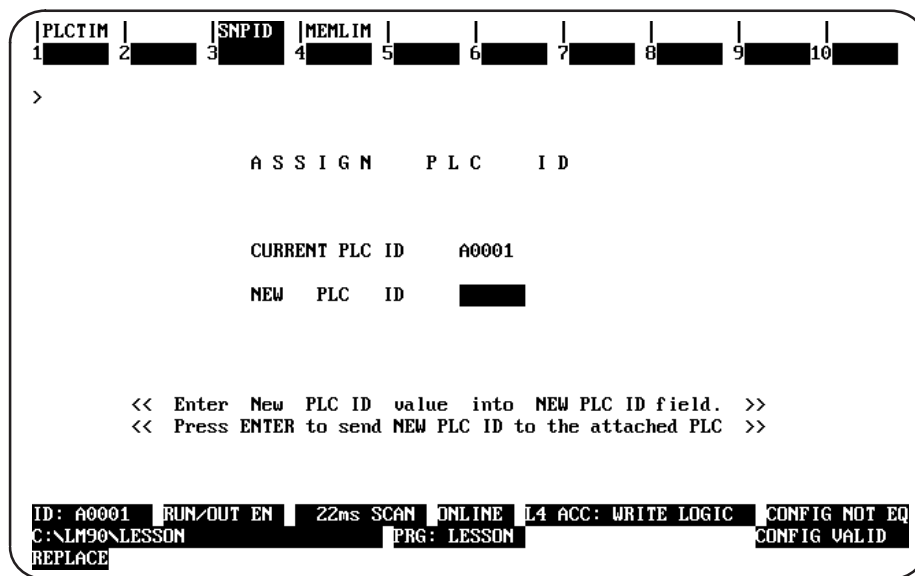
If hours are specified, entering minutes and seconds is optional. Partial times must, however, end with a colon. For example, HH: or HH:MM:

3. Press the **Enter** or **Escape** key. The change in the currently highlighted field is sent to the PLC and will appear in the *Current PLC Values* fields on the screen.

## SNP ID

This SNP ID is for the Power Supply port only (see Note below). For multidrop configurations, each CPU connected to the system must have a unique identification name consisting of 1 to 6 characters. With 6.6 and later CPUs, you have the option of using 1 to 7 characters. All previous releases allow up to 6 characters. The SNP ID is not required for peer-to-peer communications.

The current SNP ID name is displayed in **ONLINE** or **MONITOR** mode. To display the current SNP ID or assign a new name, press **SNP ID (F3)**. The entry for the *New SNP ID* field can only be changed in **ONLINE** mode with communications. If the programmer is in **OFFLINE** mode or is not communicating with the CPU, asterisks are displayed in the *Current SNP ID* field.



### Note

On Release 7 351 and 352 PLCs, Port 1 and Port 2 have their IDs stored in the configuration (see page 10-18 and following for 351/352 port configuration and each port's SNP ID).

### Changing the SNP ID Name

To change the identification of the PLC:

1. Enter a new name in the *New SNP ID* field, using the alphanumeric characters (A – Z, 0 – 9) or special characters (-, @, \_, #, \$, %, <, >, =, +, &). The first character must be alphabetic.

### Note

Lower-case characters may be entered using the Hand-Held Programmer; however, Logicmaster 90-30/20/Micro software supports only upper-case characters.

2. To clear the SNP ID, enter all blank characters.
3. Press the **Enter** key to send this new name to the attached SNP device.

## PLC Memory Limits

To display the reference and logic memory limits for the PLC, press **Memory Limits (F4)**. The limits are fixed, based on the model of the CPU. This screen shows the fixed memory limits for a CPU 331.

1	PLCTIM	2	SNPID	3	MEMLIM	4		5		6		7		8		9		10	
>																			
P L C   M E M O R Y   C O N F I G U R A T I O N																			
Type	Discrete Input %I	System Use %S	Discrete Output %Q	Internal Discrete %M	Temporary Status %T	Global Genius %G													
Size (Bits)	512 512	128 128	512 512	1024 1024	256 256	1280 1280	CFG PLC												
Type	Analog Input %AI (Words)	Analog Output %AQ (Words)	Register Memory %R (Words)	CPU Memory Total (Bytes)	CPU Model Number														
Size	128 128	64 64	2048 2048	16384 16384	331 331	CFG PLC													
ID: A0001   RUN/OUT EN   22ms SCAN   ONLINE   L4 ACC: WRITE LOGIC   CONFIG NOT EQ																			
C:\LM90\LESSON   PRG: LESSON   CONFIG VALID																			
REPLACE																			

The current memory allocations stored in the program folder are displayed in all operating modes. To display the PLC values, place the computer in **ONLINE** or **MONITOR** mode.

The upper portion of this screen shows the maximum values for discrete references (%I, %Q, %M, %T, %S, and %G). These values are set by the system and will change only when the CPU model number is changed.

The lower portion of this screen shows values for register references (%AI, %AQ, and %R). These values are also set by the system and will change only when the CPU model number is changed. The sum of CPU memory, according to the particular CPU model is displayed in the *CPU Memory Total* field.

### Configurable Memory

Beginning with Release 9.02, all model 351 and higher CPUs have configurable %R, %AQ, and %AI memory. Refer to page 10-24 for information about this feature.

This appendix contains a tutorial for the programming software package. The lesson includes:

- Creating a program folder.
- Creating a program.
- Entering a variable declaration.
- Adding ladder logic to the program.
- Printing the program.
- Exiting the programmer.

## **Help Screens**

Logicmaster 90-30/20/Micro software includes detailed Help screens. These Help screens are loaded onto the hard disk of your programmer during the software installation procedure and are readily accessible. To access the Help screens, press **ALT-H** for help, **ALT-I** for instruction mnemonic help, or **ALT-K** for key help.

## **Starting the Lesson**

Before you can start, the programming software must be installed and started. If that has not been done yet, please turn back to chapter 2, "Operation," for instructions.

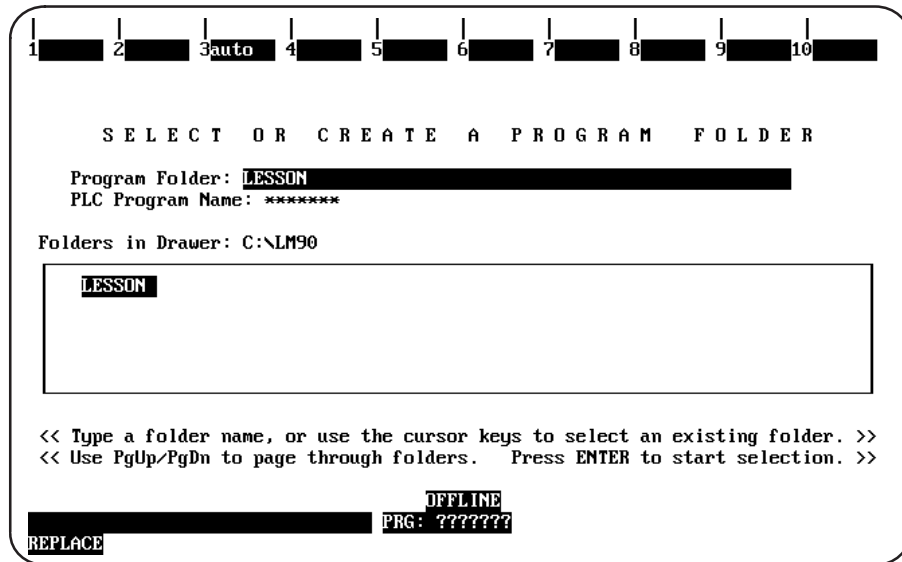
## **Exiting the Programmer**

You can exit the programmer at any time by pressing the **CTRL-Break** keys.

## Creating a Program Folder

For this lesson, you will create and use a program folder named **LESSON**.

When Logicmaster 90-30/20/Micro software is started (if the current default directory is an existing program folder), a screen similar to the one shown below appears. If the software is already running, you can display a similar screen from the Program Folder menu by pressing **Folder (F8)** or **Shift-F8** from the Programming Software main menu and then **Select (F1)**.





1. If the current default directory is not an existing program folder, type in a name of seven characters or less for the folder. For this lesson, type **LESSON**.
2. Press the **Enter** key. The following prompt will appear at the top of the screen:

Program folder does not exist; create new folder? (Y/N)

3. To create the program folder, enter **Y** (Yes). The software will accept the name and display the Programming Software main menu.

```
|PROGRAM |TABLES |STATUS | | | |SETUP |FOLDER |UTILITY |PRINT
1|program 2|tables 3|status 4| 5| 6| 7|setup 8|folder 9|utility10|print
>
SERIES 90-30 / 90-20 PROGRAMMING SOFTWARE
Version 4.01 Direct Serial - COM
F1 ..... Program Display/Edit
F2 ..... Reference Tables
F3 ..... PLC Control and Status
F7 ..... Programmer Mode and Setup
F8 ..... Program Folder Functions
F9 ..... Utility: Load/Store/etc.
F10 ..... Print Functions
<< Press ALT-K at any time to see special key assignments >>
OFFLINE
C:\LM90\LESSON PRG: LESSON
REPLACE
```

The name of your new program folder (**LESSON**) should be displayed at the bottom of the screen.

## Creating a Program

Create a program in this program folder by pressing **Program (F1)**. The following screen will be displayed:

```

PROGRAM | TABLES | STATUS | | | | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3modify 4search 5 6 7option 8goto 9more 10zoom
>
[ START OF LD PROGRAM LESSON ] (* *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]
[ END OF PROGRAM LOGIC ]

OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: _MAIN SIZE: 123 RUNG 0004
REPLACE : :
```

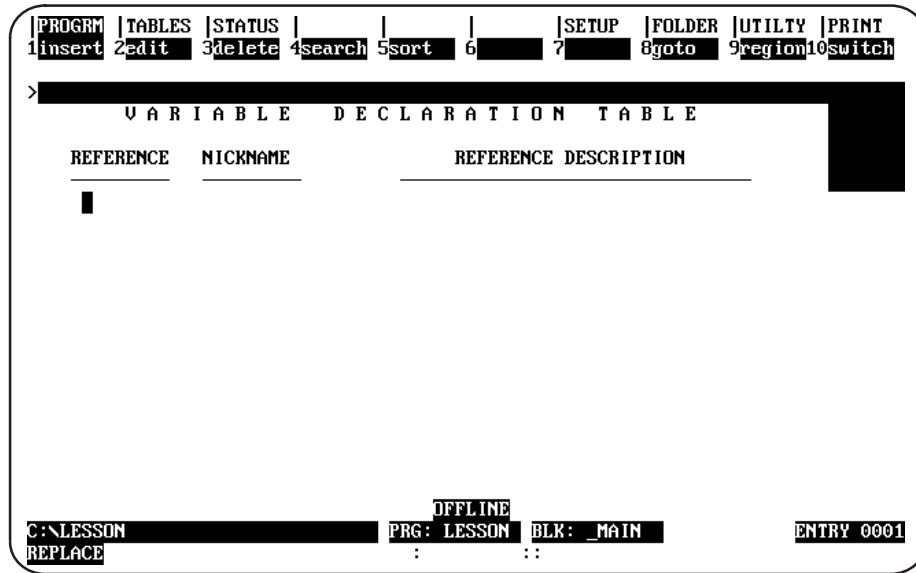
The screen shows a **[VARIABLE DECLARATIONS]** marker where variable declarations can be inserted. Variable declarations are used to list nicknames and reference descriptions for the block. At the **\_MAIN** block, the nicknames listed in the variable declaration table are known throughout all blocks; in a subroutine block, the nicknames are local and known only to that particular block.

Some programs will contain no declarations. This lesson shows how to enter a variable declaration.

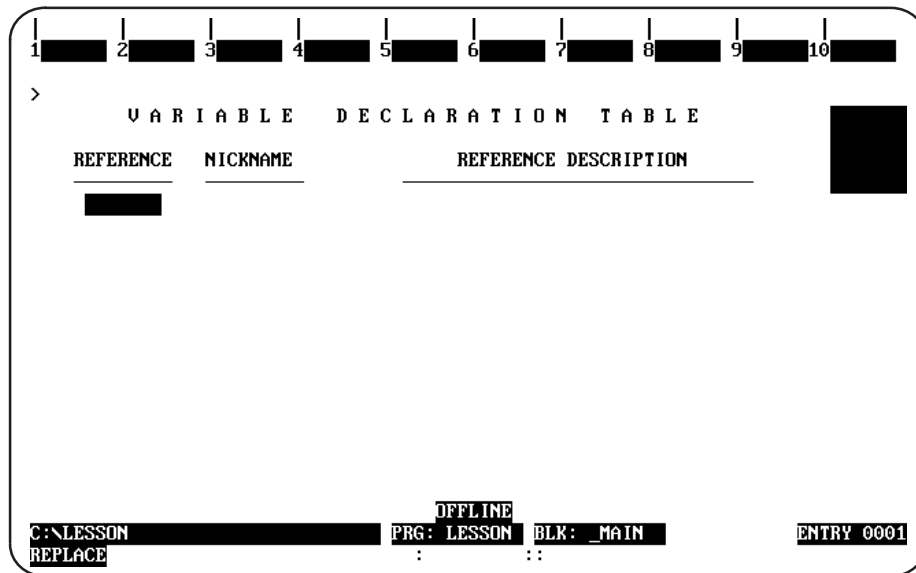
## Entering a Variable Declaration

To enter a variable declaration:

1. Use the cursor keys to move the cursor block to the [VARIABLE DECLARATIONS] marker.
2. Press **Zoom (F10)** to display a table where variables for the program can be entered.



3. Press **Insert (F1)** to begin entering information in the *Reference* field.



4. Enter the machine reference which the nickname will represent. For this lesson, type %I33 or 33I.
5. Press the **Enter** key. The cursor block moves to the *Nickname* field.
6. Enter the nickname *count*, and press the **Enter** key. The cursor block moves to the *Reference Description* field.

```
1 [ ] 2 [ ] 3 [ ] 4 [ ] 5 [ ] 6 [ ] 7 [ ] 8 [ ] 9 [ ] 10 [ ]
>
  V A R I A B L E   D E C L A R A T I O N   T A B L E
  REFERENCE  NICKNAME  REFERENCE DESCRIPTION
  -----  -
  %I0033    count
           [REDACTED]
```

OFFLINE  
C:\LESSON PRG: LESSON BLK: \_MAIN ENTRY 0001  
REPLACE :

7. For this lesson, we will not use the *Reference Description* field. Press the **Escape** key to save your entries and leave **INSERT** mode. Note that when you press the **Escape** key, the software automatically changes your entry in the *Nickname* field to uppercase letters.

```
PROGRAM | TABLES | STATUS | search | sort | SETUP | FOLDER | UTILITY | PRINT
1 insert | 2 edit | 3 delete | 4 goto | 5 region | 6 switch
>
  V A R I A B L E   D E C L A R A T I O N   T A B L E
  REFERENCE  NICKNAME  REFERENCE DESCRIPTION
  -----  -
  %I0033    COUNT
           [REDACTED]
```

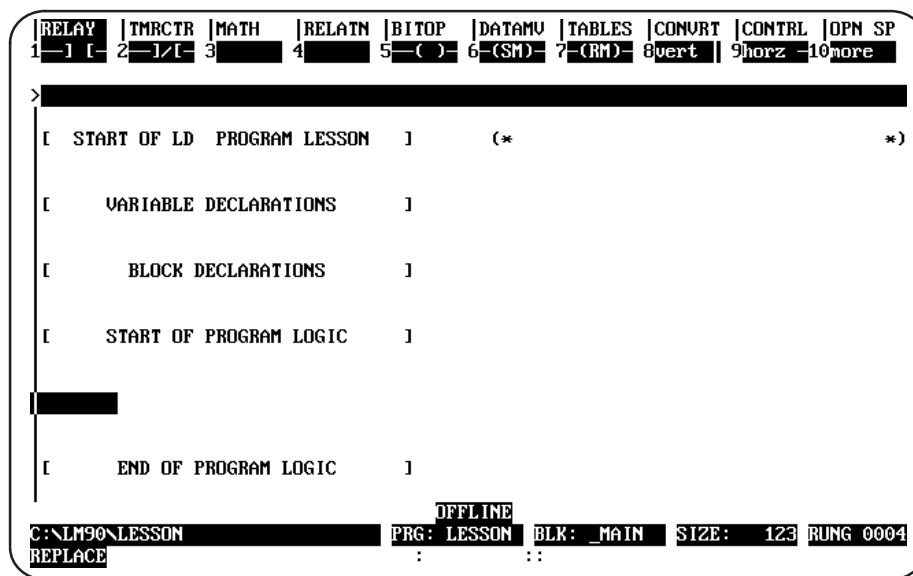
OFFLINE  
C:\LESSON PRG: LESSON BLK: \_MAIN ENTRY 0001  
REPLACE :

8. Press the **Escape** key again to return to the program ladder logic display.

## Adding Ladder Logic to the Program

To create logic for the example program:

1. On the program screen, use the Down cursor key to move the cursor to the [END OF PROGRAM LOGIC] marker.
2. Press **Insert (F1)**. You can now insert ladder logic at the cursor location.



The top line of the display shows these ladder logic functions:

- Relay contacts and coils.
- Timers and counters.
- Math functions.
- Relational (comparison) functions.
- Logical bit operation functions.
- Data move functions.
- Table functions
- Conversion functions.
- Control functions.
- Open space operations.

All of these functions are described in chapter 3, "Program Editing." For additional information, refer to the *Series 90-30/20/Micro Programmable Controllers Reference Manual*, GFK-0467.

### Note

**ALT-E** can be used to toggle the display of the status area at the bottom of the screen. Press **ALT-E** to remove the status information; then, press **ALT-E** again to display it again. For more information on the status area, refer to chapter 2, section 6, "Screen Format."

- The reverse-video block on the top line shows that the currently selected type of function is RELAY. Press F1 to start the rung with a normally open contact.

```

RELAY | TMRCTR | MATH | RELATN | BITOP | DATAMU | TABLES | CONVRT | CONTRL | OPN SP
1-] [- 2-] / [- 3- 4- 5-( )- 6-(SM)- 7-(RM)- 8vert | 9horz -10more
>
[ START OF LD PROGRAM LESSON ] (* *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]
???????
[ END OF PROGRAM LOGIC ]
OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: _MAIN SIZE: 123 RUNG 0004
REPLACE : ::

```

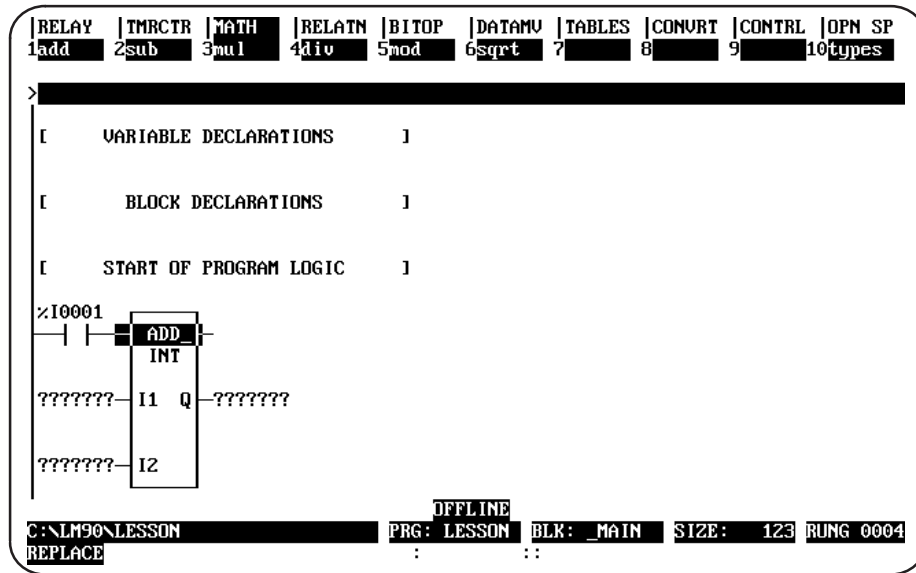
- Type in %I as a reference for the contact. Note that typing 1I has the same result. The reference appears in the command line at the top of the screen.
- Press the **Enter** key. The software automatically changes your entry to the correct format and moves the cursor to the next location in the rung.

```

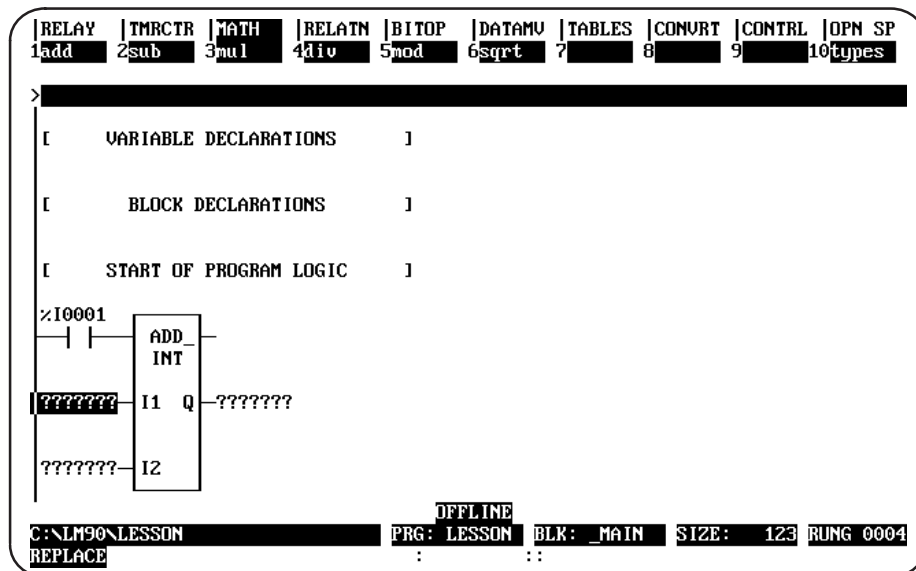
RELAY | TMRCTR | MATH | RELATN | BITOP | DATAMU | TABLES | CONVRT | CONTRL | OPN SP
1-] [- 2-] / [- 3- 4- 5-( )- 6-(SM)- 7-(RM)- 8vert | 9horz -10more
>
[ START OF LD PROGRAM LESSON ] (* *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]
%I0001
[ END OF PROGRAM LOGIC ]
OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: _MAIN SIZE: 123 RUNG 0004
REPLACE : ::

```

6. Add a math function to the rung. Press **Math (Shift-F3)** to select the math functions. At the top of the screen, the reverse video block moves to MATH. The second line shows the types of math functions that are currently available. Press **Add (F1)** to place an addition function in the rung.



7. Use the **Tab** key to move the cursor to the first input parameter (I1). This represents the first of the two numbers to be added.



8. Specify the first number to be added. For this lesson, enter **%R17** or **17R**. This represents register reference **%R0017**. Then, press the **Enter** key.

- Use the **Tab** key to move the cursor to the second input parameter (I2). This represents the number to be added to the current value of the reference %R0017.

The screenshot shows a software interface for editing a PLC program. At the top, there is a menu bar with options: RELAY, TMRCTR, MATH, RELATN, BITOP, DATAMV, TABLES, CONVRT, CONTRL, and OPN SP. Below the menu, there are several menu items: 1add, 2sub, 3mul, 4div, 5mod, 6sqrt, 7, 8, 9, and 10types. The main area displays a ladder logic diagram with an ADD\_INT instruction. The first input is labeled %I0001. The second input is labeled I1 Q-???????. The third input is labeled I2. The status bar at the bottom shows: C:\LM90\LESSON, PRG: LESSON, BLK: \_MAIN, SIZE: 123, RUNG 0004, and REPLACE. The word OFFLINE is also visible in the status bar.

- Enter the number 4 as the value to be added. Press the **Enter** key. Because you did not type a % symbol before the number, the software knows it is a constant.

This screenshot is identical to the previous one, but the value for the second input parameter I2 has been changed to a constant. The label for I2 is now CONST +00004. The rest of the interface, including the menu bar, menu items, and status bar, remains the same.



11. Use the **Tab** key to move the cursor to the next parameter (Q). Q is an output reference which represents the location for the sum (output) of the addition. Enter the reference **%R18**, and press the **Enter** key.

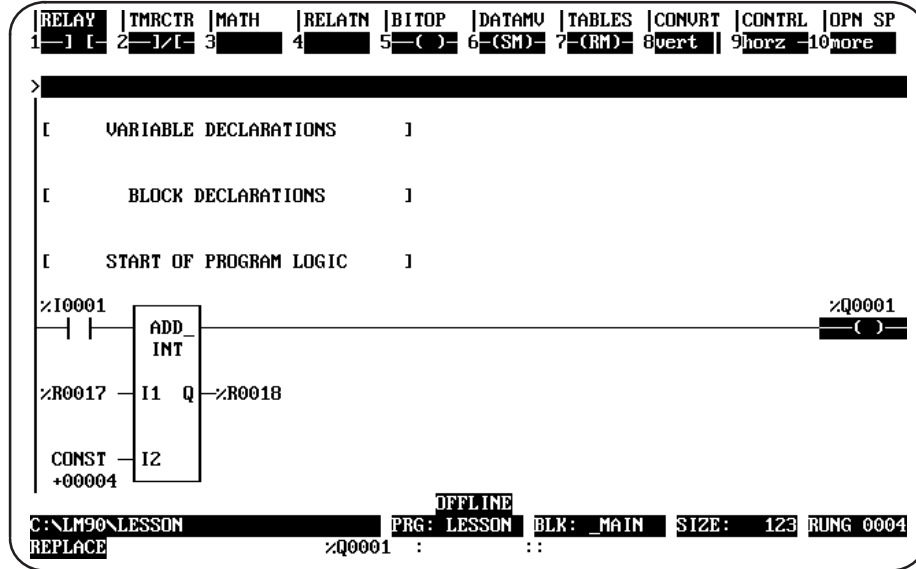
The screenshot shows a software interface for programming a PLC. At the top, there is a menu bar with options: RELAY, TMRCTR, MATH, RELATN, BITOP, DATAMU, TABLES, CONVRT, CONTRL, and OPN SP. Below the menu, there are several sections: VARIABLE DECLARATIONS, BLOCK DECLARATIONS, and START OF PROGRAM LOGIC. The main area displays a ladder logic rung. The rung starts with a normally open contact labeled %I0001. This is followed by an instruction box labeled 'ADD\_INT'. The instruction has two inputs: 'I1' and 'I2'. The input 'I1' is connected to a contact labeled %R0017. The input 'I2' is connected to a constant value '+00004'. The output of the instruction is labeled 'Q' and is currently empty. Below the instruction box, there is a status bar that reads 'OFFLINE' and 'PRG: LESSON'. At the bottom, there is a command line that says 'REPLACE %R0018 : ::'.

12. Use the **Tab** key to move the cursor block up to the top rung line. To complete the rung by programming a coil, press **Shift-F1** to select the relay functions.

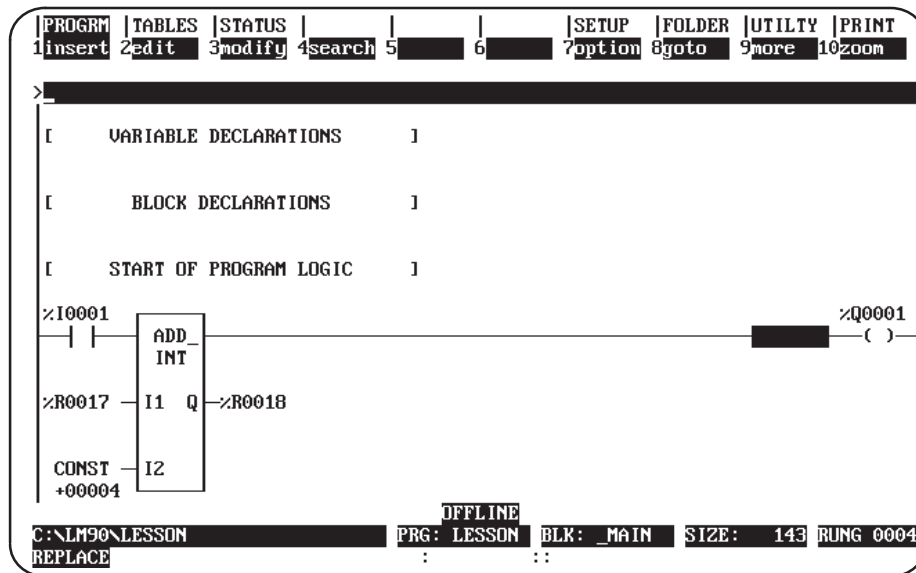
This screenshot is identical to the previous one, but the output 'Q' of the 'ADD\_INT' instruction is now filled with a coil symbol, representing a set coil. The status bar at the bottom still shows 'OFFLINE' and 'PRG: LESSON'. The command line at the bottom now shows 'REPLACE : ::'.

13. Enter **%Q1** into the command line. This will be the reference for the coil.

- Press **F5** to enter a coil at the end of the rung. Notice that the reference can be entered either before selecting the function as shown here, or after selecting the function as shown earlier.



- To accept the rung, press the **Enter** key (or the Plus (+) key on the numeric keypad). This accepts the completed rung, and the cursor moves downward so you can insert another line of logic.
- Select the control functions by pressing **Shift-F9**. Press **Comment (F8)** to add a comment to the program.



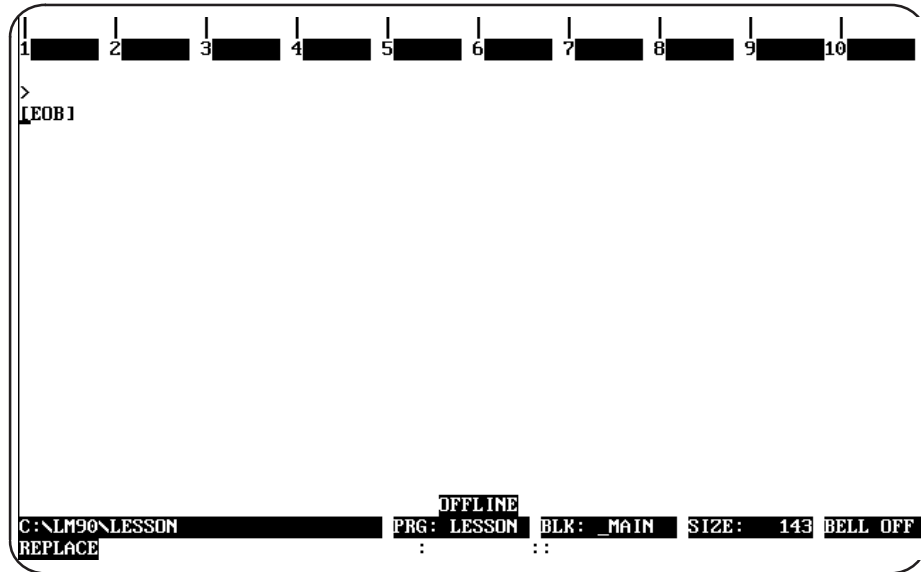
- 17. Press the **Enter** key and then the **Escape** key to return these key functions to the top of the screen as shown below:

```
PROGRAM | TABLES | STATUS | _____ | _____ | _____ | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3modify 4search 5 _____ 6 _____ 7option 8goto 9more 10zoom
>
[ START OF PROGRAM LOGIC ]
%I0001 ----- %Q0001
| | ADD INT ( )
| |
%R0017 -- I1 Q -- %R0018
CONST -- I2
+00004
(* COMMENT *)
[ END OF PROGRAM LOGIC ]
OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: MAIN SIZE: 143 RUNG 0006
REPLACE : ::
```

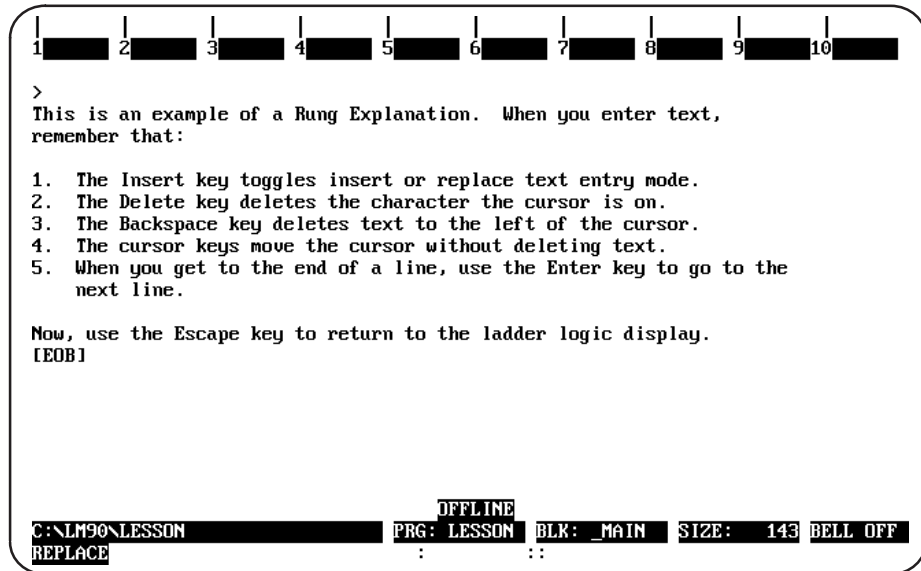
- 18. Press the **Up Cursor Movement** (or **Up Arrow**) key to move the cursor up to highlight the COMMENT instruction as shown below:

```
PROGRAM | TABLES | STATUS | _____ | _____ | _____ | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3modify 4search 5 _____ 6 _____ 7option 8goto 9more 10zoom
>
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]
%I0001 ----- %Q0001
| | ADD INT ( )
| |
%R0017 -- I1 Q -- %R0018
CONST -- I2
+00004
(* COMMENT *)
OFFLINE
C:\LM90\LESSON PRG: LESSON BLK: MAIN SIZE: 143 RUNG 0005
REPLACE : ::
```

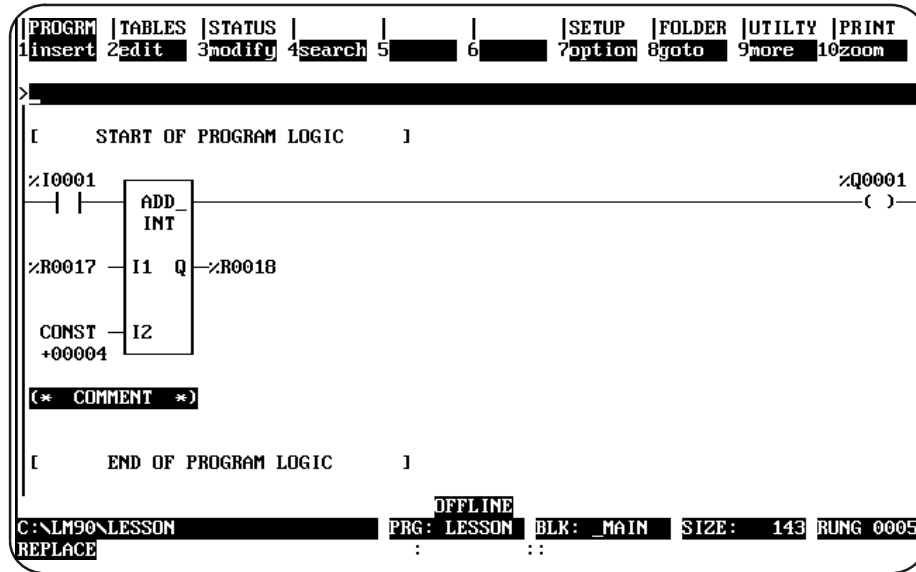
19. Press **Zoom (F10)**. The program logic is replaced with a blank screen where you can enter text.



20. Beginning at the **[EOB]** (End of Buffer) symbol, enter the text shown below:



21. Press the **Escape** key to save the comment and return the ladder logic display to the screen.



22. You could display the comment again by pressing **Zoom (F10)** with the cursor located at the comment rung.
23. That completes the lesson. Press the **Escape** key to return to the main menu. The following information pertains to subroutines. If you want to practice creating a subroutine, continue below. If you want to print this program now, continue on page A-23.

## Creating a Subroutine Block

Subroutines are declared through the block declaration editor.

### Note

Subroutine blocks are *not* available for the Series 90-20 PLC nor for Micro PLCs.

1. To create a subroutine declaration, position the cursor on the [ **BLOCK DECLARATIONS** ] marker.

```

PROGRAM | TABLES | STATUS | _____ | _____ | _____ | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3modify 4search 5_____ 6_____ 7option 8goto 9more 10zoom
> _____
[ START OF LD PROGRAM LESSON ] (* *)
[ VARIABLE DECLARATIONS ]
[ BLOCK DECLARATIONS ]
[ START OF PROGRAM LOGIC ]
[ END OF PROGRAM LOGIC ]

OFFLINE
C:\LESSON PRG: LESSON BLK: _MAIN SIZE: 123 RUNG 0002
REPLACE : ::

```

2. Then, press the **Zoom (F10)** key.

```

PROGRAM | TABLES | STATUS | _____ | _____ | _____ | SETUP | FOLDER | UTILITY | PRINT
1insert 2edit 3delete 4search 5lock 6_____ 7_____ 8goto 9_____ 10zoom
> _____
-[ START OF BLOCK DECLARATIONS ]
-[ END OF BLOCK DECLARATIONS ]

OFFLINE
C:\LESSON PRG: LESSON BLK: _MAIN ENTRY
REPLACE : ::

```

- Enter the name of the subroutine on the command line. For this lesson, type in the name `SHIP_IT`.

```

|PROGRAM |TABLES |STATUS |      |      |      |SETUP |FOLDER |UTILITY |PRINT
1insert 2edit 3delete 4search 5lock 6      7      8goto 9      10zoom
>SHIP_IT
-[ START OF BLOCK DECLARATIONS 1
-[ END OF BLOCK DECLARATIONS ]

OFFLINE
C:\LESSON PRG: LESSON BLK: _MAIN ENTRY
REPLACE : :

```

- Then, press the **Insert (F1)** key.

```

|      |      |      |      |      |      |      |      |      |      |
1      2      3      4      5      6      7      8      9      10
>
-[ START OF BLOCK DECLARATIONS 1
SUBR 1 SHIP_IT LANG: (*)
-[ END OF BLOCK DECLARATIONS 1

OFFLINE
C:\LESSON PRG: LESSON BLK: _MAIN ENTRY
REPLACE : SHIP_IT :

```

- An explanation of up to 32 characters can also be entered at this time. Type the description "THIS IS A BLOCK" on the command line.

```

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10
>"THIS IS A BLOCK"
-[ START OF BLOCK DECLARATIONS ]
    SUBR 1 SHIP_IT LANG:      (*)
-[ END OF BLOCK DECLARATIONS ]

OFFLINE
C:\LESSON PRG: LESSON BLK: MAIN ENTRY
REPLACE : SHIP_IT ::

```

- Press the **Enter** key.

```

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10
>
-[ START OF BLOCK DECLARATIONS ]
    SUBR 1 SHIP_IT LANG:      (* THIS IS A BLOCK *)
-[ END OF BLOCK DECLARATIONS ]

OFFLINE
C:\LESSON PRG: LESSON BLK: MAIN ENTRY
REPLACE : SHIP_IT :: THIS IS A BLOCK

```



- Then, press the **Escape** key.

```

|PROGRAM|TABLES|STATUS| | | |SETUP|FOLDER|UTILITY|PRINT
1insert 2edit 3delete 4search 5lock 6 7 8goto 9 10zoom
>
-[ START OF BLOCK DECLARATIONS ]
      SUBR 1 SHIP_IT LANG:      (* THIS IS A BLOCK *)
-[ END OF BLOCK DECLARATIONS ]

OFFLINE
C:\LESSON PRG: LESSON BLK: _MAIN ENTRY
REPLACE : ::

```

There are two alternate methods of entering this information. Both methods combine the previous step and this step into one operation. The first method is to type the name **SHIP\_IT** and the description **"THIS IS A BLOCK"** together on the command line. Press the **Insert (F1)** key, and then press the **Escape** key. The second method is to first press the **Insert (F1)** key, type the name and description together on the command line. Then, press the **Enter** key, followed by the **Escape** key.

- Use the cursor keys to position the cursor on the subroutine block declaration named **SHIP\_IT**. Then, press **Zoom (F10)**. Note that the name of the block (**BLK: \_MAIN**) in the status area of the previous screen has now been changed to the name of the subroutine block (**SUB: SHIP\_IT**).

```

|PROGRAM|TABLES|STATUS| | | |SETUP|FOLDER|UTILITY|PRINT
1insert 2edit 3modify 4search 5 6 7option 8goto 9more 10zoom
>
[ START LD SUBROUTINE SHIP_IT ]
[ VARIABLE DECLARATIONS ]
[ START OF SUBROUTINE LOGIC ]
[ END OF SUBROUTINE LOGIC ]

OFFLINE
C:\LESSON PRG: LESSON SUB: SHIP_IT SIZE: 123 RUNG 0003
REPLACE : ::

```

- You can create logic for the subroutine block on this screen. Position the cursor on the [END OF SUBROUTINE LOGIC] marker and press **Insert (F1)**. You can now insert ladder logic at the cursor location.

```

RELAY | TMRCTR | MATH | RELATN | BITOP | DATAMV | TABLES | CONVRT | CONTRL | OPN SP
1-1 | 2-1/I- | 3- | 4- | 5-( )- | 6-(SM)- | 7-(RM)- | 8vert | 9horz | 10more
>
[ START LD SUBROUTINE SHIP_IT ]
[ VARIABLE DECLARATIONS ]
[ START OF SUBROUTINE LOGIC ]
[ END OF SUBROUTINE LOGIC ]

OFFLINE
C:\LESSON PRG: LESSON SUB: SHIP_IT SIZE: 123 RUNG 0003
REPLACE : ::

```

- Press **F1** to start the rung with a normally open contact. Type in the reference **%I1**, and press the **Enter** key. The software automatically changes your entry to the correct format and moves the cursor to the next location in the rung.

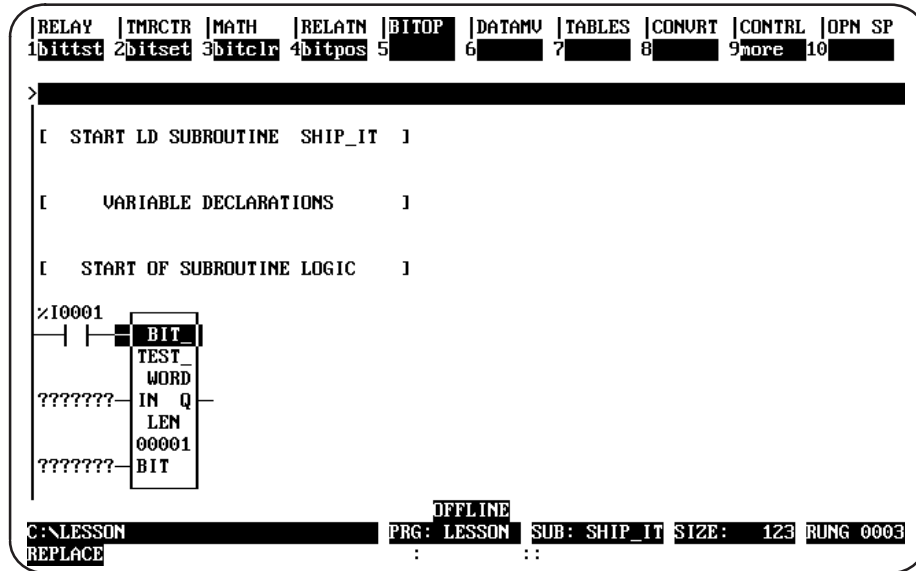
```

RELAY | TMRCTR | MATH | RELATN | BITOP | DATAMV | TABLES | CONVRT | CONTRL | OPN SP
1-1 | 2-1/I- | 3- | 4- | 5-( )- | 6-(SM)- | 7-(RM)- | 8vert | 9horz | 10more
>
[ START LD SUBROUTINE SHIP_IT ]
[ VARIABLE DECLARATIONS ]
[ START OF SUBROUTINE LOGIC ]
%I0001
| |
[ END OF SUBROUTINE LOGIC ]

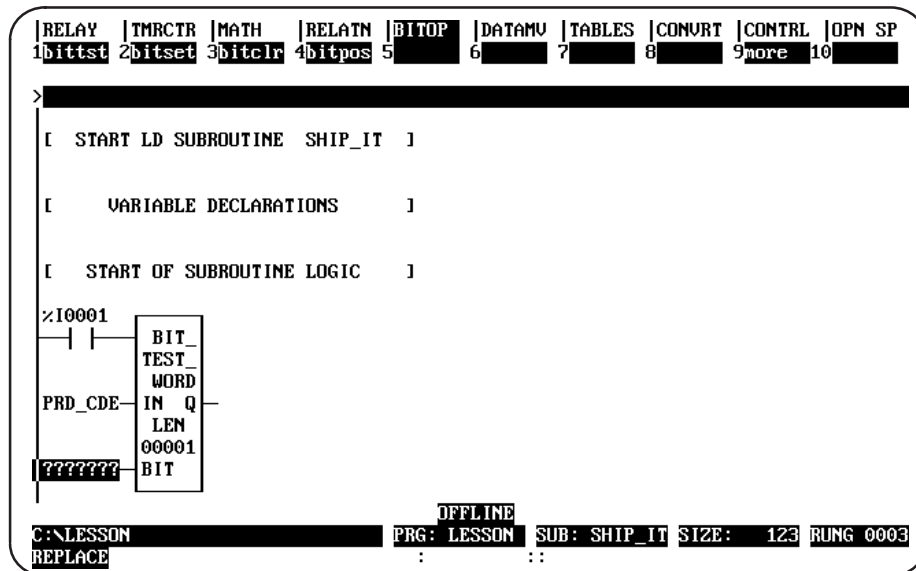
OFFLINE
C:\LESSON PRG: LESSON SUB: SHIP_IT SIZE: 123 RUNG 0003
REPLACE : ::

```

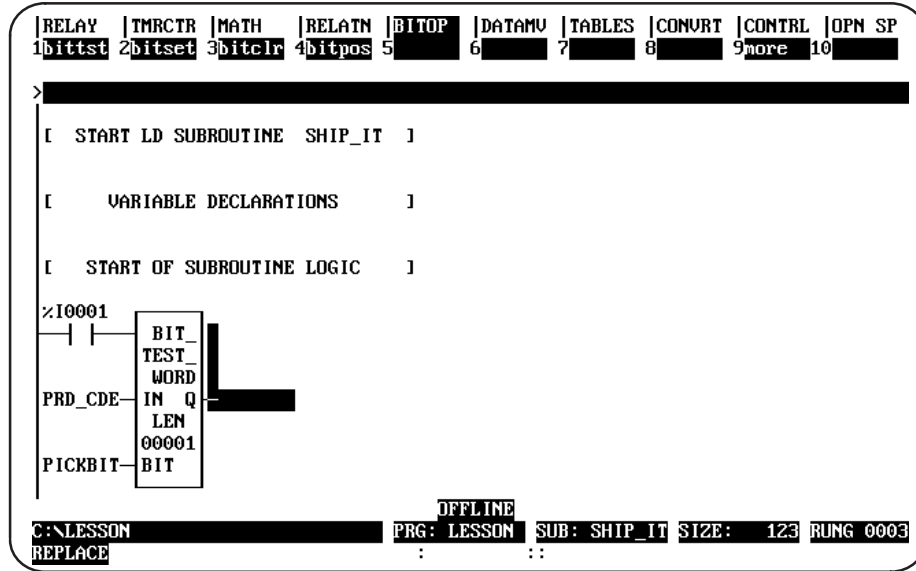
11. Add a bit operation function to the rung. Press **Bit Operation (Shift-F5)** to select the functions. At the top of the screen, the reverse video block moves to BITOP. The second line shows the types of bit operation functions that are currently available. Press **More (F9)** to display additional bit operation functions you can select. Then, press **Bit Test (F1)** to insert a function in the rung that will test a bit within a bit string.



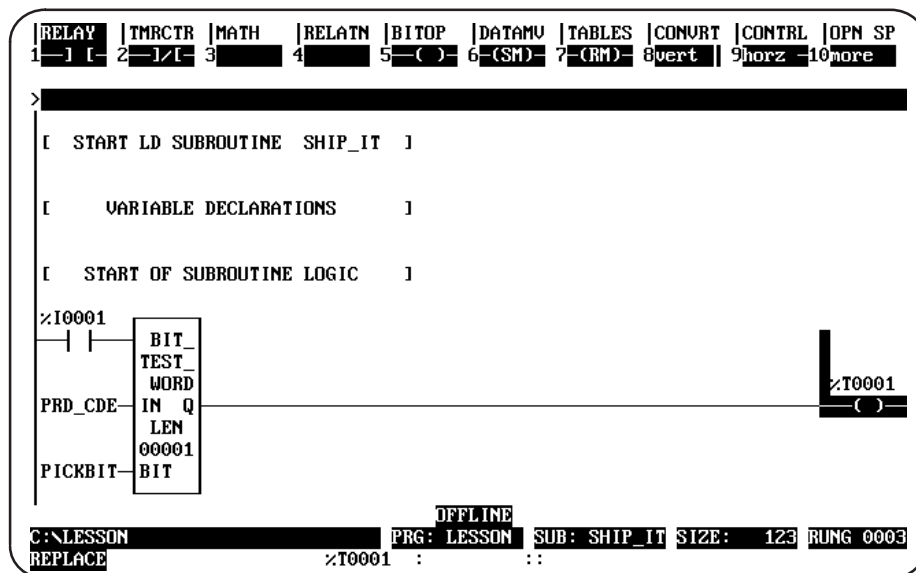
12. Use the **Tab** key to move the cursor to the parameter IN. IN contains the first word of the data to be operated on. For this lesson, enter **PRD\_CDE** on the command line. Because you have not previously entered the nickname PRD\_CDE in the variable declaration table, you will also need to enter a reference address for the nickname now. After typing **PRD\_CDE** on the command line, type **%R1** for the reference address. Then, press the **Tab** key. The software will enter the reference address in the logic and automatically move the cursor to the next parameter (BIT).



- BIT contains the bit number of IN that should be tested. For this lesson, enter **PICKBIT** and the reference address **%R2** on the command line. (If the nickname PICKBIT had previously been entered in the variable declaration table, you would not need to enter it here on the command line.) Then, press the **Tab** key. Again, the software will enter the reference address in the logic and automatically move the cursor to the next parameter (Q).



- Output Q is energized if the bit tested is a 1. Press **Relay (Shift-F1)** to display the relay functions. Then, type the reference **%T1** on the command line, and press **F5** to enter a coil at the end of the rung.

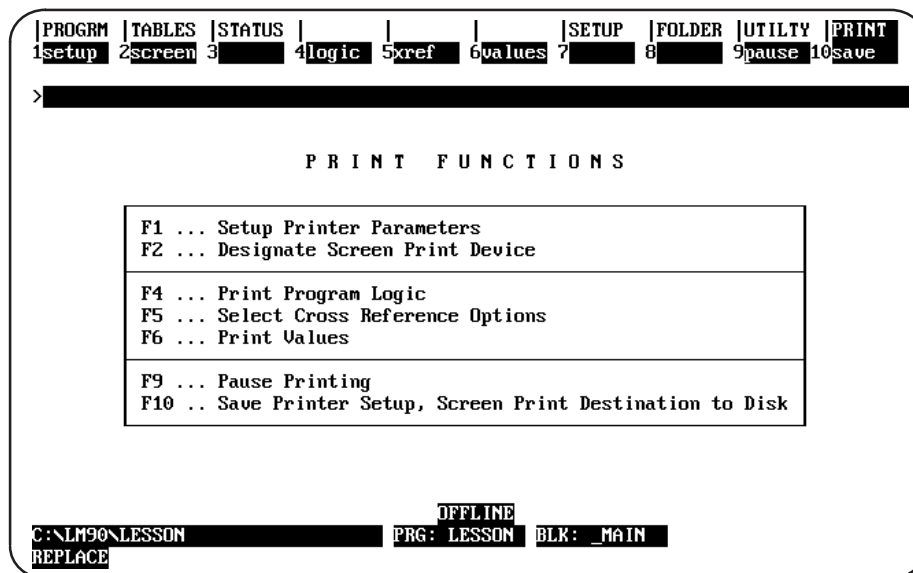


15. The LEN parameter represents the number of words in the string to be tested. This is set at 1 and does not need to be changed for this lesson. However, if you wanted to change the number of words to 5, for example, you would position the cursor on the function, enter 5 on the command line, and press the **Enter** key.
16. That completes the example subroutine. In this subroutine, whenever input %I0001 is set, the bit at the location contained in reference PICKBIT is tested. The bit is part of string PRD\_CDE. If it is 1, output Q passes power flow and coil %T0001 is turned on.
17. Press the **Escape** key a total of four times to return to the main menu. If you want to print this program now, continue below.

## Printing the Program

If you have a printer connected to the computer, and the Logicismaster 90-30/20/Micro software has been set up to communicate with it, you can print the lesson program now. The software assumes a default printer setup. If that is not correct for your printer, you can enter changes on the Printer Parameters screen, as explained in chapter 9. If the printer has not already been used to print Logicismaster 90 programs, you should check chapter 9, "Print Functions," before continuing.

1. To display the Print Function menu, press **Print (Shift-F10)**.



This menu is used to set printer parameters (F1 and F2), request printing (F4, F5, and F6), pause printing (F9), and save the printer setup (F10). Default printer setup values are supplied with the software, but are easily changed if needed. Chapter 9 explains printer setup in detail.

- Press **Logic (F4)** to print program logic. Define the printout content on the screen that appears.

```

|PROGRAM |TABLES |STATUS | | | |SETUP |FOLDER |UTILITY |PRINT
|1setup  |2screen  |3      |4logic  |5xref  |6values |7      |8      |9pause  |10save
-----
>
      PRINT PROGRAM LOGIC
-----
TITLE: ████████████████████████████████████████████████████████████████████████████
SUBTITLE:
-----
HEADER PAGE          Y (Y/N)  LOGIC              Y (Y/N)
VARIABLE TABLE     N          REFERENCE LIST    N
ALL BLOCKS          N          NICKNAME + REFERENCE N
IL LOGIC            N          REFERENCE DESCRIPTION N
                                RUNG COMMENTS     N
-----
FROM RUNG   0   TO RUNG  9999   STARTING PAGE NUMBER   1
-----
      << * Press ENTER Key to Start Printout * >>
-----
PORT: LPT1 (LPT1, COM1, LPT2, COM2, FILE)
FILE NAME:
-----
                                OFFLINE
C:\LM90\LESSON          PRG: LESSON  BLK: _MAIN
REPLACE
  
```

- Enter a title for the printout, such as: **PRACTICE PROGRAM**.
- Both the title and subtitle entries are optional. Skip the subtitle entry.
- The illustration above shows the default selections for the printout options. To include the example rung explanation in the printout, move the cursor to: "RUNG COMMENTS N". Use the Tab key to toggle the selection from **N** (No) to **Y** (Yes).
- The other entries shown are suitable for this lesson, so they can be skipped.
- The listing destination (lower part of the screen) allows you to direct the printout to a serial or parallel port, or to a file. If the destination shown is not correct, use the Down cursor key to move the cursor to **PORT**. Enter the correct port designation.
- When you are ready to print the program, press the **Enter** key. While the program is being printed, the software displays the message:

Listing in progress ... (Listing CAN be paused or aborted).

## Ending the Lesson

The program you created was saved when you exited the display/edit program function.

# Appendix B

## Configuration Lesson for the Series 90-30 PLC

---

---

This appendix contains a tutorial for the configuration software package. The lesson includes:

- Creating a program folder.
- Configuring a CPU module.
- Configuring a 90-30 I/O module.
- Configuring a Genius Communications Module.

You can exit and save the lesson at any time, as explained below. It is not necessary to complete all of the steps.

- To exit and save the lesson, press the **Escape** key.
- To exit without saving the lesson, press **CTRL-Break**.

### Help Screens

Logicmaster 90-30/20/Micro software includes detailed Help screens. These Help screens are loaded onto the hard disk of your programmer during the software installation procedure and are readily accessible. To access the Help screens, press **ALT-H** for help, **ALT-I** for instruction mnemonic help, or **ALT-K** for key help.

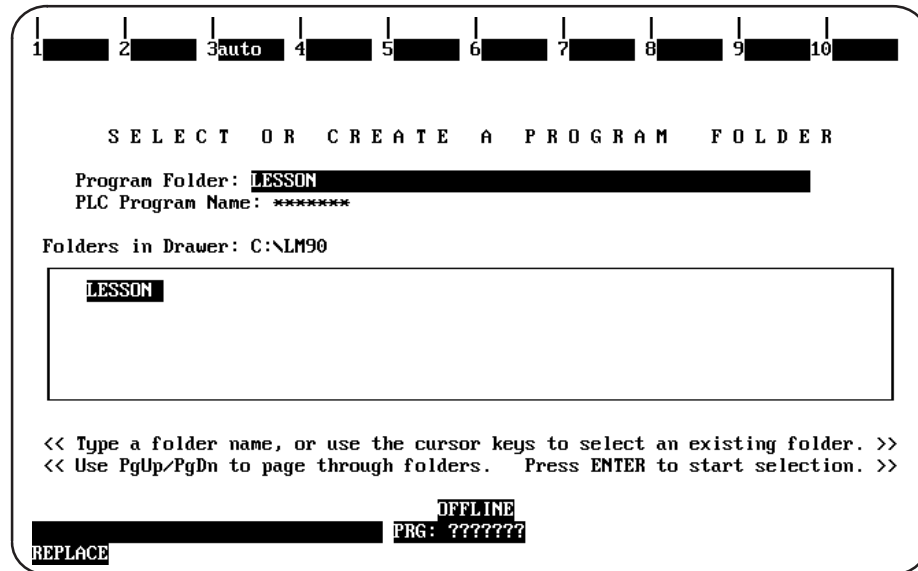
### Starting the Lesson

Before you can start, the configuration software must be installed and started up. If that has not been done yet, please turn back to chapter 2, "Operation," for instructions.

## Creating a Program Folder

Each program and its associated configuration are stored as many files. These files are stored in a subdirectory called a program folder. For this lesson, you will use a program folder named **LESSON**.

When Logicmaster 90-30/20/Micro software is started (if the current default directory is an existing program folder), a screen similar to the one shown below appears. If the software is already running, you can display a similar screen from the Program Folder menu by pressing **Folder (F8 or Shift-F8)** from the Configuration Software main menu and then **Select (F1)**.

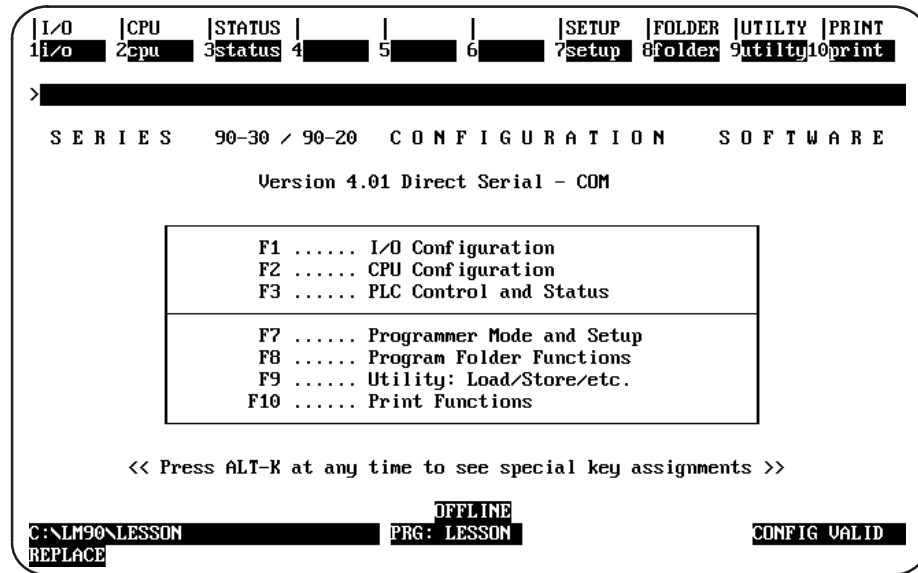




1. If the current default directory is not an existing program folder, type in a name of seven characters or less for the folder. For this lesson, type **LESSON**.
2. Press the **Enter** key. The following prompt will appear at the top of the screen:

Program folder does not exist; create new folder? (Y/N)

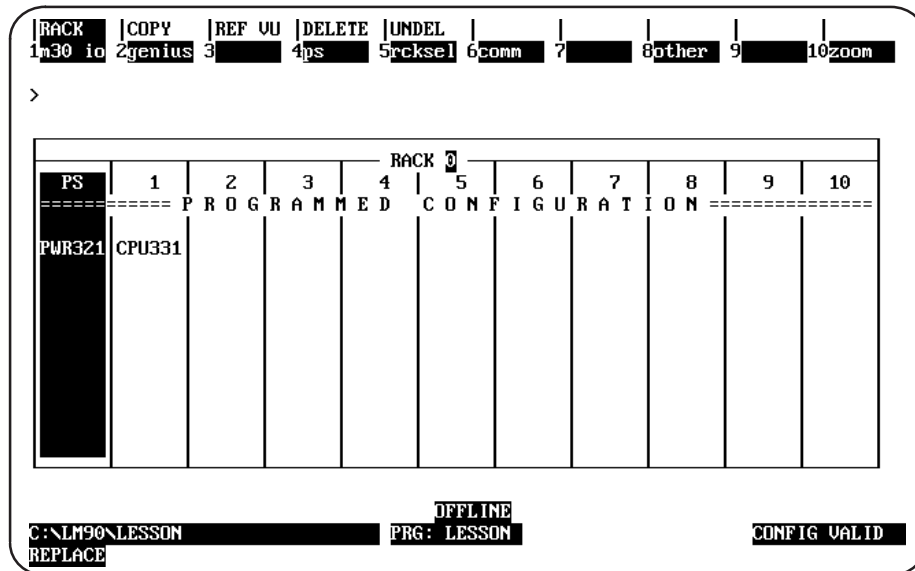
3. To create the program folder, enter **Y** (Yes). The software will accept the name and display the Configuration Software main menu.



The name of your new program folder (**LESSON**) should be displayed at the bottom of the screen.

## Displaying the Rack Configuration

From the main menu, press I/O (F1 or Shift-F1 from the Folder screen) to display the beginning screen.

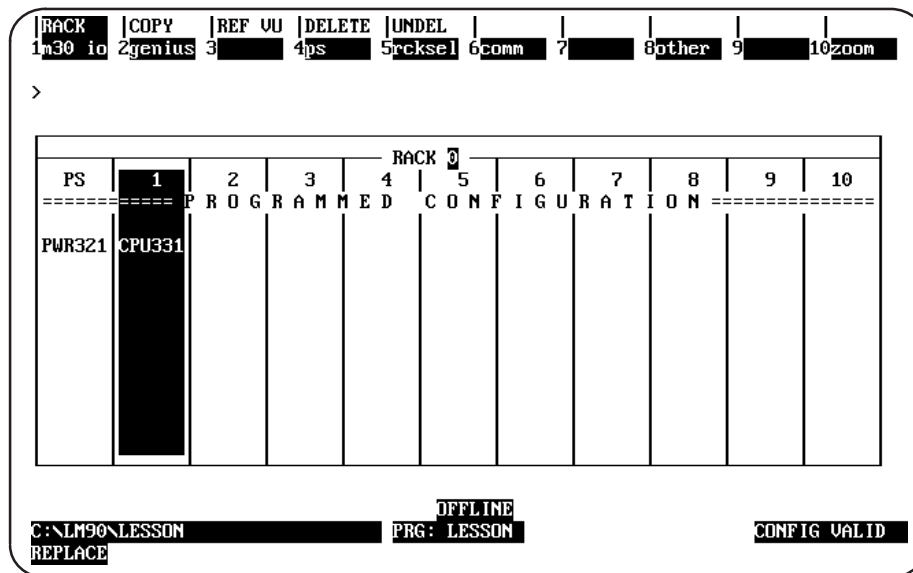


This screen shows the slots available in rack 0, which is the first rack in the system. Use the Page Up or Page Down key, or the Up or Down cursor key, to display the configuration of another rack.

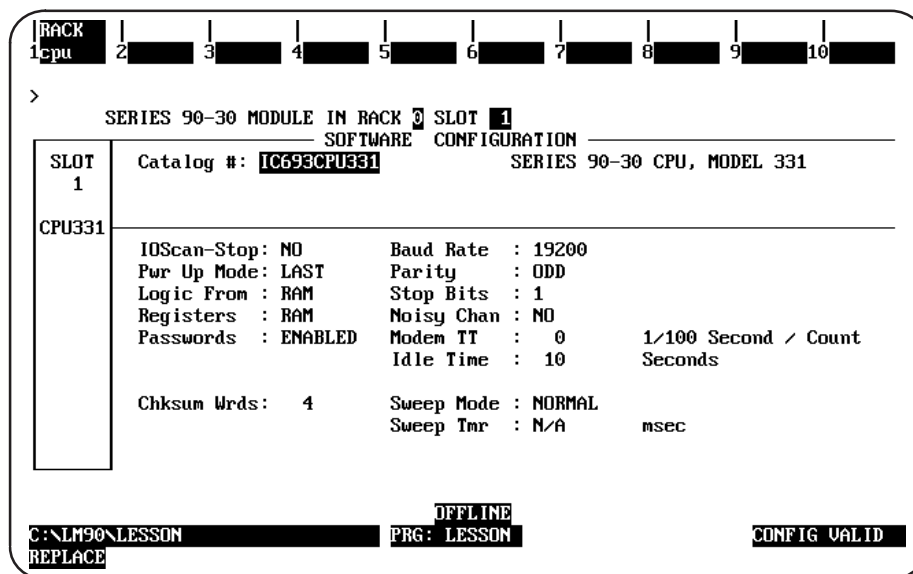
## Configuring a CPU Module

The left slot contains the power supply. Although it requires no additional configuration, its configuration may be changed. Slot 1 contains the CPU module, for which you will enter an example configuration.

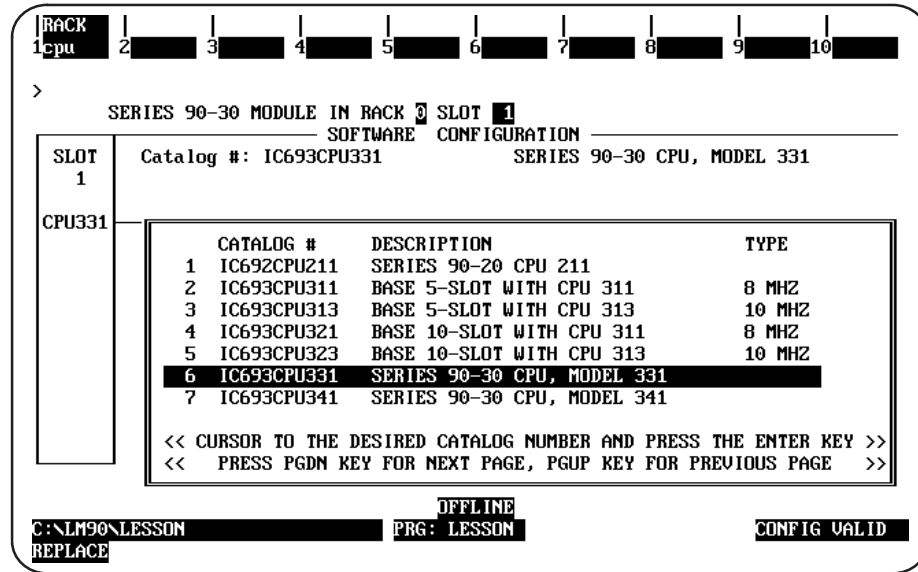
1. Press the **Right** cursor movement key once to highlight slot 1.



2. Press **Zoom (F10)** to display this detail screen.



- 3. When each of the parameters on the screen has been entered, press **CPU (F1)** to display the list of available CPU modules and their catalog numbers.



- 4. If you are configuring another type of CPU module, move the cursor to the desired module and press the **Enter** key. Then, enter **Y** (Yes) after the prompt:

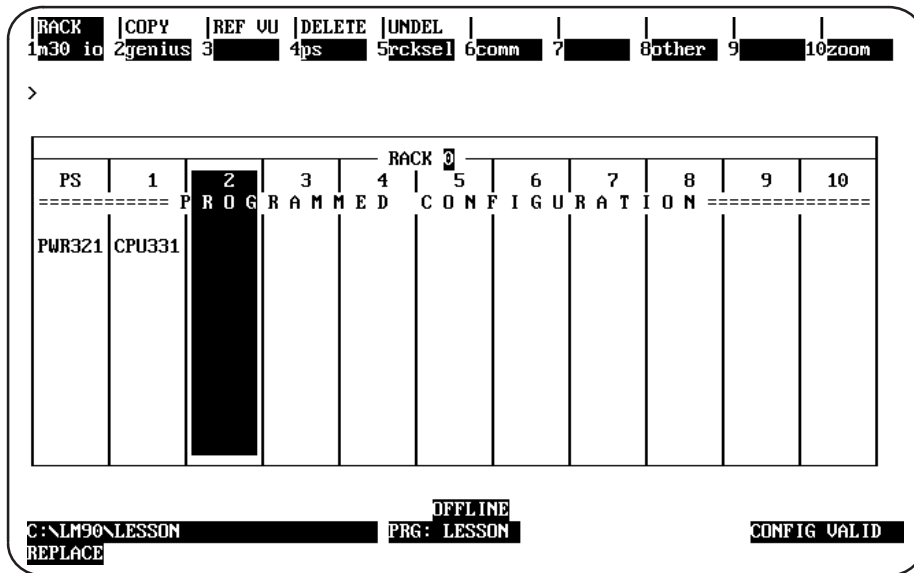
REPLACE displayed module? (Y/N)

- 5. Press **Rack (Shift-F1)** or the **Escape** key to exit from the detail screen and return to the rack display.

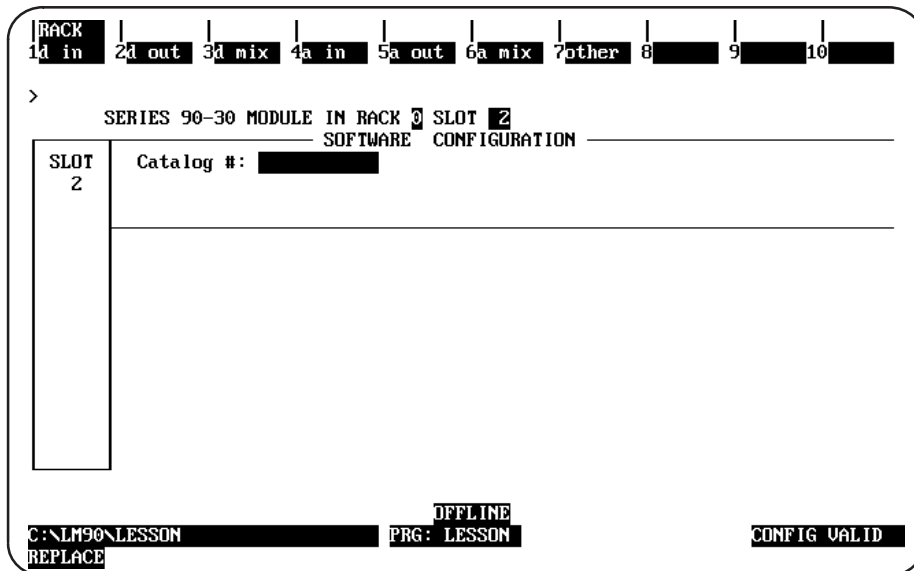
## Configuring a 90-30 I/O Module

Next, you will configure slot 2 of the main rack for a 90-30 I/O module.

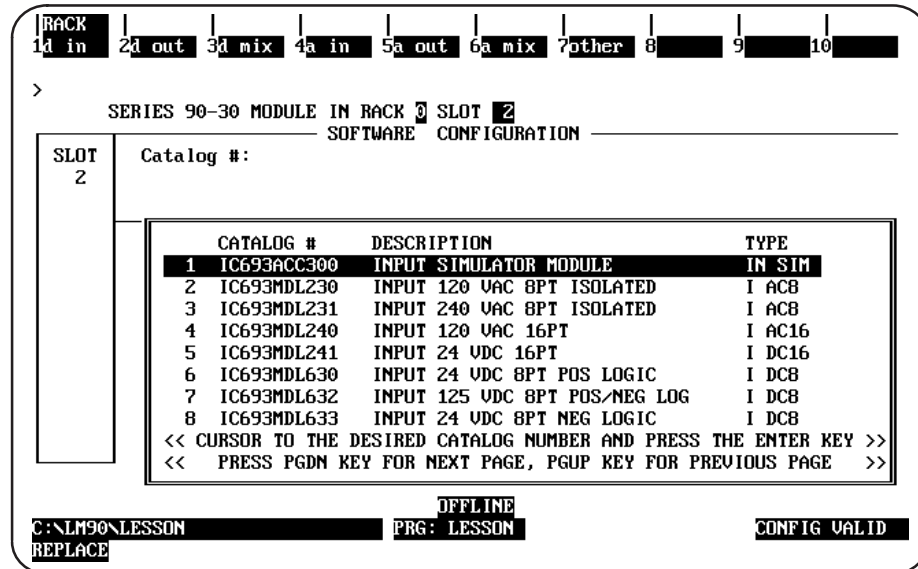
1. Use the Right cursor key to highlight slot 2.



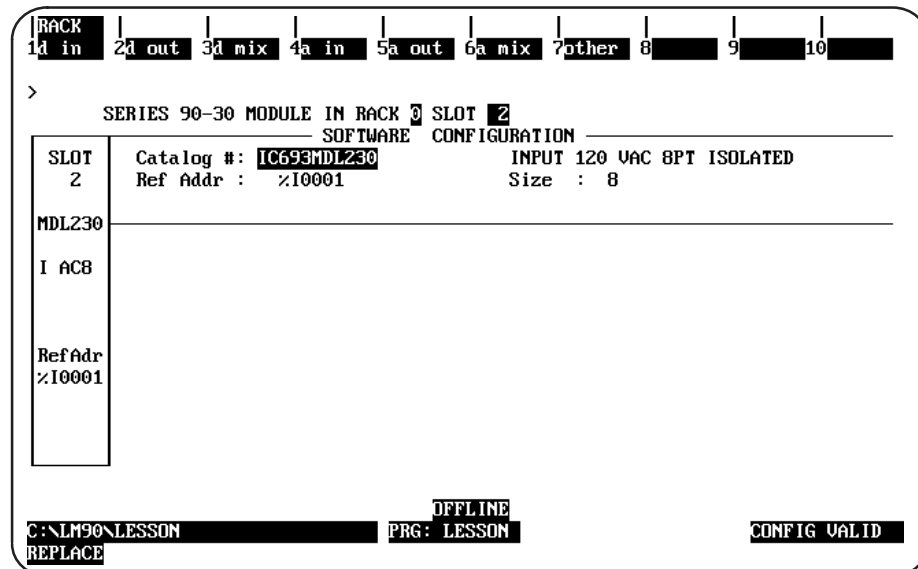
2. Press **Module 30 I/O (F1)** to configure a 90-30 I/O Module in slot 2. The software zooms into the detail screen.



- The types of modules that might be placed in this slot are shown by the key functions at the top of the screen. For this example, select a Discrete Input module by pressing **F1**. A list of modules is displayed.



- Select a module by moving the cursor to any entry and pressing the **Enter** key. Its catalog number will appear in reverse video in the proper field. For this example, move the cursor to **IC693MDL230** and press the **Enter** key. The following screen is displayed:



5. Assign a beginning I/O reference address to this module. Move the cursor to:

**Ref Addr : %I00001**

For this example, you will purposely enter an incorrect I/O reference. First, move the small cursor to the right in the *Reference Address* field by pressing the **CTRL** and Right cursor keys. Enter **1** and then **6** as the last two digits. Now the reference address should look like this:

**Ref Addr : %I00016**

This reference address is incorrect because it does not begin on a byte boundary. (A byte boundary is a number which is one greater than an integer multiple of 8; for example, 1, 9, 17, or 25).

6. Watch the reference address. Notice that when you press the Down cursor key once, the software automatically changes the reference address to the next lowest byte boundary:

**Ref Addr : %I00009**

and the message "The Reference Address has been adjusted to be byte aligned" will be displayed.

7. For this lesson, the rest of the entries on this screen need not be changed. Press the **Escape** key to return to the rack display.

## Configuring a Genius Communications Module

Next, you will configure slot 3 of the main rack for a Genius Communications Module.

1. Use the Right cursor key to highlight slot 3.

RACK	COPY	REF UU	DELETE	UNDEL	comm	other	zoom
1io	2genius	3	4ps	5rcksel	6	7	10

PS	1	2	3	4	5	6	7	8	9	10
PROGRAMMED CONFIGURATION										
PWR321	CPU331	MDL230								
		I ACB								
		RefAdr								
		%I0001								

OFFLINE  
C:\LM90\LESSON REPLACE PRG: LESSON CONFIG VALID

2. Press **Genius (F2)** from the I/O Rack Configuration screen. Then, press **GCM (F2)** to display the catalog number of the GCM.

SERIES 90-30 MODULE IN RACK 0 SLOT 3  
SOFTWARE CONFIGURATION

SLOT 3 Catalog #:

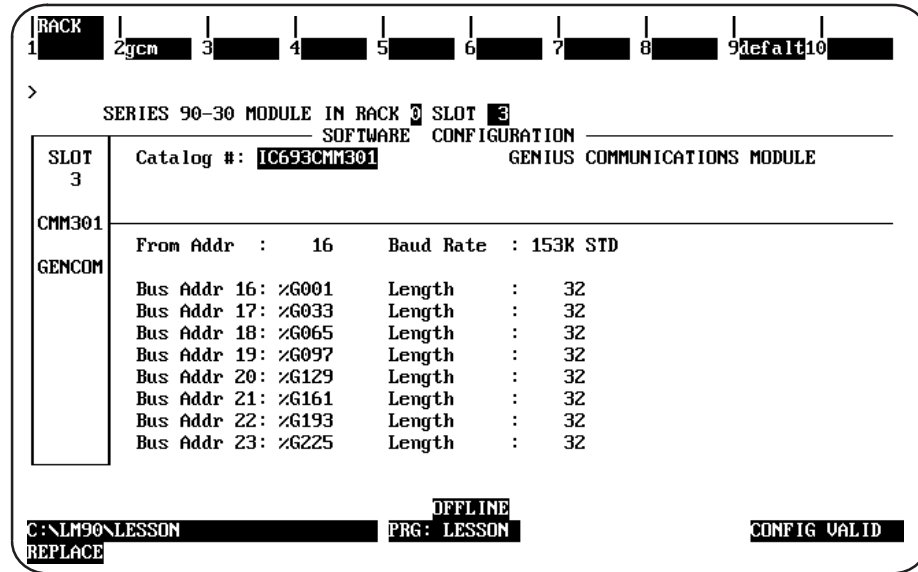
CATALOG #	DESCRIPTION	TYPE
1	IC693CMM301 GENIUS COMMUNICATIONS MODULE	GENCOM
2	IC693CMM302 ENHANCED GENIUS COMM MODULE	GENCOM

<< CURSOR TO THE DESIRED CATALOG NUMBER AND PRESS THE ENTER KEY >>  
<< PRESS PGDN KEY FOR NEXT PAGE, PGUP KEY FOR PREVIOUS PAGE >>

OFFLINE  
C:\LM90\LESSON REPLACE PRG: LESSON CONFIG VALID



3. Press the **Enter** key to enter the catalog number shown in reverse video. The following detail screen is displayed.



4. Complete the detail screen. To do this, you must be familiar with information in the *Genius Communications Module Manual*, GFK-0412. For this example, we will assume the default entries are correct.
5. Press **Rack (Shift-F1)** or the **Escape** key to return to the rack display.
6. Press the **Escape** key from the rack display (or **CTRL-U**) to save the created configuration.

## Ending the Lesson

The configuration created was saved when you pressed **Rack (Shift-F1)** or the **Escape** key to return to the rack display. This is the end of the lesson.

# Appendix C

## Programmer Environment Setup

A default setup file is created during installation. Only in special circumstances will this file need to be changed. Follow the steps below in order to edit the setup file.

### Displaying the Setup Screen

1. After completing the installation and restarting the computer as described, at the MS-DOS prompt, type **LM90** and press the **Enter** key. A menu of Series 90 PLCs and functions is displayed.

```
MICRO 90-20 90-30
1Program 2Config 3 PCM 4 APM 5 OI 6 7 Util 8Comenu 9Setup 10 Exit

LOGICMASTER 90 SOFTWARE
FOR SERIES 90 (c) PROGRAMMABLE CONTROLLERS

Shift-F1 ... Series 90 Micro Programmable Controller
Shift-F2 ... Series 90-20 Programmable Controller
Shift-F3 ... Series 90-30 Programmable Controller

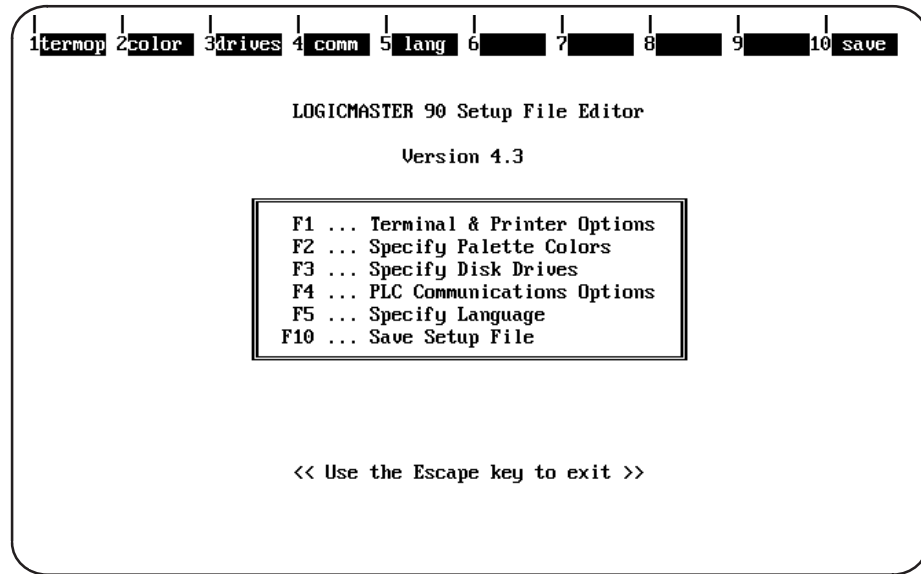
F1 ... Logicmaster 90 Programmer Package
F2 ... Logicmaster 90 Configuration Package
F3 ... PCM Development Package (PCOP)
F4 ... Axis Positioning Module Package
F5 ... Operator Interface Utilities
F7 ... Logicmaster 90 Utilities
F8 ... User Command Menu
F9 ... Logicmaster 90 Setup Package
F10 ... Exit to DOS

Use the Shift-function keys to select PLC type.
Use the function keys to start software package.
```

### Note

Unlike the setup features discussed in chapter 6, you must begin the setup steps detailed in this appendix by entering the “Logicmaster 90 Setup Package” by pressing **F9** from the Logicmaster main menu, not from within the programmer.

2. Press **Setup (F9)** to select the setup function. The following menu is displayed:



## Selecting Terminal and Printer Options

To select terminal and printer options, press **Terminal Options (F1)** from the Setup File Editor menu.

**Terminal & Printer Options**

Country code: 1

Programmer Mode Keyswitch:	<b>DISABLED</b>	(ENABLED,DISABLED)
CGA Video Snow Suppression:	ON	(ON,OFF)
IBM-compatible Graphics Printer:	NO	(NO,YES)

<< Use Tab/Shift-Tab keys to adjust field contents.>>  
 << Use the ←↑↓ arrow keys to select a field.>>

Use the cursor keys to move from one option field to another. Selected fields are shown in reverse video on the display screen. Then, use the **Tab** and **Shift-Tab** keys to toggle each selection.

Field	Description
Programmer Mode Keyswitch	Logicmaster 90-30/20/Micro software may be run on machines with or without a keyswitch to specify the operational mode ( <b>ONLINE</b> , <b>MONITOR</b> , or <b>OFFLINE</b> ). Choices for this field are <b>ENABLED</b> or <b>DISABLED*</b> .  This switch should normally be left at the factory default <b>DISABLED</b> position. It can be used to <b>ENABLE</b> the <b>OFF-LINE/MONITOR/ONLINE</b> keyswitch on the Workmaster I or original CIMSTAR I industrial computer. Enabling this value will disable the use of the <b>Alt-M</b> key to change Modes and will cause an error message on all current personal computers..
CGA Video Snow Suppression	This switch is used to prevent screen flickering on early 1980's CGA (Color Graphic Adapter) video cards. The default <b>ON*</b> is for the original CGA cards while <b>OFF</b> can be used for later cards. This switch is not used for later EGA or VGA video cards.
IBM Compatible Graphics Printer	If your printer is capable of producing IBM compatible graphics characters, you may improve the appearance of your ladder logic printouts by enabling this table has no title this option. If you printout displays contacts or lines as capital letters, your printer does not support the IBM graphic characters. You must set this switch to <b>NO</b> . Choices for this field are <b>YES</b> or <b>NO*</b> .

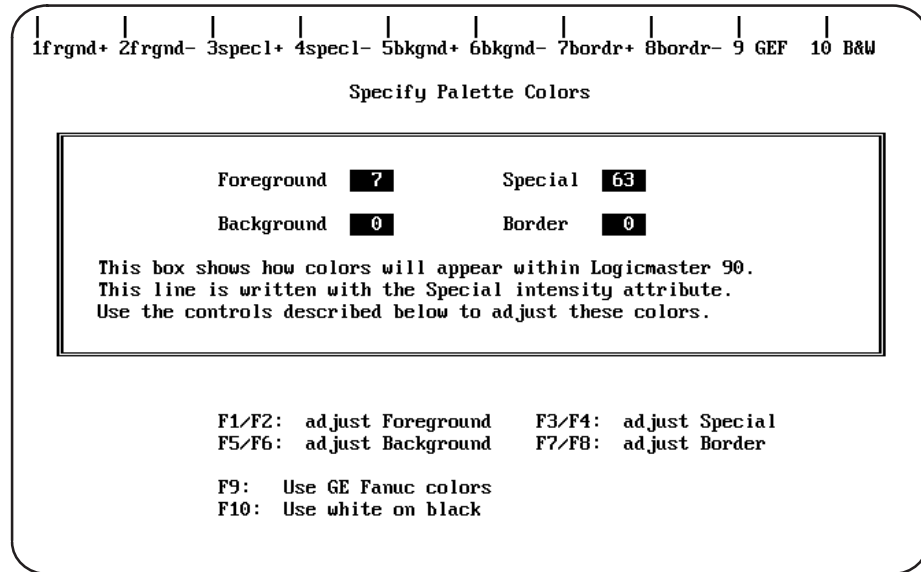
\* Default Selection

When the selections are complete, press the **Escape** key to return to the Setup File Editor menu.

## Specifying Palette Colors

If your personal computer has a color monitor and the video adapter is an EGA (Enhanced Graphics Adapter) or VGA (Video Graphics Array), you may select the colors that will be used in the Logicmaster 90 displays.

To specify palette colors, press **Colors (F2)** from the Setup File Editor menu.



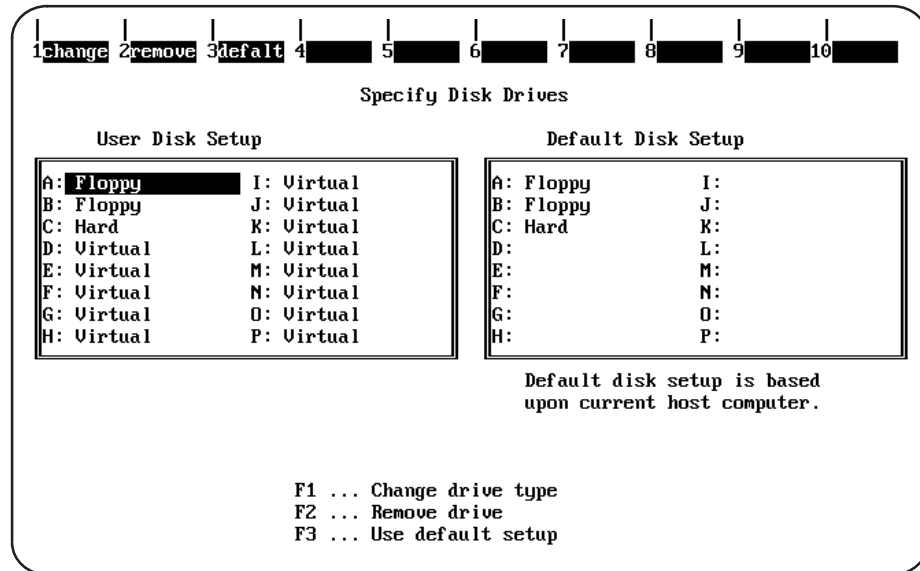
The middle portion of the screen, enclosed in a box, provides a sample display using the currently selected colors. The numeric values stored in the palette registers control the foreground, background, special, and screen border colors.

The color values in the palette registers may be changed by pressing the function keys (F1 through F10), as described in the lower portion of the screen. The function keys are paired; one key increases the value while the other decreases it. The values may be adjusted within the range 0 to 63. As the values are changed, the colors in the sample display box are updated accordingly.

You may wish to experiment with different color combinations. When you are satisfied with your selections, press the **Escape** key to return to the Setup menu. The color values will be saved along with the other setup information; these color values will be used each time you start up the Logicmaster 90-30/20/Micro software.

## Completing the Disk Drive Setup

To complete the disk drive setup, press **Drives (F3)** from the Setup File Editor menu.



1. If the drives shown on the left of your screen are correct, press **Default (F3)**. If the current selections are not correct, change them as instructed.
2. Press the **Escape** key to return to the Setup File Editor menu.

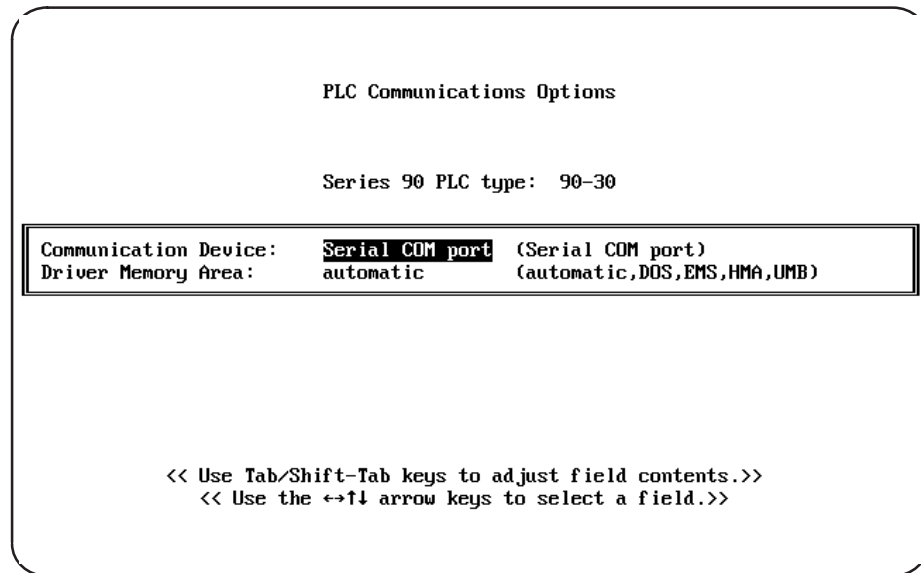
## PLC Communications Options

The PLC Communications Options screen enables you to select whether to communicate with the PLC through a WSI Board or serial COM port and to select one of four different interrupt lines on the PC backplane. To display this screen, press **PLC Communications Options (F4)** from the Setup File Editor menu.

Remember, unlike the serial port setup discussed in chapter 6, to access this screen you must enter **Setup (F9)** from the main menu which takes you to the Setup File Editor screen, then press **PLC Communications Options (F4)**—see pages C-1–C-2 for more information.

### Standard Serial COM Port Installed

In the following example screen, only the standard serial communications version of software is installed.



Field	Description
Communication Device	When both the WSI and standard serial communications versions of Logimaster 90-30/20/Micro software are installed, you must choose whether to communicate with the PLC through a <b>WSI Board*</b> or <b>serial COM port</b> .
Driver Memory Area	This field shows the current selection for which area of memory the communications driver will load into. The default and recommended setting is <b>“automatic”</b> as shown above.

\* Default Selection

### Standard Serial COM Port and Driver Memory Area

To change the serial COM memory area, use the **Down Arrow** key to move the cursor to the “Driver Memory Area” field. (Selected fields are shown in reverse video on the display screen.) Then, use the **Tab** and **Shift-Tab** keys to toggle each selection. As noted above, the “automatic” setting is recommended; this lets Logimaster determine which area of memory it is best to place the driver. For information about the different memory configuration options, read pages 6-6 and following in this manual.

## PLC Communications Options— Both WSI and Serial COM Port Installed

In the following example, both the WSI and Standard Serial Communications versions of Logicmaster 90-30/20/Micro software are installed, and the serial COM port was chosen as the communication driver.

PLC Communications Options

Series 90 PLC type: 90-30

Communication Device:	Serial COM port	(WSI card, Serial COM port)
Driver Memory Area:	automatic	(automatic, DOS, EMS, HMA, UMB)

<< Use Tab/Shift-Tab keys to adjust field contents.>>  
<< Use the ←→↑↓ arrow keys to select a field.>>

### Note

To change the selection of “Communication Device,” use the **Tab** and **Shift-Tab** keys to toggle the selection between “Serial COM port” and “WSI.” This, of course, is only applicable *if* you have both the WSI and the Standard Serial COM Port installed.



## PLC Communications— WSI Version with Configurable Interrupt Request

In the next example, the WSI version of software is installed and selected, and a WSIB2 Board was used. Note that the default value **IRQ3** appears in the “WSIB Interrupt Line” field. This value can be changed by using the **Tab** and **Shift-Tab** keys to toggle each selection.

**PLC Communications Options**

Series 90 PLC type: 90-30

Communication Device:	<b>WSI card</b>	(WSI card, Serial COM port)
WSIB Interrupt Line:	<b>IRQ3</b>	(IRQ7, IRQ3, IRQ4, IRQ5 )

<< Use Tab/Shift-Tab keys to adjust field contents.>>

Field	Description
Communication Device	When both the WSI and standard serial communications versions of Logicmaster 90-30/20/Micro software are installed, you must choose whether to communicate with the PLC through a <b>WSI Board*</b> or <b>serial COM port</b> .
WSIB Interrupt Line	In a PC or PC-AT computer, the WSI Board can be configured to use four different interrupt lines on the PC backplane. Choices are <b>IRQ3</b> , <b>IRQ4</b> , <b>IRQ5</b> , or <b>IRQ7*</b> . This option is available only if you are using the WSIB2 Board for communications. If you have the older WSIB1 Board, the interrupt line is displayed as IRQ3 and is not changeable. If you have selected a serial COM port for communications, this option is not displayed.

\* Default Selection

### Note

When the WSI Version of Release 4 or later of Logicmaster 90-30/20/Micro software is used on a Workmaster II industrial computer or other microchannel personal computer with a WS9A2 board, the selection of an interrupt line on the PLC Communications Options screen must be set to IRQ3. Failure to select IRQ3 will result in improper setup of the board, possible lockup of the Logicmaster software during startup, and failure to communicate with the PLC.

## PLC Communications— WSI Version with Non-Configurable Interrupt Request

In this example, again the WSI version of software is installed, but a non-configurable WSIB1 Board was used. The interrupt line IRQ3 is displayed in the “WSIB Interrupt Line” field and cannot be changed. Again, there are no options to select; the screen is presented only to provide information.

PLC Communications Options

Series 90 PLC type: 90-30

Communication Device:	WSI card	(WSI card, Serial COM port)
WSIB Interrupt Line:	IRQ3	

<< Use Tab/Shift-Tab keys to adjust field contents.>>

## Saving the Programmer Environment Setup

To save the setup information in a file for future use, first press **Esc** (the Escape key) to return to the “LOGICMASTER 90 Setup File Editor” screen. Then press **Save (F10)** from the Setup File Editor menu. After the file is saved, the software displays the message:

**Setup File saved successfully as c:\lm90\lm90.prd.**

Press any key to return to the Setup File Editor menu. Then, press the **Escape** key to return to the menu of Series 90 PLCs and functions.

# Appendix D

## Instruction Mnemonics

In program display/edit mode, you can quickly enter or search for a programming instruction by typing the ampersand (&) character followed by the instruction's mnemonic. For some instructions, you can also specify a reference address or nickname, a label, or a location reference address.

This appendix lists the mnemonics of the programming instructions for Logicmaster 90-30/20/Micro programming software. At any time during programming, you can display a help screen with these mnemonics by pressing the **ALT** and **I** keys (i.e., hold down the **ALT** key and then tap the **I** key).

Function Group	Instruction	Mnemonic						
		All	INT	DINT	BIT	BYTE	WORD	
Contacts	Any Contact Normally Open Contact Normally Closed Contact Continuation Contact	&CON &NOCON &NCCON &CONC	&CON &NOCON &NCCON &CONC					
Coils	Any Coil Normally Open Coil Negated Coil Positive Transition Coil Negative Transition Coil SET Coil RESET Coil Retentive SET Coil Retentive RESET Coil Retentive Coil Negated Retentive Coil Continuation Coil	&COI &NOCOI &NCCOI &PCOI &NCOI &SL &RL &SM &RM &NOM &NCM &COILC	&COI &NOCOI &NCCOI &PCOI &NCOI &SL &RL &SM &RM &NOM &NCM &COILC					
Links	Horizontal Link Vertical Link	&HO &VE	&HO &VE					
Timers	On Delay Timer Elapsed Timer Off Delay Timer	&ON &TM &OF	&ON &TM &OF					
Counters	Up Counter Down Counter	&UP &DN	&UP &DN					

Function Group	Instruction	Mnemonic							
		All	BCD-4	INT	DINT	BIT	BYTE	WORD	REAL
Math	Addition Subtraction Multiplication Division Modulo Square Root Sine Cosine Tangent Inverse Sine Inverse Cosine Inverse Tangent Base 10 Logarithm Natural Logarithm Power of <i>e</i> Power of <i>x</i>	&AD &SUB &MUL &DIV &MOD &SQ &SIN &COS &TAN &ASIN &ACOS &ATAN &LOG &LN &EXP &EXPT		&AD_I &SUB_I &MUL_I &DIV_I &MOD_I &SQ_I	&AD_DI &SUB_DI &MUL_DI &DIV_DI &MOD_DI &SQ_DI				&AD_R &SUB_R &MUL_R &DIV_R &MOD_R &SQ_R
Relational	Equal Not Equal Greater Than Greater or Equal Less Than Less Than or Equal	&EQ &NE &GT &GE &LT &LE		&EQ_I &NE_I &GT_I &GE_I &LT_I &LE_I	&EQ_DI &NE_DI &GT_DI &GE_DI &LT_DI &LE_DI				&EQ_R &NE_R &GT_R &GE_R &LT_R &LE_R
Bit Operation	AND OR Exclusive OR NOT Bit Shift Left Bit Shift Right Bit Rotate Left Bit Rotate Right Bit Test Bit Set Bit Clear Bit Position Masked Compare	&AN &OR &XO &NOT &SHL &SHR &ROL &ROR &BT &BS &BCL &BP &MCM						&AN_W &OR_W &XO_W &NOT_W &SHL_W &SHR_W &ROL_W &ROR_W &BT_W &BS_W &BCL_W &BP_W &MCM_W	
Data Move	Move Block Move Block Clear Shift Register Bit Sequencer Communications Request	&MOV &BLKM &BLKC &SHF &BI &COMMR		&MOV_I &BLKM_I		&MOV_BI  &SHF_BI		&MOV_W &BLKM_W  &AR_W	&MOV_R &BLKM_R
Table	Array Move Search Equal Search Not Equal Search Greater Than Search Greater Than or Equal Search Less Than Search Less Than or Equal	&AR &SRCHE &SRCHN &SRCHGT &SRCHGE &SRCHLT &SRCHLE		&AR_I &SRCHE_I &SRCHN_I &SRCHGT_I &SRCHGE_I &SRCHLT_I &SRCHLE_I	&AR_DI &SRCHE_DI &SRCHN_DI &SRCHGT_DI &SRCHGE_DI &SRCHLT_DI &SRCHLE_DI	&AR_BI	&AR_BY &SRCHE_BY &SRCHN_BY &SRCHGT_BY &SRCHGE_BY &SRCHLT_BY &SRCHLE_BY	&AR_W &SRCHE_W &SRCHN_W &SRCHGT_W &SRCHGE_W &SRCHLT_W &SRCHLE_W	
Conversion	Convert to Integer Convert to Double Integer Convert to BCD-4 Convert to REAL Convert to WORD Truncate to Integer Truncate to Double Integer	&TO_INT &TO_DINT &BCD4 &TO_REAL &TO_W &TRINT &TRDINT	&TO_INT_BCD4	&TO_BCD4_I &TO_REAL_I	&TO_REAL_DI			&TO_REAL_W	&BCD4_R

Function Group	Instruction	Mnemonic						
		All	INT	DINT	BIT	BYTE	WORD	REAL
Control	Call a Subroutine Do I/O SER-Sequential Event Recorder PID – ISA Algorithm PID – IND Algorithm SFCRES – SFC Reset End Rung Explanation System Services Request Master Control Relay End Master Control Relay Nested Master Control Relay Nested End Master Cntl Relay Jump Nestfed Jump Label Nested Label	&CA &DO &SER &PIDIS &PIDIN &SFCRES &COMME &SV &MCR &ENDMCR &MCRN &ENDMC RN &JUMP &JUMPN &LABEL &LABELN	&JUMP &JUMPN &LABEL &LABELN					

# Appendix E

## Key Functions

The following table lists the keyboard functions that are active in the Logicmaster 90-30/20/Micro software environment. This information may also be displayed on the programmer screen by pressing **ALT-K** to access key help.

Key Sequence	Description	Key Sequence	Description
<i>Keys Available throughout the Software Package</i>			
ALT-A	Abort.	CTRL-Break	Exit package.
ALT-C	Clear field.	Esc	Zoom out.
ALT-M	Change programmer mode.	CTRL-Home	Previous command line contents.
ALT-R	Change PLC <b>RUN/STOP</b> state.	CTRL-End	Next command line contents.
ALT-E	Toggle status area.	CTRL-←	Cursor left within the field.
ALT-J	Toggle the command line.	CTRL-→	Cursor right within the field.
ALT-L	List directory files.	CTRL-D	Decrement reference address.
ALT-P	Print screen.	CTRL-U	Increment reference address.
ALT-H	Help.	Tab	Change/increment field contents.
ALT-K	Key help.	Shift-Tab	Change/decrement field contents.
ALT-I	Instruction mnemonic help.	Enter	Accept field contents.
ALT-T	Start <b>TEACH</b> mode.	CTRL-E	Display last system error.
ALT-Q	Stop <b>TEACH</b> mode.	F12 (or Keypad -)	Toggle discrete reference.
ALT-n	Playback file n (n = 0 thru 9).	F11 (or Keypad *)	Override discrete reference.
<i>Keys Available in the Program Editor Only</i>			
ALT-B	Toggle text editor bell.	Keypad +	Accept rung.
ALT-D	Delete rung element.	Enter	Accept rung.
ALT-S	Store block to PLC and disk.	CTRL-PgUp	Previous rung.
ALT-X	Display zoom level.	CTRL-PgDn	Next rung.
ALT-N	Toggle nickname/reference address.	~	Horizontal shunt.
ALT-U	Update disk.		Vertical shunt.
ALT-V	Variable table window.	Tab	Go to the next operand field.
<i>Special Keys</i>			
ALT-O	Password override. Available only on the Password screen in the configuration software.		

The Help card on the next page contains a listing of the key help and also the instruction mnemonics help text for Logicmaster 90 software. This card is printed in triplicate and is perforated for easier removal from the manual.

Print side 1 of GFJ-055C on this page.



Print side 2 of GFJ-055C on this page.

# Appendix F

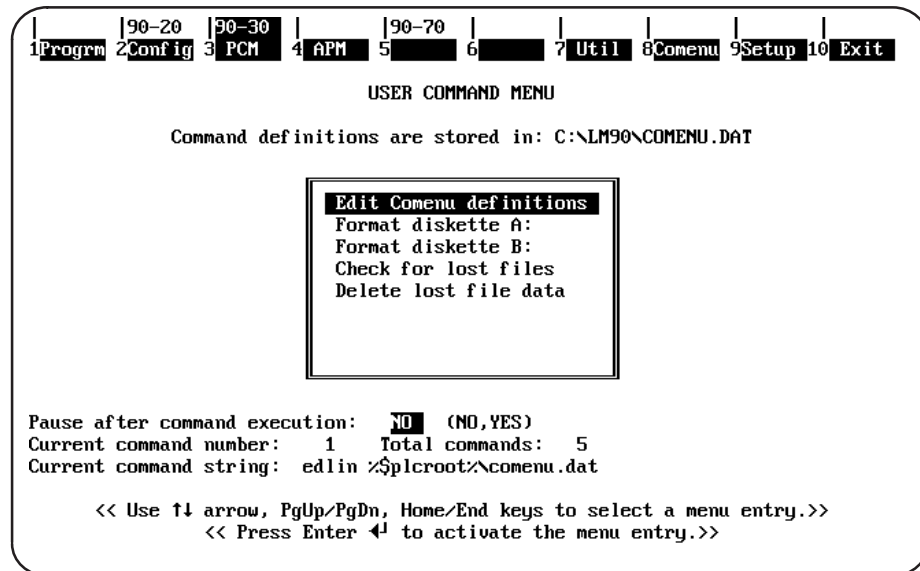
## User Command Menu

You can maintain a file of MS-DOS executable commands outside of the Logimaster software packages. This command definition file named COMENU.DAT contains a list of task entries, each with a unique display label and associated MS-DOS command. The file is stored in the Logimaster home directory (normally \LM90) and can be edited using an MS-DOS compatible text editor such as EDLIN. The MS-DOS commands in this file can be used to run other software packages and perform routine MS-DOS functions, such as disk maintenance.

### Accessing the User Command Menu

The MS-DOS executable commands maintained in the COMENU.DAT file are accessed from the User Command menu called *Comenu*.

1. To display this menu, press **Comenu (F8)** from the menu of Series 90 PLCs and functions. A sample Comenu is shown below.



2. When **F8** is pressed, the Logicmaster startup program reads the Comenu definition file and creates a display similar to the one shown above.
3. Use the cursor keys, the **Page Up** and **Down** keys, or the **Home** and **End** keys to select a command. The label for the currently selected entry is shown in reverse video. If more command definitions are available than can be displayed on the screen, use the cursor keys to scroll through the entries. The number of the currently selected command and the total number of commands are displayed beneath the labels window. The MS-DOS command string for the current entry is shown beneath the labels window for verification of the command text associated with the entry.
4. Move the cursor to the desired command label and press the **Enter** key. The menu program will clear the screen and submit the selected command to MS-DOS for execution. Subsequent screens are controlled by MS-DOS and the selected command.
5. When the command is completed, control will normally return immediately to the Comenu display. However, you may wish to pause after the execution of the command in order to study the display generated by the command. If a pause is desired, enter **Y** (Yes) in the *Pause after command execution* field, or use the **Tab**, **Shift-Tab**, **Space**, or **Backspace** key to toggle the value of the field from **No** to **Yes**.
6. After the pause, MS-DOS will prompt you to press any key in order to continue. When a key is pressed, the User Command menu is displayed again.
7. Press **Exit (F10)** or the **Escape** key to return to the menu of Series 90 PLCs and functions.

## Creating a COMENU.DAT file

A default command definition file is provided with the Logicmaster 90-30/20/Micro software. The default file provides example definitions and includes entries for performing disk formatting.

Definitions may be entered in the COMENU.DAT file in either of two forms:

```
<display label> = <DOS command string>
```

or

```
<display label> =  
<DOS command string>
```

according to these guidelines:

1. Blank lines and lines beginning with a semicolon are ignored.
2. Lead and trailing space and tab characters around the label and command are ignored.
3. The label may be any text up to 50 characters long. Labels which exceed this length are truncated.
4. The equal sign (=) must appear on the same line as the label. A carriage return and line feed may follow the equal sign if you want the command text to start on a separate line.

5. Each command string is terminated by a carriage return and line feed, or by the end-of-file. The command string itself should not contain any carriage returns, even if the string is longer than can be displayed on a single line.
6. The command string is limited by MS-DOS to a maximum length of 127 characters. Commands which exceed this length are ignored, and the associated label is not displayed.
7. A maximum of 100 definitions can be processed; any additional definitions beyond the limit are ignored.
8. You may include in the definition file a command for editing the definition file itself. This will allow you to modify the file even while remaining within the Logicmaster environment.

## Example Comenu Definition File

The following is an example Comenu definition file used to create the Comenu shown above.

```
; This is the Comenu definition file.
; Blank lines and lines starting with a semicolon are ignored.

; The following entry may be used to start the editor on this file,
; so that additional revisions may be made without leaving the LM90 shell.

Edit Comenu definitions =
    edit \LM90\COMENU.DAT

; Disk utilities:

Format diskette A:    = format a: /v
Format diskette B:    = format b: /v
Check for lost files = chkdsk /f
Delete lost file data = del \file0*.chk
Duplicate floppy A:   = diskcopy a: a:

; Third-party packages:

Cadeпа grafсet package = \SWN\CADEPA

; User applications:

Application 1 = \appl\appl.bat
Application 2 = \appl\app2.bat
```

# Appendix G

## *Files Created with Logicmaster 90-30/20/Micro Software*

---

---

This appendix lists the files created with Logicmaster 90-30/20/Micro software. The files are identified here by their extensions. The content of each file is briefly discussed, along with an explanation of when the file is created. None of these files is created during the INSTALL process.

### **Caution**

**Do not use MS-DOS to copy individual files from one folder to another or to delete files. Doing so may produce unexpected results. MS-DOS may only be safely used to copy an entire program folder to another program folder of the same name.**

**You cannot rename a folder with a DOS copy because the folder name is contained in some of the files. There is also a folder marker file, LMFOLDER.30, which must be present in the folder directory.**

**If you have a copy of all files in a folder, but do not know the folder name, you can determine it by examining the \_MAIN.DEC file with DOS DEBUG or a file browser program. For example, go to the folder directory with DOS and enter DEBUG \_MAIN.DEC at the DOS prompt. At the dash (-) prompt, enter the letter D (for Display) followed by the Enter key. Examine the ASCII data shown at the right side of the screen. The folder name starts at the third character on the fourth line of the ASCII data. Type Q (for Quit) and press Enter to leave DEBUG. Create a directory with the same name as the folder and copy all the files to it and Logicmaster should now be able to access this new folder.**

## Files in the Program Folder

The following files are associated with program logic. They are created as a result of editing the program.

### Note

.PDT and .STE are new file extensions available with Release 3.50 of Logicmaster 90-30/20 software. .PDT combines .PRG and .DAT files from earlier releases into one file. .STE combines .SYM and .NXP files from earlier releases into one file.

Extension	File Description	When Logicmaster 90-30/20/Micro Software Creates the File
.PDT	This file contains the program logic.	The _MAIN.PRG file is created during the first edit session in the program. When a subroutine is declared, its .PDT file is created.
.DEC	This file contains program block declarations, as well as other data including information about the coil usage and retentive sense of the %Q and %M references.	The _MAIN.DEC file is created during the first edit session in the program. There is one .DEC file for the entire program.
.STE	This file contains the nicknames, reference descriptions, and identifiers associated with a particular subroutine.	The _MAIN.STE file is created during the first edit session. The .STE file for a subroutine is created when a nickname, reference description, or identifier is first defined within that subroutine.
.EXP	This file contains the text for comment rungs.	The .EXP file is created when the first comment rung for the program is programmed.
.LH1	This file contains the header from each .PDT file.	The .LH1 file is created during the first edit session. There is one .LH1 file per folder.
.SDE	This file contains material that was selected and written to a program segment.	The .SDE file is created when you press <b>Write (F5)</b> . Unlike the other files listed above, .SDE files may be written to any MS-DOS directory. The default destination is the current folder.

When specifying values using the Reference table function, Logicmaster 90-30/20/Micro software creates files to contain this information. The folder may contain these two files for each reference type. Only reference types which permit overrides may have override files. (The %I reference is used here only as an example.)

Extension	File Description	When Logicmaster 90-30/20/Micro Software Creates the File
.I	The _MAIN.I file contains initial values for %I references.	The _MAIN.I file is created when data is first entered for any %I reference. There is only one .I file per folder.
.IO	The _MAIN.IO file contains initial override data for %I references.	The _MAIN.IO file is created when data is first entered for any %I reference. There is only one .IO file per folder.

When specifying formats using the reference table function, Logicmaster 90-30/20/Micro software creates files to contain this information.

Extension	File Description	When Logicmaster 90-30/20/Micro Software Creates the File
.RDF	The .RDF files contain information specifying the formats for reference data.	The FIXED.RDF and MIXED.RDF files are created when a reference format is specified for any of the fixed or mixed tables, respectively. A folder may contain up to two of these files.

The configurator packages creates files to contain the configuration data.

Extension	File Description	When Logicmaster 90-30/20/Micro Software Creates the File
.CFG	The IOCFG.CFG file contains information about the I/O configuration for the program. The CPUCFG.CFG file contains information about the CPU configuration for the program.	The IOCFG.CFG and CPUCFG.CFG files are created when any configuration data is first edited.

Teach files are used to store a sequence of keystrokes that may be repeated.

Extension	File Description	When Logicmaster 90-30/20/Micro Software Creates the File
.DEF	This file contains keys that were pressed while in Teach mode.	The .DEF file is created when Teach mode is entered and a file name is specified.

## Files in the Logicmaster 90-30/20/Micro Home Directory

For your convenience, Logicmaster 90-30/20/Micro software creates files that store information to be used as defaults the next time the software is booted.

Extension	File Description	When Logicmaster 90-30/20/Micro Software Creates the File
.DAT	The modem auto dial feature writes information in the MODEM.DAT file. This file is only used with Logicmaster 90-30 software.	The .DAT file is created using Logicmaster 90-30 software when modem information is saved.
.PSU	These files contain setup information that is specified using the Programmer Setup functions. A file name may be specified for these setup files. Default file names are %COM1.PSU or %COM2.PSU, %PLC030.PSU, and %WSI030.PSU. These files contain information that can be used as defaults on the Printer Serial Port Setup, Select SNP Connections, and Programmer WSI Serial Port Setup screens, respectively.	The .PSU files are created when Save (F7) is pressed.
.SET	This file contains information that is specified using the Print functions.	A PRINT.SET file is created when information on the Setup Printer Parameters screen is saved. The SCRPRINT.SET file is associated with information on the Select Screen Print Destination screen.
.FLD	The LAST30.FLD file contains the name of the selected folder.	The .FLD file is created when the first folder is created when booting the software. It is used as the default the next time the software executes.

## Cross Reference Data Files

Print function cross reference data files are not automatically deleted when you exit the print function. You can, however, specify that the files be deleted at the end of a cross reference listing by setting the *Delete Files After Use* field on the Cross Reference screen to **Y** (Yes). (Refer to chapter 9, section 3, "Print Program," for more information.)

Extension	File Description	When Logicmaster 90-30/20/Micro Software Creates the File
.XRF	The .XRF file contains cross reference data for a single logic block.	A .XRF file is created for each logic block when printing a listing containing cross reference tables, reference use tables, or in-ladder cross references.
.XOV	The PRINTXOV file contains data for managing the .XRF files.	A PRINTXOV file is created when the all blocks option is selected while printing a listing containing cross reference tables, reference use tables, or in-ladder cross references.



### **Error Message: “File System Error”**

If the message, “File System Error”, occurs, the FILES allocated in the CONFIG.SYS file is probably too low. Change or add this line to the CONFIG.SYS file:

**FILES=20**

You must then reboot your computer in order to activate the new value for FILES in the modified CONFIG.SYS file. To reboot your computer, press CTRL-ALT-Delete.

If this line already exists, try changing the number to 30 (i.e., FILES=30) and rebooting your computer.

### **Error Message: “Read/Write PLC Initialization Aborted”**

If the message, “Read/Write PLC Initialization Failure”, occurs while attempting to run Logicmaster 90-30/20/Micro software, the WSI Board may not be seated properly. Re-install the WSI Board in the computer slot or try another slot. If this does not correct the problem, the board may be bad.

### **Error Message: “Constant Out of Range”**

If the message, “Constant Out of Range” occurs while trying to enter hexadecimal constants, you may be trying to enter hex constants into integer type functions. Use the Types (F10) function softkey to change the function type.

### **Error Message: “Invalid CPU”**

You have connected to a different type of PLC, such as a Series 90-70 PLC rather than a Series 90-30. Exit to the top level Logicmaster menu and press **Shift-F3** or **Shift-F5** to select the proper type of PLC

## Error Message: "No Communications"

This message is displayed when you press **Alt-M** to go into Monitor or On-Line Mode and Logicmaster fails to communicate to the PLC. Check that the cable and converter are connected to the proper computer port and that the computer and PLC are using the same data rate and parity. Check the computer settings under **F7** Programmer Mode and Setup, then **F4** PLC Comm. Serial Port Setup. Check the PLC settings by selecting the Configuration package with **F2**, then **F1** for I/O Config, left arrow to the CPU module and <F10> to zoom into the port settings.

A small loopback test program, LB, in the LM90 directory is very helpful in checking serial ports. Go to the LM90 directory under DOS (i.e., **CD \LM90**) and type **LB**, then select the COM port you are using, 1 or 2. Unplug the far end of the RS-232 cable and short pins 2 and 3 with a paper clip or wire and the TD/RD (Transmit/Receive) should Pass. Short pins 7 and 8 on a 9 pin cable (or 4 to 5 on a 25 pin cable) and the RTS/CTS should Pass. Press any key to exit the LB program.

Both TD/RD and RTS/CTS signals must pass for Logicmaster to communicate. If both fail, try the other COM port. If RTS/CTS 'passes' with no jumper wire, it may be a modem port.

## Cannot Install the Software

You must have at least 1.4 Megabytes of hard disk space and an additional 600K bytes of temporary hard disk space if Lotus/Intel/Microsoft expanded memory (LIM 3.0 or higher) is not available for Logicmaster 90-30/20 software. Also, be sure the CONFIG.SYS file has files set to at least 20, i.e., FILES=20.

If the message, "The unarchive program failed (PK11)", occurs, check the CONFIG.SYS file. It should have the line FILES=20. If it does, try changing that to FILES=30, reboot the computer, and try the INSTALL procedure again.

If the message, "The unarchive program failed (PK1)", occurs, try installing the software again. If the INSTALL procedure still fails, the INSTALL diskette is probably bad.

## Error Message: Comm Driver Not Loaded

If the message, "Comm Driver Not Loaded" occurs, you may be trying to run the standard serial communications version of Logicmaster 90-30/20/Micro software without the correct version of MS-DOS and/or without a memory manager. Refer to chapter 1, "Introduction," for a list of what you will need in order to run the Logicmaster 90-30/20/Micro software.

## Printer Output is Garbled

If the *Specify Graphics Printer* option was enabled in the Logicmaster 90 setup package, make sure that the attached printer supports the IBM graphics character set.

If you have a serial printer, make sure the appropriate serial port has been configured with the MS-DOS mode command to match the printer settings. The MS-DOS mode command should be used to configure the serial printer port before starting up the Logicmaster 90 software package. For more information, see chapter 6, "Programmer Setup."

## Error Message: Port/File Access Denied

Attempting to print logic to a virtual drive which is full results in the error "Port/File access denied." To correct the problem, delete some files from your virtual disk.

## System Software Error ID: 0000 EX: 0000

Logicmaster 90-30/20/Micro software does not have enough available RAM to perform the operation. Check the AUTOEXEC.BAT and CONFIG.SYS files to remove any device drivers or Terminate and Stay Resident (TSR) programs in order to free more RAM.

## Busy Message Displayed After Load

The message, "Loading . . ." will continue to be displayed after a load operation fails because of incompatible files. To clear the busy message, press any key.

## Error Message: Error Detected in WSI Board

Some users of IBM-clone computers have experienced difficulty using Logicmaster 90-30/20/Micro software. The computer backplane clock rate for the WSI Board slot should not be greater than 8 MHz.

In addition, certain device drivers installed by CONFIG.SYS when DOS is booted can prevent the expansion bus from performing 8-bit data transfers with the Work Station Interface (WSI) Board. When this situation occurs, the Logicmaster 90-30/20/Micro software displays the error message, "Error detected in WSI Board," during initialization.

To correct the problem for a COMPAQ computer:

1. Copy CONFIG.SYS to CONFIG.SAV to preserve the original version of the CONFIG.SYS file.
2. Examine the CONFIG.SYS file (type CONFIG.SYS) for a DEVICE statement similar to one of these statements:

```
device=xxxxxx.sys bus 16
device=c:xxx/sys bus 16
device=c:xxx.sys <other parameters> bus 16
```

where *<other parameters>* represents any number of words or numbers separated by spaces.

3. Using a text editor such as EDLIN, remove "bus 16" from the statement.
4. Save the modified version.
5. Power the computer off and on. A warm boot (i.e., pressing CTRL-ALT-Delete) will not reset the hardware for correct 8-bit operation.

If some other application program requires the unmodified form of the device driver, different versions of the CONFIG.SYS file can be used for it and for Logicmaster 90-30/20/Micro software. A good method for managing this technique is to use a separate batch file to copy each version when it is needed from its permanent, distinctive file name (e.g., CONFIG.LM) to CONFIG.SYS.

## **Trouble Communicating with the Series 90-30 or Series 90-20 PLC**

**Error Message: Error Detected in WSI Board Port**

**Error Message: Error Loading Code into WSI Board**

Certain boards installed in the computer can conflict with the interrupt used by the Work Station Interface Board. The WSI Board uses the expansion bus IRQ3 interrupt request line. IRQ3 is assigned to the COM2 serial port and is also used by other communication adapters. If the WSI Board is installed in a computer where COM2 is used by an application which requires the use of interrupts (e.g., serial mouse), then neither the WSI Board nor the COM2 application will operate properly. The WSIB2 Board has a software-configurable interrupt request line. Refer to the information on PLC communications options in appendix C, *Programmer Environment Setup*, for information on moving the interrupt request line to one that does not conflict.

The WSI Board uses the MS-DOS memory addresses ranging from CE00:0 to CE00:1FFF. Other applications must avoid this address range. It may be possible to configure the application to use a different address range than that of the WSI Board. Consult the documentation for the conflicting application for more information.

The WSI Board uses the I/O port addresses 310 hex to 313 hex. Other applications must avoid these port addresses in order to prevent conflicts.

# Appendix I

## *Variable Declaration Table Import/Export Using Comma Separated Variable (CSV) Format*

---

---

### **Comma Separated Variable (CSV) Format and SNF Format**

To import to and export from the Variable Declaration Table, Logicmaster 6.0 and later uses an extension of the industry-standard Comma Separated Variable (CSV) format called Shared Name File (SNF) format. Importing an SNF into the Variable Declarations Table gives you the ability to define nicknames ahead of time in a spreadsheet program. Exporting an SNF from the Variable Declarations Table gives you the ability to use the exported file with CIMPLICITY™ and third party operator interfaces.

The Shared Name File (or SNF) format is an extension of the industry-standard Comma Separated Variable (CSV) format. If you are planning on importing a file into the Variable Declarations Table as discussed on page 3-41, you must ensure that your file is in SNF format. When exporting Variable Declarations as discussed on page 3-43, Logicmaster places the files into SNF format (see page 3-41 for more information on SNF format).

SNF files contain two types of information, Comments and Records. Comment lines begin with the Comment Line Sequence ## (i.e., two consecutive pound characters as the first non-blank characters in the line). Records, except for Header Section records (see below), do not begin with the Comment Line sequence.

### **Note**

The ability to import and export CSV files is a Release 6 or later feature.

### **General SNF Format Rules**

- Shared Name Files shall have a default extension of “.SNF”. The recommended naming convention is that the filename be the same as the source file (folder) from which the SNF is generated.
- Comment lines begin with the double pound sign (“##”) character sequence as the first non-blank character.
- Comment lines may appear anywhere within the SNF, but may not occur inside a record.
- Records continued over several lines may not have intervening comment lines.

- The SNF will consist of three sections which must occur in order:
  - (1) an optional Header section
  - (2) a required Field Names section
  - (3) a required Data section.
- All sections are made up of comma separated variable (CSV) records. Each CSV record is composed of a list of fields separated by commas. Individual fields may be enclosed within double quotation marks, and fields that contain commas must be enclosed within quotes.
- All records in a section must include the same fields, and the fields must be in the same order in each record. Fields in records may be left empty, but empty fields must be delimited with commas like all other fields.
- Spaces surrounding the comma delimiters in a record are ignored. To include leading or trailing spaces in a field, surround the field with quotation marks.
- Records may span one or more lines of text in an SNF file. A line of text in an SNF file ends with the new line character ( or character sequence) appropriate for the operating system. The last line of text in the file must include a new line character.
- Any line beginning with a double colon sequence (::) is considered a continuation of the record started on the previous line. Fields must be contained within a single line, and their trailing comma delimiter must follow the field on the same line as the field (i.e., lines of text followed by continuation lines must end in a non-quoted comma).
- Blank lines are not allowed.
- The maximum line size is 2000 characters.
- The maximum record size is 2000 characters not counting continuation characters and new line characters.
- The maximum number of fields in a record is 100.

### **Header Section Rules**

- Header section records may not span multiple lines.
- The Header section contains structured comments which provide some context for the file.
- All header section entries must be accepted by any package using the SNF format, but they need only interpret and use the ones they recognize.

### **Field Names Section Rules**

- The Field Names Section contains one and only one record.
- The Field Names Section begins with the first non-comment line in the file.
- Field Names are case insensitive.
- PT\_ID must be the first field in the Field Names section.
- No empty field are permitted in the Field Names section.

## Data Section Rules

- The Data Section follows the Field Names Section.
- The Data Section begins on the first non-comment line after the Field Name section and continues to the end of the file.
- Predefined Keywords in the Data section are case insensitive.
- PT\_IDs in the Data Section are case insensitive.
- Empty fields are permitted in the Data section, but they must be delimited by commas.
- If the PT\_ID field is empty, the ADDR field must be present and not empty.
- The format of numbers shall be whatever Excel 5.0 uses as a default when creating a CSV files.

## The Header Section

This section of the SNF contains fields which define the context of the SNF. The Header section has four keywords currently defined. It is expected that additional keywords will be defined by GE Fanuc and in addition customers will wish to add their own keywords.

<i>##&amp;&amp;Creator</i>	This identifies the program which specified the content of this SNF file. Three possible values generated by GE Fanuc programs are as follows: "SIMPLICITY" or "LOGICMASTER". Other values for this field will not cause an error.
<i>##&amp;&amp;CreateDate</i>	This specifies the date when this SNF file was created. The date format shall be in the format "(d)d-mmm-yyyy", for example, 31-MAY-1994, 4-NOV-1986, or 11-Sep-1994
<i>##&amp;&amp;ProgDate</i>	The date of the creation of the program with which this SNF file is associated. If Logicmaster is creating the SNF file, this field contains the date the program was last updated.
<i>##&amp;&amp;FileType</i>	This identifies what type of data is contained within this SNF. The possible values for this field are as follows: "I/O" and "I/O_WITH_FAULTS." ("I/O" is the only value currently implemented; "I/O WITH FAULTS" is reserved for future reference.)
<i>##&amp;&amp;DelVarDecl</i>	This identifies the blocks for which the Variable Declaration Table should be deleted before the import occurs. The possible values are "ALLBLOCKS" to delete the Variable Declaration of all blocks, "_MAIN" to delete the program's global Variable Declaration, or the name of the desired blocks can be listed (as in the second example shown below).

### EXAMPLES

```

#&&DelVarDecl, _MAIN
    or
#&&DelVarDecl, _MAIN, Block1, SLIDE
    or
#&&DelVarDecl, ALLBLOCKS

```

## The Field Names Section

This section of the SNF is used to specify the names of the fields in the Data section, and the order of these named fields. The order of the fields in the field names section is the order to be used in all records of the Data section. The field names recognized by Logicmaster are shown below.

Note: Cimplicity recognizes these field names plus field names specified in the *Cimplicity Systems Import/Export Utility Operation Manual* (GFK-0923).

Field Name	Meaning of This Field	Logicmaster Equivalent
PT_ID	Tag Name of this point *	Nickname
ADDR	A mnemonic reference to a PLC memory location.	Reference
DESC	An ASCII string describing what data this point represents.	Reference Description
PT_TYPE	What type of data is being described by this point	No Logicmaster equivalent (I/O is the only valid entry at this time.)

\*There are additional field names used by Cimplicity. For those field names refer to the Cimplicity documentation. Additional fields which are not supported by the importing program will be ignored without error.

The field name order as shown in the above table is the preferred order.

## Specifying a Block Other Than \_MAIN

To specify a block in which to place the variable declaration during import, use the following syntax, "**reference\_address,block\_name**" (the quotation marks are required). If the block name exceeds the seven (7) character maximum, the import will abort. In addition, if the specified block name does not exist, the import will be aborted.

The following example demonstrates the syntax needed:

```
PT_ID, ADDR, DESC
Input1, %I1, "First input"
Input2, "%I2, BLOCK1", "Second input"
Output1, %Q1, "First output"
Output2, "%Q2, BLOCK1", "Second output"
```

If the import of this file is initiated in \_Main, Input1 and Output1 will be defined in the variable declarations for \_MAIN while Input2 and Output2 will be defined in the variable declarations for BLOCK1.



### Example of Variable Declaration Table

VARIABLE DECLARATION TABLE

REFERENCE	NICKNAME	REFERENCE DESCRIPTION
%I00001	SNERT	Starting point
%I00002	I2	Begin
%Q00001	BLIVET	Allowable over
%R00001	TIC101	Temperature Setpoint 101
%AI0001	REBLIV	Analog input
%AQ0001	DIVOT	Output

### Example of an SNF File with All of Variable Declaration Table Selected for Export

```
##&&Creator, LOGICMASTER
##&&CeateDate,2-June-95
##&&FileType,I/O
PT_ID,ADDR,DESC,PT_TYPE
SNERT,%I00001,Starting point,
I2,%I00002,Begin,
BLIVET,%Q00001,Allowable over,
TIC101,%R00001,Temperature Setpoint 101,
REBLIV,%AI0001,analog input,
DIVOT,%AQ0001,Output,
```

### Example of an Excel Spreadsheet Which Can Be Saved as a CSV for Import into Logicmaster

##Example of Excel spreadsheet which can be saved as a CSV file to be imported				
##&&Creator	Excel			
##&&CreateDate	2-June-95			
##&&FileType	I/O			
PT_ID	ADDR	DESC	PT_TYPE	
SNERT	%I00001	Starting point		
I2	%I00001	Begin		
TIC101	%R00001	Temperature Setpoint 101		
REBLIV	%AI0001	Analog input		
DIVOT	%AQ0001	Output		
##PT_ID is the equivalent of Nickname				
## ADDR is the equivalent of Reference				

After saving this spreadsheet as a CSV file and then importing it into the Variable Declaration Table, the fields will be the same as the first example shown above.

## Longer Example CSV (SNF) File

The following is an example of the CSV format (i.e., SNF format) created by Logicmaster. This example was exported from Logicmaster as noted in the first four lines. The double pound sign and double ampersand denotes commented lines. If you had created this in a spreadsheet program, it would look the same without the commented lines.

```
##&&Creator,LOGICMASTER
##&&CreateDate,17-MAR-1995
##&&ProgDate,21-FEB-1995
##&&FileType,I/O
PT_ID,ADDR,DESC,PT_TYPE
D_HATCH,%I00001,Dry Hatch Status,BOOL
W_HATCH,%I00002,Wet Hatch Status,BOOL
A_RESET,%I00008,Alarm Reset,BOOL
ALM_ACK,%Q01000,Alarm Acknowledge from BCS,BOOL
BCS_INT,%Q01001,BCS initialization,BOOL
NEXT_BT,%Q01002,Next Batch,BOOL
AGIT_ON,%M01001,Agitator Status,
BULK_ON,%M01002,Bulk Ingrid Valve Status,
STEAM_O,%M01003,Steam Valve Status,
AGIT_SP,%R00011,Agitate Speed Setpnt,INT
TEMP_SP,%R00012,Reactor Temp Setpnt,INT
AGIT_T,%R00013,Agitate Time (seconds),INT
BULK_AD,%R00014,Bulk Add Measure (lbs),INT
PHCTL1A,%R00021,BulkAddPhase Control Reg #1,INT
PHCTL1B,%R00022,BulkAddPhase Control Reg #2,INT
PH_CB1A,%R00023,BulkAddPhase Ctl But Reg #1,INT
PH_CB1B,%R00024,BulkAddPhase Ctl But Reg #2,INT
INTL_R1,%R00025,BulkAddPhase Intlock Reg,INT
PHSTS_1,%R00026,BulkAddPhase Status Reg,INT
BK_WGT,%R00027,Bulk Added Weight,INT
R_TEMP,%AI0001,Reactor Temp,INT
LIQ_LEV,%AI0002,ReactorLiquid Level,INT
AGIT_S,%AI0003,Agitate Speed,INT
ST_F_RT,%AI0004,Steam Flow Rate,INT
AGIT_SC,%AQ0001,Agitate Speed Command,INT
ST_V_PO,%AQ0002,Steam Valve Position,INT
```

## Numbers

- 350 and higher CPU modules, configuring the Run/Stop switch, 10-17
- 351 CPUs, storing a program created for a 341 or lower CPU to a 351 CPU, 8-7
- 351, 352, and 363 and CPU modules, configuring the SNP ports, 10-18
- 351/2 CPUs, selecting a folder created for another CPU, 7-4

## A

- Access level, 2-29, 5-4
- ADC module, 10-82
- Address overlap, 10-6
- Alphanumeric display coprocessor module, 10-82
- ALT keys, 1-3, 2-24, E-1
- Analog Combo, Configuring, 10-94
- Analog Output, High Density, 10-90
- Analog, configuring an Analog Combo, 10-94
- Annotation, 1-7, 3-30, 7-2
  - comment, 3-30
  - nickname, 3-30
  - reference description, 3-30
- Annotation files, 8-3
- APM
  - APM commands, 10-80
  - configuring a single axis APM, 10-74
  - configuring a two-axis APM, 10-73
  - configuring an APM, 10-73
- ASCII characters, 4-14
- ASCII display format, 4-13
- ASCII string entry, 4-5
- Auto dial feature, 2-13
- Auto-next highest reference, 3-90
- AUTOEXEC.BAT file, 2-4
- Automatically inserting references, 3-96
- Axis positioning module
  - APM commands, 10-80
  - configuring a single axis APM, 10-74
  - configuring a two-axis APM, 10-73

configuring an APM, 10-73

## B

- Backing up program folders, 7-8
- BAS/CCM mode, 10-62
- Base rack, selecting the, 10-26
  - power supply, 10-29
- Base selection for the CPU 211, 10-33
- BASIC mode, 10-62
- BCD-4, 3-15
- Binary display format, 4-13
- Bit, 3-15
- Bit operation functions, 3-10
- Block declarations, 3-18
- Block edit, 3-104
  - bumpless run mode store function, 3-104
- Block locking feature, 3-60
  - displaying the lock status of the subroutine, 3-64
  - EDITLOCK, 3-60
  - permanently locking a subroutine, 3-64
  - PERMEDITLOCK, 3-60, 3-62
  - PERMVIEWLOCK, 3-60, 3-62
  - unlocking a subroutine, 3-63
  - VIEWLOCK, 3-60
- Block memory usage, 5-17
  - block checksum, 5-17
  - block name, 5-17
  - block's explanation text size, 5-17
  - block's logic memory size, 5-17
  - block's symbol table size, 5-17
- Borders, 3-50
- Bumpless run mode store function, 3-104
- Byte, 3-15

## C

- Cabling, 2-2
- CCM ONLY mode, 10-61, 10-69
- CCM/PROG mode, 10-62
- CCM/RTU mode, 10-70
- CCM/SNP mode, 10-72

- CFG file, G-3
- CGA video snow suppression, C-3
- Changing the configuration of a slot, 10-7
- CHKDSK (MS-DOS) command, 2-7
- Clearing PLC memory, 8-11
- Clearing program folders, 7-12
- CMM
  - configuring a CMM, 10-67
    - CCM ONLY mode, 10-69
    - CCM/RTU mode, 10-70
    - CCM/SNP mode, 10-72
    - RTU ONLY mode, 10-70
    - RTU/CCM mode, 10-70
    - RTU/SNP mode, 10-72
    - SNP ONLY mode, 10-71
    - SNP/CCM mode, 10-72
    - SNP/RTU mode, 10-72
  - modes of configuration, 10-67
  - selecting the configuration mode, 10-68
- Coil checking, 3-92
  - MULTIPLE, 3-93
  - SINGLE, 3-93
  - WARN MULTIPLE, 3-93
- Coil Reference Printing, 9-16
- Colon, double, 3-31
- Colors, palette, C-4
- COM1, 1-9, 6-4, 6-10, 6-23
  - port requirements, 6-5
- COM2, 1-9, 6-4, 6-10, 6-23
  - port requirements, 6-5
- Comenu, F-1
- Comm driver not loaded error, H-2
- Comma Separated Variable (CSV) format, 3-41, I-1
  - exporting variable declarations, 3-43
  - guidelines for format and field names, I-1
  - importing variable declarations, 3-41
- Command line, 2-28
- Comment, 3-30
- COMMENT instruction, 3-46
- Comment, rung, creating footers in listings, 3-48–3-50
- Comments, rung, 3-46
  - adding text, 3-47
  - creating borders, 3-50
  - creating longer comments, 3-51
  - inserting a rung comment, 3-46
  - printing a title and subtitle, 3-51
  - starting a new page of comments, 3-51
- Communications driver, 6-6
  - communications driver load order, 6-8
  - expanded memory, 6-7
  - extended memory, 6-7
  - MS-DOS memory areas defined, 6-7
  - MS-DOS system memory, 6-7
- Communications driver load order, 6-8
- Communications module
  - configuring a CMM, 10-67
    - CCM ONLY mode, 10-69
    - CCM/RTU mode, 10-70
    - CCM/SNP mode, 10-72
    - RTU ONLY mode, 10-70
    - RTU/CCM mode, 10-70
    - RTU/SNP mode, 10-72
    - SNP ONLY mode, 10-71
    - SNP/CCM mode, 10-72
    - SNP/RTU mode, 10-72
  - configuring a TCP/IP Ethernet module, 10-63
  - modes of configuration, 10-67
  - selecting the configuration mode, 10-68
- Compliant memory managers, 6-13
- CONFIG.SYS file, 2-4
- Configurable memory limits, 10-24
- Configuration
  - CPU, 11-1
  - I/O, 10-1
- Configuration lesson, B-1
- Configuration of 350 and higher CPUs, Memory protection, 10-18
- Configuration of 351 and higher CPUs, user-configurable memory, 10-24
- Configuration of 364 CPUs, TCP/IP Ethernet configuration, 10-21
- Configuration reference view, 10-101
  - detail screen, 10-103
  - displaying the rack screen, 10-104
  - overlapping references, 10-105
  - reference view table, 10-102
- Configuration software, 1-4, 2-21
  - accessing the print configuration screen, 9-19
  - configuration lesson, B-1
  - CPU configuration, 1-4
  - exiting the software, 2-18

- functions, 2-21
  - I/O configuration, 1-5
  - main menu, 2-21
  - menu tree, 2-22
  - print configuration, 9-18
  - printing configuration, 1-10
  - starting the software, 2-17
  - version number, 2-21
  - Configurations, tested, 6-18
  - Configuring memory for the communications driver, 6-6
  - Conflicts, port usage, 6-12
  - Constant out of range error, H-1
  - Continuation coil, 3-73
  - Continuation contact, 3-73
  - Control functions, 3-13
  - Conventional memory, 6-16
  - Conversion functions, 3-12
  - Copying configuration, 10-6
  - Copying program folders, 7-14
  - Copying program folders to diskettes, 7-15
  - Copying program folders to the hard drive, 7-14
  - Counters, 3-5
  - CPU 211, configuring the, 10-31
  - CPU configuration, 1-4, 11-1
    - changing the PLC date and time, 11-3
    - changing the SNP ID name, 11-4
    - fault display and clearing, 1-4
    - PLC date and time, 11-2
    - PLC memory allocation, 1-4
    - PLC memory limits, 11-5
    - PLC time and date, 1-4
    - SNP ID, 11-4
    - storing the CPU configuration to the PLC, 11-1
    - system response to faults, 1-4
  - CPU module, configuring the, selecting a different CPU module, 10-13
  - CPU, configuring the, 10-9
  - Creating a program, 1-6
  - Creating program folders, 7-4
  - Cross references, 9-10
    - deleting files after printing, 9-13
    - including as part of the ladder logic, 9-11
    - including explicit and/or implicit cross references in the table, 9-12
    - instruction symbols, 9-10
    - parameters, 9-11
    - print coil references, 9-16
    - print reference tables, 9-14
    - printing separate reference use tables or one single table, 9-12
    - printing separate tables or one single table, 9-11
    - selecting reference types for printing cross references, 9-12
    - specifying a directory to store cross reference data files, 9-13
  - CSV and SNF formats, I-1
  - CSV format, 3-41, I-1
    - exporting variable declarations, 3-43
    - guidelines for formatting and field names, I-1
    - importing variable declarations, 3-41
  - CTRL keys, 1-3, 2-24, E-1
  - Current program, status of, 2-28
  - Cutting selected rungs, 3-78
- ## D
- DAT file, G-4
  - Data move functions, 3-11
  - Data types, 3-15
    - BCD-4, 3-15
    - bit, 3-15
    - byte, 3-15
    - int, 3-15
    - real, 3-15
    - word, 3-15
  - Data zoom feature, 3-16
  - Date, 11-2
    - changing the PLC date, 11-3
  - Date, PLC, 1-4
  - DEC file, G-2
  - Decrement reference address, 3-88
  - DEF file, G-3
  - Default configuration, 10-2
  - Default reference table format, 4-18
  - Delete column function, 3-85

- Delete instruction function, 3-87
- Delete row function, 3-86
- Deleting program folders, 7-7
- Deleting the configuration of a module, 10-7
- DINT, 3-15
- Direct connection, 6-3
- Direction of the search, 3-99
- Discrete reference tables, 4-11
- Discrete references, 4-4
  - changing the values of a word of discrete references, 4-8
  - forcing a discrete reference, 4-8
  - removing overrides, 4-10
  - using overrides, 4-9, 4-10
- Disk drive setup, C-5
- Display all mode, 3-52
- Display formats, 4-11, 4-13
  - ASCII, 4-13
  - binary, 4-13
  - changing the display format, 4-13, 4-15
  - changing the format of a table, 4-15
  - defining a mixed table, 4-22
  - deleting a line of a mixed reference table, 4-23
  - editing the title of a mixed reference table, 4-21
  - hexadecimal, 4-13
  - mixed reference table, 4-20
  - mixed user-defined table, 4-13
  - moving the cursor in a mixed reference table, 4-24
  - real integer, 4-13
  - returning to default values, 4-18
  - signed double integer, 4-13
  - signed integer, 4-13
  - timer/counter format, 4-17
  - timer/counter format in a mixed reference table, 4-24
  - tmrctr format, 4-13
- Displaying file settings, 6-12
- Displaying port settings, 6-11
- Double colon, 3-31
- Double precision integer, 3-15
- Drawer, 7-1

## E

- Editing a program, 1-6, 3-1
  - auto-next highest reference, 3-90
  - automatically inserting references, 3-96
  - block edit, 3-104
  - changing register values, 3-114
  - changing the display mode, 3-52
  - coil checking, 3-92
  - continuation coil, 3-73
  - continuation contact, 3-73
  - cutting selected rungs, 3-78
  - delete column, 3-85
  - delete instruction, 3-87
  - delete row, 3-86
  - deleting a rung, 3-76
  - editing a rung, 3-67, 3-69
  - editor options, 3-92
  - forcing and overriding discrete references, 3-113
  - generic search by reference type, 3-103
  - implicit search, 3-103
  - including rungs from a file, 3-81
  - increment/decrement reference address, 3-88
  - inserting or editing rungs, 3-104
  - ladder logic program elements, 3-2
  - modifying a reference address or constant, 3-111
  - modifying instructions, 3-110
  - move logic down, 3-84
  - move logic right, 3-83
  - online editing/monitoring, 3-104
  - open space functions, 3-74, 3-82
  - pasting previously cut rungs, 3-79
  - program annotation, 3-30
  - program entry, 3-23
  - program format, 3-18
  - quick search for a coil, 3-103
  - rung comments, 3-46
  - search and replace function, 3-101
  - search function, 3-98
  - selecting rungs, 3-77
  - subroutine blocks, 3-57
  - substitution groups, 3-106
  - substitutions, 3-105
  - variable declaration table, 3-33
  - viewing variable declarations, 3-76
  - writing selected rungs to a file, 3-80
- Editing functions, 1-7
- EDITLOCK, 3-60
- Editor options, 3-92
  - automatically inserting references, 3-96

coil checking, 3-92  
 EEPROM, 8-13  
 Embedded high speed counter, Series 90-20  
   configuring the embedded HSC for a CPU 211, 10-56  
   configuring the embedded HSC module, 10-50  
 Enhanced GCM, 10-83, 10-85  
 Enhanced Genius communications module, 10-83, 10-85  
 Entry teach file, 2-26  
 Equality of logic, 2-29  
 Error detected in WSI board, H-3  
 Error detected in WSI board port, H-4  
 Error loading code into WSI board, H-4  
 Error messages, H-3  
   cannot install the software, H-2  
   comm driver not loaded, H-2  
   constant out of range, H-1  
   error detected in WSI board, H-3  
   error detected in WSI board port, H-4  
   error loading code into WSI board, H-4  
   file system error, H-1  
   invalid CPU, H-1  
   no communications, H-2  
   port/file access denied, H-3  
   printer output is garbled, H-2  
   read/write PLC initialization aborted, H-1  
   system software error ID: 0000 EX: 0000, H-3  
   trouble communicating with the Series 90-30 or 90-20 PLC, H-4  
 Errors, common user, H-3  
 Exiting the software, 2-16  
 EXP file, G-2  
 Expanded memory, 6-7, 6-16  
 Expansion rack, selecting an, 10-29  
 Explicit cross references, 9-12  
 Explicit search, 3-99  
 Extended memory, 6-7

## F

Fault display and clearing, 1-4, 1-7

Fault table  
   I/O, 5-12  
   PLC, 5-8  
 File system error, H-1  
 Files, G-1  
   .CFG file, G-3  
   .DAT file, G-4  
   .DEC file, G-2  
   .DEF file, G-3  
   .EXP file, G-2  
   .FLD file, G-4  
   .I file, G-3  
   .IO file, G-3  
   .LH1 file, G-2  
   .PDT file, G-2  
   .PSU file, G-4  
   .RDF file, G-3  
   .SDE file, G-2  
   .SET file, G-4  
   .STE file, G-2  
   .XOV file, G-4  
   .XRF file, G-4  
   cross reference data files, G-4  
   in the home directory, G-4  
   in the program folder, G-2  
 FLD file, G-4  
 Folder functions, 7-2, 7-3  
   auto-select function, 7-5  
   automatic folder selection, 7-6  
   backing up program folders, 7-8  
   clearing program folders, 7-12  
   copying program folders, 7-14  
   copying program folders to diskettes, 7-15  
   deleting program folders, 7-7  
   locking/unlocking program folders, 7-13  
   renaming program folders, 7-11  
   restoring program folders, 7-9  
   selecting/creating program folders, 7-4  
 Folder functions, program, 1-9  
 Folders, program, 1-9, 2-18, 7-1  
   annotation files, 7-2  
   auto-select function, 7-5  
   automatic folder selection, 7-6  
   backing up program folders, 7-8  
   clearing program folders, 7-12  
   copying program folders, 7-14  
   copying program folders to diskettes, 7-15  
   deleting program folders, 7-7  
   drawer, 7-1  
   locking/unlocking program folders, 7-13

- names, 7-2
- program folder functions, 7-2
- renaming program folders, 7-11
- restoring program folders, 7-9
- selecting/creating program folders, 7-4
- TEMP program folder, 7-2

Folders, programs, considerations when using a 351 or 352 CPU, 7-4

Footers in listings, 3-48–3-50

Forcing a discrete reference, 4-8

Forcing and overriding discrete references, 3-113

Foreign (third-party) modules, configuring, 10-99

Function key assignments, 2-27

## G

Gateway address, 10-21

GBC, 10-88

GCM, 10-83

Generic I/O modules, 10-47

Generic search, 3-103

Genius Bus Controller, 10-88

- configuring, 10-88

Genius communications module, 10-83

Grounding, 2-2

## H

Hardware, 1-2

- cabling, 2-2
- grounding, 2-2
- keyboards, 2-24
- standard serial communications version setup, 2-3
- WSI board, 2-2
- WSI version setup, 2-2

Help (ALT-H), 1-3

Help screens, 1-3

- help (ALT-H), 1-3
- key help (ALT-K), 1-3
- mnemonic help (ALT-I), 1-3

Hexadecimal display format, 4-13

High Density Analog Output, 10-90

High memory area, 6-14

High speed counter module, Series 90-30,

- configuring the HSC module, 10-50

HSC module, 10-50

## I

I file, G-3

I/O configuration, 1-5, 10-1

- APM commands, 10-80
- changing the configuration of a slot, 10-7
- configuration reference view, 10-101
- configuration validation, 10-6
- configuring 90-30 I/O modules, 10-45
- configuring a CMM, 10-67
- configuring a GBC, 10-88
- configuring a GCM, 10-83
- configuring a PCM, 10-57, 10-58
- configuring a single axis APM, 10-74
- configuring a two-axis APM, 10-73
- configuring an ADC module, 10-82
- configuring an APM, 10-73
- configuring an embedded HSC for a CPU 211, 10-56
- configuring an enhanced GCM, 10-83
- configuring an I/O link interface module, 10-48
- configuring an I/O link master module, 10-49
- configuring the CPU 211, 10-31
- configuring the HSC module, 10-50
- configuring third-party modules, 10-99
- copying configuration from slot to slot, 10-6
- CPU 331, 10-4
- CPU module, 10-9
- CPU311, 10-slot rack, 10-5
- CPU311, 5-slot rack, 10-5
- default configuration, 10-2
- deleting the configuration of a module, 10-7
- displaying the rack screen, 10-4
- enhanced GCM, 10-85
- I/O base selection for the CPU 211, 10-33
- moving the configuration of a module to another slot, 10-8
- rack screen, 1-5
- saving the configuration to disk, 10-8
- selecting a different CPU module, 10-13
- selecting an expansion rack, 10-29
- selecting the base rack, 10-26



selecting the configuration mode of a CMM, 10-68  
 selecting the configuration mode of a PCM, 10-59  
 I/O fault table, 5-12  
   clearing the fault table, 5-15  
   date and time faults were last cleared, 5-12  
   entries overflowed, 5-12  
   fault description, 5-12  
   fault table entries, 5-13  
   number of faults in the fault table, 5-13  
   PLC time/date, 5-12  
   top fault displayed, 5-12  
   total faults, 5-12  
   zooming into the fault table, 5-14  
 I/O link interface module, configuring an, 10-48  
 I/O link master module, configuring an, 10-49  
 I/O modules, configuring 90-30, 10-45  
   generic I/O modules, 10-47  
 IBM-compatible graphics printer, C-3  
 Identifier table, 3-35, 3-40  
 Implicit cross references, 9-12  
 Implicit search, 3-99, 3-103  
 Including rungs from a file, 3-81  
 Increment reference address, 3-88  
 Instruction mnemonics, D-1  
 Instructions, programming mnemonics, D-1  
 INT, 3-15  
 IO file, G-3  
 IP address, 10-21  
   Configuration, 10-21  
   Isolated network, 10-21

## K

Key functions, 1-3, 2-24  
 Key help (ALT-K), 1-3  
 Key Switch configuration for 350 and higher CPUs, 10-17  
 Keyboard functions, 2-24  
 Keyboard macros, 2-24

Keyboard status, 2-28  
 Keyboards, 2-24  
 Keyswitch, 1-8, 2-30, 6-1, C-3

## L

Ladder logic, 3-18  
   accepting a rung, 3-26  
   block declarations, 3-18  
   creating/editing program logic, 3-19  
   data zoom feature, 3-16  
   entering nicknames and reference descriptions, 3-31  
   exiting rung entry, 3-29  
   inserting functions, 3-27  
   inserting logic elements, 3-24  
   language rules, 3-20  
   markers, 3-18  
   power source, 3-18  
   program entry, 3-23  
   program format, 3-18  
   program logic, 1-6, 3-18  
   rungs, 3-18  
   structure of a rung, 3-19  
   using mnemonics, 3-23  
   variable declarations, 1-6, 3-18  
 Ladder logic program elements, 3-2  
   bit operation functions, 3-10  
   control functions, 3-13  
   conversion functions, 3-12  
   data move functions, 3-11  
   data types, 3-15  
   data zoom feature, 3-16  
   math functions, 3-6  
   relational functions, 3-9  
   relay functions, 3-3  
   table functions, 3-11  
   timers and counters, 3-5  
 Left to right reference display mode (ALT-N), 4-7  
 LH1 file, G-2  
 Loading from PLC to programmer, 8-3  
 Locking program folders, 7-13

## M

Markers, 3-18  
 Math functions, 3-6  
 Memory allocation, 1-4

- Memory limits, configurable, 10-24
  - Memory limits, PLC, 11-5
  - Memory manager
    - compliant memory managers, 6-13
    - restrictions, 6-13
    - specifications, 6-13
    - tested configurations, 6-18
    - using conventional memory on an 80386 or higher computer, 6-16
    - using expanded memory on an 80386 or higher computer, 6-16
    - using the high memory area on an 80386 or higher computer, 6-14
    - using the upper memory block on an 80386 or higher computer, 6-15
    - using video RAM, 6-17
  - Memory protection configuration for 350 and higher CPUs, 10-18
  - Memory usage, 5-17
  - Memory, PLC, 5-16
    - amount of memory available, 5-16
    - amount of program memory occupied, 5-16
    - amount of program memory remaining, 5-16
    - model number, 5-16
    - PLC ID, 5-16
    - software revision, 5-16
  - Memory, program, 1-8
  - Message line, 2-28
  - Micro PLC, 10-36
  - Micro PLC, Configuring, 10-36
  - Mixed reference table, 4-20
    - defining a mixed table, 4-22
    - deleting a line, 4-23
    - editing the title, 4-21
    - moving the cursor, 4-24
    - timer/counter format, 4-24
  - Mixed table display format, 4-13
  - Mnemonic help (ALT-I), 1-3
  - Mnemonics, 3-23
  - Mnemonics, instruction, D-1
  - Model 90-30 I/O modules, generic I/O modules, 10-47
  - Model 90-30 I/O modules, configuring, 10-45
  - Modem, 2-13
  - Modifying a reference address or constant, 3-111
  - Modifying instructions, 3-110
  - Monitor mode, 1-8, 2-30, 6-2
  - Move logic down function, 3-84
  - Move logic right function, 3-83
  - Moving the configuration of a module to another slot, 10-8
  - MS-DOS, 1-3
    - CHKDSK command, 2-7
    - memory areas defined, 6-7
    - system memory, 6-7
  - MS-DOS memory areas
    - expanded memory, 6-7
    - extended memory, 6-7
    - MS-DOS system memory, 6-7
  - Multidrop configuration, 6-3, 11-4
  - MULTIPLE coil check, 3-93
- ## N
- Name Server IP address, 10-21
  - Names, 7-2
  - Nickname, 3-30
  - NULL (blank) string, 5-7
- ## O
- OEM protection, 5-7
  - Off-line mode, 1-8, 2-30, 6-2
  - Online editing/monitoring, 3-104
    - block edit, 3-104
    - changing register values, 3-114
    - forcing and overriding discrete references, 3-113
    - inserting or editing rungs, 3-104
    - modifying a reference address or constant, 3-111
    - modifying instructions, 3-110
    - substitution groups, 3-106
    - substitutions, 3-105
  - Online mode, 1-8, 2-30, 6-2
  - Open space functions, 3-74, 3-82
    - delete column, 3-85
    - delete instruction, 3-87
    - delete row, 3-86

- move logic down, 3-84
    - move logic right, 3-83
  - Operation, 2-1
    - auto dial feature, 2-13
    - COMENU.DAT file, F-1
    - exit, 2-16
    - exiting the programming or configuration software, 2-18
    - keyboard functions, 2-24
    - keyboards supported, 2-24
    - screen format, 2-27
    - setup package, 2-16
    - software installation, 2-4
    - standard serial communications version hardware setup, 2-3
    - starting the programming or configuration software, 2-17
    - startup, 2-16
    - teach mode (keyboard macros), 2-24
    - user command menu, F-1
    - using a modem, 2-13
    - using a teach file to enter the software, 2-26
    - WSI version hardware setup, 2-2
  - Option modules
    - configuring a CCM module, 10-67
    - configuring a GCM module, 10-83
    - configuring a Genius Bus Controller module, 10-88
    - configuring a High Speed Counter module, 10-50
    - configuring a PCM module, 10-57
    - configuring a TCP/IP Ethernet module, 10-63
    - configuring an ADC module, 10-82
    - configuring an APM module, 10-73
  - Output, High Density Analog, 10-90
  - Overlapping references, 10-6, 10-105
  - Overrides, 4-9, 4-10
  - Overriding discrete references, 3-113
- ## P
- Pagination guidelines, 9-20
  - Palette colors, C-4
  - Password protection, 1-8, 5-3
    - access description, 5-4
    - access level, 5-4
    - changing the privilege level, 5-4
    - creating, changing, removing passwords, 5-5
    - current level, 5-4
    - enabling/disabling passwords, 5-6
    - OEM protection, 5-7
    - password active, 5-4
    - privilege levels, 1-8
  - Pasting previously cut rungs, 3-79
  - Pause mode, 8-8
  - PCM, 10-57
    - configuration modes, 10-57
    - configuring a PCM, 10-58
      - BAS/CCM mode, 10-62
      - BASIC mode, 10-62
      - CCM ONLY mode, 10-61
      - CCM/PROG mode, 10-62
      - PCM CFG mode, 10-59
      - PROG PRT mode, 10-60
      - PROG/CCM mode, 10-62
    - selecting the configuration mode, 10-59
  - PCM CFG mode, 10-59
  - PCM configuration software, 2-17
  - PDT file, G-2
  - Periodic subroutines, 3-65
  - PERMEDITLOCK, 3-60, 3-62
  - PERMVIEWLOCK, 3-60, 3-62
  - PID function, 3-16
  - PLC
    - changing the PLC date and time, 11-3
    - changing the SNP ID name, 11-4
    - clearing memory, 8-11
    - date and time, 11-2
    - fault display and clearing, 1-4
    - PLC and program memory information, 1-8
    - PLC memory allocation, 1-4
    - SNP ID, 11-4
    - starting/stopping PLC execution, 1-7
    - sweep time display and change, 1-7
    - system response to faults, 1-4
    - time and date, 1-4
  - PLC communications options, C-6
  - PLC communications serial port setup, 6-4
    - displaying file settings, 6-12
    - displaying port settings, 6-11
    - port usage conflicts, 6-12
    - saving the port setup, 6-12
    - setting up a port, 6-11
    - standard serial COM port, 6-4

- WSI serial port setup, 6-19
- PLC control and status, 5-1
  - block memory usage, 5-17
  - changing the privilege level, 5-4
  - configured reference sizes, 5-18
  - creating, changing, removing passwords, 5-5
  - enabling/disabling passwords, 5-6
  - I/O fault table, 5-12
  - OEM protection, 5-7
  - password protection, 5-3
  - PLC CPU sweep control, 5-19
  - PLC fault table, 5-8
  - PLC memory used, 5-16
  - run/stop PLC, 5-2
  - security access level, 5-4
- PLC CPU sweep control, 5-19
  - changing the sweep mode, 5-20–5-22
- PLC fault table, 5-8
  - clearing the fault table, 5-11
  - date and time faults were last cleared, 5-8
  - entries overflowed, 5-8
  - fault descriptions, 5-9
  - fault location, 5-9
  - fault table entries, 5-9
  - number of faults in the fault table, 5-10
  - PLC time/date, 5-8
  - top fault displayed, 5-8
  - total faults, 5-8
  - zooming into the fault table, 5-11
- PLC memory, 5-16
  - amount of memory available, 5-16
  - amount of program memory occupied, 5-16
  - amount of program memory remaining, 5-16
  - model number, 5-16
  - PLC ID, 5-16
  - software revision, 5-16
- PLC memory limits, 11-5
- PLC state, 2-29
- PLC/programmer status, 2-29
  - access level, 2-29
  - current program, 2-28
  - keyboard status, 2-28
  - logic equality, 2-29
  - PLC state, 2-29
  - programmer operating mode, 2-29
  - scan time, 2-29
  - SNP ID, 2-29
- Port configuration for 351, 352, and 363 CPUs, 10-18
- Port requirements, 6-5
- Port usage conflicts, 6-12
- Power source, 3-18
- Power supply, configuring the, 10-29
- Print configuration
  - accessing the print configuration screen, 9-19
  - CPU configuration data, 9-20
  - destination of the listing, 9-20
  - detailed configuration screens, 9-20
  - I/O rack, 9-20
  - number of first rack to print, 9-20
  - number of last rack to print, 9-20
  - pagination guidelines, 9-20
  - parameters, 9-20
  - reference view tables, 9-20
  - specifying a file name, 9-20
  - starting page number, 9-20
  - subtitle, 9-20
  - title, 9-20
- Print functions, 1-10, 9-1
  - examples, 9-21
  - menu, 9-2
  - print coil references, 9-16
  - print configuration, 9-18
  - print program, 9-7
  - print reference tables, 9-14
  - printer parameters, 9-3, 9-4
  - selecting a screen print device, 9-5
  - sending screen prints to a printer, 9-5
  - sending screens to a file, 9-6
- Print program, 9-7
  - all blocks, 9-8
  - cross reference parameters, 9-11
  - cross references, 9-10
  - deleting files after printing cross references, 9-13
  - file name, 9-9
  - from/to rung, 9-9
  - header page, 9-8
  - IL logic, 9-8
  - including cross references as part of the ladder logic, 9-11
  - including explicit and/or implicit cross references in the cross reference table, 9-12
  - logic, 9-8
  - port, 9-9
  - print program parameters, 9-8
  - printing separate reference use tables or one single table, 9-12

- printing separate tables of cross references or one single table, 9-11
- selecting reference types for printing cross references, 9-12
- shortcuts for printing program logic, 9-9
- specifying a directory to store cross reference data files, 9-13
- starting page number, 9-9
- subtitle, 9-8
- title, 9-8
- variable declaration table, 9-8
- Print reference tables
  - address range, 9-15
  - header page, 9-15
  - overrides, 9-15
  - printing value tables, 9-15
  - specifying a file name, 9-15
  - specifying the destination of the listing, 9-15
  - starting page, 9-15
  - subtitle, 9-15
  - title, 9-15
- Printer options, C-3
- Printer parameters, 9-3, 9-4
  - changing setup printer parameters, 9-3
  - control characters, 9-4
  - line feed with carriage return, 9-4
  - number of lines per page, 9-4
  - paper width, 9-4
  - printer setup sequences, 9-4
- Printer serial port setup, 6-23
- Privilege levels, 1-8, 5-4
- PROG PRT mode, 10-60
- PROG/CCM mode, 10-62
- Program block declarations, 1-6
- Program edit run mode store, storing a changed block, 3-104
- Program editing, 3-1
- Program entry, 3-23
  - accepting a rung, 3-26
  - automatically inserting references, 3-40
  - entering nicknames and reference descriptions, 3-31
  - entering variable declarations, 3-36
  - exiting rung entry, 3-29
  - inserting functions, 3-27
  - inserting logic elements, 3-24
  - selecting a reference table, 3-29
  - using mnemonics, 3-23
- Program folder functions, 1-9, 7-2, 7-3
- Program folder names, 7-2
- Program folders, 1-9, 2-18, 7-1
  - annotation files, 7-2
  - auto-select function, 7-5
  - automatic folder selection, 7-6
  - backing up program folders, 7-8
  - clearing program folders, 7-12
  - copying program folders, 7-14
  - copying program folders to diskettes, 7-15
  - deleting program folders, 7-7
  - drawer, 7-1
  - locking/unlocking program folders, 7-13
  - names, 7-2
  - program folder functions, 7-2
  - renaming program folders, 7-11
  - restoring program folders, 7-9
  - selecting/creating program folders, 7-4
  - TEMP program folder, 7-2
- Program format, 3-18
  - block declarations, 3-18
  - creating/editing program logic, 3-19
  - ladder logic language rules, 3-20
  - markers, 3-18
  - power source, 3-18
  - Program block declarations, 1-6
  - program logic, 1-6, 3-18
  - rungs, 3-18
  - structure of a ladder logic rung, 3-19
  - variable declarations, 1-6, 3-18
- Program instructions
  - bit operation functions, 3-10
  - control functions, 3-13
  - conversion functions, 3-12
  - data move functions, 3-11
  - data types, 3-15
  - data zoom feature, 3-16
  - math functions, 3-6
  - relational functions, 3-9
  - relay functions, 3-3
  - table functions, 3-11
  - timers and counters, 3-5
- Program logic, 1-6, 3-18, 3-19
- Program memory, 1-8
- Program utility functions, 1-10, 8-1
  - clearing PLC memory, 8-11
  - loading from PLC to programmer, 8-3
  - pause mode, 8-8
  - read/write/verify EEPROM, 8-13
  - run mode store function, 8-8
  - STOP-to-RUN transition, 8-8

- storing to PLC from programmer, 8-5
  - verifying a program with the PLC, 8-9
  - Programmable coprocessor module
    - configuring a PCM, 10-57, 10-58
      - BAS/CCM mode, 10-62
      - BASIC mode, 10-62
      - CCM ONLY mode, 10-61
      - CCM/PROG mode, 10-62
      - PCM CFG mode, 10-59
      - PROG PRT mode, 10-60
      - PROG/CCM mode, 10-62
    - modes of configuration, 10-57
    - selecting the configuration mode, 10-59
  - Programmer environment setup, C-1
    - disk drive setup, C-5
    - palette colors, C-4
    - PLC communications options, C-6
    - saving the setup, C-9
    - terminal and printer options, C-3
  - Programmer keyswitch, 1-8, 2-30, 6-1, C-3
  - Programmer operating mode, 1-8, 2-29, 2-30, 5-2, 6-2
    - mode selection, 6-2
  - Programmer setup, 2-12, 6-1
    - menu, 6-1
    - PLC communications serial port setup, 6-4
    - printer serial port setup, 6-23
    - programmer operating mode, 6-2
    - selecting SNP connections, 6-3
    - standard serial COM port, 6-4
    - view modes setup, 6-21
    - WSI serial port setup, 6-19
  - Programming instruction mnemonics, D-1
  - Programming lesson, A-1
  - Programming software, 1-6, 2-19
    - annotation, 1-7
    - automatically inserting references, 3-96
    - block declarations, 3-18
    - coil checking, 3-92
    - creating/editing a program, 1-6
    - displaying tables of reference values, 1-7
    - editing functions, 1-7
    - exiting the software, 2-18
    - fault display and clearing, 1-7
    - functions, 2-19
    - main menu, 2-19
    - menu tree, 2-20
    - PLC and program memory information, 1-8
    - PLC sweep time display and change, 1-7
    - printing programs, 1-10
    - program folder functions, 1-9
    - program folders, 1-9
    - program logic, 1-6, 3-18
    - program utility functions, 1-10
    - programmer operating mode, 6-2
    - programming lesson, A-1
    - selecting the programmer operating mode, 1-8, 2-30
    - setting up PLC communications using a WSI board, 1-9
    - setup for PLC communications using standard COM serial ports, 1-9
    - starting the software, 2-17
    - starting/stopping PLC execution, 1-7
    - status information, 2-20
    - system security, 1-8
    - transferring programs, 1-10
    - variable declarations, 1-6, 3-18
    - version number, 2-19
  - PSU file, G-4
- ## Q
- Quick search, 3-103
- ## R
- Rack screen, 10-4
  - RAM, 2-7
    - video RAM, 6-17
  - RDF file, G-3
  - Read EEPROM, 8-13
  - Read/write PLC initialization aborted error, H-1
  - Real, 3-15
  - Real format, 4-13
  - Reference description, 3-30
  - Reference display mode (ALT-N), 4-7
  - Reference overlap, 10-6
  - Reference sizes, configured, 5-18
  - Reference table values, changing, 4-4
  - Reference tables, 1-7, 4-1
    - ASCII characters, 4-14
    - ASCII format, 4-13
    - ASCII string entry, 4-5

- binary format, 4-13
  - changing a register reference, 4-4
  - changing display formats, 4-11
  - changing reference table values, 4-4
  - changing the display format, 4-13, 4-15
  - changing the format of a table, 4-15
  - changing the reference display mode (ALT-N), 4-7
  - changing the values of a word of discrete references, 4-8
  - defining a mixed table, 4-22
  - deleting a line of a mixed reference table, 4-23
  - discrete and register references, 4-4
  - discrete reference tables, 4-11
  - display formats, 4-13
  - displaying a reference table, 4-2, 4-3
  - editing the title of a mixed reference table, 4-21
  - forcing a discrete reference, 4-8
  - hexadecimal format, 4-13
  - mixed reference table, 4-20
  - mixed user-defined table, 4-13
  - moving the cursor in a mixed reference table, 4-24
  - moving the cursor in a reference table, 4-3
  - real format, 4-13
  - register reference tables, 4-12
  - removing overrides, 4-10
  - returning to default values, 4-18
  - signed double integer format, 4-13
  - signed integer format, 4-13
  - system reference table, 4-19
  - timer/counter format, 4-17
  - timer/counter format in a mixed reference table, 4-24
  - tmrctr format, 4-13
  - using overrides, 4-9, 4-10
  - using the cursor to select a reference table, 4-3
- Register reference tables, 4-12
  - Register references, 4-4
    - changing a register reference, 4-4
  - Register values, changing, 3-114
  - Relational functions, 3-9
  - Relay functions, 3-3
  - Renaming program folders, 7-11
  - Replacement target, 3-99
  - Restoring program folders, 7-9
  - RTU ONLY mode, 10-70
  - RTU/CCM mode, 10-70
  - RTU/SNP mode, 10-72
  - Run mode store function, 8-8
  - Run PLC, 5-2
  - Run/Stop switch configuration for 350 and higher CPUs, 10-17
  - Rung, 3-18
  - Rung comments, 3-46
    - adding text, 3-47
    - creating borders, 3-50
    - creating longer comments, 3-51
    - inserting a rung comment, 3-46
    - printing a title and subtitle, 3-51
    - starting a new page of comments, 3-51
  - Rung edit, 3-67, 3-69
    - auto-next highest reference, 3-90
    - changing register values, 3-114
    - continuation coil, 3-73
    - continuation contact, 3-73
    - cutting selected rungs, 3-78
    - delete column, 3-85
    - delete instruction, 3-87
    - delete row, 3-86
    - deleting a rung, 3-76
    - forcing and overriding discrete references, 3-113
    - including rungs from a file, 3-81
    - increment/decrement reference address, 3-88
    - modifying a reference address or constant, 3-111
    - modifying instructions, 3-110
    - move logic down, 3-84
    - move logic right, 3-83
    - open space functions, 3-74, 3-82
    - pasting previously cut rungs, 3-79
    - selecting rungs, 3-77
    - substitution groups, 3-106
    - substitutions, 3-105
    - viewing variable declarations, 3-76
    - writing selected rungs to a file, 3-80

## S

- Saving the configuration to disk, 10-8
- Saving the port setup, 6-12
- Scan time, 2-29
- Scope of the search, 3-99

- Screen format, 2-27
  - command line, 2-28
  - function key assignments, 2-27
  - message line, 2-28
  - PLC/programmer status, 2-29
  - status information, 2-28
- Screen print device, 9-5
  - sending screen prints to a printer, 9-5
  - sending screens to a file, 9-6
- SDE file, G-2
- Search and replace function, 3-101
- Search function, 3-98
  - action to perform on the search target, 3-99
  - direction of the search, 3-99
  - explicit vs. implicit search, 3-99
  - generic search by reference type, 3-103
  - implicit search, 3-103
  - legitimate replacement items, 3-100
  - quick search for a coil, 3-103
  - replacement for target, 3-99
  - scope of the search, 3-99
  - starting point of the search, 3-99
  - target to search for, 3-99
- Security, 1-8, 5-3
  - privilege levels, 1-8
- Selecting a different CPU module, 10-13
- Selecting a reference table, 3-29
- Selecting program folders, 7-4
- Selecting rungs, 3-77
- Serial number, 2-9
- Serial ports, 6-23
  - displaying file settings, 6-12, 6-20
  - displaying port settings, 6-11, 6-20
  - PLC communications serial port setup, 6-4
  - port usage conflicts, 6-12
  - saving the port setup, 6-12, 6-20
  - setting up a port, 6-20
  - setting up a port for PLC communications, 6-11
  - setting up PLC communications using a WSI board, 1-9
  - setup for PLC communications using standard COM serial ports, 1-9
  - standard serial COM port, 6-4
  - WSI serial port setup, 6-19
- SET file, G-4
- Setting up a port, 6-11
- Setup package, 2-16
- Shared Name File format
  - exporting variable declarations to SNF, 3-43
  - importing variable declarations, 3-41
- Shortcuts for printing program logic, 9-9
- Signed double integer display format, 4-13
- Signed integer, 3-15
- Signed integer display format, 4-13
- Simple isolated network configuration, 10-21
- SINGLE coil check, 3-93
- SNF format
  - exporting variable declarations to SNF, 3-43
  - importing variable declarations, 3-41
- Snow suppression, C-3
- SNP connections, 6-3
  - file name, 6-3
  - port connection, 6-3
  - SNP ID, 6-3
  - SNP ID numbers, 6-3
- SNP ID, 2-29, 11-4
  - changing the SNP ID name, 11-4
- SNP ONLY mode, 10-71
- SNP port configuration for 351, 352, and 363 CPUs, 10-18
- SNP/CCM mode, 10-72
- SNP/RTU mode, 10-72
- Software error ID: 0000 EX: 0000, H-3
- Software installation, 2-4
  - AUTOEXEC.BAT and CONFIG.SYS files, 2-4
  - installation instructions, 2-8
  - programmer setup, 2-12
  - RAM required, 2-7
  - serial number, 2-9
  - subdirectories created, 2-4
- Standard COM serial ports, 1-9
- Standard serial COM port, 6-4
  - communications driver load order, 6-8
  - compliant memory managers, 6-13
  - configuring memory for the communications driver, 6-6
  - displaying file settings, 6-12
  - displaying port settings, 6-11



- DOS memory areas defined, 6-7
    - expanded memory, 6-7
    - extended memory, 6-7
    - memory manager restrictions, 6-13
    - memory manager specifications, 6-13
    - MS-DOS system memory, 6-7
    - port requirements, 6-5
    - port usage conflicts, 6-12
    - requirements, 6-5
    - running Logicmaster 90 software, 6-9
    - saving the port setup, 6-12
    - setting up a port for PLC communications, 6-11
    - setting up the computer's serial port, 6-10
    - tested configurations, 6-18
    - using conventional memory on an 80386 or higher computer, 6-16
    - using expanded memory on an 80386 or higher computer, 6-16
    - using the high memory area on an 80386 or higher computer, 6-14
    - using the upper memory block on an 80386 or higher computer, 6-15
    - using video RAM, 6-17
  - Starting up the software, 2-16
    - using a teach file to enter the software, 2-26
  - Status address location, 10-21
  - Status functions, 1-8, 5-1
    - block memory usage, 5-17
    - configured reference sizes, 5-18
    - I/O fault table, 5-12
    - OEM protection, 5-7
    - password protection, 5-3
    - PLC CPU sweep control, 5-19
    - PLC fault table, 5-8
    - PLC memory used, 5-16
    - run/stop PLC, 5-2
    - security access level, 5-4
  - Status information, 2-20, 2-28
    - access level, 2-29
    - current program, 2-28
    - Keyboard status, 2-28
    - logic equality, 2-29
    - PLC and programmer information, 2-28
    - PLC state, 2-29
    - programmer operating mode, 2-29
    - scan time, 2-29
    - SNP ID, 2-29
  - STE file, G-2
  - Stop PLC, 5-2
  - STOP-to-RUN transition, 8-8
  - Storing the CPU configuration to the PLC, 11-1
  - Storing to PLC from programmer, 8-5
    - considerations when storing to a 351 CPU, 8-7
  - Subnet mask, 10-21
  - Subroutine blocks, 3-57
    - adding subroutine block declarations, 3-58
    - deleting subroutine block declarations, 3-59
    - displaying the lock status of the subroutine, 3-64
    - editing subroutine block declarations, 3-58
    - locking/unlocking subroutines, 3-60
    - periodic subroutines, 3-65
    - permanently locking a subroutine, 3-64
    - searching for subroutine block declarations, 3-59
    - unlocking a subroutine, 3-63
    - using goto to move the cursor, 3-59
    - zooming into subroutine block logic, 3-59
  - Substitution groups, 3-106
  - Substitutions, 3-105
  - Sweep control
    - active, 5-19
    - online adjustments, 5-19
  - Sweep mode, configured, 10-10, 10-32
  - Sweep time, 1-7
  - Symbols, cross reference instruction, 9-10
  - System configuration (CONFIG.SYS) file, 2-4
  - System reference table, 4-19
- ## T
- Table functions, 3-11
  - Target, search, 3-99
  - TCP/IP Ethernet Configuration on 364 CPUs, 10-21
  - TCP/IP Ethernet Module, configuration, 10-63
  - Teach mode, 2-24
    - creating teach sequences, 2-25
    - pausing the playback of a teach sequence, 2-25

- playing back a teach sequence, 2-25
- using a teach file to enter the software, 2-26
- TEMP program folder, 7-2, 7-5
- Terminal options, C-3
- Third-party modules, configuring, 10-99
- Time, 11-2
  - changing the PLC time, 11-3
- Time, PLC, 1-4
- Timer interrupts (periodic subroutines), 3-65
- Timer/counter format, 4-17
- Timers, 3-5
  - Instruction mnemonics and description, 3-5
  - Substitution groups, 3-106
- TMRCTR format, 4-13
- Transferring programs, 1-10
- Troubleshooting, H-3

## U

- Unlocking program folders, 7-13
- Upper memory block, 6-15
- Use tables, 9-12
- User command menu, F-1
  - accessing the user command menu, F-1
  - comenu, F-1
  - creating a COMENU.DAT file, F-2
  - example COMENU definition file, F-3
- User errors, common, H-3
- User reference sizes, 5-18
- User-configurable memory limits, 10-24
- Utilities, menu, 8-1
- Utility functions, 8-1
  - clearing PLC memory, 8-11
  - clearing reference override tables, 8-12
  - loading from PLC to programmer, 8-3
  - pause mode, 8-8
  - read/write/verify EEPROM, 8-13
  - run mode store function, 8-8
  - STOP-to-RUN transition, 8-8
  - storing to PLC from programmer, 8-5
  - verifying a program with the PLC, 8-9
- Utility functions, program, 1-10

## V

- Valid configuration, 10-6
- Variable declaration table, 3-33
  - automatically inserting references, 3-40
  - CSV format files, 3-41
  - cut/pasting variable declarations, 3-39
  - deleting variable declarations, 3-39
  - displaying the table, 3-34
  - editing variable declarations, 3-39
  - entering variable declarations, 3-36
  - exporting CSV files, 3-43
  - exporting SNF files, 3-43
  - identifier table, 3-35
  - importing CSV format files, 3-41
  - importing SNF files, 3-41
  - searching for variable declarations, 3-39
  - using goto, 3-39
  - viewing the identifier table, 3-40
  - window, 3-35
- Variable declarations, 1-6, 3-18
- Verify EEPROM, 8-13
- Verifying a program with the PLC, 8-9
- Version of software, 2-19, 2-21
- Video RAM, 6-17
- Video snow suppression, C-3
- View mode (ALT-N), 6-21
- View modes setup, 6-21
  - changing the view mode, 6-22
- VIEWLOCK, 3-60

## W

- WARN MULTIPLE coil check, 3-93
- Word, 3-15
- Work station interface (WSI) board, 2-2
- Write EEPROM, 8-13
- Writing selected rungs to a file, 3-80
- WSI board, 1-9, 2-2
- WSI serial port setup, 6-19
  - baud rate, 6-20
  - displaying file settings, 6-20
  - displaying port settings, 6-20
  - file name, 6-20
  - modem turnaround time, 6-20
  - parity, 6-20
  - saving the port setup, 6-20

setting up a port, 6-20  
stop bits, 6-20

## **X**

XOV file, G-4

XRF file, G-4