Contents

About This Guide
Notational Conventions ................................................................. x
Service Information ........................................................................ x
    Symbol Support Center ........................................................... xi

Chapter 1. Getting Started
Installation ...................................................................................... 1-1
About MCL-Designer ........................................................................ 1-2
    Introduction ............................................................................... 1-2
    Operator Usage ......................................................................... 1-2
    MCL-Designer Features ........................................................... 1-2
Starting MCL Designer ................................................................. 1-3
Setting up a New Project ............................................................. 1-3
Scanner Setup .................................................................................. 1-5
    System Setup ............................................................................. 1-7
    Memory Mapping ...................................................................... 1-8
    Scanner Setup .......................................................................... 1-10
    Phaser RS232 Setup .................................................................. 1-17
MCL-Designer Setup ....................................................................... 1-21

Chapter 2. Quick Tour
Project Setup ..................................................................................... 2-1
Menu Setup ....................................................................................... 2-4
Designing a Program ....................................................................... 2-10
Saving Your Design ......................................................................... 2-20
Simulating and Testing Your Design .............................................. 2-20
Editing Your Design ........................................................................ 2-23
# Chapter 3. Designing a Main Menu

How to Create a Main Menu ................................................................. 3-1
  Adding Text to the Menu ................................................................. 3-1
  Adding Variables to the Menu ......................................................... 3-4
Setting Menu Options ........................................................................ 3-6

# Chapter 4. Designing Programs

Main Program Window ........................................................................ 4-1
  Screen Description ........................................................................... 4-5
  Display Text ..................................................................................... 4-6
  Display Variable .............................................................................. 4-7
  Barcode Input Dialog Box ................................................................. 4-13
Working With Fields ........................................................................ 4-19
  Placing Fields on a Screen ............................................................... 4-19
  Editing Fields on a Screen ............................................................... 4-20
  Moving Fields .................................................................................. 4-20
  Copying Fields ................................................................................ 4-21
Working With Screens ....................................................................... 4-22
  Inserting a Blank Screen ................................................................. 4-22
  Copying a Screen ............................................................................ 4-23
  Moving a Screen ............................................................................. 4-24
  Deleting a Screen ............................................................................ 4-24
  Deleting All Screens ....................................................................... 4-25
  Quick View of All Screens ............................................................... 4-25
Working With Processes ................................................................... 4-26
  Adding a Process ............................................................................. 4-28
  Process List ...................................................................................... 4-28
  Variable Group .............................................................................. 4-30
  Branch Group .................................................................................. 4-40
  Set Label ......................................................................................... 4-46
  Files Group ...................................................................................... 4-49
  MCL-Link Communications Group ................................................. 4-61
  Remote File Check .......................................................................... 4-68
  Serial Communications Group ....................................................... 4-73
  Others Group .................................................................................. 4-78
Printing Reports ................................................................................ 4-84
  Changing the Printer Setup ............................................................. 4-84
  Printing a Screen ............................................................................ 4-84
  Printing Options ............................................................................. 4-84
Working With Files ........................................... 4-87
  Files Window .............................................. 4-87
Working with Variables .................................... 4-92
  Global View of Variables ................................ 4-93

Chapter 5. Simulating the Designed Project
Testing and Simulating ..................................... 5-1
Working Items of the Simulator ......................... 5-2
  Break ......................................................... 5-3
  Bar Code Input ........................................... 5-3
  Variable List ............................................ 5-4
  File Viewer ............................................... 5-5
  MCL-Code Display ....................................... 5-5

Chapter 6. Downloading A Project
Downloading to a Scanner ................................ 6-1

Chapter 7. MCL-Code Reference Guide
About MCL-Code ............................................ 7-1
  MCL-Code Program ....................................... 7-1
  Typographic Conventions ................................ 7-2
Introduction ................................................ 7-3
  Overview of the MCL-Code ............................. 7-3
Variables ...................................................... 7-3
  Using Variables .......................................... 7-3
Files and Records ......................................... 7-5
  General ..................................................... 7-5
  Structure Of The Files ................................. 7-6
MCL Commands ............................................. 7-7
  Backlight (BL) ............................................ 7-10
  Beep (BP) .................................................. 7-11
  ComPute CRC (CP CRC) ................................ 7-13
  ComPute XOR (CP XOR) ................................. 7-15
  ComPute ConVert (CP CVT) ............................. 7-16
  Display Clear (DC) ....................................... 7-19
  Display Message (DM) ................................... 7-20
  File Delete (delete record) (FD) ...................... 7-21
  File Erase (delete file) (FE) ........................... 7-22
  File checK (FK) .......................................... 7-23
  File Open (FO) ............................................ 7-24
  File Print (FP) ............................................ 7-26
Programs and File Sizes ................................................................. 7-95
MCL-Code Error Messages .............................................................. 7-96
System Setup Variables ................................................................. 7-98
PY Parameter Numbers ................................................................. 7-103

Feedback
About This Guide

MCL-Designer® is user-friendly 32-bit Windows software, which allows you to create batch applications. It is a complete, ready-to-work application development system.

This guide contains all the necessary information required to work with MCL-Designer:

- **Chapter 1, Getting Started**
  Introduces the design of a project. Learn how to set up the system, memory, RF, scanner and communication settings.

- **Chapter 2, Quick Tour**
  Teaches the basics of designing with MCL-Designer. Learn how to make a menu and a small program, to add a process, and modify your design.

- **Chapter 3, Designing a Main Menu**
  Shows how to create a menu depending on the designed programs.

- **Chapter 4, Designing Programs**
  Working methods are explained. Learn how to work with programs, variables, fields and files.

- **Chapter 5, Simulating the Designed Project**
  Describes how to simulate the project on an integrated scanner simulator.

- **Chapter 6, Downloading A Project**
  Explains how to download the entire project to the scanner, after it has been created and tested.

- **Chapter 7, MCL-Code Reference Guide**
  Provides a reference guide for MCL-Code commands.

MCL-Designer runs under Windows 95 or Windows NT with at least 16 MB RAM and 20 MB of free hard drive space. A 133 Mhz processor and 16 bit color VGA monitor are recommended.
Notational Conventions

The following conventions are used in this document:

- “Operator” and “User” refer to anyone using the MCL-Designer software.
- “PC” refers to the IBM personal computer or compatible system that you are using to develop applications.
- “Scanner” refers to various types of Symbol scanners.
- “You” refers to the administrator or person who is using this guide as a reference aid to install, configure, or operate the software.
- Keystrokes in bold type indicate non-alphanumeric keystrokes. For example: Select the <F1> key on the scanner to access on-line help.
- **Bold** type is used to identify menu items and input or text fields on a scanner screen.
- *Italics* are used:
  - for the names of parameters in function prototypes and variable names in usage and syntax descriptions.
  - to highlight specific items in the general text.
  - to identify chapters and sections in this and related documents.
- Square brackets [ ] in a command line enclose optional command line parameters.
- The piping symbol (|) has the effect of “or” when it is used to separate inline parameters on a command line; i.e., it separates alternative values for parameters.
- Bullets (•) indicate:
  - action items.
  - lists of alternatives.
  - lists of required steps that are not necessarily sequential.
  - Sequential lists (e.g., those that describe step-by-step procedures) appear as numbered lists.

Service Information

If you have a problem with your software, contact the Symbol Support Center.
Call the Support Center from a phone near the development PC so that the service person can try to talk you through your problem. If the software is found to be working properly, the Support Center may request a sample of your code for analysis at our plant.

**Symbol Support Center**

For service information, warranty information or technical assistance contact or call the Symbol Support Center in:

- **United States**
  Symbol Technologies, Inc.
  One Symbol Plaza
  Holtsville, New York 11742-1300
  1-800-653-5350

- **Canada**
  Symbol Technologies Canada, Inc.
  2540 Matheson Boulevard East
  Mississauga, Ontario, Canada L4W 4Z2
  905-629-7226

- **United Kingdom**
  Symbol Technologies
  Symbol Place
  Winnersh Triangle, Berkshire RG41 5TP
  United Kingdom
  0800 328 2424 (Inside UK)
  +44 208 945 7529 (Outside UK)

- **Asia/Pacific**
  Symbol Technologies Asia, Inc.
  230 Victoria Street #04-05
  Bugis Junction Office Tower
  Singapore 188024
  337-6588 (Inside Singapore)
  +65-337-6588 (Outside Singapore)

- **Australia**
  Symbol Technologies Pty. Ltd.
  432 St. Kilda Road
  Melbourne, Victoria 3004
  1-800-672-906 (Inside Australia)
  +61-3-9866-6044 (Outside Australia)

- **Austria**
  Symbol Technologies Austria GmbH
  Prinz-Eugen Strasse 70
  Suite 3
  2.Haus, 5.Stock
  1040 Vienna, Austria
  1-505-5794 (Inside Austria)
  +43-1-505-5794 (Outside Austria)

**Europe/Mid-East Distributor Operations**

Contact your local distributor or call
+44 118 945 7360
If you purchased your Symbol product from a Symbol Business Partner, contact that Business Partner for service.
Installation

MCL Designer is part of the MCL Collection of software. It can be installed separately or as part of the whole MCL Collection series. To install MCL Designer:

1. Insert the MCL-Collection CD-ROM disk into the CD-ROM drive.
2. Click the Start button on the task bar.
4. Type D:\mcl.exe. and click OK (replace D: with the CD-ROM drive letter).
5. Click Finish when the introductory screen appears.
6. Click Yes to accept the license agreement.
7. Click Next to accept the default directory or click Browse to select a different directory. Click OK, then Next.
8. Click Next to install all the MCL programs, or select MCL Designer and de-select all the other options.
9. Click Next to accept the MCL program folder.
10. Setup installs the selected programs onto the hard drive. A screen that monitors the progress appears. When installation is complete, click Finish.
About MCL-Designer

Introduction

MCL-Designer is a Windows based development tool that generates MCL-Code programs that can be downloaded to the scanner and executed by the MCL-Code runtime installed on the scanner.

Operator Usage

The program development can be divided into three steps:

1. designing the program
2. simulating the program on a PC
3. downloading the program to the scanner.

MCL-Designer Features

MCL-Designer offers a number of features, which enable you to design:

- total project designs
- main menu
- eight programs with 32 screens per program
- eight local data files (for capture and look up functions)
- 103 variables for temporary data storage
- full data input (keyboard, bar code) control
- generated MCL-Code source file
- full paper report on the project
- full graphic simulator.
Starting MCL Designer

To start the MCL Designer program, double-click on the MCL Designer icon on the Windows desktop or:

1. Click on the Start button on the Task bar.
2. Select Programs.
3. Select MCL.
4. Select MCL Designer. The MCL Designer window appears.

Setting up a New Project

Before designing programs, you must create a project. A project can contain a maximum of eight programs. Refer to Chapter 4, Designing Programs for information on designing programs. To set up a project:

1. In the MCL Designer window, click on the Project tab.

![MCL Designer Window](image)

Figure 1-1. MCL Designer Window
Note: If you want to keep notes about the project, click the Notes button.

Table 1-1. Button Bar Descriptions

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![New Icon]</td>
<td>New</td>
<td>Starts a new project.</td>
</tr>
<tr>
<td>![Open Icon]</td>
<td>Open</td>
<td>Opens an existing project.</td>
</tr>
<tr>
<td>![Save Icon]</td>
<td>Save</td>
<td>Saves the current project.</td>
</tr>
<tr>
<td>![Trash Icon]</td>
<td>Trash</td>
<td>Deletes a project.</td>
</tr>
<tr>
<td>![Simulate Icon]</td>
<td>Simulate</td>
<td>Starts project simulation.</td>
</tr>
<tr>
<td>![Verification Icon]</td>
<td>Verification</td>
<td>Verifies a project. This process check maintains a coherence between programs, files and the setup. Clicking this button gives you an error report on screen and files.</td>
</tr>
<tr>
<td>![Print Icon]</td>
<td>Print</td>
<td>Enters print definitions. Refer to Chapter 4, Designing Programs.</td>
</tr>
<tr>
<td>![Send Icon]</td>
<td>Send</td>
<td>Sends the project to the scanner. Refer to Chapter 6, Downloading A Project.</td>
</tr>
<tr>
<td>![Check Icon]</td>
<td>Check</td>
<td>Accepts entered information or steps.</td>
</tr>
<tr>
<td>![Cancel Icon]</td>
<td>Cancel</td>
<td>Terminates the current operation or closes a dialog box or window.</td>
</tr>
</tbody>
</table>

2. Choose New from the File menu or click ![New Icon].
3. In the Project Name field, Enter a project name.
4. Click ![Check Icon] to accept the project name.
Scanner Setup

To set up the scanner information, click on Setup in the Project tab. The Phaser Setup dialog box appears.

Figure 1-2. Phaser Setup - P360/P460 Memory Scanner

The Phaser Setup dialog box allows you to configure:

- the Phaser model
- the version of firmware in the Phaser
- system settings
- the mapping of the data file memory
- the scanner
- the RS232 settings.
The Phaser Setup dialog box contains fields to change the settings of the scanner.

- **Model**
  Sets the Phaser model for which you are designing the project. Click on the list box to select the available model.

- **Firmware**
  Sets the firmware version number. Click on the list box to select the version number.

- **System**
  Sets the system settings. Click on the System button to open the Phaser System Setup dialog box.

- **Memory (P360/P460 Memory Scanner)**
  Sets the amount of memory the scanner is configured with. Click on the Memory button to open the Memory Mapping dialog box.

- **Scanner**
  Sets the default decode parameters. Click on the Scanner button to open the Scanner Setup dialog box.

- **RS232**
  Sets the RS232 settings. Click on the RS232 button to open the Phaser RS232 Setup dialog box.

- Click ✅ to accept the default setting
  or
- Enter the appropriate information to change the values and click ✅ to apply the changes.
- Click ❌ to close the dialog box without saving the changes.
System Setup

To set up the scanner’s system click on the System button. The Phaser System Setup dialog box allows you to set up the configuration of the system (Figure 1-3).

![Phaser System Setup Dialog Box]

Figure 1-3. Phaser System Setup Dialog Box

Phaser System Setup dialog box contains fields to change the settings of the system.

- **Date Format**: Sets the format for the date that is used in the scanner. Click on the list box to display the available options.
- **Time Format**: Sets the format for the time that is used in the scanner. Click on the list box to display the available options.
- **Decimal**: Sets the decimal number format to either a comma or a period. Click the appropriate radio button to change the setting.
- **Beeper Volume**: Sets the beeper volume to high, medium or low, or to a keyboard click sound when data is entered. Click on the appropriate radio button to change the setting.
- **Power Save Time**: Sets the amount of time that the scanner is not in use before the scanner enters the sleep-mode. Use the arrow buttons to select the time in seconds.
Memory Mapping

To set up the data file memory allocation, click on the Memory button on the Phaser Setup dialog box (Figure 1-2). The Memory Mapping dialog box appears.

Figure 1-4. Memory Mapping Dialog Box
Getting Started

Memory Size

Sets the amount of memory the scanner is configured with. Click the appropriate radio button to change the setting.

Data File Memory Allocation

Size KB (A - H)

Sets the amount of memory to be allocated to each data file.

Available Memory

Displays the total memory available.

Allocated Memory

Displays the total amount of memory allocated to the data files.

• Click to accept the default setting or
• Enter the appropriate information to change the values and click to apply the changes.
• Click to close the dialog box without saving the changes.

Note: As a general rule, allocate as much room as possible for each data capture file. If a project has look-up files and a capture file, first allocate the necessary amount of memory to the look-up file, and then allocate all remaining memory available to the capture file. Also, if there are multiple capture files, divide all of the available memory among the capture files in proportion to their expected relative sizes. Thus, if one file (e.g., transfer items) is expected to contain only 10% as much data as another file (e.g., received items), allocate 10 times more memory to the larger file than to the smaller file. These steps will help to maximize the life of the flash memory by minimizing the number of write operations to any given memory block.
**Scanner Setup**

To configure the default decode parameters, click on the **Scanner** button in the **Setup** dialog box. The **Scanner Setup** dialog box allows you to specify the bar code types and certain parameters for the scanner.

![Scanner Setup Window - General Tab](image)

**Figure 1-5. Scanner Setup Window - General Tab**

- **Enable Symbology Decoder**
  - Sets the specific type of Symbology Decoder. Click the appropriate check box to enable the selected decode type.

- **Laser On Timeout**
  - Sets the Laser On Timeout period (1-10 seconds).
### Linear Security Level
Sets the Linear Security Level, which determines how many times a code must be successfully read before being decoded.

### Bi-directional Redundancy
Enables or disables Bi-directional Redundancy. This parameter is only valid when a Linear Security Level has been enabled. When this parameter is enabled, a bar code must be successfully scanned in both directions (forward and reverse) before being decoded.

### Beep on Good Decode
Enables or disables an audible beep when a good decode is achieved.

### Add Code ID
Allows adding a Code ID (Symbol or AIM), or no Code ID (None).

- Click ✅ to accept the default setting
- Enter the appropriate information to change the values and click ✅ to apply the changes.
- Click ❌ to close the dialog box without saving the changes.
**UPC/EAN Tab**

To set up the UPC/EAN options, click on the UPC/EAN tab in the Scanner Setup dialog box.

![Scanner Setup Window - UPC/EAN Tab](image)

**Figure 1-6. Scanner Setup Window - UPC/EAN Tab**

**UPC Settings**

- **Transmit Check Digit**: Sets which UPC bar codes are transmitted with or without check digits. Click the appropriate check box to enable check digits with UPC-A, UPC-E and/or UPC-E1.

- **System Character Preamble**: Sets which UPC bar codes are decoded with or without system character preamble. Click the appropriate check box to enable system character preamble for UPC-A, UPC-E and/or UPC-E1.

- **Country Code Preamble**: Sets which UPC bar codes are decoded with or without country code preamble. Click the appropriate check box to enable country code preamble for UPC-A, UPC-E and/or UPC-E1.
Getting Started

Convert to UPC-A
Converts UPC-E and/or UPC-E1 to UPC-A format. Click the appropriate check box to convert to UPC-A format.

EAN/UPC Settings

EAN-8 Zero Extend
Sets EAN-8 Zero Extend option. If this parameter is enabled, five leading zeros are added to decoded EAN-8 symbols to make them compatible in format to EAN-13 symbols. Click on the check box to enable EAN-8 Zero Extend.

EAN-8 to EAN-13 Type
Enables EAN-8 to EAN-13 option. Click on the check box to enable EAN-8 to EAN-13 option.

Enable Bookland EAN
Enables Bookland EAN option. Click on the check box to enable Bookland EAN.

UPC Security
Sets the UPC security level (0-3).

Supplementals
Sets whether the UPC/EAN bar codes are decoded with or without supplemental characters. If Ignore Supp. is selected, the UPC/EAN is decoded and the supplemental characters ignored. If Decode Supp. only is selected, UPC/EAN symbols without supplemental characters are not decoded. An autodiscriminate (Auto-d) option is also available. If this option is selected, choose an appropriate Redundancy value. A value of 5 or more is recommended.

Redundancy
Sets Redundancy value (2-20). This option adjusts the number of times a symbol without supplementals is decoded before transmission.

- Click ✔ to accept the default setting
- Enter the appropriate information to change the values and click ✔ to apply the changes.
- Click ✗ to close the dialog box without saving the changes.
Code 39 Tab

To set up the Code 39 options, click on the Code 39 tab in the Scanner Setup dialog box.

![Scanner Setup Window - Code 39 Tab](image)

**Figure 1-7. Scanner Setup Window - Code 39 Tab**

**Code 39 Settings**
Sets the format for Code 39 bar codes. Click the appropriate check box to change the setting.

**Check Digit**
Sets the Check Digit options for Code 39 bar codes. When enabled, the Verify Code 39 Check Digit parameter checks the integrity of a Code 39 symbol to ensure it complies with specified algorithms. When selected, the Transmit Check Digit selection allows you to transmit the check digit with the data. Click the appropriate check box to change the setting.

**Trioptic**
Sets the Trioptic options for Code 39 bar codes. Trioptic Code 39 symbols always contain six characters. Click the appropriate check box to change the setting.
Note: Trioptic Code 39 and Code 39 Full ASCII cannot be enabled simultaneously. If you get an error beep when enabling Trioptic Code 39, disable Code 39 Full ASCII and try again.

- Click ✔ to accept the default setting

or

- Enter the appropriate information to change the values and click ✔ to apply the changes.

- Click ✗ to close the dialog box without saving the changes.

Others Tab
To set up the options of Other types of bar codes, click on the Others tab in the Scanner Setup dialog box.

![Scanner Setup Window - Others Tab](image)

Figure 1-8. Scanner Setup Window - Others Tab
### Code 128 Settings
Sets the options for Code 128 bar codes. Click the appropriate check box to change the settings.

### CODABAR Settings
Sets the options for Codabar bar codes. When CLSI is enabled, it strips the start and stop characters and inserts a space after the first, fifth, and tenth characters of a 14-character Codabar symbol. When Notis is enabled, it strips the start and stop characters from decoded Codabar symbol. Click the appropriate check box to change the settings.

### Code 11 Settings
Sets the options for Code 11 bar codes. Click the appropriate check box to change the settings.

### Interleaved 2 of 5 settings
Sets the options for Interleaved 2 of 5 bar codes. Click the appropriate check box to change the settings.

### MSI Settings
Sets the options for MSI bar codes. Click the appropriate check box to change the settings.

- Click ✅ to accept the default setting
- Enter the appropriate information to change the values and click ✅ to apply the changes.
- Click ❌ to close the dialog box without saving the changes.
**Phaser RS232 Setup**

To set up the RS232 communication settings, click on the RS232 button in the *Phaser Setup* dialog box (Figure 1-2). The *Phaser RS232 Setup* dialog box will appear.

![Phaser RS232 Setup Window](Image)

**Figure 1-9. Phaser RS232 Setup Window**

### Speed

Sets the speed (baud rate) of the RS-232 port. Baud rate is the number of bits of data transmitted per second. The scanner’s baud rate setting should match the data rate setting of the host device. If not, data may not reach the host device or may reach it in distorted form. Click on the list box to select the RS-232 port speed.

### Parity

Sets the type of parity (none, odd, even, mark or space). Click on the list box to display the parity options.

- **None**
  - If no parity is required, select **NONE**.
- **Even**
  - If you select **EVEN** parity, the parity bit has a value 0 or 1, based on data, to ensure than an even number of 1 bits are contained in the coded character.
- **Odd**
  - If you select **ODD** parity, the parity bit has a value 0 or 1, based on data, to ensure than an odd number of 1 bits are contained in the coded character.
- **Mark**
  - If you select **MARK** parity, the parity bit is always 1.
- **Space**
  - If you select **SPACE** parity, the parity bit is always 0.
**Data Bits**
Sets the number of data bits for the RS-232 port. Click the appropriate radio button to change the setting.

**Stop Bits**
Sets the number of stop bits for the RS-232 port. The stop bit(s) at the end of each transmitted character marks the end of transmission of one character and prepares the receiving device for the next character in the serial data stream. Click the appropriate radio button to change the setting.

**Software H.**
Selects the Software Handshaking type to be asserted. Click on the list box to select the software handshaking desired.

<table>
<thead>
<tr>
<th>None</th>
<th>When this option is selected, data is transmitted immediately.</th>
</tr>
</thead>
<tbody>
<tr>
<td>XON/XOFF</td>
<td>When this option is selected, an XOFF character turns the scanner transmission off until the scanner receives an XON character. If XOFF is received before the scanner has data to send, the scanner waits up to 2 seconds for an XON character before transmission. If the XON character is not received within this time, the scanner issues an error indication and discards the data. If XOFF is received during a transmission, data transmission stops after sending the current byte. When the scanner receives an XON character, it sends the rest of the data message. The scanner waits indefinitely for the XON character.</td>
</tr>
<tr>
<td>ACK/NAK</td>
<td>When this option is selected, after transmitting data, the scanner expects either an ACK or NAK response from the host. Whenever a NAK is received, the scanner transmits the same data again and waits for either an ACK or NAK. After three unsuccessful attempts to send data when NAKs are received, the scanner issues an error indication and discards the data. The scanner waits up to the programmable Host Serial Response Timeout to receive an ACK or NAK. If the scanner does not get a response in this time, it issues an error indication and discards the data. There are no retries when a timeout occurs.</td>
</tr>
</tbody>
</table>
ENQ Only  When this option is selected, the scanner waits for an ENQ character from the host before transmitting data. If an ENQ is not received within 2 seconds, the scanner issues an error indication and discards the data. The host must transmit an ENQ character at least every 2 seconds to prevent transmission errors.

ACK/NAK with ENQ  Combines the ACK/NAK and ENQ options.

Hardware H.  Selects the Hardware Handshaking type to be asserted. Click on the list box to select the hardware handshaking desired.

None  Select None if no hardware handshaking is desired.

Standard  Selects standard RTS/CTS hardware handshaking.

Option 1  When RTS/CTS Option 1 is selected, the scanner asserts RTS before transmitting and ignores the state of CTS. The scanner deasserts RTS when the transmission is complete.

Option 2  When RTS/CTS Option 2 is selected, RTS is always high or low. However, the scanner waits for CTS to be asserted before transmitting data. If CTS is not asserted within 2 seconds (default), the scanner issues an error indication and discards the data.

Option 3  When RTS/CTS Option 3 is selected, the scanner asserts RTS prior to any data transmission, regardless of the state of CTS. However, the scanner waits for CTS to be asserted before transmitting data. If CTS is not asserted within 2 seconds (default), the scanner issues an error indication and discards the data. The scanner deasserts RTS when transmission is complete.

Host RS-232  Selects the Host type of the RS-232 port. Click on the list box to select the Host type.

Beep on Bell  Enables or disables the Beep on Bell option. When enabled, the scanner issues a beep when a bell character is detected on the RS-232 line. Click on the check box to select or deselect the option.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parity Check</td>
<td>Enables or disables the Parity Check option. Click the appropriate radio button to enable or disable parity check.</td>
</tr>
<tr>
<td>RTS State</td>
<td>Sets the idle state of the Serial Host RTS line to low or high. Click the appropriate radio button to change the setting.</td>
</tr>
<tr>
<td>Intercharacter Delay</td>
<td>Sets the Intercharacter Delay (in 100 msec intervals). The intercharacter delay gives the host system time to service its receiver and perform other tasks between characters. Click the arrow box to select the desired delay.</td>
</tr>
<tr>
<td>Response Timeout</td>
<td>Sets the Response Timeout (in 100 msec intervals). This parameter specifies how long the scanner waits for an ACK, NAK or CTS before determining that a transmission error has occurred. Click the arrow box to select the desired timeout.</td>
</tr>
</tbody>
</table>

- Click ![✓](https://example.com/checkmark) to accept the default setting or
- Enter the appropriate information to change the values and click ![✓](https://example.com/checkmark) to apply the changes.
- Click ![✗](https://example.com/cancel) to close the dialog box without saving the changes.
MCL-Designer Setup

To set up the MCL-Designer program, choose Designer Setup from the Tool menu. The MCL Designer Setup dialog box (Figure 1-10) appears:

![MCL Designer Setup Window](image)

**Figure 1-10. MCL Designer Setup Window**

- **Project Library** Select the project library directory.
- **Simulator Station ID** Sets the scanner ID in simulation mode. Use the up and down arrow to set the station ID number.
- **Data files Directory** Select the directory to store the data files during simulation. Type the directory path in the text box.
- **Assign Aux Port to** This field allows you to assign the terminal’s auxiliary port to a PC communication port (COM1 - COM4).
- **Simulation Startup Options** Sets the simulation options at start-up. Click the appropriate check box to:
  - Delete capture files when starting the simulator.
  - Start the simulation in step by step mode.

- Click **✓** to accept the default setting
- Enter the appropriate information to change the values and click **✓** to apply the changes.
- Click **✗** to close the dialog box without saving the changes.
Quick Tour takes you through the steps required to design a simple project using the MCL Designer program. The steps include:

- setting up the project
- creating a menu
- creating a small program
- saving the project
- simulating the project
- editing the project design.

**Project Setup**

Before designing a new project, you must first set up the project: To set up a project:

1. Click on the *Project* tab in the *MCL Designer* window.
2. Click ☐ to create a new project.
3. In the *Project* text box, enter: **QTOUR**.
4. Click ☑ to accept the project name.
5. Click [Notes] to add information about the project. The Notes window appears:

![Notes Window](image)

**Figure 2-2. Notes Window**

6. In the **Author** text box, type: **My Name**.
7. In the **Notes** text box, type: **Quick Tour**.
8. Click [✓] to accept the information.
9. In the *MCL Designer* window, click [Setup] to specify the settings for the scanner.

10. Click on the *Phaser Model* list box to select the scanner type. Select **P360/P460 Memory Scanner**.

11. Select the firmware version. Select **MCL Ver 2.xx**.

![Phaser Setup Screen](image)

**Figure 2-3. Phaser Setup Screen**

12. Click [ ] to accept the information.
Menu Setup

To create a short menu for your designed project:

1. In the MCL Designer window, click on the Main tab. The Main Tab window appears.

![Figure 2-4. MCL Designer Window - Main Tab](image)

2. Click on the Main Program button. The Program Designer QTOUR/MAIN screen appears.

![Figure 2-5. Program Designer Window - QTOUR / MAIN](image)
3. Click \[\text{Display Text}\] button to display the \emph{Display Text} dialog box.

4. In the \emph{Display Text} box, type: \textbf{QTOUR MAIN MENU}.

5. Choose the \textit{Normal} radio button to display black text on a white background.

6. Click on the desired check boxes for beep options.

![Display Text Dialog Box](image)

**Figure 2-6. Display Text Dialog Box**

7. Click \[\text{OK}\] to accept the information.

8. A cursor with pages attached appears on the screen. Position the cursor on the first line in screen 1.

9. Click the left mouse button to place the text box on the screen.

10. To edit text in the Main User Menu screen, double-click on the text. The Display Text dialog box appears with the selected text. The text can then be edited.
Display Variable

1. Click to display the Display Variable dialog box.

![Display Variable Dialog Box (Date)](image)

2. Click on the Display Variable list box and select System Date.
3. Click on the desired check boxes for beep options.
4. Click to accept the information.
5. A cursor, with pages attached, appears on the Menu Screen. Position the cursor over the second line in screen 1, to the extreme left.
6. Click the left mouse button to place the text box on the screen.
7. Click to display the Display Variable dialog box again.
8. Click on the Display Variable list box and select Time HH:MM.
9. Select the desired format, length on screen and beep options.
Figure 2-8. Display Variable Dialog Box (Time)

1. Click to accept the information.
2. A cursor, with pages attached, appears on the Menu Screen. Position the cursor to the right of the date.
3. Click the left mouse button to place the text box on the screen.

Display Text

1. Click to switch to Screen 2.
2. Click to display the Display Text dialog box.
3. In the *Display Text* box, type: **F1. SCAN DATA**.

4. Click **✓** to accept the information.

5. A cursor, with pages attached, appears on the Menu Screen. Position the cursor over the first line in screen 2.

6. Click the left mouse button to place the text box on the screen. Figure 2-10 illustrates screen 1 of the Menu Screen at this time.
Menu Options

1. Click to set the scanner menu keys. The Menu List dialog box (Figure 2-11) appears.

![Menu List Window](image)

Figure 2-11. Menu List Window

2. In the Menu Options area, type SCAN DATA.
3. In the Shortcut area, click on the list box and select F1.
4. In the Goto Program area, click on the list box and select S02 - SCREEN 2. This enables the SCAN DATA program to run when the F1 key is pressed.
5. Click to accept the information.
6. The Menu design is complete and a program can now be designed.
Designing a Program

This section provides the steps required to create a small program. This program allows the operator to enter a barcode and input keyboard data.

1. In the MCL Designer window, click the Main tab.

2. Click on the Main Program button for the first Program (1). The Program Designer window (Figure 2-13) appears.

Figure 2-12. MCL Designer - Main Tab
Figure 2-13. Program Designer Window

Display Text - Screen 1

1. In the Program Designer window, click to display the Display Text dialog box (Figure 2-14).
2. In the Display Text box, type: SCAN BARCODE.

Figure 2-14. Display Text Dialog Box
3. Click to accept the information.
4. A cursor, with pages attached, appears on the Menu Screen. Position the cursor over the first line.
5. Click the left mouse button to place the text box on the screen.

**Display Text - Screen 2**

1. Click to switch to Screen 2.
2. Click to display the *Display Text* dialog box.
3. In the *Display Text* box, type: **Code:**.

![Display Text Dialog Box](image)

Figure 2-15. Display Text Dialog Box

4. Click to accept the information.
5. A cursor, with pages attached, appears on the Menu Screen. Position the cursor over the left side of the first line in Screen 2.
6. Click the left mouse button to place the text box on the screen.
Barcode Input

1. Click to open the *Barcode Input* dialog box.

![Barcode Input Dialog Box - Input Tab](image)

Figure 2-16. Barcode Input Dialog Box - Input Tab

2. In the *Input* tab, click on \(\text{ }\) to open the *Variables* dialog box.

![Variables Dialog Box](image)

Figure 2-17. Variables Dialog Box

3. Click on the A0 cell and in the *Variable Name* column, type: **CODE**.
4. Click ✓ to accept the information.
5. In the Select Entry Type(s) area, click on the UPC -A, UPC-E0, UPC-E1, EAN 13, EAN 8, PDF 417 and Keyboard check boxes.
6. In the Input Length area, select 1 for Minimum and 13 for Maximum
7. In the Option area, click on the Numeric only check box.
8. In the Validation area, click on the Auto Enter radio button to allow barcode inputs without clicking the Enter key.

**Hot Keys**

1. In the Barcode Input dialog box, choose the Hot Keys tab.

![Barcode Input Dialog Box - Hot Keys Tab](image)

2. In the Other Keys area, click on the Clear list box and select Exit.
3. Click ✓ to accept the information.
4. A cursor, with pages attached, appears on the screen. Position the cursor to the right of the word CODE in the first line in Screen 2.
5. Click the left mouse button to place the text box on the screen.
Display Text - Screen 3

1. Click to switch to Screen 3.
2. Click to display the Display Text dialog box.

![Display Text Dialog Box]

Figure 2-19. Display Text Dialog Box

3. In the Display Text box, type: Qty:
4. Click to accept the information.
5. A cursor, with pages attached, appears on the screen. Position the cursor over the left side of the first line in Screen 3.
6. Click the left mouse button to place the text box on the screen.
Keyboard Input

1. Click to open the Keyboard Input dialog box.

![Keyboard Input Dialog Box - Input Tab](image)

Figure 2-20. Keyboard Input Dialog Box - Input Tab

2. In the Input tab, click to open the Variables dialog box.

![Variables Window](image)

Figure 2-21. Variables Window

3. Click on the A1 cell and in the Variable Name column, type: QTY.
4. Click to accept the information.
5. In the Input Type area, click on the Numeric radio button.
6. In the Default Value area, type: 1.
7. In the *Input Length* area, select **1** for Minimum and **5** for Maximum.
8. In the *Field Validation* area, click on the *Enter Key* radio button to allow the next input only after pressing the Enter key.
9. Click ☑️ to accept the information.
10. A cursor, with pages attached, appears on the screen. Position the cursor to the right of the word QTY: in the first line in Screen 3.
11. Click the left mouse button to place the text box on the screen.

**Adding a Process**

1. Click 🔄 (Process Out) to add a process. The *Process Designer* dialog box appears.

![Figure 2-22. Process Designer Dialog Box](image)

2. Click 🔄 to insert a goto line. The *Goto* dialog box appears.

![Figure 2-23. Goto Dialog Box](image)

3. In the *Go To* list box, select **Current Screen**.
4. Click ☑️ to accept the information.
6. Click ☑️ to accept the information.
Screen Description

1. To insert a description of the screen, click to display the Screen Description dialog box.

![Screen Description Dialog Box](image)

Figure 2-24. Screen Description Dialog Box.

2. In the Screen Name text box, type: **SCAN BARCODE**.

3. Click to accept the information.

Figure 2-25 shows Screen 1 of the program that was just designed.

![Program Designer Window - Screen 1](image)

Figure 2-25. Program Designer Window - Screen 1
Figure 2-26 shows Screen 2 of the program that was just designed.

Figure 2-26. Program Designer Window - Screen 2

Figure 2-27 shows Screen 3 of the program that was just designed.

Figure 2-27. Program Designer Window - Screen 3
Saving Your Design

After creating the program design, it must be saved. To save your design, choose Save, or Save As from the File menu, or click .

Simulating and Testing Your Design

After the program is designed and saved, it can be tested using the simulator. To test the program:

1. In the MCL Designer window or the Program Designer window, click to display the Simulator window. This window displays the first Menu screen that was designed with Quick Tour.

Figure 2-28. Simulator Window
2. In the *Simulator* window, click the FUNC and the 1 keys with the left mouse button, or press the F1 key on the computer to start the program simulation.

![Simulator Window - Code Screen](image)

*Figure 2-29. Simulator Window - Code Screen*

3. Using the computer keyboard, enter the code and press ENTER. The next simulator window appears (*Figure 2-30*).
4. Using the computer keyboard, enter the quantity and press ENTER.

Now that the simulation of the program was successful, it can be downloaded to the scanner. Refer to Chapter 6, *Downloading A Project* for procedures on how to download a program to a scanner.
Editing Your Design

Here are some additional items to work with to edit your design.

- If you want to open a project, choose Open from the File menu or .
- If you want to delete an entire project, click and confirm.
- To delete or modify the menu:
  - To delete a field, click on the field and press the delete key on the keyboard.
  - To modify a field, double-click on the field and type its value.
- To delete, rename, copy or import a program, click on the Program Tab. Then click with your left mouse button on a specific program name. A menu pops up with the options to:
  - Edit, which has the same function as .
  - Remove, to place the program temporarily away.
  - Copy, to copy the program to another place.
  - Rename, to give the program another name.
  - Import, to insert a saved program into a selected program slot.
- To modify a designed screen:
  - To delete a field, click on the field and press the delete key on the keyboard.
  - To modify a field, double-click on the field and type its value.

These are only a few basics to modify your design. If you need to know more about editing and modifying, refer to Chapter 4, Designing Programs.
Chapter 3
Designing a Main Menu

How to Create a Main Menu

A menu is like a table of contents for programs designed within your project. The operator can select a program to work with from this menu screen. Designing a menu consists of placing text and variables on the screen and setting up scanner keys to activate stored programs.

Adding Text to the Menu

To place text on the menu screen:

1. In the MCL Designer window, click on the Main tab. The Main tab window appears.
2. Click on the Main Program button. The Program Designer SAMPLE / MAIN screen appears.
3. Click \( \text{ } \) to add text. The *Display Text* dialog box appears.

![Display Text Dialog Box](image)

**Figure 3-3. Display Text Dialog Box**

4. In the *Display Text* field, type in some text (e.g., *Main Menu*).

5. Click on the *Normal* radio button if you want black text on a white screen, or on the *Reverse* radio button if you want white text on a black screen. For this example click on *Normal*.

   **Note:** To add spaces in reverse mode, use the underscore character.  
   *Example:* _Main Menu_

6. Click on the desired check boxes to select beep options.

7. Click \( \text{✓} \) to accept the information.

8. A cursor, with pages attached, appears on the screen. Position the cursor in the extreme left side of the first row in screen 1.

9. Click the left mouse button to place the text box on the screen.

10. Repeat steps 3 through 7, to enter the text “F1. Program 1” in normal mode.

11. Position the cursor in the extreme left side of the second row in screen 1.
12. Click the left mouse button to place the text box on the screen.

Adding Variables to the Menu

To insert variables, such as a date, a time, etc. in the menu screen:

1. Click ![Variable Icon](image) to add a variable. The Display Variable dialog box appears.
2. Click on the *Display Variable* list box and select: *System Date*.
3. Click on the *Normal* radio button to set black text on a white background.
4. Click ✓ to accept the information.
5. A cursor, with pages attached, appears on the screen.
6. Position the cursor in the first row, next to the words MAIN MENU.
7. Click the left mouse button to place the text box on the screen.

![Figure 3-6. Variable Placed on the Screen](image)
Setting Menu Options

Special keys on the scanner can be specified when working with the Menu. To set these keys, click the MCL Designer screen. The *Menu List* dialog box appears:

![Menu List Screen](image)

- **Column Width**  
  Sets the number of characters in the function name to be displayed on the menu screen. The maximum is 20.

- **Default Position**  
  Sets the default position of the characters to be displayed on the menu screen (1-8).

- **Up/Down Key Mode**  
  Sets the up/down key mode to either scrolling or rolling.

- **Menu Options**  
  Sets the menu option names entered.

- **Shortcut**  
  Sets the programs to start either with the Function Keys (F1,F2,...) or Numeric Keys (1,2,...). F1 or 1 starts Program 1, F2 or 2 starts Program 2, etc.

- **Goto Program**  
  Sets the action to be performed, or program screen to be started.

- Click ✓ to accept the default setting or
- Enter the appropriate information to change the values and click ✓ to apply the changes.
- Click ❌ to close the dialog box without saving the changes.
Chapter 4
Designing Programs

Each project can contain up to eight programs. Programs are designed using the Program Designer window. In each program you create a set of screens, variables and processes that perform specific functions on your scanner.

**Main Program Window**

The Program Designer window is the workplace for designing programs. To start the Program Designer, click on the Programs tab in the MCL Designer window.

![MCL Designer Window - Programs Tab](image)

*Figure 4-1. MCL Designer Window - Programs Tab*
Click on next to one of the programs. The Program Designer window opens (Figure 4-2). Table 4-1 lists the buttons available and a description of each.

![Program Designer Window]

**Figure 4-2. Program Designer Window**

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Folder]</td>
<td>New</td>
<td>Creates a new program. Corresponds to the New item in the File menu.</td>
</tr>
<tr>
<td>![Folder]</td>
<td>Open</td>
<td>Opens an existing program. Corresponds to the Open item in the File menu.</td>
</tr>
<tr>
<td>![Folder]</td>
<td>Save</td>
<td>Saves a program. Corresponds to the Save item in the File menu.</td>
</tr>
<tr>
<td>![Scissors]</td>
<td>Cut</td>
<td>Cuts a field (text, variable) from the screen. Corresponds to the Cut item in the Edit menu.</td>
</tr>
<tr>
<td>![Clipboard]</td>
<td>Copy</td>
<td>Copies a field on a screen. Corresponds to the Copy item in the Edit menu.</td>
</tr>
<tr>
<td>![Clipboard]</td>
<td>Paste</td>
<td>Pastes what was cut or copied onto the screen. Corresponds to the Paste item in the Edit menu.</td>
</tr>
</tbody>
</table>
### Table 4-1. Button Descriptions (Continued)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Duplicate Field]</td>
<td>Duplicate Field</td>
<td>Duplicates a field on the screen. Corresponds to the Duplicate item in the Edit menu.</td>
</tr>
<tr>
<td>![Undo]</td>
<td>Undo</td>
<td>Undoes the last action. Corresponds to the Undo item in the Edit menu.</td>
</tr>
<tr>
<td>![Redo]</td>
<td>Redo</td>
<td>Redoes an undone action. Corresponds to the Repeat item in the Edit menu.</td>
</tr>
<tr>
<td>![Duplicate Screen]</td>
<td>Duplicate Screen</td>
<td>Copies a screen of the program to the same or another program.</td>
</tr>
<tr>
<td>![Insert]</td>
<td>Insert</td>
<td>Inserts a new screen.</td>
</tr>
<tr>
<td>![Move]</td>
<td>Move</td>
<td>Moves a screen to another location.</td>
</tr>
<tr>
<td>![Delete]</td>
<td>Delete</td>
<td>Deletes a screen.</td>
</tr>
<tr>
<td>![Variable List]</td>
<td>Variable List</td>
<td>Displays a list with all the variables in the project.</td>
</tr>
<tr>
<td>![View]</td>
<td>View</td>
<td>Displays a general view of all the screens.</td>
</tr>
<tr>
<td>![Simulate]</td>
<td>Simulate</td>
<td>Simulates the project design.</td>
</tr>
<tr>
<td>![Accept]</td>
<td>Accept</td>
<td>Accepts the entered information.</td>
</tr>
<tr>
<td>![Cancel]</td>
<td>Cancel</td>
<td>Cancels the current operation.</td>
</tr>
</tbody>
</table>
## Table 4-1. Button Descriptions (Continued)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Notes Icon" /></td>
<td>Notes</td>
<td>Click this button to type notes about the screens. Refer to <em>Screen Description</em> on page 4-5.</td>
</tr>
<tr>
<td><img src="image" alt="Display Text Icon" /></td>
<td>Display Text</td>
<td>Click this button to enter text.</td>
</tr>
<tr>
<td><img src="image" alt="Display Variable Icon" /></td>
<td>Display Variable</td>
<td>Click this button to enter a variable like date, time,…</td>
</tr>
<tr>
<td><img src="image" alt="Input Keyboard Icon" /></td>
<td>Input Keyboard</td>
<td>Click this button to enter a keyboard variable.</td>
</tr>
<tr>
<td><img src="image" alt="Input Barcode Icon" /></td>
<td>Input Barcode</td>
<td>Click this button to enter a barcode variable.</td>
</tr>
<tr>
<td><img src="image" alt="Choice Menu Icon" /></td>
<td>Choice Menu</td>
<td>Click this button to edit menu list options.</td>
</tr>
<tr>
<td><img src="image" alt="Choice Input Icon" /></td>
<td>Choice Input</td>
<td>Click this button to edit input list options.</td>
</tr>
<tr>
<td><img src="image" alt="Process In Icon" /></td>
<td>Process In</td>
<td>Click this button to add a process before the current screen.</td>
</tr>
<tr>
<td><img src="image" alt="Process Out Icon" /></td>
<td>Process Out</td>
<td>Click this button to add a process after the current screen.</td>
</tr>
<tr>
<td><img src="image" alt="First Screen Icon" /></td>
<td>First Screen</td>
<td>Displays the first screen of the program.</td>
</tr>
<tr>
<td><img src="image" alt="Previous Screen Icon" /></td>
<td>Previous Screen</td>
<td>Displays the previous screen.</td>
</tr>
<tr>
<td><img src="image" alt="Next Screen Icon" /></td>
<td>Next Screen</td>
<td>Displays the next screen.</td>
</tr>
<tr>
<td><img src="image" alt="Last Screen Icon" /></td>
<td>Last Screen</td>
<td>Displays the last screen of the program.</td>
</tr>
</tbody>
</table>
**Screen Description**

Each screen can have a descriptive name associated with it. To assign a name to a screen, click to display the Screen Description dialog box (Figure 4-3).

![Screen Description Dialog Box](image)

**Figure 4-3. Screen Description Dialog Box**

The Screen Description dialog box displays and sets the screen description. These settings are contained in two text boxes and two check boxes.

- **Screen Name**: Enter a description of the current screen. This name appears above the main screen in the Program Designer window.
- **Start With Clear Screen**: Determines if the previous screen remains or if the screen becomes blank. If Clear Screen check box is selected, the screen becomes blank. If the Clear Screen check box is de-selected, the previous screen will remain. This allows an overlap between screens.
- **Start With Beep**: If selected, enables the scanner to emit a beep when entering the screen.
- **Notes**: Allows the user to write comments about the program being designed.

- Click ✅ to accept the default setting
- Enter the appropriate information to change the values and click ✅ to apply the changes.
- Click ❌ to close the dialog box without saving the changes.
Display Text

To insert text onto the screen use the Display Text dialog box. Click  to open the Display Text dialog box (Figure 4-4).

![Display Text Dialog Box](image)

**Figure 4-4. Display Text Dialog Box**

- **Display Text**: Enter the text that appears on the screen.
- **Normal/Reverse**: Sets the text and background colors. Click on the Normal radio button to display the text in black on a white background or click on the Reverse radio button to display the text in white on a black background.
- **Display First**: Sets this text to display first on the screen when the screen appears.
- **After Display**: Sets a beep to sound on the scanner after the text displays. Click on the list box to select a short, medium, or long beep (or combination of beeps), and the number of beeps. Click  to test the beep options. Click the Pause check box to insert a pause after the text displays, and select the number of seconds of the pause. Click the Enter to Continue check box to make the operator press the Enter key before continuing.
Enter the appropriate information to change the values and click ✓ to accept the information. A cursor with pages attached appears on the screen. Position the cursor to the desired location on the screen and click the left mouse button to place the text on the screen.

Click × to close the dialog box without saving the changes.

**Display Variable**

The *Display Variable* dialog box enables you to insert a variable on the screen. This variable can be a date format, a time format, scanner ID, or a variable that has already been created. Click ✓ to insert a variable. The *Display Variable* dialog box appears.

---

![Display Variable Dialog Box](image)

**Figure 4-5. Display Variable Dialog Box**

- **Display Variable**: Sets the type of variable.
- **Normal/Reverse**: Sets the text and background colors. Click on the *Normal* radio button to display the variable in black on a white background or click on the *Reverse* radio button to display the variable in white on a black background.
- **Display First**: Sets this variable to display first on the scanner screen when this screen appears.
• Enter the appropriate information to change the values and click ✅ to accept the information. A cursor with pages attached appears on the screen. Position the cursor to the desired location on the screen and click the left mouse button to place the text on the screen.

• Click ✗ to close the dialog box without saving the changes.

**Keyboard Input**

The *Keyboard Input* dialog box enables you to insert a keyboard variable (enter the data into a variable using the keyboard). Click ✂ to open the *Keyboard Input* dialog box. The Keyboard Input dialog box contains:

- Input tab
- Advanced tab
- Hot Keys tab.
Input Tab

The Input tab (Figure 4-6) enables you to insert a keyboard variable (enter the data variable using the keyboard).

![Keyboard Input - Input Tab](image)

**Figure 4-6. Keyboard Input - Input Tab**

**Input Variable Name**
Enter or select a variable where the input data will be stored. Click on ... to open the Variable dialog box to create a new variable or select an existing variable. Refer to *Working with Variables* on page 4-92 for information on variables.

**Input Type**
Sets the keyboard input variable type. Click on the appropriate radio button to set the variable type. The Secret radio button indicates a secret code, like a password.

**Default value**
Sets a default value that always appears in the variable field. This is a variable that is used regularly.
### Input Length
Sets the length of the keyboard input. The length can be either an interval or a fixed length (minimum=maximum). Use the up and down arrows to select the lengths.

### Field Validation
Assigns an action to take place after data input has occurred. Click on the appropriate radio button to select the action.

### End of field
When the number of characters entered equals the maximum length, the program goes to the next step.

### Enter Key
When the operator presses the Enter key, the program goes to the next step.

### Next Scan
Each barcode input is scanned without having to press the Enter key.

- Click to accept the default setting
- Enter the appropriate information to change the values and click to apply the changes.
- Click to close the dialog box without saving the changes.
Advanced Tab

The *Advanced* tab contains fields that set extra options for the data input.

![Keyboard Input - Advanced Tab](image)

- **Format Field**
  - Fills the input field with zeroes or blanks on the left or right side. This is used when the input data is smaller than the required input.

- **Check Range**
  - This option makes it possible to make a selection in the data input. If the input corresponds with the range, it will be read. Otherwise, the input will be ignored.

- **Check File**
  - This part of the window makes it possible to select that the entered data is or is not in the local data (capture or look up) file. The first field of the file must be the data to check.

- Click ☑ to accept the default setting
- or
- Enter the appropriate information to change the values and click ☑ to apply the changes.
- Click ✗ to close the dialog box without saving the changes.
Hot Keys Tab

The *Hot Keys* tab configures some of the keys on the scanner keyboard.

![Keyboard Input - Hot Keys Tab](image)

**Figure 4-8. Keyboard Input - Hot Keys Tab**

**Time Out**
Sets a time interval when the scanner will go to another screen or exit if there is no action.

**Other Keys**
Assigns another function to several keys on the scanner keyboard. These keys include Clear, Up, Down, F+Up, F+Dw and Star (*). When the key is pressed, the selected action occurs.

**Function Keys**
Assigns another function to the scanner’s function keys. When the key is pressed, the selected action occurs.

- Click **✓** to accept the default setting
- Enter the appropriate information to change the values and click **✓** to apply the changes.
- Click **✗** to close the dialog box without saving the changes.
Barcode Input Dialog Box

The Barcode Input dialog box enables you to insert a barcode variable (enter data from a scanner). There are four tab windows:

- Input Tab
- Advanced Tab
- Hot Keys Tab
- Branch Tab.

Input Tab

The Input tab (Figure 4-9) enables you to insert a bar code variable (enter the data into a variable using the keyboard).

![Barcode Input - Input Tab](image)

Figure 4-9. Barcode Input - Input Tab
### Input Variable Name
Selects the variable that the input barcode data gets stored into. Click on ![variable] to open the Variable dialog box to create a new variable or select an existing variable. Refer to [Working with Variables](#) on page 4-92 for information on variables.

### Select Entry Type(s)
Selects one or more bar code symbologies. The Keyboard check box allows the operator the opportunity to enter bar code data on the keyboard.

### Input Length
Sets the length of the bar code input. You can set an interval or a fixed length (minimum=maximum) for the input.

### Option
- **Numeric Only**
  - Enter only numeric characters for the barcode variable.

### Validation
- **Auto Enter**
  - Sets the program to continue after each scan without having to enter a keystroke.

- **Enter Key**
  - Sets the program to continue after the operator presses the Enter key.

- **Click ![checkmark] to accept the default setting**
- **Enter the appropriate information to change the values and click ![checkmark] to apply the changes.**
- **Click ![x] to close the dialog box without saving the changes.**
Advanced Tab

The Advanced tab sets extra options for the input bar code.

Figure 4-10. Barcode Input - Advanced Tab

**Check Preamble**

Sets a preamble. These preamble characters are compared with the first characters of the input barcode. The data input is rejected if the preamble does not match the data introduction. If the data is not rejected, you can choose to remove these characters.

**Fill On Max. Length**

Fills the input field with zeroes or blanks on the left or right side. This is used when the input data length is smaller than the required input.

**Propose Default Value**

Selects a default input value. The default value can be a fixed value or a variable, and can be overwritten at the time of input.

**Check File**

Selects if the entered data is or is not in the local data file (capture or look up).

**Note:** The first field of the file must be the data to check.
Store Barcode Descriptor

Selects the variable that the barcode descriptor gets stored into. The barcode descriptor is an internal variable which is required to output barcode data to certain host systems (especially RS232 variants and Synapse cables). Click on \( \) to open the **Variable** dialog box to create a new variable or select an existing variable.

- Click \( \) to accept the default setting
- Enter the appropriate information to change the values and click \( \) to apply the changes.
- Click \( \) to close the dialog box without saving the changes.

**Hot Key Tab**

The **Hot Keys** tab configures some of the keys on the scanner keyboard.

![Barcode Input - Hot Keys Tab](image)

**Figure 4-11. Barcode Input - Hot Keys Tab**
Time Out
Sets a time interval to go to another screen or exit if there is no action.

Other Keys
Assigns another function to several keys on the scanner keyboard. These include Clear, Up, Down, F+Up, F+Dw and Star (*).

Function Keys
Assigns another function to the scanner’s function keys.

- Click ✓ to accept the default setting or
- Enter the appropriate information to change the values and click ✓ to apply the changes.
- Click ✗ to close the dialog box without saving the changes.

Branch Tab
The Branch tab enables the user to set up a case with the bar code type, bar code length or bar code value.

Figure 4-12. Barcode Input - Branch Tab
Enable Test: Enables or disables this test.

*IF* Selects a variable or value to test.

*Equal* Selects a variable or value to test against the value in the *IF* list box.

*Goto* Selects the action to perform or the screen to go to if the test is true.

*Otherwise Goto* Selects the action that occurs if the test is false.

- Click ✔ to accept the default setting
  - Enter the appropriate information to change the values and click ✔ to apply the changes.
  - Click ❌ to close the dialog box without saving the changes.

---

**Note:** The values of system variables, in addition to user-defined variables, can be specified using the ampersand (“&”) character plus the name of the system variable to be used. (These system variables are listed in *MCL-Code Reference Guide.* For example, system variable 33 (“power source”) indicates whether the scanner is operating off battery power (value=0), in a cradle (value=1), or connected to a host (value=2). This variable can thus be used in a Case-and-Branch to switch automatically between different applications, depending on which of these operating modes the scanner is in.
Working With Fields

Fields are variables, text boxes, or lines that are placed on the screen. The following paragraphs provide information on how to edit these fields.

Placing Fields on a Screen

To place fields (display fields or input fields) on the screen:

1. In the Program Designer window, click on one of the buttons listed in Table 4-2.

2. Set the appropriate settings for that field.

3. Click to accept the information.

A cursor, with pages attached, appears on the screen (Figure 4-13).

<table>
<thead>
<tr>
<th>Placing Display Fields</th>
<th>Placing Input Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>to enter text</td>
<td>to enter keyboard input</td>
</tr>
<tr>
<td>to enter a variable</td>
<td>to enter barcode input</td>
</tr>
</tbody>
</table>

Figure 4-13. Place a Field on the Screen
4. Position the cursor over a box on the screen.
5. Click the left mouse button to place the field on the screen.

**Editing Fields on a Screen**

To edit a field on the screen:

1. Position the cursor over a field.
2. Double-click on the field. The appropriate dialog box appears.
3. Change or edit the settings.
4. Click ✅ to accept the new settings. The field is updated on the screen.

**Moving Fields**

To move a field on the same screen:

1. Position the cursor over a field.

2. Drag the left mouse button.
3. Move the cursor to a new position.

4. Release the mouse button.
**Copying Fields**

To copy a field to the same screen or another screen:

<table>
<thead>
<tr>
<th>Using Cut-and-Paste</th>
<th>Using Duplicate Button</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Select the field to copy.</td>
<td>1. Select the field to copy.</td>
</tr>
<tr>
<td>2. Click (copy).</td>
<td>2. Click (duplicate).</td>
</tr>
<tr>
<td>3. Click (paste).</td>
<td>3. Move the field to the new position.</td>
</tr>
<tr>
<td>4. Place the text on the screen by dragging the mouse.</td>
<td>4. Click on the left mouse button.</td>
</tr>
<tr>
<td>5. Click on the left mouse button.</td>
<td></td>
</tr>
</tbody>
</table>

*Note: The icons (copy) and (paste) represent the respective actions.*
Working With Screens

Each program in a project can contain up to 32 screens. All screens do not have to be used in a program. When the program goes to a screen, the execution process is as follows:

- a. execute *Process In* processes
- b. clear screen (optional)
- c. beep (optional)
- d. display screen content
- e. process input fields as they are encountered
- f. execute *Process Out* processes.

Screen execution begins in the top left corner and continues line by line until the lower right hand corner is reached.

Inserting a Blank Screen

To insert a blank screen choose *Insert A Blank Screen* from the *Edit* menu or click ![Insert Button](image) to display the *Insert* dialog box.

![Insert Dialog Box](image)

*Figure 4-14. Insert Dialog Box*
Designing Programs

Insert Blank Screen Before
Selects the screen that a blank screen will go before. Click on the list box to select the screen.

Update Screen Link
Maintains the previous links (goto) for the next screen.

• Click  to accept the default setting
or
• Enter the appropriate information to change the values and click  to apply the changes.
• Click  to close the dialog box without saving the changes.

Copying a Screen
To copy the current screen to a new screen, click  . The Copy dialog box appears.

Figure 4-15. Copy Dialog Box

1. Click on the Copy Screen To list box and select the screen that the current screen will be copied to.

Note: When a screen is copied, it over-writes the selected screen.

2. Click one or all of the check boxes to select the parts of the screen to copy:
   • Copy Process In check box to copy the process before the screen
   • Copy Screen check box to copy the screen with the fields
   • Copy Process Out check box to copy the process after the screen.
3. Click  to copy the screen.
Moving a Screen

To move the current screen to a new screen, click \( \text{Move} \). The Move dialog box appears.

![Figure 4-16. Move Dialog Box](image)

1. Click on the Move Screen To list box and select the screen that the current screen will be moved to.

   **Note:** When a screen is moved, it over-writes the selected screen.

2. Click the Update Screen Links check box to maintain the previous links for the next screen.
3. Click \( \text{Move} \) to move the screen.

Deleting a Screen

To delete a screen, choose Erase a screen from the Edit menu or click \( \text{Erase} \). The Delete dialog box appears.

![Figure 4-17. Delete Dialog Box](image)
1. Click on the *Delete Screen* list box and select the screen that you want to delete.
2. Click the *Update Screen Links* check box to maintain the previous links for the next screen.
3. Click ✅ to delete the screen.

**Deleting All Screens**

To delete all screens, choose *Erase all screens* from the *Edit* menu. A Confirm dialog box appears.

*Note:* *If you erase all the screens, all data will be permanently lost.*

Click Yes to confirm or No to Cancel.

**Quick View of All Screens**

To display a Quick View of all screens, click ✏️. The Quick View window appears.

*Figure 4-18. Confirmation Dialog Box*

*Figure 4-19. Quick View Window*

Use the arrow button on the bottom of the screen to scroll between screens.
Working With Processes

A process is a sequence of operations that perform before and/or after a screen. Figure 4-20 illustrates the process sequence.

![Figure 4-20. Process Operation](image)

The processes are executed sequentially, unless a specific goto function is defined in the process. The Process (In or Out) window (Figure 4-21) lists the process number, the process, and a description for each process. Table 4-3 lists and describes the buttons in the Process window.

![Figure 4-21. Process (In or Out) Window](image)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Insert Line" /></td>
<td>Insert Line</td>
<td>Inserts a new process line.</td>
</tr>
<tr>
<td><img src="image" alt="Edit" /></td>
<td>Edit</td>
<td>Edits a process line.</td>
</tr>
<tr>
<td><img src="image" alt="Delete" /></td>
<td>Delete</td>
<td>Deletes a line in the process screen.</td>
</tr>
</tbody>
</table>
### Table 4-3. Process Window Button Descriptions (Continued)

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Copy]</td>
<td>Copy</td>
<td>Copies a process line.</td>
</tr>
<tr>
<td>![Paste]</td>
<td>Paste</td>
<td>Pastes a process line.</td>
</tr>
<tr>
<td>![GoTo]</td>
<td>GoTo</td>
<td>Opens the <em>Goto</em> dialog box.</td>
</tr>
<tr>
<td>![Set a Label]</td>
<td>Set a Label</td>
<td>Opens the <em>Label</em> dialog box.</td>
</tr>
<tr>
<td>![Test Branch]</td>
<td>Test Branch</td>
<td>Opens the <em>Test and Branch</em> dialog box.</td>
</tr>
<tr>
<td>![Key Branch]</td>
<td>Key Branch</td>
<td>Opens the <em>Key and Branch</em> dialog box.</td>
</tr>
<tr>
<td>![Assign a Variable]</td>
<td>Assign a Variable</td>
<td>Opens the <em>Variable Assign</em> dialog box.</td>
</tr>
<tr>
<td>![Arithmetic]</td>
<td>Arithmetic</td>
<td>Opens the <em>Variable Arithmetic</em> dialog box.</td>
</tr>
<tr>
<td>![Beep]</td>
<td>Beep</td>
<td>Opens the <em>Beep</em> dialog box.</td>
</tr>
<tr>
<td>![Warning Message]</td>
<td>Warning Message</td>
<td>Opens the <em>Warning Message</em> dialog box.</td>
</tr>
<tr>
<td>![Comment Line]</td>
<td>Comment Line</td>
<td>Opens the <em>Comment</em> dialog box.</td>
</tr>
<tr>
<td>![Variable List]</td>
<td>Variable List</td>
<td>Opens the project’s <em>Variables</em> list window.</td>
</tr>
<tr>
<td>![OK]</td>
<td>OK</td>
<td>Confirms a selection.</td>
</tr>
</tbody>
</table>
Adding a Process

To add or edit a process click  (Process In) or click  (Process Out), depending upon where to add the process. The Process (In or Out) window appears (Figure 4-21 on page 4-26).

Process List

Processes are divided into six groups with each group containing a number of commands. Table 4-4 lists the process groups and their associated commands.

Table 4-4. Process Groups

<table>
<thead>
<tr>
<th>Group</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>Assign, Combine, Format, Substring, Extract, Length, + - x / % (Arithmetic)</td>
</tr>
<tr>
<td>Branch</td>
<td>Goto, Key &amp; Branch, Test &amp; Branch, Case &amp; Branch, Control &amp; Branch, Set Label, Gosub</td>
</tr>
</tbody>
</table>
### Table 4-4. Process Groups (Continued)

<table>
<thead>
<tr>
<th>Group</th>
<th>Commands</th>
</tr>
</thead>
</table>
| Files                                | Write a Record  
Read a Record  
Delete a Record  
Review File  
Erase File  
Seek  
File Check |
| MCL-Link Communications              | Send Packet  
Receive Packet  
Send File  
Receive File  
Host Look-Up  
MCL-Link Start  
Remote File Check  
Remote File Operation  
Start EXE Program on Host  
Modem Command |
| Serial Communications                | String Serial Output  
Scanner Serial Output  
Receive Data  
Print File |
| Other                                | Pause  
Beep  
Display Data  
Warning Message  
Clear Screen  
MCL Code  
Comment |
To insert a process, click \( \text{Edit} \) to edit a process line or click \( \text{New} \) to insert a new process line. The *Insert A Process* dialog box appears.

![Insert Process Dialog Box](image)

**Figure 4-22. Insert Process Dialog Box**

Click on a Group in the *Group* list. The *Command* list then displays the commands for the selected group. Double-click on a command or click \( \text{Insert} \) to insert the process.

**Variable Group**

*Note:* The resulting variable can be one of the input variables for the following examples.

**Assign Variable**

In the *Insert A Process* dialog box, click on Group *Variable* and then on Command *Assign*. Click on the \( \text{Assign} \) button to open the *Variable Assign* dialog box. This dialog box enables you to assign a value to a variable.

![Variable Assign Dialog Box](image)

**Figure 4-23. Variable Assign Dialog Box**
Designing Programs

Value
Enter the value to assign to a variable.

Result Variable Name
Selects the variable that the value is assigned to. Click on the to open the Variable dialog box to create a new variable. Refer to Working with Variables on page 4-92 for information on variables.

• Click to accept the default setting
  or
• Enter the appropriate information to change the values and click to apply the changes.
• Click to close the dialog box without saving the changes.

Combine
In the Insert A Process dialog box, click on Group Variable and then on Command Combine. Click on the to open the Variable Combine dialog box. This dialog box enables you to combine all or part of variables into a new variable.

![Variable Combine Dialog Box](image)

Figure 4-24. Variable Combine Dialog Box
To combine variables:

1. Click on the *Data String* list box to select a variable to combine (e.g., V_PROD).

   **Note:** This data string can also be a fixed value entered directly in the field (e.g., 134).

2. Click the *Extract* check box if only a portion of this variable is required.
3. If so, choose the position of the characters to extract using the *From* and *To* counters (e.g., 1 to 6).
4. Repeat steps 1 through 3 for the next variable (e.g., V_LOCAT).
5. Do not select *Extract* check box if all of the variable is to be used.
6. Click to select the variable that contains the combined variables.
Format

In the Insert A Process dialog box, click on Group Variable and then on Command Format. Click on the to open the Variable Format dialog box. This dialog box enables you to format a variable with a smaller length than the required length.

![Variable Format Dialog Box](image)

Figure 4-25. Variable Format Dialog Box

- **Input Variable** selects the input variable to format.
- **Length** selects the new length of the variable.
- **Fill With** sets the open spaces between the required length and the variable length. Click on the appropriate radio button to set the fill type.
- **On the** sets the side of the variable that the fill characters will go on. Click on the appropriate radio button to select the side.
- **Result Variable** selects the variable that the value is assigned to. Click on to open the Variable dialog box to create a new variable or select an existing variable. Refer to Working with Variables on page 4-92 for information on variables.

- Click to accept the default setting
- Enter the appropriate information to change the values and click to apply the changes.
- Click to close the dialog box without saving the changes.
To format a variable:

1. Click on the *Input Variable* list box and select the input variable to format (e.g., V_CODE).
2. Using the up and down arrow button, select the new length for the variable (e.g., 10).
3. Click on *Zero* or *Blank* radio button to fill the open spaces with zeros or blanks.
4. Click on *Left* or *Right* radio button to place the open spaces on the left or the right of the variable.
5. Click to select a variable name to put the new format in.

**Substring**

In the *Insert A Process* dialog box, click on Group *Variable* and then on Command *Substring*. Click on to open the *Variable Substring* dialog box. This dialog box enables you to remove a certain number of characters from a variable.

![Variable Substring Dialog Box](image)

**Figure 4-26. Variable Substring Dialog Box**

- **Input Variable**
  - Selects the input variable to remove characters from.

- **Substring**
  - Selects the starting character and ending character to remove from the variable. Use the up and down arrow keys to select the character position.

- **Result Variable**
  - Selects the variable that the value is assigned to. Click on to open the *Variable* dialog box to create a new variable or select an existing variable. Refer to *Working with Variables* on page 4-92 for information on variables.
• Click to accept the default setting or
• Enter the appropriate information to change the values and click to apply the changes.
• Click to close the dialog box without saving the changes.

To remove characters from a variable:

1. Click on the Input Variable list box to select the variable to remove characters from (e.g., V_BARC).
2. Select From position and the To position (e.g., From position 1 to position 10).
3. Click to select the variable to save the string value.

Extract

In the Insert A Process dialog box, click on Group Variable and then on Command Extract. Click on to open the Variable Extract dialog box. This dialog box enables you to extract a specified part out of a variable.

![Variable Extract Window]

Figure 4-27. Variable Extract Window
**Input Variable**
Selects the input variable to extract a field from.

**Extract**
Selects the field number and the separator of the variable. Use the up and down arrow keys to select the field number.

**Result Variable**
Selects the variable that the value is assigned to. Click on the [ ] to open the Variable dialog box to select a variable. Refer to *Working with Variables* on page 4-92 for information on variables.

- Click [✓] to accept the default setting
  or
- Enter the appropriate information to change the values and click [✓] to apply the changes.
- Click [ ] to close the dialog box without saving the changes.

To extract a field from a variable:

1. Click on the **Input Variable** list box to select the variable to extract (e.g., V_BARC). Input variable is: 123,abc,456,def.
2. Select the field to extract (e.g., 2).
3. Select the separator between the fields (e.g., ,) Comma).
4. Click [ ] to select the variable name to put the extract in. The result variable contains abc.
Length

In the Insert A Process dialog box, click on Group Variable and then on Command Length. Click on ✓ to open the Variable Length dialog box. This dialog box enables you to count the length of a variable.

![Variable Length Dialog Box](image)

**Figure 4-28. Variable Length Dialog Box**

- **Variable to Measure** Selects the variable whose characters will be counted by this process.
- **Result Variable** Selects the variable that the value is assigned to. Click on ✓ to open the Variable dialog box to create a new variable or select an existing variable. Refer to Working with Variables on page 4-92 for information on variables.

- Click ✓ to accept the default setting or
- Enter the appropriate information to change the values and click ✓ to apply the changes.
- Click ✗ to close the dialog box without saving the changes.

To determine the length of a variable:

1. Click on the Variable to Measure list box to select the variable to count (e.g., V_PROD).
2. Click to select the variable to save the length value.
Arithmetic

In the Insert A Process dialog box, click on Group Variable and then click on Command Arithmetic. Click on \(\checkmark\) to open the Variable Arithmetic dialog box. This dialog box enables you to perform the following operations on a variable:

- add
- subtract
- multiply
- divide
- percent.

Figure 4-29. Variable Arithmetic Dialog Box

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value 1</td>
<td>Selects the first variable to perform the operation.</td>
</tr>
<tr>
<td>+, -, X, /, %</td>
<td>Sets the operation to perform on Value 1 variable.</td>
</tr>
<tr>
<td>Value 2</td>
<td>Selects the second variable to perform the operation.</td>
</tr>
<tr>
<td>Result Variable</td>
<td>Selects the variable that the value is assigned to. Click on (\checkmark) to open the Variable dialog box to create a new variable or select an existing variable. Refer to Working with Variables on page 4-92 for information on variables.</td>
</tr>
<tr>
<td>Integer</td>
<td>Sets the integer format for the resulting variable.</td>
</tr>
</tbody>
</table>
Designing Programs

Decimals
Sets the decimal format for the resulting variable.

Fill With
Fills the input field with zeroes or blanks. This is used when the input data is smaller than the required input.

Add Sign
Adds the sign to the result variable.

- Click ✔ to accept the default setting or
- Enter the appropriate information to change the values and click ✔ to apply the changes.
- Click ✗ to close the dialog box without saving the changes.

To perform an arithmetic operation on a variable:

1. Click on the Value 1 list box and select one of the two variables/values (e.g., V_QUAN).
2. Click on one of the operation radio buttons (+, -, X, / or %).
3. Click on the Value 2 list box and select the other variable/value (e.g., 12).
4. Click ✔ to select the variable to save the result.
5. Select the Integer and Decimal formats for the resulting variable.
Branch Group

Goto

In the Insert A Process dialog box, click on Group Branch and then click on Command Goto. Click on \( \boxed{\text{✓}} \) to open the Goto dialog box. This dialog box enables you to set a goto function:

- a jump to another screen
- a jump to a label (enter the label name directly)
- a jump to a specific process
- exit (or return to the main Menu).

![Goto Dialog Box](image)

**Figure 4-30. Goto Dialog Box**

- Click \( \boxed{\text{✓}} \) to accept the default setting
- Click \( \boxed{\text{✓}} \) to close the dialog box without saving the changes.
Key & Branch

In the Insert A Process dialog box, click on Group Branch and then click on Command Key & Branch. Click on ✓ to open the Key & Branch dialog box. This dialog box enables you to assign another function to several keys.

Figure 4-31. Key & Branch Dialog Box

- **Time Out Seconds** Assigns a time-out interval. If there is no activity after the time-out period, then the action or screen selected in the Goto list box takes effect.

- **Goto** Selects the action to perform or the screen to goto if the time-out occurs.

- **Other Keys** Assigns another function to keys on the scanner.

- **Key/Goto** Assigns another function to the Function keys on the scanner.

- Click ✓ to accept the default setting or

- Enter the appropriate information to change the values and click ✓ to apply the changes.

- Click ✗ to close the dialog box without saving the changes.
Test & Branch

In the Insert A Process dialog box, click on Group Branch and then click on Command Test & Branch. Click on ☑ to open the Test & Branch dialog box. This dialog box enables you to set up a test and, depending on the test result, branch to a specified screen.

![Test & Branch Dialog Box]

**Figure 4-32. Test & Branch Dialog Box**

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>If Value 1</td>
<td>Sets variable for test.</td>
</tr>
<tr>
<td>Test Type</td>
<td>Determines the operation to be performed.</td>
</tr>
<tr>
<td>Force Character Comparison</td>
<td>Forcing the comparison between Value 1 variable and Value 2 variable.</td>
</tr>
<tr>
<td>Then Goto</td>
<td>Sets the action to perform or the screen to go to if the test result is true.</td>
</tr>
<tr>
<td>Else Goto</td>
<td>Sets the action to perform or the screen to go to if the test result is false.</td>
</tr>
</tbody>
</table>

- Click ☑ to accept the default setting or
- Enter the appropriate information to change the values and click ☑ to apply the changes.
- Click ✗ to close the dialog box without saving the changes.
To test a variable and branch upon a condition:

1. Click on the *If Value 1* list box to select the variable to compare (e.g., V_PRICE).
2. Click on a *Test Type* radio button. Available choices are: equal to (=), less than (<), and greater than (>).
3. Click on the *Value 2* list box to select the variable to compare (e.g., 120).
4. Click on the *Then Goto* list box to select a screen/action to go to if the comparison is true.
5. Click on the *Else Goto* list box to select a screen/action to go to if the comparison is false.

**Case & Branch**

In the *Insert A Process* dialog box, click on Group *Branch* and then click on Command *Case & Branch*. Click on ‟ to open the *Case & Branch* dialog box. This dialog box enables you to compare the value of a variable input with other variables or values.

![Case & Branch Window](image)

**Figure 4-33. Case & Branch Window**
**Case Value**
Sets variable for testing against the five *Equal* variables.

**Equal**
Sets the variables or value to compare to *Case Value*. Up to five variables or values can be compared to *Case Value* variable.

**Goto**
Sets the action to perform or screen to go to if the test result are true.

**Otherwise Goto**
Sets the action to perform or screen to go to if the test results is false.

- Click ✓ to accept the default setting or
- Enter the appropriate information to change the values and click ✓ to apply the changes.
- Click ✗ to close the dialog box without saving the changes.

The value of variable *Case Value* is compared with the *Equal* variables or values. When the value of the *Case Value* equals one of these set *Equal* values, the action or screen in the *Goto* list box is performed. If the comparison is false, the action or screen in the *Otherwise Goto* list box is performed.

**Control & Branch**

In the *Insert A Process* dialog box, click on Group *Branch* and then click on Command *Control & Branch*. Click on ✓ to open the *Control & Branch* dialog box. This dialog box enables you to compare a character pattern of a variable.

![Control & Branch Window](image)

**Figure 4-34. Control & Branch Window**
If Value

Follows Pattern

Sets the pattern to compare to the If Value. This pattern can exist out of:

- Letters A to Z to specify a specific letter.
- Numbers 0 to 9 to specify a specific number.
- $ to specify numeric characters.
- " to specify text.
- ? to specify any character
- * to specify all other characters.

Then go to

Sets the action to perform or the screen to go to if the test result is true.

Else Goto

Sets the action to perform or the screen to go to if the test result is false.

- Click ✅ to accept the default setting or
- Enter the appropriate information to change the values and click ✅ to apply the changes.
- Click ❌ to close the dialog box without saving the changes.

For example:

B$$"?4* matches with B21C14ABC123.

If the comparison is true, the process follows the screen or action in the Then Goto list box. If the comparison is false, the process follows the screen or action in the Else Goto list box.
Set Label

In the Insert A Process dialog box, click on Group Branch and then click on Command Set Label. Click on ✓ to open the Label dialog box. This dialog box enables you to set a label. A label is a reference that can be set to go directly to a certain position, like a specific line in a process. Enter a label name, and it always refers to that position.

![Label Dialog Box](image)

**Figure 4-35. Label Dialog Box**

- **Label** Sets the label name.

---

**Note:** The label name must be unique for the program and must begin with an alphabetic character (A to Z).

- Click ✓ to accept the default setting or
- Enter the appropriate information to change the values and click ✓ to apply the changes.
- Click ✗ to close the dialog box without saving the changes.
Figure 4-36 illustrates the use of a label in a process.

**Figure 4-36. Label Process**

**Gosub Window**

In the *Insert A Process* dialog box, click on Group *Branch* and then click on Command *Gosub*. Click on ✔️ to open the *Gosub* dialog box. This dialog box enables you to go to a subroutine and then return.

**Figure 4-37. Gosub Dialog Box**

- **Program**: Sets the program to go to.
- **Gosub**: Selects the screen of the program to go to.
Figure 4-38 illustrates the use of a Gosub in a process.

1. Start from line 3. Enter Gosub S20-Screen 20.
2. MCL-Designer goes to process line 20 to undertake a certain action.
3. When the action is finished, the exit enables the process to turn back to line 4.

---

**Note:** The Gosub can go to a label, a screen or another program.

**Always use Gosub with an exit. Refer to Goto on page 4-40.**
Files Group

Files store the data input on a permanent basis in the scanner. Due to the interaction between the files and programs through processes, MCL-Designer enables you to read, write and delete the data in a file. Refer to Files Window on page 4-87 for more information on working with files.

Write a Record

In the Insert A Process dialog box, click on Group Files and then click on Command Write a Record. Click on to open the File Write dialog box. This dialog box enables you to write the input of variables into the fields of a file.

Figure 4-39. Write File Dialog Box

- **File Name**: Selects the file to write the fields to.
- **If Write Error Goto**: Selects the action to perform or screen to go to if a write error occurs. A write error can occur when there is not enough memory to write a record or when a problem is detected on the file itself.
Fields To Write

Field Name  Shows the fields in the selected file. A file can contain up to 16 fields.

Length    Shows the length of the fields in the selected file.

Value     Sets the variable or value to the field. Click on the list box to select the variable or enter a value in the box.

- Click ✔️ to accept the default setting
  or
- Enter the appropriate information to change the values and click ✔️ to apply the changes.
- Click ❌ to close the dialog box without saving the changes.

To write variables into the fields of a file:

1. Click on the Name list box to select the file to write to (e.g., <A>FILE). The dialog box displays the fields with their lengths. Refer to Files Window on page 4-87.
2. Click on the Value list box to select the variable or value required to write the value or content (of the variable) to the selected field: (e.g., V_PROD to field 1, V_QUAN to field 2, V_BARC to field 3.)

Note: Click 🌐 to go to the Data File Window. For more information, refer to Working With Files on page 4-87.
Read a Record

In the Insert A Process dialog box, click on Group Files and then click on Command Read a Record. Click on ✓ to open the File Read dialog box. This dialog box enables you to read the fields of a record in a file and assign the fields content to a variable.

**Figure 4-40. File Read Dialog Box**

**File Name**
Selects the file to read a record from.

**If Record Not Found**
Sets the action to perform or the screen to go to if the record is not found.

**Search Record**

**Mode**
Selects the search mode:
- First: first record
- Last: last record
- Next: next record
- Previous: previous record
- Current: current record
- Key: If the file is sorted, sets a key to search the record
- Record #: If you know the record number in the file, search using Key 1 list box (= record number).
Key1/Key2 Selects a variable to search for when Key is selected in the Mode list box. Enter a record number when Record# is selected in the Mode list box.

Fields To Read
Field Name Shows the fields in the selected file.
Length Shows the length of the fields in the selected file.
Variable to Update Selects the variable to be assigned to the field. Click on to open the Variable dialog box to create a new variable or select an existing variable. Refer to Working with Variables on page 4-92 for information on variables.

- Click to accept the default setting
- Enter the appropriate information to change the values and click to apply the changes.
- Click to close the dialog box without saving the changes.

To read fields from a file and store them into variables:

1. Click on the Name list box to select the file in which to read the record (e.g. <A>FILE).
2. Click on the Goto list box to select the screen to goto if the record is not found.
3. Click on the Mode list box to select the mode to search. If Key or Record# is selected, select the appropriate variables in the Key list boxes.
4. Click to select the variables to store the input in.
**Delete a Record**

In the *Insert A Process* dialog box, click on Group *Files* and then click on Command *Delete a Record*. Click on ✓ to open the *File Delete* dialog box. This dialog box enables you to delete a record in a file.

![File Delete Dialog Box](image)

**Figure 4-41. File Delete Dialog Box**

- **File Name**: Selects the file to delete a record from.
- **If Record Not Found Goto**: Sets the action to perform or the screen to go to if the record is not found.
- **Search Record**
  - **Mode**: Selects the search mode:
    - First: first record
    - Last: last record
    - Next: next record
    - Previous: previous record
    - Current: current record
  - **Key**: If the file is sorted, you can set a key to search the record
  - **Record #**: If you know the record number in the file, you can search using *Key 1* (= record number).
- **Key1/Key2**: Selects a variable to search for when *Key* is selected in the *Mode* list box. Enter a record number when *Record#* is selected in the *Mode* list box.
To delete a record from a file:

1. Click on the File Name list box to select the file that holds the record that needs to be deleted (e.g., <A> FILE).
2. Click on the Mode list box to select the search mode (e.g., Record V_PROD).

In order to prevent an excessive number of writes to a single block of flash memory, the Phaser does not support record updates in place. Modification of information previously written to a record is accomplished by a delete-and-append operation in the following way:

- perform a “Files: Read a Record” process, storing to temporary variables any field values that are not being changed,
- perform a “Files: Delete a Record” process to delete the incorrect record,
- perform a “Files: Write a Record” process to create a new record with the correct data.

**Review File**

In the Insert A Process dialog box, click on Group Files and then click on Command Review a File. Click on to open the File Review dialog box. This dialog box enables you to view a file on the screen (browse records using the up and down keys).

![Figure 4-42. File Review Dialog Box](image)
To review a file on the screen:

1. Click on the *File Name* list box and select the file to review (e.g., `<A>`Inventory).
2. Click in the appropriate *View* check box to select the fields to review.
3. In the *Field Title* text box, enter a title for that field. The default value is Field X, where X is the field number.
4. Click the [Options] to open the File Review 2 dialog box.

**Figure 4-43. File Review 2 Dialog Box**

- **Start Review On..** Selects the field to start the review on. Click on the appropriate radio button to change the setting.

- **Fields Presentation** Sets the number of lines for the display of the file review. The file review can contain a maximum of 20 characters for each line. Therefore select:
  - 1 Line/Field if the length of the field name is less than 20.
  - Auto to arrange the view according to the length of the field name and its value.
  - Space Separator to insert a space to separate fields.

- **Allowed Functions** Sets keys to delete and search for a specific record in the file. Click the **Delete Current Record** check box to allow the user to select the key to delete a record. Click the **Search a Record** check box to allow the user to select the key to search for a record.
Erase File

In the Insert A Process dialog box, click on Group Files and then click on Command Erase a File. Click on  to open the File Erase dialog box. This dialog box enables you to erase an entire file.

Figure 4-44. File Erase Dialog Box

File Name Selects the name of the file to erase.

Note: Click  to go to the Data File Window. For more information, refer to Working With Files on page 4-87.

• Click  to accept the default setting or
• Enter the appropriate information to change the values and click  to apply the changes.
• Click  to close the dialog box without saving the changes.
Seek

In the Insert A Process dialog box, click on Group Files and then click on Command Seek. Click on \( \checkmark \) to open the File Seek dialog box. This dialog box enables you to look at a specific record in a file without reading it.

![File Seek Dialog Box](image)

**Figure 4-45. File Seek Dialog Box**

- **File Name**
  - Selects the file to view.

- **If Record Not Found Goto**
  - Sets the action to perform or the screen to go to if the record is not found.

**Search Record**

- **Mode**
  - Selects the search mode:
    - First: first record
    - Last: last record
    - Next: next record
    - Previous: previous record
    - Current: current record
    - Key: If the file is sorted, you can set a key to search the record

- **Key1/Key2**
  - Selects a variable to search for when Key is selected in the Mode list box.

- **Store Record number into Variable**
  - Writes the record number into a variable. Click on \( \text{\ldots} \) to open the Variable dialog box to create a new variable, or select an existing variable. Refer to Working with Variables on page 4-92 for information on variables.
Note: Click  to go to the Data File Window. For more information, refer to Working With Files on page 4-87.

- Click  to accept the default setting or
- Enter the appropriate information to change the values and click  to apply the changes.
- Click  to close the dialog box without saving the changes.

To look at a record in a file:

1. Click on the File Name list box to select a file (e.g. <A>FILE).
2. Click on the Mode list box to select the search mode.
3. Click on the Store Record number into Variable list box to store the record number of the selected record into a variable.
4. Click  to select the variable.

Local File Check

In the Insert A Process dialog box, click on Group Files and then click on Command Local File Check. Click on  to open the Local File Check dialog box. This dialog box enables you to check if a local file (in the scanner) exists.

Figure 4-46. Local File Check Dialog Box
**Local Filename**

Selects one of eight files to check.

**If File Empty Goto**

Selects the action to perform or the screen to go to when the file is empty.

**If File Not Empty Goto**

Selects the action to perform or the screen to go to when the file is not empty.

---

**Note:** *Click [ ] to go to the Data File Window. For more information, refer to Working With Files on page 4-87.*

---

- Click [ ] to accept the default setting
  or

- Enter the appropriate information to change the values and click [ ] to apply the changes.

- Click [ ] to close the dialog box without saving the changes.
**MCL-Link Communications Group**

**Send Packet**

In the *Insert A Process* dialog box, click on Group *MCL-Link Comm.* and then click on Command *Send Packet*. Click on ✅ to open the *Send Packet to Host* dialog box. This dialog box enables you to send a packet of variable values to the host computer.

![Send Packet To Host Dialog Box](image)

**Figure 4-47. Send Packet To Host Dialog Box**

**Send Packet To**

- **Host ID**
  - Selects the host identifier to send the packet to.

- **Packet Type**
  - Selects a name for the packet.

- **Fields Sep.**
  - Selects a field separator for the packet.

- **If Communication Error**
  - **Goto**
    - Selects the action if an error occurs during the sending.

- **Packet Fields**
  - Selects the variable or fixed data to send. Click on the list box to select a variable or enter a value into the list box. Use scroll bar to view additional packet fields.
• Click ✓ to accept the default setting
  or
• Enter the appropriate information to change the values and click ✓ to apply the changes.
• Click ❌ to close the dialog box without saving the changes.

Receive Packet
In the Insert A Process dialog box, click on Group MCL-Link Comm. and then click on Command Receive Packet. Click on ✓ to open the Receive Packet dialog box. This dialog box enables you to select a packet with variable values to work with.

![Receive Packet Dialog Box]

**Figure 4-48. Receive Packet Dialog Box**

*Receive Packet From*
- **Host ID**
  Selects the host identifier to receive the packet from.
- **Packet Type**
  Selects a name for the packet.
- **Fields Sep.**
  Selects a field separator for the packet.
- **If Communication Error**
  Selects the action to perform or the screen to go to if an error occurs while receiving the packet.

- **Goto**
  Selects the action to perform or the screen to go to if an error occurs while receiving the packet.
Packet Fields to Variables Assigns a variable to each received field. Click on to open the Variable dialog box to create a new variable or select an existing variable. Refer to Working with Variables on page 4-92 for information on variables. Use scroll bar to view additional packet fields.

- Click to accept the default setting or
- Enter the appropriate information to change the values and click to apply the changes.
- Click to close the dialog box without saving the changes.

Send File
In the Insert A Process dialog box, click on Group MCL-Link Comm. and then click on Command Send File. Click on to open the Send File dialog box. This dialog box enables you to send an entire file to a scanner.

![Send File Dialog Box]

Figure 4-49. Send File Dialog Box
Send File To

- Host ID: Selects the host identifier to send the file to.
- Local File Name: Selects the local file to send to the host.
- Remote File Name: Accepts a new file name for the host application (e.g., FILESEND or C:\DATA\FILESEND).

Options

- Check Cradle: Sets the scanner to check for a connection to a cradle.
- Delete After Send: Sets the scanner to delete the file after it is sent to the host.
- Toolbar On Bottom Line: Sets a toolbar to appear on the bottom of the screen when reviewing the file. Not supported on Phaser.
- If Communication Error: Selects the action to perform or the screen to go to if an error occurs while sending a file.

Note: Click 🗣 to go to the Data File Window. For more information, refer to Working With Files on page 4-87.

- Click ✔️ to accept the default setting or
- Enter the appropriate information to change the values and click ✔️ to apply the changes.
- Click ✗ to close the dialog box without saving the changes.
Receive File

In the Insert A Process dialog box, click on Group MCL-Link Comm. and then click on Command Receive File. Click on to open the Receive File dialog box. This dialog box enables you to ask the host to send an entire file to the scanner.

![Receive File Dialog Box]

**Figure 4-50. Receive File Dialog Box**

**Receive File From**

- **Host ID**
  
  Selects the host identifier to receive the file from.

- **Remote File Name**
  
  Accepts the file name for the host application (e.g., FILESEND or C:\USER\FILERCV).

- **Local File Name**
  
  Selects the local file name.

**Options**

- **Check Cradle**
  
  Sets the scanner to check for a connection to a cradle.

- **Toolbar On Bottom Line**
  
  Sets a toolbar to appear on the bottom of the screen when reviewing the file. Not supported on Phaser.

**If Communication Error**

- **Goto**
  
  Selects the action to perform or the screen to go to if an error occurs while receiving the file.
Note: Click to go to the Data File Window. For more information, refer to Working With Files on page 4-87.

- Click to accept the default setting
- Enter the appropriate information to change the values and click to apply the changes.
- Click to close the dialog box without saving the changes.

Host Look-up

In the Insert A Process dialog box, click on Group MCL-Link Comm. then click on Command Host Look-Up. Click to open the Host Look-Up dialog box. This dialog box enables you to look at a remote file on the host. (This file must be “text” file in ASCII format.)

![Host Look-Up Window](image)

**Figure 4-51. Host Look-Up Window**

*Host Look-Up*

*Host ID*

Selects the host identifier to look up the file.
### File Name
Accepts the file name for the host application (e.g., FILESEND or C:\USER\FILERCV).

### Search With
Selects a variable containing the search key.

### If Key Not Found Goto
Selects the action to perform or the screen to go to if the key is not found.

### If Communication Error Goto
Selects the action to perform or the screen to go to if an error occurs during the communication.

### Received Fields
Assigns the variables to store the fields from the remote file. Click on to open the Variable dialog box to create a new variable or select an existing variable. Refer to Working with Variables on page 4-92 for information on variables.

### Fields Sep.
Selects a separator between the fields.

- Click ✓ to accept the default setting or
- Enter the appropriate information to change the values and click ✓ to apply the changes.
- Click ✗ to close the dialog box without saving the changes.

---

**Note:** On the host side, the file must have a structure similar to:

"123456789,abcdef,hello world" as one record with a comma (,) as a separator.

The key to search must be the first field:"123456789".

The returned fields are abcdef and hello world.

---

### MCL-Link Start
In the Insert A Process dialog box, click on Group MCL-Link Comm. and then click on Command MCL-Link Start. This command runs the communication program on the scanner in Wait mode. The scanner waits for a command from the host. To exit this mode and continue the program, click on the Cancel key.
Remote File Check

In the Insert A Process dialog box, click on Group MCL-Link Comm. and then click on Command Remote File Check. Click on to open the Remote File Check dialog box. This dialog box enables you to check if a file exists on a remote host and get the file size and file date.

![Remote File Check Dialog Box](image)

**Figure 4-52. Remote File Check Dialog Box**

- **Host ID**
  Selects the host identifier to look up the file.

- **File Name**
  Accepts the file name for the host application (e.g., FILESEND or C:\USER\FILERCV).

- **If File Exists Goto**
  Selects the action to perform or the screen to go to if the file is found.

- **If File Does Not Exist Goto**
  Selects the action to perform or the screen to go to if the file is not found.
Store Remote File Info

File Size (Bytes) Selects the variable to store the file size information in. Click on \( \text{\textasciitilde} \) to open the Variable dialog box to create a new variable or select an existing variable. Refer to Working with Variables on page 4-92 for information on variables.

File Date+Time Selects a variable to store the “make” date and time of the file. Click on \( \text{\textasciitilde} \) to open the Variable dialog box to create a new variable or select an existing variable. Refer to Working with Variables on page 4-92 for information on variables.

If Communication Error Goto Selects the action to perform or the screen to go to if an error occurs during the communication.

- Click \( \checkmark \) to accept the default setting
- Enter the appropriate information to change the values and click \( \checkmark \) to apply the changes.
- Click \( \times \) to close the dialog box without saving the changes.

**Start EXE Program on Host**

In the Insert A Process dialog box, click on Group MCL-Link Comm. and then click on Command Start EXE Program On Host. Click on \( \checkmark \) to open the Start EXE Program On Host dialog box. This dialog box enables you to start a program (with extension EXE) on the host.
**Figure 4-53. Start EXE Program On Host Dialog Box**

- **Host ID**
  Selects the host identifier to run the executable.

- **Remote Program Name**
  Accepts the program name to run.

- **Argument Sep.**
  Selects a separator to use between the arguments.
  Example:
  INVENTORY.EXE 123123123 050
  where the arguments are: 123123123 and 050 and the separator is a blank.

- **If Communication Error Goto**
  Selects the action to perform or the screen to go to if an error occurs during the communication.

- **Program Arguments**
  Selects the arguments (or variables) to give to the program.

- **Click** [✓] to accept the default setting
  or

- **Enter** the appropriate information to change the values and **click** [✓] to apply the changes.

- **Click** [✗] to close the dialog box without saving the changes.
**Modem Command**

In the *Insert A Process* dialog box, click on Group *MCL-Link Comm.* and then click on Command *Modem Command*. Click on  to open the *Modem Command* dialog box. This dialog box enables you to send a command to modem.

![Modem Command Dialog Box](image)

**Figure 4-54. Modem Command Dialog Box**

- **Select Command**
  - *Init.* Enter the initialization command for the modem.
  - *Dial* Enter the telephone number to dial.
  - *Disconnect* Enter the disconnect command.
  - *Auto Answer* Enter the Auto Answer Hayes Command to put the modem in auto answer mode.
  - *Other* Enter any other modem command.

Select Command
If Error Goto

Selects the action to perform or the screen to go to if an error occurs. Possible errors are: no answer from modem or no connection with the remote site.

Options

Check Cradle

Sets the scanner to check for a connection to a cradle.

Toolbar On Bottom Line

Sets a toolbar to appear on the bottom of the screen when reviewing the file. Not supported on Phaser.

• Click ✔ to accept the default setting
  or

• Enter the appropriate information to change the values and click ✔ to apply the changes.

• Click ✗ to close the dialog box without saving the changes.
Serial Communications Group

String Serial Output

In the Insert A Process dialog box, click on Group Serial Comm and then click on Command String Serial Output. Click on to open the String Serial Output dialog box. This dialog box enables you to select a variable to send through the serial port to a serial output device.

Figure 4-55. String Serial Output Dialog Box

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>Sets the STX or SOH character to the beginning of a variable.</td>
</tr>
<tr>
<td></td>
<td>STX: Start of Text (02 ASCII).</td>
</tr>
<tr>
<td></td>
<td>SOH: Start Output Header</td>
</tr>
<tr>
<td>Data/Variable to Send</td>
<td>Selects the variable or data to send. Click on the list box to select a</td>
</tr>
<tr>
<td></td>
<td>variable or enter a value in the list box.</td>
</tr>
<tr>
<td>Footer</td>
<td>Sets the ETX, CR or CR/LF character to the end of the variable.</td>
</tr>
<tr>
<td></td>
<td>ETX (End of Text) - 03 ASCII.</td>
</tr>
<tr>
<td></td>
<td>CR (Carriage Return) - hard return.</td>
</tr>
<tr>
<td></td>
<td>LF (Line Feed) - a new line (13 and 10 ASCII).</td>
</tr>
<tr>
<td>If Communication Error Goto</td>
<td>Selects the action to perform or screen to go to if an error occurs.</td>
</tr>
</tbody>
</table>
Click \( \checkmark \) to accept the default setting or Enter the appropriate information to change the values and click \( \checkmark \) to apply the changes. Click \( \times \) to close the dialog box without saving the changes.

**Scanner Serial Output**

In the *Insert A Process* dialog box, click on Group *Serial Comm.* and then click on Command *Scanner Serial Output*. Click on \( \checkmark \) to open the *Scanner Serial Output* dialog box. This dialog box enables you to select a variable to send through the serial port to a scanner output device.

![Scanner Serial Output Dialog Box](image)

*Figure 4-56. Scanner Serial Output Dialog Box*
Designing Programs

Add Barcode
Sets the default or specific descriptor to be added to the barcode.

Descriptor
Click on the list box to select a descriptor or enter a value in the list box.

Data/Variable to Send
Selects the variable or data to send. Click on the list box to select a variable or enter a value in the list box.

Add Key Code
Sets the key code to be sent. Click on the list box to select a code or enter a value in the list box.

If Communication Error Goto
Selects the action to perform or screen to go to if an error occurs.

• Click \( \checkmark \) to accept the default setting or
• Enter the appropriate information to change the values and click \( \checkmark \) to apply the changes.
• Click \( \times \) to close the dialog box without saving the changes.

Receive Data
In the Insert A Process dialog box, click on Group Serial Comm. and then click on Command Receive Data. Click on \( \checkmark \) to open the Serial Input dialog box. This dialog box enables you to receive data from the auxiliary port when specific characters are received.

![Figure 4-57. Serial Input Dialog Box](image-url)
Start Character  Selects the start character. Click on the appropriate radio button to select the start character.

None: Do not wait for start character.

STX: Start of Text (02 ASCII).

Other: Enter any character in the text box provided.

Stop Char.  Selects stop character. Click on the appropriate radio button to select the stop character. Data input will stop when the selected character is received.

ETX: End of Text.

CR: Carriage Return.

Other: Enter any character in the text box provided.

Length: Ends data input when a selected number of character is reached.

Result Variable  Selects the variable that the value is assigned to. Click on $\Rightarrow$ to open the Variable dialog box to create a new variable or select an existing variable. Refer to Working with Variables on page 4-92 for information on variables.

Time Out/Clear Key  Selects an action to be performed or a screen to go to if the set time interval (seconds) is exceeded.

- Click $\checkmark$ to accept the default setting or
- Enter the appropriate information to change the values and click $\checkmark$ to apply the changes.
- Click $\times$ to close the dialog box without saving the changes.
Print File

In the Insert A Process dialog box, click on Group Serial Comm. and then click on Command Print File. Click on ✅ to open the Print File dialog box. This dialog box enables you to print a file.

![Print File Dialog Box](image)

Figure 4-58. Print File Dialog Box

- **Print File**
  - **File** 
    - Selects the file to print.
- **Replace Variables**
  - Replaces all the variable codes found in file with the actual values of these variables.
- **If Error Goto**
  - Selects the action to perform or the screen to go to if an error occurs.

**Note:** Click 🚗 to go to the Data File Window. For more information, refer to Working With Files on page 4-87.

- Click ✅ to accept the default setting or
- Enter the appropriate information to change the values and click ✅ to apply the changes.
- Click ✗ to close the dialog box without saving the changes.
**Others Group**

**Pause**

In the *Insert A Process* dialog box, click on Group *Other* and then click on Command *Pause*. Click on ✔️ to open the *Pause* dialog box. This dialog box enables you to add a pause interval.

![Pause Dialog Box](image)

**Figure 4-59. Pause Dialog Box**

- **Seconds**
  Sets the pause interval. Use the up and down arrow keys to select the pause interval.

- **ShortCut Delay**
  Sets the shortcut delay interval. Click on the check box to enable shortcut delay.

- Click ✔️ to accept the default setting or
- Enter the appropriate information to change the values and click ✔️ to apply the changes.
- Click ❌ to close the dialog box without saving the changes.
Beep

In the Insert A Process dialog box, click on Group Other and then click on Command Beep. Click on \( \checkmark \) to open the Beep dialog box. This dialog box enables you to insert a beep.

![Beep Dialog Box](image)

**Figure 4-60. Beep Dialog Box**

**Beep Type**

Selects the type of beep tone to be inserted. Click on the arrow box to select the combination of beeps desired. Click \( \checkmark \) to test the beep option selected.

- Click \( \checkmark \) to accept the default setting
- Enter the appropriate information to change the values and click \( \checkmark \) to apply the changes.
- Click \( \times \) to close the dialog box without saving the changes.
Display Data

In the Insert A Process dialog box, click on Group Other and then click on Command Display Data. Click on ✔ to open the Display Data dialog box. This dialog box enables you to display data on the scanner.

![Display Data Dialog Box]

**Figure 4-61. Display Data Dialog Box**

- **Line**: Sets the display data to appear on 1 or 2 lines. Click on the up or down arrow to change the number of lines.

- **Column**: Sets the display data column width (number of characters per line - maximum 20). Click on the up or down arrow to change the number of columns.

- **Video Mode**: Sets the text and background colors. Click on the Normal radio button to display the variable in black on a white background or click on the Reverse radio button to display the variable in white on a black background.

- **Data**: Sets the type of data to appear. Click on the arrow box to select the data type.
Warning Message

In the *Insert A Process* dialog box, click on Group *Other* and then click on Command *Warning Message*. Click on ✔️ to open the *Warning Message* dialog box. This dialog box enables you to insert a 2-line message without using a full screen.

**Figure 4-62. Warning Message Dialog Box**

<table>
<thead>
<tr>
<th><strong>Warning Message</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Line 1/2</td>
<td>Enter the text that displays as a warning.</td>
</tr>
<tr>
<td>Beep</td>
<td>Sets a beep to sound when the warning screen displays. Click on the arrow box to select the combination of beeps desired. Click ✔️ to test the beep option selected.</td>
</tr>
<tr>
<td>Pause</td>
<td>Sets a pause to occur when the warning screen display. Set Enter to Continue check box to enable to operator to continue after the Enter key is pressed.</td>
</tr>
</tbody>
</table>

---

**Note:** *The current screen is cleared before displaying the message.*

- Click ✔️ to accept the default setting or
- Enter the appropriate information to change the values and click ✔️ to apply the changes.
- Click ✗ to close the dialog box without saving the changes.
Clear Screen

In the Insert A Process dialog box, click on Group Other and then click on Command Clear Screen. Click on to open the Clear Screen dialog box. This dialog box enables you to clear the whole screen or just selected lines of the screen.

![Figure 4-63. Clear Screen Dialog Box](image)

Line(s) To Clear

<table>
<thead>
<tr>
<th>Line 1</th>
<th>Sets the specific line to clear.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line 2</td>
<td></td>
</tr>
</tbody>
</table>

- Click to accept the default setting
  or
- Enter the appropriate information to change the values and click to apply the changes.
- Click to close the dialog box without saving the changes.
MCL-Code

In the *Insert A Process* dialog box, click on Group *Other* and then click on Command *MCL Code*. Click on ✔ to open the *MCL Code* dialog box. This dialog box enables you to insert a process line directly coded in MCL code. Refer to Chapter 7 for more information on MCL-Code.

![MCL Code Dialog Box](image)

**Figure 4-64. MCL-Code Dialog Box**

---

**Note:** Click ✔ to go to the Variables window. For more information, refer to *Global View of Variables* on page 4-93.

Comment

In the *Insert A Process* dialog box, click on Group *Other* and then click on Command *Comment*. Click on ✔ to open the *Comment* dialog box. This dialog box enables you to insert a comment.

![Comment Dialog Box](image)

**Figure 4-65. Comment Dialog Box**
Printing Reports

Changing the Printer Setup
To change the printer setup:

1. Choose Print Setup from the File menu in the MCL Designer window to display the standard Windows printer setup window.
2. Change the settings.
3. Click ☑ to accept the changes.

Printing a Screen
To print the design:

1. Choose Print from the File menu.
2. The current screen is sent to a printer connected to your computer.

Printing Options
To specify the parts of a project to print, click ☑ to open the Print dialog box.

Figure 4-66. Print Dialog Box
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Info</td>
<td>Sets the print report to print the current project.</td>
</tr>
<tr>
<td>Project Description</td>
<td>Enables the print report to contain a description of the project.</td>
</tr>
<tr>
<td>Phaser Setup</td>
<td>Enables the print report to contain the Phaser setup information.</td>
</tr>
<tr>
<td>Programs Info</td>
<td>Sets the print report to print program information.</td>
</tr>
<tr>
<td>Programs</td>
<td>Enables all of the programs or one of the programs to be included in the print report.</td>
</tr>
<tr>
<td>Screen Pictures</td>
<td>Enables pictures of each screen to be included in the print report.</td>
</tr>
<tr>
<td>Program Details</td>
<td>Enables program details to be included in the print report.</td>
</tr>
<tr>
<td>Variables List</td>
<td>Enables the variable list to be included in the print report.</td>
</tr>
<tr>
<td>MCL Code</td>
<td>Enables the MCL Code to be included in the print report.</td>
</tr>
<tr>
<td>Print To</td>
<td>Selects where to print to.</td>
</tr>
<tr>
<td></td>
<td>Click on the Screen radio button to print the project to the screen.</td>
</tr>
<tr>
<td></td>
<td>A window appears and the project display on the computer’s screen.</td>
</tr>
<tr>
<td></td>
<td>Click the Printer radio button to print to a printer connected to the computer. The Printer button opens the standard Windows printer setup dialog box for changing the printer or setting for the printer.</td>
</tr>
<tr>
<td>Data Files Info</td>
<td>Sets the print report to print data file information.</td>
</tr>
<tr>
<td>Data Files</td>
<td>Enables all of the data files or one of the data files of the project to be printed.</td>
</tr>
<tr>
<td>File Structure</td>
<td>Enables the print report to contain the structure of the data file.</td>
</tr>
<tr>
<td>File Content</td>
<td>Enables the print report to contain the contents of the data file.</td>
</tr>
</tbody>
</table>
• Click ☑️ to accept the default setting or
• Enter the appropriate information to change the values and click ☑️ to apply the changes.
• Click ✗ to close the dialog box without saving the changes.
Working With Files

Files store the input data on a permanent basis in the scanner. Due to the interaction between the files and programs through processes, MCL-Designer enables you to read, write, update and delete the data in a file.

You can define a maximum of 8 files in the scanner. Each program of the project can access any of these data files. Data files are fixed record lengths except for the Print file. Up to 16 fields can be stored in a record. The maximum record length is 255 bytes, the maximum field length is 128 bytes and the maximum file size depends on the memory of the scanner (up to 3.2 MB). The default field separator is the pipe (|) (124 decimal). Refer to System Setup on page 1-7.

Files Window

In the MCL Designer window, click on the Files tab.

Figure 4-67. MCL Designer Files Tab

1. Click next to the file number. Up to eight files can be created for a project.
2. If this is a new file, the Create Data File dialog box appears.
3. Enter a file name in the Data File Name field.
4. Click to accept the file name. The Data File Description dialog box appears.
Figure 4-68. Data File Description Dialog Box

File Name  Shows the name of the current file.
Type of File  Selects the type of file.

**Capture:** Creates the file and fills it by the scanner using the *Write a Record* command. Refer to *Write a Record* on page 4-49. This file is used to store data captured on the scanner.

**Look up:** Use the file for search and verification. It is normally a read-only file coming from the host, but it can also be created locally on the scanner.

**Print:** This is typical a read-only file with no specific field organization. It is a variable record length file.

Sort on  Selects the type of sort.

**None:** Each record is opened at the end of the file (typical capture file).

**Field 1:** The file will be sorted on the first field in ascending order. This option is used for a fast search on field 1.
Fields 1+2: This option makes it possible to sort on the first field and the second field in ascending order. This option is used for a fast search on both fields.

Note: The search function is not allowed on field 2.

Allocated Memory
Shows the amount of memory allocated to the data files, and how many records exist.

Fields Description
Field Name
Shows and accepts the name of the field.
Length
Shows and accepts the length of the field.

• Click to accept the default setting or
• Enter the appropriate information to change the values and click to apply the changes.
• Click to close the dialog box without saving the changes.

Note: Click to go to the Memory Mapping dialog box. For more information, refer to Memory Mapping on page 1-8.

The Look up file with sorting on field 1 or field 1+2 is used for a fast read-only search operation. The Capture file, sorted or non-sorted, is used to store input data and usually consists of small files (<1000 records). A search function in a sorted file is faster, but a write function is slower. A write function in a non-sorted file is faster, but a search function is slower.

Note: A look up file defined as “sorted” must be sorted on the host prior to being sent to the scanner.
1. Click  to open the *File View* dialog box.

![File View Dialog Box](image)

*Figure 4-69. File View Dialog Box*

This window shows the contents of the file on the hard disk. In this case, Field 1, Field 2, and Field 3 are the 3 field names of the records, and the numbers in parenthesis are the lengths.

**Table 4-5. Button Descriptions**

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Add" /></td>
<td>Add</td>
<td>Adds a new record.</td>
</tr>
<tr>
<td><img src="image" alt="Cut" /></td>
<td>Cut</td>
<td>Deletes a record.</td>
</tr>
<tr>
<td><img src="image" alt="Copy" /></td>
<td>Copy</td>
<td>Copies a record.</td>
</tr>
<tr>
<td><img src="image" alt="Paste" /></td>
<td>Paste</td>
<td>Pastes a copied record.</td>
</tr>
<tr>
<td><img src="image" alt="Duplicate" /></td>
<td>Duplicate</td>
<td>Duplicates a selected record.</td>
</tr>
<tr>
<td><img src="image" alt="Sort" /></td>
<td>Sort</td>
<td>Sorts the records.</td>
</tr>
</tbody>
</table>
• Click ☑ to accept the default setting or
• Enter the appropriate information to change the values and click ☑ to apply the changes.
• Click ✗ to close the dialog box without saving the changes.
Working with Variables

A variable is a memory area in the scanner, where data is stored temporarily. There can be up to 100 variables of 64 bytes each, plus three 512 byte variables. All variables are global for one project. All 8 programs can access (read/write) the variables. The content of a variable is an ASCII character string that can be displayed on the scanner screen at any moment.

A variable is never empty if no data is stored in that variable. The content will be the “?” character. Internally, the variables are named A0 to A3,...Y0 to Y3, Z0 to Z2. A comprehensive name can be entered in the MCL-Designer variable table.

**Note:** All the variables are initialized to “?” after a scanner boot.

It is possible to specify a variable content using “&” and the variable code in any data fields.

Figure 4-70. Variable Content Using “&”

123&A0: will display 123ABC if ABC is the content of the variable A0.
Global View of Variables

The Variable List icon in the Program Designer window displays a list of all the variables that were created in the current project.

![Variables List Window]

- The Var. column lists the variable number. (A0 through Z3).
- Columns 1 through 8 represent the 8 programs in the project. If a variable is used in the program, an “X” appears in that column.
- The Variable Name column lists the name given to the particular variable.

Figure 4-71. Variables List Window
Chapter 5
Simulating the Designed Project

Testing and Simulating

MCL-Designer features an integrated simulator that can test and simulate the newly designed project. The simulator offers all the functionalities of the scanner, so downloading the project is not necessary.

Click to open the simulation window.

Figure 5-1. Simulator QTOUR Window
The Simulator window displays the menu design. To start one of the programs, click the key that corresponds to the program, as defined in the Menu options. Refer to Chapter 3, Designing a Main Menu, where the function or numeric menu key was selected.

After selecting the program to work with, test the program by entering the required data. For use of the simulator keyboard, refer to the scanner’s Quick Reference Guide.

**Note:** You can use the mouse to click on the scanner keyboard or type on the keyboard of your PC.

### Working Items of the Simulator

Table 5-1 lists the buttons available on the Simulator window that control the simulation process.

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Run" /></td>
<td>Run</td>
<td>Processes to the next step in the program. If the button is green, you must click it for the program to run until the next step. This button works with the Break button.</td>
</tr>
<tr>
<td><img src="image" alt="Break" /></td>
<td>Break</td>
<td>Sets a break condition, when you can only go to the next step by clicking on the Run button.</td>
</tr>
<tr>
<td><img src="image" alt="Barcode" /></td>
<td>Barcode</td>
<td>Simulates the scanner input.</td>
</tr>
<tr>
<td><img src="image" alt="Variable" /></td>
<td>Variable</td>
<td>Displays the list of the different variables in the project. Every time a variable is updated, it appears on this list. A new input overwrites the old value.</td>
</tr>
<tr>
<td><img src="image" alt="Entries" /></td>
<td>Entries</td>
<td>Shows the field in the data files.</td>
</tr>
<tr>
<td><img src="image" alt="MCL Code" /></td>
<td>MCL Code</td>
<td>Displays the MCL code of the current program.</td>
</tr>
</tbody>
</table>

Once the project has been successfully tested, download it to the scanner. Refer to Chapter 6, Downloading A Project on how to download the project.
**Break**

Click on the Break button to display the *Break* dialog box (Figure 5-2). You can set a break on:

- each MCL-Code line or program line
- each variable update or each variable that is read
- one specific variable update.

The traffic light icon on the Simulator window switches to red if a break is defined. If desired, you can also click on the *Succeed/Fail Dialog Box* check box to display a dialog box indicating successful simulation or a failure.

![Break Dialog Box](image)

**Figure 5-2. Break Dialog Box**

**Bar Code Input**

To simulate a bar code input, click on . The *Barcode Input* dialog box appears (Figure 5-3). Select the type of bar code that will be simulated. Then type the required input in the *Scanned Data* window.

---

**Note:** *Only select the bar codes that were previously selected in the Barcode Input Window. Refer to Chapter 4, Designing Programs.*
Variable List

Displays the variables used during the simulation. Click on the to display the Variable List dialog box (Figure 5-4). This window can remain open while the simulator is running. The Search field allows for a quick search of a variable. Enter the first character of the variable and the first variable matching that character displays.
**File Viewer**

Displays the data files used during the simulation. Click on the 📄 to display the File Viewer dialog box (Figure 5-5). This window can remain open while the simulator is running. Select the data files to view using the File Name list box.

![Figure 5-5. File Viewer Dialog Box](image)

**MCL-Code Display**

The MCL-Code of the project can be displayed in the MCL-Code Viewer dialog box (Figure 5-6). The viewer displays each line that is performed.

![Figure 5-6. MCL Code Viewer Dialog Box](image)

The dialog box displays:

- project file name
- current program name
- MCL-Code line number and command line.
Chapter 6
Downloading A Project

Downloading to a Scanner

To download the project design to a Version 2.XX scanner, select Host Mode on the scanner (system menu option 1, suboption 0), then click on .

To download the project design to a Version 1.XX scanner, select Load New System on the scanner (system menu option 1, suboption 2), then click on . The Send Project dialog box appears.

![Send Project Dialog Box]

**Figure 6-1. Send Project Dialog Box**

- **Project**
  - Displays the name of the current project.
- **Comm Port**
  - Selects the host’s communication port.
• Click ✔️ to accept the default setting
  or
• Enter the appropriate information to change the values and click ✔️ to apply the changes.
• Click ❌ to close the dialog box without saving the changes.
Chapter 7
MCL-Code Reference Guide

About MCL-Code

The Macro Command Language (MCL-Code) is a collection of simple, high-level commands used to quickly program Symbol scanners. An MCL-Code program is a standard ASCII file containing easy-to-read macro command lines. The program is loaded into the scanner’s memory and is interpreted by the MCL-Code run-time kernel (also loaded in the scanner). Programs and File Sizes on page 7-95 explains how memory is organized and what limitations there are on programs and file size.

MCL-Code Program

The user may design up to nine MCL-Code programs (0.MCL to 8.MCL). Each MCL-Code program can contain up to 512 lines; however, the physical restraint on a program is 4KB.

Each MCL program can call subroutines, which may be defined in the same file, or in a different file. In this way, applications which exceed the 4 KB limit may still be implemented.
**Typographic Conventions**

The following typographic conventions are used in this chapter:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Used for</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>Commands and switches that follow them. You must type the commands and their switches exactly as they appear.</td>
</tr>
<tr>
<td><em>italics</em></td>
<td>Place-holders that represent information you must provide. For example, if you are asked to type filename, you should type the actual name of a file instead of the word shown in italics.</td>
</tr>
<tr>
<td>`</td>
<td>Argument field separator within the command line or file. This is a reserved character and should not be used for any other purpose.</td>
</tr>
<tr>
<td>&lt; Key &gt;</td>
<td>Keyboard key symbols. Examples: &lt; Enter &gt;, &lt; F9 &gt;, &lt; Spacebar &gt;</td>
</tr>
<tr>
<td>[ arg ]</td>
<td>Optional arguments in a command line.</td>
</tr>
<tr>
<td>&amp;</td>
<td>A variable quantity or expression. Example: &amp;V0 = contents of variable V0</td>
</tr>
</tbody>
</table>
Introduction

Overview of the MCL-Code

An MCL-Code program is a sequence of command lines. All MCL command lines consist of a number of tokens, separated by a vertical bar ('|'). The first token in a command line is the 2 character command name, and the remaining tokens are parameters for that command.

MCL-Code commands can be divided into nine logical groups:

- display
- keyboard & scanner input + special input
- files
- variable operations (add, subtract,...)
- serial communication
- input/output
- labels and branch
- parameterization
- miscellaneous.

Variables

Using Variables

A variable:

- exists in a memory zone
  (typically 64 bytes except Z0 to Z2 = 512 bytes)
- has a name (A0 to Z0, A1 to Z1,......, A3 to Y3)
- can be written and read
- is global for all programs in the scanner.

When a variable is written, the user has to specify the name of the variable.

ex: VM | A1 | HELLO WORLD

The command writes HELLO WORLD into variable A1.
When a variable is read, the user specifies the contents of the variable.

ex: DM | 1 | 1 | N | &A1
This command displays HELLO WORLD. &A1 represents the contents of A1.

Any argument of a command can be a variable.

**Types Of Variables**
The MCL-Code recognizes three types of variables:

- system variables
- small variables
- large variables.

**System Variables**
Numbered from 00 – 99, these are parameters and function return codes. The length of the variable depends on the data represented by the variable. In general, these variables do not represent actual memory locations, but require an underlying process to read or set. This underlying process is transparent to the user. *System Setup Variables* on page 7-98 lists the value represented by each System Variable.

**Small Variables**
Labeled A0, A1, A2, A3; B0-B3; …; Y0-Y3; for a total of 100 variables, each capable of storing 64 bytes of data. Small Variables store strings only; any integer data to be manipulated is converted to a string prior to storage, and back to an integer prior to processing. User variables are global variables. They can be accessed from any program in the scanner.

**Large Variables**
Labeled Z0 – Z2, for a total of 3 variables, each capable of storing 512 bytes of data. Like Small Variables, Large Variables store only strings.
Files and Records

General

Within an MCL-Code program, you can use local files.

Note: The P370/P470 RF scanners do not support files. Scanned data may be manipulated using any of the functions available in MCL. However, the resultant data must be transmitted to the host, since there is no facility for storing data long-term in the scanner.

The files are named A to H. These files can be downloaded from the host computer to the scanner. MCL file commands allow the user to write, read, update, delete, or search into these files. Files can be unformatted ASCII text. Refer to Programs and File Sizes on page 7-95 for instructions on sizing the files you will use in your application.

On the scanner, a file is simply an area of the FLASH reserved to hold a specific set of data. A file allocation table (FAT) will be maintained, which specifies for each file, the start and end locations, as well as the number of fields, and the size of each field.

Table 7-2. Sample FAT

<table>
<thead>
<tr>
<th>Name</th>
<th>Start</th>
<th>End</th>
<th>Keys</th>
<th># Fields</th>
<th>Len 1</th>
<th>Len 2</th>
<th>...</th>
<th>Len 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1100</td>
<td>10FF</td>
<td>0</td>
<td>10</td>
<td>7</td>
<td>6</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1100</td>
<td>7FFF</td>
<td>0</td>
<td>3</td>
<td>80</td>
<td>7</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>8000</td>
<td>FFFF</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>H</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
</tbody>
</table>

Files are declared using a File Size command (FZ). This command specifies the name of the file to allocate, and its start and end addresses. The following commands would be used to allocate the above described files:

FZ[A]1000]10FF
FZ[B]1100]7FFF
FZ[C]8000]FFFF
See **File Size (FZ)** on page 7-34 for a complete description of the FZ command.

The File Open (FO) command specifies the field structure of a file. If zero (0) fields are defined, the file is treated as a sequential ASCII file. See **File Open (FO)** on page 7-24 for the full specification of the FO command.

**Structure Of The Files**

The files can have two different structures: variable or fixed length records. Fixed length records can have a key (a sort field).

**Sequential Files**

Sequential files are defined as having 0 fields. The file is simply interpreted as a string of unformatted ASCII data. Sequential files may be used for writing activity logs and such. They CANNOT be searched by field. Sequential file records are delimited by a Carriage Return (0x0D) or Carriage Return/Line Feed (0x0D/0x0A) pair.

**Direct Files**

Files containing fixed length records are called direct files, and may or may not have keys. These files are opened using command FO | filename | Keys | Field1… (refer to **File Open (FO)** on page 7-24).

- Maximum values for direct file: Record length = 255 characters.
- Number of keys = 9.
- Number of fields = 20.
- Field length = 255 characters.

Key fields refer to the fields on which the file is sorted, and on which fields searches may be performed. If no key fields are defined, the file is considered to be unsorted, and any search for data in the file is performed in a linear fashion. If 1 or more key fields are defined, then the file is considered to be sorted on the concatenation of all the key fields. Any keyed search on key \( n \) must specify a value for all keys from 1 to \( n \).

Keyed files cannot be appended to, as no sorting of written records is possible. The file MUST be presorted on the host prior to downloading to the scanner; the scanner has no capability to sort files. If an unsorted file is declared with keys, unexpected results may occur.

When one or more keys are defined, the file is maintained in ascending order on the defined key(s); every record can be accessed directly by the use of its key(s).
### MCL Commands

Table 7-3 lists the available MCL code commands. These commands are described on the pages following as referenced in the table.

#### Table 7-3. Logical Group, Command, and Description

<table>
<thead>
<tr>
<th>Logical group</th>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>BL Backlight</td>
<td>Sets Backlight On/Off.</td>
<td>7-10</td>
</tr>
<tr>
<td></td>
<td>DC Display Clear</td>
<td>Clears line(s) on the scanner display.</td>
<td>7-19</td>
</tr>
<tr>
<td></td>
<td>DM Display Message</td>
<td>Displays a message on the scanner display.</td>
<td>7-20</td>
</tr>
<tr>
<td>Keyboard and Scanner Input</td>
<td>IC Input Cash</td>
<td>Waits for input with cash register format.</td>
<td>7-36</td>
</tr>
<tr>
<td></td>
<td>ID Input Date</td>
<td>Waits for input with selected date format.</td>
<td>7-38</td>
</tr>
<tr>
<td></td>
<td>IH Input Time</td>
<td>Waits for input with selected time format.</td>
<td>7-40</td>
</tr>
<tr>
<td></td>
<td>IK Input Key</td>
<td>Waits for key input.</td>
<td>7-41</td>
</tr>
<tr>
<td></td>
<td>IL Input List</td>
<td>Sets a variable based on a selection from a scrolling list of options.</td>
<td>7-43</td>
</tr>
<tr>
<td></td>
<td>IX Extended Input</td>
<td>Waits for specific input from keyboard or scanner.</td>
<td>7-45</td>
</tr>
<tr>
<td></td>
<td>ML Menu List</td>
<td>Branches based on a selection from a scrolling list of options.</td>
<td>7-50</td>
</tr>
<tr>
<td>Modem</td>
<td>MD MoDem</td>
<td>Outputs a string of characters to the modem and waits for a response from the modem.</td>
<td>7-49</td>
</tr>
<tr>
<td>Files</td>
<td>FD File Delete</td>
<td>Deletes current record.</td>
<td>7-21</td>
</tr>
<tr>
<td></td>
<td>FE File Erase</td>
<td>Delete complete file.</td>
<td>7-22</td>
</tr>
<tr>
<td></td>
<td>FK File Check</td>
<td>File Check.</td>
<td>7-23</td>
</tr>
<tr>
<td></td>
<td>FO File Open</td>
<td>Opens file and defines the data structure.</td>
<td>7-28</td>
</tr>
<tr>
<td></td>
<td>FP File Print</td>
<td>Print a file to serial port.</td>
<td>7-26</td>
</tr>
<tr>
<td></td>
<td>FR File Read</td>
<td>Reads the current record.</td>
<td>7-28</td>
</tr>
<tr>
<td></td>
<td>FS File Seek</td>
<td>Positions the file pointer on the requested record.</td>
<td>7-30</td>
</tr>
<tr>
<td></td>
<td>FU File Update</td>
<td>Reads a record at the current pointer position of the file, updates the fields with the passed parameters, writes the record at the end of the file and deletes the current record</td>
<td>7-32</td>
</tr>
<tr>
<td></td>
<td>FW File Write</td>
<td>Writes a new record in the file.</td>
<td>7-33</td>
</tr>
<tr>
<td></td>
<td>FZ File SiZe</td>
<td>Declare storage for a file.</td>
<td>7-34</td>
</tr>
<tr>
<td>Logical group</td>
<td>Command</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------</td>
<td>----------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Variable Operations</td>
<td>VC Variable Compare</td>
<td>Compares text with pattern.</td>
<td>7-78</td>
</tr>
<tr>
<td></td>
<td>VE Variable Extract</td>
<td>Extracts characters from defined record.</td>
<td>7-80</td>
</tr>
<tr>
<td></td>
<td>VF Variable Format</td>
<td>Formats data as characters or a number.</td>
<td>7-81</td>
</tr>
<tr>
<td></td>
<td>VJ Variable Justify</td>
<td>Justifies data in variable to left or right.</td>
<td>7-84</td>
</tr>
<tr>
<td></td>
<td>VL Variable Length</td>
<td>Computes the size of text.</td>
<td>7-86</td>
</tr>
<tr>
<td></td>
<td>VM Variable Move</td>
<td>Replaces current contents of variable with new value.</td>
<td>7-87</td>
</tr>
<tr>
<td></td>
<td>VO (% + - * /) Variable Operation</td>
<td>Variable Operations: arithmetic.</td>
<td>7-88</td>
</tr>
<tr>
<td></td>
<td>VS Variable Substring</td>
<td>Extracts from a string.</td>
<td>7-90</td>
</tr>
<tr>
<td></td>
<td>VX Variable operation</td>
<td>Variable operations (+ - * / %) with numeric formatting.</td>
<td>7-92</td>
</tr>
<tr>
<td>Serial Communication</td>
<td>SI Serial Input</td>
<td>Waits for characters coming from the serial port</td>
<td>7-69</td>
</tr>
<tr>
<td></td>
<td>SO Serial Output</td>
<td>Sends characters to the serial port.</td>
<td>7-73</td>
</tr>
<tr>
<td></td>
<td>SO Serial Output</td>
<td>Sends characters to the serial port.</td>
<td>7-75</td>
</tr>
<tr>
<td></td>
<td>(P360/P460 Memory Scanner)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(P370/P470 RF Scanner)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Labels and Branch</td>
<td>LB Label definition</td>
<td>Assigns a label to the current program address.</td>
<td>7-48</td>
</tr>
<tr>
<td></td>
<td>SK Skip</td>
<td>Jumps to a label.</td>
<td>7-72</td>
</tr>
<tr>
<td></td>
<td>IF Condit. Branch</td>
<td>Compares data and jumps to labels.</td>
<td>7-39</td>
</tr>
<tr>
<td></td>
<td>RM Run MCL Program</td>
<td>Calls a program and begins executing that program.</td>
<td>7-67</td>
</tr>
<tr>
<td></td>
<td>SR Skip and Return</td>
<td>Jumps to a label and returns after QX</td>
<td>1.</td>
</tr>
<tr>
<td></td>
<td>QX Quit or eXit</td>
<td>Terminates program execution.</td>
<td>7-66</td>
</tr>
</tbody>
</table>
### Table 7-3. Logical Group, Command, and Description (Continued)

<table>
<thead>
<tr>
<th>Logical group</th>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Protocol Support</td>
<td>NI <strong>Network Input</strong></td>
<td>Waits for a command from the network and stores it in a variable.</td>
<td>7-52</td>
</tr>
<tr>
<td></td>
<td>NO <strong>Network Output</strong></td>
<td>Sends a frame to the host via RS-232 using MCL-Link protocol.</td>
<td>7-55</td>
</tr>
<tr>
<td></td>
<td>NO <strong>Network Output</strong></td>
<td>Sends a frame to the host via RS-232 using MCL-Link protocol.</td>
<td>7-57</td>
</tr>
<tr>
<td></td>
<td>NS <strong>MCL-Link Start</strong></td>
<td>Calls MCL-Link communication program.</td>
<td>7-59</td>
</tr>
<tr>
<td>Parameters</td>
<td>PI <strong>Parameterize Input</strong></td>
<td>Sets the input time-out value and exit keys for all subsequent input operations.</td>
<td>7-60</td>
</tr>
<tr>
<td></td>
<td>PT <strong>Parameterize Time</strong></td>
<td>Sets the date and time.</td>
<td>7-62</td>
</tr>
<tr>
<td></td>
<td>PX <strong>Change Syst. Var.</strong></td>
<td>Changes System variable.</td>
<td>7-63</td>
</tr>
<tr>
<td></td>
<td>PY <strong>Change Parameter</strong></td>
<td>Changes scanner parameter based on parameter number and value.</td>
<td>7-64</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>BP <strong>Beep</strong></td>
<td>Emits a “beep” tone during a user-defined time interval.</td>
<td>7-11</td>
</tr>
<tr>
<td></td>
<td>WT <strong>Wait</strong></td>
<td>Suspends program execution for a pre-defined delay time.</td>
<td>7-94</td>
</tr>
<tr>
<td>Compute Commands</td>
<td>CP CRC <strong>ComPute CRC</strong></td>
<td>Compute various CRCs of the given string as a 5 digit integer from 00000 to 65535.</td>
<td>7-13</td>
</tr>
<tr>
<td></td>
<td>CP XOR <strong>ComPute XOR</strong></td>
<td>Compute XOR checksum of given string as a five digit integer.</td>
<td>7-15</td>
</tr>
<tr>
<td></td>
<td>CP CVT <strong>ComPute ConVerT</strong></td>
<td>This function provides useful conversions.</td>
<td>7-16</td>
</tr>
</tbody>
</table>
**Backlight (BL)**

**Function**
Controls the backlight on the scanner screen.

**Syntax**
BL | *parameter*

where:
- Parameter = backlight setting.
  - ON = backlight ON.
  - OFF = backlight OFF.
  - time = backlight on for specified number of seconds (1 to 255 seconds).

**Affected System Variables**
None

**Note**
This command is reserved for future use; the present Phaser design does not incorporate a backlight.

**Examples**

- **BL | 5** Sets backlight ON for 5 seconds.
- **BL | ON** Sets backlight ON.
- **BL | OFF** Sets backlight OFF.
**Beep (BP)**

**Function**
Emits a specified beep sequence.

**Syntax**
BP | *index*

where:

Index = index into the predefined table.

**Affected System Variables**
None

**Parameters**

*index*: Index into the predefined table.

**Notes**
The beep table is defined as follows:

<table>
<thead>
<tr>
<th>Table 7-4. Beep Sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Beep Description</strong></td>
</tr>
<tr>
<td>1 Short High Beep</td>
</tr>
<tr>
<td>2 Short High Beeps</td>
</tr>
<tr>
<td>3 Short High Beeps</td>
</tr>
<tr>
<td>4 Short High Beeps</td>
</tr>
<tr>
<td>5 Short High Beeps</td>
</tr>
<tr>
<td>1 Short Low Beep</td>
</tr>
<tr>
<td>2 Short Low Beeps</td>
</tr>
<tr>
<td>3 Short Low Beeps</td>
</tr>
<tr>
<td>4 Short Low Beeps</td>
</tr>
<tr>
<td>5 Short Low Beeps</td>
</tr>
</tbody>
</table>
### Table 7-4. Beep Sequences (Continued)

<table>
<thead>
<tr>
<th>Beep Description</th>
<th>Beep Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Long High Beep</td>
<td>10</td>
</tr>
<tr>
<td>2 Long High Beeps</td>
<td>11</td>
</tr>
<tr>
<td>3 Long High Beeps</td>
<td>12</td>
</tr>
<tr>
<td>4 Long High Beeps</td>
<td>13</td>
</tr>
<tr>
<td>5 Long High Beeps</td>
<td>14</td>
</tr>
<tr>
<td>1 Long Low Beep</td>
<td>15</td>
</tr>
<tr>
<td>2 Long Low Beeps</td>
<td>16</td>
</tr>
<tr>
<td>3 Long Low Beeps</td>
<td>17</td>
</tr>
<tr>
<td>4 Long Low Beeps</td>
<td>18</td>
</tr>
<tr>
<td>5 Long Low Beeps</td>
<td>19</td>
</tr>
<tr>
<td>4 Short Beeps ---- High – Low – High – Low</td>
<td>20</td>
</tr>
<tr>
<td>4 Long Beeps ---- High – Low – High – Low</td>
<td>21</td>
</tr>
<tr>
<td>2 Short Beeps ---- High – Low</td>
<td>22</td>
</tr>
<tr>
<td>2 Short Beeps ---- Low – High</td>
<td>23</td>
</tr>
<tr>
<td>3 Short Beeps ---- High – Low – High</td>
<td>24</td>
</tr>
<tr>
<td>3 Short Beeps ---- Low – High – Low</td>
<td>25</td>
</tr>
<tr>
<td>4 Long Beeps ---- High – High – Low – Low</td>
<td>26</td>
</tr>
<tr>
<td>4 Long Beeps ---- Low – High – Low</td>
<td>27</td>
</tr>
<tr>
<td>3 Long Beeps ---- Low – High – Low</td>
<td>28</td>
</tr>
<tr>
<td>3 Long Beeps ---- High – Low – Low</td>
<td>29</td>
</tr>
<tr>
<td>3 Long Beeps ---- Low – High – High</td>
<td>30</td>
</tr>
<tr>
<td>4 Long Beeps ---- High – High – High – Low</td>
<td>31</td>
</tr>
<tr>
<td>Decode Beep</td>
<td>32</td>
</tr>
</tbody>
</table>

### Examples

**BP | 32**  
Generate a decode beep.
**Compute CRC (CP CRC)**

**Function**
Compute various CRCs of the given string as a 5 digit integer from 00000 to 65535.

**Syntax**
```
CP | CRC | variable| string | option
```

where:
- **Variable** = variable receiving value
- **String** = input character string
- **Option** = specific CRC algorithm
  - 0: MCL-CRC16
  - 1: CCITT
  - 2: CRC16
  - 3: Alternative CRC-16 (with sum initialized to 0x0000)
  - *default: MCL-CRC16

(If no option is selected, the terminal defaults to the MCL-CRC16 algorithm.)

**Affected System Variables**
None

**Note**
Use the command CP|CVT to convert the result of this operation in the desired format.

**Examples**
```
CP | CRC | A0 | Mouse House | 0
After this operation, the variable A0 will contain 59010

CP | CRC | A0 | Mouse House
After this operation, the variable A0 will contain 59010

CP | CRC | A0 | The hazy sky | 1
```
After this operation, the variable A0 will contain 59802

CP | CRC | A0 | 02343499 | 2

After this operation, the variable A0 will contain 64622

CP | CRC | A0 | Mouse House | 3

After this operation, the variable A0 will contain 64891
**ComPute XOR (CP XOR)**

**Function**
Compute XOR checksum of given string as a five digit integer. (Use CP | CVT to convert in another format.)

**Syntax**
\[ \text{CP} \mid \text{XOR} \mid \text{variable} \mid \text{string} \]

where:
- Variable = variable receiving value
- String = input character string

**Affected System Variables**
None

**Note**
Use the command CP | CVT to convert the result of this operation in the desired format.

**Examples**
- CP | XOR | A0 | Mouse House
  After this operation, the variable A0 will contain 00218
- CP | XOR | A0 | 900345
  After this operation, the variable A0 will contain 00244
**ComPute ConVert (CP CVT)**

**Function**
This function provides useful conversions.

**Syntax**

```
CP | CVT | variable | type | string
```

where:

- **Variable** = variable receiving value
- **Type** = type of conversion
  - 0 = convert a char to this ASCII value or convert a string to the sum of ASCII values of each character
  - 1 = get a number as an ASCII code and convert it to a character
  - 2 = convert a number to a string of 2 hexadecimal digits
  - 3 = convert a number to a string of 4 hexadecimal digits
  - 4 = converts a number to a string of 2 characters as used by MCL-Net protocol for checksums
  - 5 = converts a number to a string of 4 characters as used by MCL-Net protocol for checksums
  - 6 = convert a string of hexadecimal digits to a decimal number (of 5 digits)
  - 7 = convert a string of MCL-Net checksum digits to a decimal number
- **String** = input character string or number
Affected System Variables

| &99 = | 0, if conversion is OK. |
|       | Variable = returns value of conversion. |
|       | 1, if the number or string cannot be converted. |
|       | Variable = "?" |

Examples

CP | CVT | A0 | 0 | X
Convert char to an ASCII code. A0 will contain the ASCII code of X = 00088

CP | CVT | A0 | 0 | ABCDE
Convert a string to the sum of ASCII codes. A0 will contain the ASCII code of X = 00335

CP | CVT | A0 | 1 | 88
Convert an ASCII code to a character. A0 will contain the character X

CP | CVT | A0 | 2 | 186
Convert to 2 hexadecimal digits. A0 will contain BA

CP | CVT | A0 | 3 | 186
Convert to 4 hexadecimal digits. A0 will contain 00BA

CP | CVT | A0 | 2 | 25761
Var 99 = 1 because 25761 cannot be converted to 2 hexadecimal digits. A0 = "?"

CP | CVT | A0 | 4 | 186
Convert to 2 MCL-Net checksum characters. A0 will contain ::

CP | CVT | A0 | 5 | 186
Convert to 4 MCL-Net checksum characters. A0 will contain 00::

CP | CVT | A0 | 4 | 25761
Var 99 = 1 because 25761 cannot be converted to 2 hexadecimal digits. A0 = "?"

CP | CVT | A0 | 5 | 25761
Convert to 4 MCL-Net checksum characters. A0 will contain 64:1

```
CP | CVT | A0 | 6 | 64A1
```

Convert from hexadecimal to integer. A0 will contain 25761

```
CP | CVT | A0 | 7 | ;;
```

Convert from MCL-Net checksum characters to integer. A0 will contain 00187
Display Clear (DC)

Function
Clears the display, or a single line of the display.

Syntax
DC | line

where:

Line = number of the line to be cleared.
(0 clears the whole screen, else the specified line is cleared).

Affected System Variables
None

Examples

DC | 0  Clears all lines on the scanner display.
DC | 1  Clears the first line only.
**Display Message (DM)**

**Function**
Displays a message on the scanner screen.

**Syntax**

```
DM | line | column | style | message
```

where:

- Line = the line number on which to display the message (1 or 2).
- Column = the column position (1 to 20).
- Style = the display attribute:
  - N = Normal Video.
  - R = Reverse Video.
- Message = the text or variable to be displayed.

**Affected System Variables**
None

**Note**

1. If the line number or column number is outside the display range, nothing is displayed.
2. An alternate way to display characters is the use of the hexadecimal conversion, ex: \41 = “A”.
3. If the first character to be displayed is a “#”, then it must be encoded as \23 so that it is not misinterpreted as a carriage return during the download process.

**Examples**

```
DM | 1 | 5 | N | MENU
```
Displays the word “MENU” on line 1, column 5.

```
DM | 1 | 7 | R | &A1
```
Displays the contents of variable A1 on line 1, column 7 in reverse video.
**File Delete (delete record) (FD)**

**Function**
Deletes the current record in a file.

**Syntax**
`FD | filename`

where:
- `Filename` = the name of the selected file (range: A - H).

**Affected System Variables**

| &99 | 0, if no error occurs or 1 in case of error. |

**Notes**

1. This command does not delete a file, it deletes a record in a file. (Use File Erase to delete a file).
2. The file pointer is placed on the next record of the file, except when the last record is being deleted. In this case, the pointer is set to the end of the file.
3. Command is supported on DIRECT files only.
4. Records are deleted by replacing them with a special string.
5. If `filename` is unknown, System Error “2” is displayed.

**Examples**

`FD | A` Deletes the current record in file “A”.
File Erase (delete file) (FE)

Function
Erases a file.

Syntax
FE | filename

where:
Filename = the name of the selected file (range: A - H).

Affected System Variables

| &99 | 0, if no error occurs or 1 in case of error. |

Notes
1. After a file erase command, the file cannot be recovered.
2. A file is erased by replacing its FAT entry with all zeros (0).
3. If filename is unknown, System Error “2” is displayed.

Examples

FE | A Erases file “A”.
**File check (FK)**

**Function**
Checks if a file is present and returns its size.

**Syntax**
FK | *filename*

where:
Filename = file to be checked (range: A-H).

**Affected System Variables**

<table>
<thead>
<tr>
<th>&amp;99</th>
<th>0, if no error occurs or 1 in case of error.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;97</td>
<td>file size (free format).</td>
</tr>
</tbody>
</table>

**Example**

FK | A  Checks if file “A” exists.
**File Open (FO)**

**Function**
Opens a file and defines the structure of the data.

**Syntax**

```
FO | filename | keys | fieldlength1 [ | fieldlength 2 ].....[ | fieldlength n]
```

where:

- **Filename** = the name of the file (range: A - H).
- **Keys** = the number of key fields in the record.
  (range: 0 - 9; if no key, enter 0).
- **Fieldlength 1** = the length of the first field (range: 0 -255).
- **Fieldlength n** = the length of the last field (range: 1 -255). Up to 20 fields may be declared.

**Affected System Variables**

<table>
<thead>
<tr>
<th>&amp;99</th>
<th>0 if no error occurs, or 1 in case of error.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>If &amp;99!=0 (if an error occurred), &amp;98 will contain the error code:</td>
</tr>
<tr>
<td></td>
<td><em>F001</em>: Illegal File Name</td>
</tr>
<tr>
<td></td>
<td><em>F002</em>: File not empty - the record structure being defined does not match the existing record structure, and the existing file is not empty. This is not allowed, as it could cause a loss of collected data.</td>
</tr>
<tr>
<td></td>
<td><em>F003</em>: File Allocation Table could not be updated. This could result from an excessive number of file redefinitions, causing FLASH memory wear.</td>
</tr>
</tbody>
</table>

**Notes**

1. To declare a sequential file, set **keys** and **fieldlength1** to 0 (zero), and do not declare any more fields. (Sequential files cannot have keys).
2. All fields in the record are text fields (numeric values are converted to ASCII representations). Size fields accordingly.
3. If filename is unknown or any I/O error, System Error “2” is displayed.
4. If the number of key fields is greater than 9, System Error “2” is displayed.
5. If the file does not exist (has not been allocated using an FZ command), System Error “2” is displayed.

6. If a file is opened with the same specification that previously existed, then any existing data remains intact. If it is opened with a new specification, and the file is not empty, an error occurs.

7. A file must be opened before any operation can take place.

8. The keys represent the number of fields to use as key. They define how the file is organized as the file is sorted following these keys.

9. Keyed files are read-only. The scanner is not able to write or update records in a sorted manner. The files MUST be sorted according to the defined keys prior to downloading to the scanner.

10. The maximum record size is 255 characters with a maximum of 20 fields.

Examples

<table>
<thead>
<tr>
<th>FO</th>
<th>A</th>
<th>2</th>
<th>6</th>
<th>4</th>
<th>20</th>
<th>8</th>
</tr>
</thead>
</table>
|     | Opens a file with name “A”.
|     | The file consists of records with 4 fields, with field lengths of 6, 4, 20 and 8 characters, respectively. The two first fields (6 and 4 characters) are defined as key fields.
|     | The file is created and maintained, in ascending order, on the two access keys. The primary sort is on the 1st access key, and the secondary sort is on the 2nd access key.
| View of the file: | 000001 | 1234 | ABCDEFGHIJKLMNOPQRST | 00020.17 CR LF
| | 000001 | 1235 | UVWXYZABCDEFHIJKLMNOPQRSTUVWXYZ | 12000.15 CR LF
| | 000074 | 7012 | HELLO WORLD | 00000.00 CR LF

<table>
<thead>
<tr>
<th>FO</th>
<th>A</th>
<th>1</th>
<th>4</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Opens a file with name “A”.
| The file consists of records with 2 fields, with field lengths of 4 and 20 characters, respectively. The first field (4 characters) is defined as a key field.
| View of the file: | 0001 | BARBARA HENDRICKX CR LF
| | 0001 | THOMAS EDISON CR LF
| | 1205 | WOLFGANG AMADEUS MZT CR LF

| FO | D | 0 | 0 |
|----|----|----|
| Opens a file with name “D” with no structure.
| View of the file: | BARBARA HENDRICKX CR LF
| | THOMAS EDISON CR LF
| | WOLFGANG AMADEUS MZT CR LF
**File Print (FP)**

**Function**
Output a file to a serial port, with no formatting and no protocol. (Dump file).

**Syntax**
```
FP | com | filename | replace
```

where:
- **Com** = Ignored (only one com port available).
- **Filename** = the name of the file (range: A - H).
- **Replace** = 
  - **N**: Transmit contents of the file as is.
  - **Y**: All variables (ex: &A0) are replaced with the contents of the referenced variable.

**Affected System Variables**

| &99   | 0 if no error occurs, or 1 in case of error. |

**Notes**

1. The FP command makes use of the existing host interface software to communicate a file with a host using either serial or Synapse communications. The file will be transferred according to the file type:

   **Direct files** (files with defined record structures) will be transferred one record at a time. If a record is too big to be transmitted by the installed host, the operation fails; therefore it is incumbent on the designer to create a record structure which can be handled by the intended host interface.

   **Sequential files** (files with no defined record structures) will be transferred in packets which are sized to the capabilities of the host interface.

   In either case, the global communication parameters (including data formatting for appending CR/LF to the data) will be applied.
2. The position of the file pointer after an FP command is indeterminable, so an FS command should be used to reposition the file at a known position prior to any subsequent accesses.

3. The file to be printed must be opened with the FO command prior to printing.

4. In order to send formatted data during an upload, a loop of SO commands should be used instead of FP. Note, however, that if this method is used during daisy-chaining, the bus arbitration will occur per record, not per file upload.

**Example**

FP | 1 | A | Y  Dumps file “A” to serial port, replacing variables.
**File Read (FR)**

**Function**
Reads the record located at the current pointer position of the file.

**Syntax**

```
FR | filename | length | variable
```

where:

Filename = the name of the sequential file (range: A - H).
Length = the number of bytes to read from the current pointer position.
Variable = the variable in which data is stored.

```
FR | filename | variable 1 [ | variable 2] ... [ | variable n ]
```

where:

Filename = the name of the DIRECT file (range: A - H).
Variable 1 = the variable in which data for the first field of the record is stored. (Can be replaced by ?? to skip this field).
Variable n = the variable in which data for the last field of the record is stored. (Can be replaced by ?? to skip this field).

**Affected System Variables**

```
&99 =
0 if no error occurs, or 1 in case of error.
```

**Notes**

1. The number of variables must be less than or equal to the number of fields contained in the record.
2. To ignore a field of a record, enter “??” instead of the variable name. For DIRECT file read only.
3. If filename is unknown, System Error “2” is displayed.
4. After reading, the pointer remains unchanged.
5. Each field is stored in variable 1, variable2,..... Last variable in command contains all remaining fields. If a variable is replaced by ??, the field is not read.

6. If a sequential read is attempted on a direct file, or a direct read is attempted on a sequential file, System Error “2” is displayed.

Examples

**FR | A | C1 | ?? | F1 | G0**  
Reads a record from the current pointer position of file “A” and stores the contents of field 1 into C1, field 3 into F1, and field 4 into G0 (field 2 is ignored).

**FR | A | 30 | C1**  
Reads 30 bytes from the current pointer position of file “A” into variable C1. (File A must be declared as a sequential file.)
**File Seek (FS)**

**Function**
Positions the file pointer to a specific record or byte, or searches a file for a specified pattern or key value.

**Syntax**

```plaintext
FS | filename | position [ | pattern]  
OR  
FS | filename | position [ | key1val ] ...[ | keynval ]
```

where:

- **Filename** = the file to be positioned (range = A-H).
- **Position** = the position at which the file pointer is to be set (if pattern and keyval are omitted, or at which the search begins. Values can be:
  - F = First record.
  - L = Last record.
  - P = Previous record.
  - N = Next record.
  - C = Current record.
  - n = An arbitrary record number.

- **Pattern** = A pattern to find beginning at the specified position. Patterns can contain a string of characters. This string will be matched to the first n characters in each record, until a match is found.
  An asterisk "*" is used as a wildcard.

- **Keynval** = The value of the key fields. The number of specified keys can be any number up to the total number of keys defined for the file. The search begins at the position specified, and locates the first record whose first n keys match the specified values.
Affected System Variables

<table>
<thead>
<tr>
<th>&amp;96</th>
<th>Current record number.</th>
</tr>
</thead>
</table>
| &99 | 0, if the requested record is found.  
1, if not found, pointer remains unchanged. |

Notes

1. The number of keys must be less than or equal to the number of keys defined in the File Open command of the related file.
2. If the filename is unknown, System Error “2” is displayed.
3. If the file is defined by FO | filename | 0 | 0 (sequential file), no key or pattern can be defined.
4. System Variable 96 returns the current record number after FS command. This can be used to determine the total number of records in a file using the Last position parameter.
5. If the requested record is not found, the current file pointer is not updated.

Examples

FS | A | F | 00001 | 1130 | &A0  Positions the pointer of file “A” on the first record with:    
key 1 = 00001.    
key 2 = 1130.    
key 3 = variable &A0.

FS | D | 18  Positions the pointer on the 18th record of file “D”.
**File Update (FU)**

**Function**
Reads a record at the current pointer position of the file updates the fields with the passed parameters, writes the record at the end of the file and deletes the current record.

**Syntax**

```
FU | filename | var1 [ | var2]...[|varn}
```

where:

- `filename` = name of the file to update (A-H).
- `var1` = the variable in which data for the first field of the record to update will be stored. (Can be replaced by “??” to skip this field.)
- `varn` = the variable in which data for the last field of the record to update will be stored. (Can be replaced by “??” to skip this field.)

**Affected System Variables**

<table>
<thead>
<tr>
<th>&amp;99</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>&quot;0&quot; if no error occurs.</td>
</tr>
<tr>
<td>1</td>
<td>&quot;1&quot; if error is detected.</td>
</tr>
</tbody>
</table>

**Notes**

1. The number of variables must be less than or equal to the number of fields contained in the record. To ignore a field without using a variable, enter “??” signs instead of the variable name.
2. If the length of the specified fields exceeds the corresponding field length of the file, the variable will be truncated to fit within the field.
3. If `filename` is unknown, System Error “2” is displayed.
4. The updated record is appended to the end of the file.
5. After updating, the file pointer is moved to the updated record which is at the end of the file.
6. Only applies to DIRECT files.

**Example**

```
FU | A | ?? | &A1
```

Update the second field of file “A” with the contents of variable A1 leaving the first field unchanged.
**File Write (FW)**

**Function**
Writes a new record in the file.

**Syntax**
FW | filename | field1 [ | field2 ] ... [ | field n ]

where:
Filename = the name of the file (range: A - H).
Field1,…n. = the text or variable for fields 1 to n.

**Affected System Variables**

| &99 = | 0 if no error occurs, or 1 in case of error. |

**Notes**
1. The number of field(s) must be equal to the number of field(s) defined within the File Open command.
2. If filename is unknown, System Error “2” is displayed.
3. Records are appended to the end of the file, regardless of the current file pointer position. The current pointer position is not changed.
4. All fields are truncated or left justified (filled with blanks on right) to the correct length.
5. It is recommended that \od (carriage return) or \od\oa (carriage return/line feed) terminate each sequential file record to facilitate later retrieval of these records.

**Example**

FW | A | &C1 | &D0 | &F1 | 123  Writes a new record into file “A”.
Stores the content of the variables C1, D0, F1 and 123 in the record.
**File Size (FZ)**

**Function**
To allocate storage for a file.

**Syntax**
FZ | *filename* | *start* | *end*

where:

- Filename = the name of the file to allocate storage for (range 'A' – 'H').
- Start = the first block number to allocate for this file. Values are hexadecimal.
- End = the last block number to allocate for this file; if 0xFFFF is specified, the entire available memory is allocated. Values are hexadecimal.

**Affected System Variables**

<table>
<thead>
<tr>
<th>&amp;99 =</th>
<th>0 if no error occurs, or 1 in case of error.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98 =</td>
<td>If an error occurs, contains the error code:</td>
</tr>
<tr>
<td>FZ01 = Bad block allocation: The start and end block numbers do not represent a legal file (start &gt; end; or end too big).</td>
<td></td>
</tr>
<tr>
<td>FZ02 = File being reallocated (allocated with different parameters than pre-existing file) is not empty. This is not allowed as data loss would occur.</td>
<td></td>
</tr>
<tr>
<td>FZ03 = File allocation would cause loss of data in another file</td>
<td></td>
</tr>
<tr>
<td>FZ04 = File Allocation Table could not be updated. This could be due to excessive wear on the FLASH memory part.</td>
<td></td>
</tr>
</tbody>
</table>
**Notes**

1. Files are allocated using block numbers. A block represents 256 bytes of FLASH memory. Block numbers are referenced to an artificial zero point which represents the first FLASH memory block available for data storage. Therefore, if the system code occupies 140 Kbytes (0x023000) of space, followed by 32 KB (0x008000) of MCL application storage, then “Block 0” starts at address 0x02B000.

2. MCL Designer can be set up for 512 KB, 1 MB, 4 MB or 8 MB scanners on the System Setup screen. 256 KB are reserved for system use (512 KB are reserved on the Extended Memory Phaser). Designer keeps track of the remaining space as files are allocated.

3. Minimum file size supported is 2 blocks.

4. On Extended Memory Phaser, file allocation requests are rounded up to the nearest whole 64 KB segment.

**Example**

```
FZ|A|1000|1255  Allocates file 'A' to begin at block 1000, and to occupy 256 blocks (64K).
```
Input Cash (IC)

Function
Inputs a number with selected format in cash register input mode (digits are scrolled from right to left) a short beep is emitted after input.

Syntax
IC | line | column | variable | integer | fraction

where:
- Line = line of the input field on the display (range: 1 - 2).
- Column = left most column of the input field.
- Variable = name of the variable to receive the input.
- Integer = size of integer (0 to 7 digits).
- Fraction = size of fractional part (0 to 4 digits).

Affected System Variables
None

Notes
1. Integer set to 0 means that it is not possible to enter a number greater than 0.9999. In that case, the decimal point is placed at the most left place.
2. Fraction set to 0 means that the entered number is an integer. In that case, the decimal point is not displayed.
3. Integer and fraction both set to 0 is not allowed.
4. Integer entry ends and fractional entry begins when either the maximum specified number of integers has been entered or ‘.’ or ‘,’ is pressed.
Active keys are:

- <Enter> Accept Input.
- ‘.’ or ‘,’ switches to fractional input.
- <Bksp> removes last entered character.
- <0> to <9> digits.

All keys defined in the last PI command

Example

IC | 2 | 6 | A0 | 4 | 2 Lets user enter a number with format 0000.00 in variable A0 at screen position (2,6).
**Input Date (ID)**

**Function**
Displays an input mask at a screen position and waits for a date input with selected format.

**Syntax**

\[
\text{ID} \mid \text{line} \mid \text{column} \mid \text{variable}
\]

where:

- **Line** = line of the input field on the display (range: 1 - 2).
- **Column** = left most column of the input field.
- **Variable** = name of the variable to receive the input.

**Affected System Variables**
None

**Notes**

1. The default date format is MM/DD/YYYY. The date format and date separator can be changed using the PY command.
2. Date is checked during input. All invalid dates are rejected.

**Examples**

<table>
<thead>
<tr>
<th>PY</th>
<th>F013</th>
<th>01</th>
<th>Sets format and separator.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PY</td>
<td>F00E</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>2</td>
<td>1</td>
<td>A0</td>
</tr>
<tr>
<td>PY</td>
<td>F013</td>
<td>00</td>
<td>Sets format and separator.</td>
</tr>
<tr>
<td>PY</td>
<td>F00E</td>
<td>02</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>2</td>
<td>1</td>
<td>A0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lets user enter a date with format DD/MM/YYYY in variable A0 at screen position (2,1).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sets user enter a date with format MM:DD:YYYY in variable A0 at screen position (2,1).</td>
</tr>
</tbody>
</table>
**IF (IF)**

**Function**
Compares a variable with text and jumps to one of two routines, depending on the outcome of the comparison.

**Syntax**
```plaintext
IF | text1 | test | text2 | true | false
```

where:
- `Text1` = the first operand to compare.
- `Test` = the condition for the comparison.
  - values for `test`: = equal to.
  - > greater than.
  - < less than.
- `Text2` = the second operand to compare.
- `True` = address or label of the next instruction if result is true.
- `False` = address or label of the next instruction if result is false.

**Affected System Variables**
None

**Note**
All comparisons are lexicographic, except when operand 1 and operand 2 are both integers.

**Examples**
- ```plaintext
  IF | &A0 | > | 999 | OVFLW | +1
  ```
  Execute the OVFLW routine if the value in A0 is greater than 999. Execute the next instruction if the value in A0 is less than or equal to 999.
- ```plaintext
  IF | &C0 | = | STOP | END | +1
  ```
  Execute the END routine if the value in C0 is equal to “STOP”. Execute the next instruction if the value in C0 is not equal to “STOP”.
Input Hour (IH)

Function
Displays an input mask and waits for a time input with selected format.

Syntax
IH | line | column | variable

where:

   Line = line of the input field on the display (range: 1 - 2 = screen line).
   Column = left most column of the input field.
   Variable = name of the variable to receive the input.

Affected System Variables
None

Note
Invalid time entry is rejected.

Example
IH | 1 | 6 | A0

   Lets user enter a hour with format HH: MM: SS in variable A0 at screen position (1,6).
**Input Key (IK)**

**Function**
Jumps to the associated label when one of the keys defined in the command argument list is pressed. (Defines HOT-KEYS).

**Syntax**
IK | key1 | label | key2 | label | ... | key n | label

where:
- Key 1 = name of the first key or time out value.
- Key n = name of the last key.
- Label = destination address, if the key is pressed.

<table>
<thead>
<tr>
<th>Key Code Values</th>
<th>Actual Keys Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>F0 to F9</td>
<td>Function keys (F0 = &lt;F10&gt;, F1 = &lt;F1&gt;, F2 = &lt;F2&gt;, etc.).</td>
</tr>
<tr>
<td>UP or DW</td>
<td>&lt;Up&gt; or &lt;Down&gt; arrow keys.</td>
</tr>
<tr>
<td>PU or PD</td>
<td>&lt;FN+UP&gt; or &lt;FN+DOWN&gt; keys.</td>
</tr>
<tr>
<td>Ax</td>
<td>Any alpha character on the keypad (AK = &lt;K&gt; key). Only uppercase is supported.</td>
</tr>
<tr>
<td>Nx</td>
<td>Any character on the numeric keypad (N9 = &lt;9&gt; key).</td>
</tr>
<tr>
<td>EN</td>
<td>&lt;Enter&gt; key.</td>
</tr>
<tr>
<td>CL</td>
<td>&lt;Fn+Bksp&gt; key.</td>
</tr>
<tr>
<td>SP</td>
<td>Special &lt;*&gt; key.</td>
</tr>
<tr>
<td>**</td>
<td>Any key (must be the last key of the list).</td>
</tr>
<tr>
<td>BS</td>
<td>Backspace key.</td>
</tr>
</tbody>
</table>
Affected System Variables

| 98 = | Set to KeyName of key pressed. |

Notes

1. Key 1 can be a TIMEOUT, with a maximum value of 9999. (see the PI command for an explanation of TIMEOUT values).
2. Maximum 20 entries (including TO).
3. If ** is used, it must be placed at the end of the list because keys defined after ** are ignored.

Examples

**IK | F1 | HELP | F8 | END | CL | +1**

Jump to the HELP label if the <F1> key is pressed.
Jump to the END label if the <F8> key is pressed.
Jump to current line +1 if the <FN+BKSP> key is pressed.

**IK | F1 | HELP | ** | OUT**

Jump to the HELP label if the <F1> key is pressed.
Jump to the OUT label if any other key is pressed.
Input List (IL)

Function
Define a list input field, where several options may be accessed using a single field, scrolling through the available options. The index of the selection is returned in a variable.

Syntax
IL | line | column | height | width | default | mode | speedkey | variable | sel1 [ | sel2 ]…[ | seln ]

where:

- line = line on which to begin field.
- column = column in which to begin field
- height = number of lines to be used (ignored by Phaser).
- width = number of columns to be used.
- default = Index of default selection.
- mode = Scrolling Mode.
  - S = Scrolling: if user attempts to scroll beyond the start or end of the list, the scanner ignores the keypress and beeps.
  - R = Rolling: if the user scrolls past the end of the list, the list rolls over back to the beginning.
- speedkey = Enable speed key processing. Note that the speedkey feature only works with numerics (0-9).
  - 1 = Pressing the key corresponding to the first character of a list item moves directly to that selection (most useful if list items are numbered).
  - 0 = Disable the speedkey feature.
- variable = Variable to receive the selection.
- sel1 = Text of first available selection.
- seln = Text of last available selection.
## Affected System Variables

<table>
<thead>
<tr>
<th>&amp;98</th>
<th>The text of the selected entry or “TO” if time-out occurs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;97</td>
<td>The index number of the selected entry.</td>
</tr>
</tbody>
</table>

## Notes
1. The PI command is active in the IL command, so all defined hot-keys are still available.
2. The selected index will be returned in the variable.
3. If <UP> (UP Arrow) or <DW> (Down arrow) are used in PI before IL command, they will perform the action defined in PI only on <UP> on first item and <DW> on last item, and then only if mode is set to S. Otherwise, <UP> and <DW> switch the current selected item in the list.

## Examples

```
IL | 2 | 2 | 1 | 18 | 1 | S | 0 | A0 | 1. Inventory | 2. POS | 3. Shipping
```

Define, at line 2, column 2, for 18 characters, an input list field which will scroll through the following selections, starting with selection 1:

1. Inventory
2. POS
3. Shipping

If the user hits the UP key when selection 1 is displayed, the program will take whatever action is currently defined in the PI command for an up key. If the user hits the DOWN key when selection 1 is displayed, selection 2 will be displayed. The chosen selection number will be stored in A0. Speedkey processing is disabled.
**Input eXtended (IX)**

**Function**
Accepts any input from the keyboard or scanner.

**Syntax**

```
IX | line | column | variable | minlength | maxlength | kb | scan | autoval | data | reset-var | scanret
```

where:

- **Line** = line of the input field on the display (range: 1 - 2 = screen line).
- **Column** = left most column of the input field.
- **Variable** = name of the variable to receive the input.
- **Minlength** = minimum length of the input field (min. = 0 char.).
- **Maxlength** = maximum length of the input field (max. = 55 char.).
- **Kb** = keyboard entry allowed (Y or N).
- **Scan** = scanning entry allowed (Y or N).
- **Autoval** = auto-validation (Y or N). If Y, scanner input is accepted without having to press <ENTER>, and keypad input of max_length characters is accepted without having to press <ENTER>. If N, <ENTER> must be pressed to accept either scanner or keypad input.
- **Data** = type of data. Valid entries are:
  - **A** = alphanumeric.
  - **N** = numeric.
  - **D** = decimal (accept “.” or “,”).
  - **S** = secret. “**” displayed in input field for every character entered.
- **Reset-var** = reset variable? (Y or N). If Y, the destination variable is cleared before the input operation, and no default value is displayed on the screen; if N, then the current value of the variable is used as the default value, and displayed in the input field prior to accepting user input.
- **Scanret** = scan-return-scan (Y or N). If Y, and scan is N, then scanning data on this field will cause the default value to be entered, and the next field to be accessed.
Affected System Variables

Normal mode

<table>
<thead>
<tr>
<th>&amp;91</th>
<th>Barcode Descriptor</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;97</td>
<td>“?”</td>
</tr>
<tr>
<td>&amp;98</td>
<td>type of code: C39, I25, C128, E08, E13, C49, UPCE0, UPCE1, UPCA, MSI, CODABAR, D25, C11, C93, E128, Unknown or KB if keyboard input.</td>
</tr>
<tr>
<td>variable</td>
<td>Contains the input data.</td>
</tr>
</tbody>
</table>

After TIMEOUT (if one has been defined in the previous PI command)

<table>
<thead>
<tr>
<th>&amp;91</th>
<th>“?”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;97</td>
<td>“?”</td>
</tr>
<tr>
<td>&amp;98</td>
<td>“TO”</td>
</tr>
<tr>
<td>variable</td>
<td>“?”</td>
</tr>
</tbody>
</table>

After EXIT KEY (defined in the previous PI command)

<table>
<thead>
<tr>
<th>&amp;91</th>
<th>“?”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;97</td>
<td>name of the pressed key (see PI command for keynames).</td>
</tr>
<tr>
<td>&amp;98</td>
<td>“KB”</td>
</tr>
<tr>
<td>variable</td>
<td>“?”</td>
</tr>
</tbody>
</table>

Notes

1. If <CL> (Clear key) is used in PI before IX command, it performs the action defined in PI only on the first input position. Otherwise, <CL> clears the current input field.

2. The Barcode Descriptor is a string describing the barcode length and the barcode symbol type. Some host interface software requires this data to properly format the output string for the host device (Synapse hosts, and some RS232 host variants, for example). If your data will be transmitted to one of these hosts, you must save the Barcode Descriptor in your data file along with the barcode data, so that the proper formatting may be performed by the interface software. (See Serial Output (SO) - P360/P460 Memory Scanner on page 7-73 and Serial Output (SO) - P370/P380 Memory Scanner on page 7-74.)
**P470 RF Scanner** on page 7-75 for a description of how to use this data in formatting your output.)

3. Setting SCAN-RET to Y overrides KB disable.

**Example**

<table>
<thead>
<tr>
<th>IX</th>
<th>2</th>
<th>10</th>
<th>A3</th>
<th>5</th>
<th>10</th>
<th>Y</th>
<th>Y</th>
<th>A</th>
<th>Y</th>
<th>N</th>
</tr>
</thead>
</table>

Accepts input data on line 2, column 10 (minimum 5 characters, maximum 10 characters), and stores it in variable A3.

The input can be entered from the keyboard or a scanner.

The entry is automatically validated (if 10 characters long).

The input is alphanumeric.

Variable A3 is reset before executing the command.
**LaBel (LB)**

**Function**
Assigns a label to the current program address.

**Syntax**

```
LB | label
```

where:

- Label = the name of the address (maximum 10 characters).

**Affected System Variables**
None

**Notes**

1. Label can be alphanumeric, but the first character must be alpha or underscore ("_") character.
2. Label name must be unique for each program.
3. Maximum length = 10 characters.

**Example**

```
LB | MAIN
```

Assigns the label "MAIN" to the current program address.
**MoDem (MD)**

**Function**

Outputs a string of characters to the modem and waits for a response from the modem.

**Syntax**

```
MD | text
```

where:

- **Text** = the string of characters to send.

**Affected System Variables**

| &99 = | “0” if no error occurs  
|       | “1” if error is detected  
| &98 = | last modem answer, if modem response received;  
|       | “NO CRADLE” if the user does not connect the scanner to a cable or cradle within 30 seconds of executing this command;  
|       | “TIMEOUT” if the modem did not respond in the time specified in the last PI command (not including the 30 second period waiting for the user to cradle the scanner);  
|       | “SER. ERROR” if an error occurs during the transmission to the modem (such as a handshaking error if handshaking is enabled). |

**Notes**

1. A <CR> character (ASCII 13) is added automatically at the end of the string.
2. The RS232 communication parameters (baud rate, parity, etc.) are used.
3. The RS232 communications must be set to use NO software handshaking.
4. Unless the Host Override Enable barcode is scanned, the user has 30 seconds after this command is encountered to place the scanner into the cradle or connect to the cable. If the user does not do so, the MD command will return an error status (&99 = 1).
5. Once the commanded string has been sent to the modem, the scanner will wait a period of time determined by the most recent PI command for the response from the modem. If no response is received, the MD command will return an error status (&99 = 1).
6. Only the first 10 characters of a response message are placed in system variable 98.
7. A valid response code (&99 = 0) does NOT indicate that the modem successfully executed the command, only that the modem received the command and responded to it. In order to determine the modem result, the user should examine &98 as well.

**Example**

```
MD | ATDT5551212
```

Dial the phone number 555-1212.
Menu List (ML)

Function
Defines a list input field, where several options may be accessed using a single field, scrolling through the available options. The label associated with the selected text is taken as a jump address.

Syntax
ML | line | column | height | width | default | mode | speedkey | sel1 | skip1 ... [ | seln | skipn ]

where:
- line = row on which to begin field.
- column = column in which to begin field.
- height = number of lines to be used (ignored by Phaser).
- width = number of columns to be used.
- default = index of default selection.
- mode = rollover mode
  - R = Roll from last to first.
  - S = Scroll from first to last.
- speedkey = enable speed key processing. Note that the speedkey feature only works with numerics (0-9).
  - 1: Pressing the key corresponding to the first character of a menu item moves directly to that selection.
  - 0: Disable the speedkey feature.
- sel1 = text of first available selection.
- skip1 label to jump to if the associated menu item is selected.
- seln = text of last available selection.
- skipn label to jump to if the associated menu item is selected.
Affected System Variables

| &98 | The code of the exit key or “TO” if time-out occurs. |

Notes

1. The PI command is active in the ML command, so all defined hot-keys are still available.
2. If <UP> (UP Arrow) or <DW> (Down arrow) are used in PI before ML command, they will perform the action defined in PI only on <UP> on first item and <DW> on last item, and then only if mode is set to S. Otherwise, <UP> and <DW> switch the current selected item in the list.

Example

ML | 2 | 2 | 1 | 18 | 1 | S | 0 | 1. Inventory | INVEN | 2. POS | POS | 3. Shipping | SHIP

Define, at row 2, column 2, for 18 characters, an input list field which will scroll through the following selections, starting with menu item 1:

1. Inventory
2. POS
3. Shipping

If the user hits the UP key when option 1 is displayed, the program will take whatever action is currently defined in the PI command for an up key. If the user hits the DOWN key when option 1 is displayed, option 2 will be displayed. The selected label associated with the selection will be jumped to. Speedkey processing is disabled.
Network Input (NI)

Function
Waits for a command from the network and stores it in a variable.

Syntax
NI | source | command | variable | variable | variable

where:
- source = the source unit identification (3 characters; usually “099”).
- command = the expected command
- variable = the variable receiving the command arguments.

Affected System Variables
Normal Mode:

| &97 = | Source Identifier. |
| &98 = | Received Command. |
| &99 = | “0” |
| variable = | Received Data. |

After TIMEOUT (if one has been defined in the previous PI command):

| &97 = | “TO” |
| &98 = | “?” |
| &99 = | “1” |
| variable = | “?” |
After KEY (defined in previous PI command) is pressed:

<table>
<thead>
<tr>
<th>&amp;97</th>
<th>“KB”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>Name of the pressed key (see PI command for key names).</td>
</tr>
<tr>
<td>&amp;99</td>
<td>“0”</td>
</tr>
<tr>
<td>variable</td>
<td>“?”</td>
</tr>
</tbody>
</table>

After an error (on the P370/P470 RF scanner):

<table>
<thead>
<tr>
<th>&amp;97</th>
<th>“?”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>Error number.</td>
</tr>
<tr>
<td>&amp;99</td>
<td>“1”</td>
</tr>
<tr>
<td>variable</td>
<td>“?”</td>
</tr>
</tbody>
</table>

Notes

1. If source is equal to “??”, command is accepted from any source.
2. If command is equal to “??”, any command received will be accepted.
3. Source and destination address are represented in either decimal digits or hex digits, depending on the setting of the MCL-Net Hex Addressing parameter (System Variable 29).
4. Commands supported via the MCL-Net protocol are defined in the MCL-Link User’s Guide (part number 70-33346-XX). The following commands documented in the MCL-Link User’s Guide are NOT supported by the P460:
   - FA: File Append
   - FC: File Copy
   - FD: File Delete
   - FN: File Rename
5. Unsupported commands (commands not listed in the MCL-Link User’s Guide) can be input if specified explicitly in the command field. This allows extensions to the MCL-Net command language without upgrades to the runtime code.
6. Error numbers reported in &98 (on the P370/P470 RF scanner) are defined as follows:

14 RF Transmission error. Previous RF transmission not yet complete.
15 RF Transmission error. Scanner is not paired properly to base or scanner is out of range.
16 RF Address error. Base to which scanner is paired has been paired with another scanner.
17 RF Transmission error. Scanner out of range.
18 MCL-Link Host unavailable. The base is connected to a Synapse cable, which cannot be used to communicate to MCL-Link.
20 Transaction error. The scanner is already busy communicating with the base.
21 Memory allocation failure. The scanner has run out of buffers necessary to manage communication. Repeated occurrence of this error should be reported to customer service.
24 RF Transmission error. The base is not responding. Repeated occurrence of this error should be reported to customer service.

Other error numbers should not occur during an NI operation, and any occurrence of other error numbers should be reported to customer service.

Example
NI | 04 | CT | A1

Wait for a CT command from address 04 and store the arguments in variable A1.
Network Output (NO) - P360/P460 Memory Scanner

Function
Sends a frame to the host via RS-232 using MCL-Link protocol.

Syntax
**NO** | destination | command | arg1 | ... | argn

where:
- destination = the destination unit identification.
- command = the command to send to the destination
- argument = the command argument(s) to send.

Affected System Variables
Normal Mode:

<table>
<thead>
<tr>
<th>&amp;99 =</th>
<th>“0”</th>
</tr>
</thead>
</table>

After connection failure (maximum retries reached):

<table>
<thead>
<tr>
<th>&amp;98 =</th>
<th>retry count</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;99 =</td>
<td>“1”</td>
</tr>
</tbody>
</table>

After Host Error (Host rejects message, or user fails to place scanner in cradle within 30 seconds):

<table>
<thead>
<tr>
<th>&amp;98 =</th>
<th>retry count</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;99 =</td>
<td>“1”</td>
</tr>
</tbody>
</table>
After KEY (defined in the most recent PI command) is pressed:

<table>
<thead>
<tr>
<th>&amp;97</th>
<th>“KB”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>Name of the pressed key (see PI command for key names).</td>
</tr>
<tr>
<td>&amp;99</td>
<td>“0”</td>
</tr>
<tr>
<td>variable</td>
<td>“?”</td>
</tr>
</tbody>
</table>

Notes
Source and destination addresses are represented in decimal or hex digits, depending on the setting of the MCL-Net Hex Addressing parameter.

Example
NO | 099 | D1 | &A0

Send the D1 command to station 99 with the value of variable A0 as an argument.
Network Output (NO) - P370/P470 RF Scanner

Function
Sends a frame to the host via RS-232 using MCL-Link protocol.

Syntax
NO | destination | command | arg1 | ... | argn

where:
- destination = the destination unit identification.
- command = the command to send to the destination
- argument = the command argument(s) to send.

Affected System Variables
Normal Mode:

<table>
<thead>
<tr>
<th>&amp;97</th>
<th>“?”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>“?”</td>
</tr>
<tr>
<td>&amp;99</td>
<td>“0”</td>
</tr>
</tbody>
</table>

After a CLEAR KEY (<FN><BS>) is pressed:

<table>
<thead>
<tr>
<th>&amp;97</th>
<th>“KB”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>“CL”</td>
</tr>
<tr>
<td>&amp;99</td>
<td>“1”</td>
</tr>
</tbody>
</table>

After an error:

<table>
<thead>
<tr>
<th>&amp;97</th>
<th>“?”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>error number</td>
</tr>
<tr>
<td>&amp;99</td>
<td>“1”</td>
</tr>
</tbody>
</table>
Notes

1. Source and destination address are represented in either decimal digits or hex digits, depending on the setting of the MCL-Net Hex Addressing parameter (System Variable 29).

2. Commands supported via the MCL-Net protocol are defined in the MCL-Link User’s Guide (part number 70-33346-XX).

3. Unsupported commands (commands not listed in the MCL-Link User’s Guide can be input if specified explicitly in the command field. This allows extensions to the MCL-Net command language without upgrades to the runtime code.

4. Error numbers reported in &98 are defined as follows:

   15 RF Transmission error. Scanner is not paired properly to base or scanner is out of range.
   16 RF Address error. Base to which scanner is paired has been paired with another scanner.
   17 RF Transmission error. Scanner out of range.
   18 MCL-Link Host unavailable. The base is connected to a Synapse cable, which cannot be used to communicate to MCL-Link.
   20 Transaction error. The scanner is already busy communicating with the base.
   21 Memory allocation failure. The scanner has run out of buffers necessary to manage communication. Repeated occurrence of this error should be reported to customer service.
   24 RF Transmission error. The base is not responding. Repeated occurrence of this error should be reported to customer service.
   247 MCL-Net Protocol error. MCL-Link responded with Busy message.
   249 MCL-Net Protocol error. MCL-Link rejected the command.
   253 MCL-Net Protocol error. MCL-Link did not respond. Base may not be properly connected to MCL-Link host.

Example

NO | 099 | D1 | &A0

Send the D1 command to station 99 with the value of variable A0 as an argument.
**MCL-Link Start (NS)**

**Function**
Calls MCL-Link communication program.

**Syntax**
NS

**Affected System Variables**

<table>
<thead>
<tr>
<th>&amp;97</th>
<th>“?”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>“?”</td>
</tr>
<tr>
<td>&amp;99</td>
<td>“0”</td>
</tr>
</tbody>
</table>

**Notes**
This command puts the scanner in MCL-Link “slave” mode. The scanner then waits for commands coming in on the serial line from the host computer. and executes them as if they had been encountered in a local MCL program.

A specific command from the host causes the MCL-Link program to terminate and return control to the calling MCL program.
**Parameterize Input (PI)**

**Function**
Sets the input time-out value and exit keys for all subsequent input operations (until another PI command is received).

**Syntax**
PI | code | label [ | code | label ] [ | code | label ] ...

where:
- **Label** = routine to execute when the defined key is used.
  (or when the time-out occurs)
- **Code** = represent a key value (see list of codes below).
  If numeric, represents a timeout in 15 msec increments.

<table>
<thead>
<tr>
<th>Key code values</th>
<th>Actual keys used</th>
</tr>
</thead>
<tbody>
<tr>
<td>F0 to F9 =</td>
<td>Function keys (F0 = &lt;F10&gt;, F1 = &lt;F1&gt;, F2 = &lt;F2&gt;, etc.).</td>
</tr>
<tr>
<td>UP or DW =</td>
<td>&lt;Up&gt; or &lt;Down&gt; arrow keys.</td>
</tr>
<tr>
<td>PU or PD =</td>
<td>&lt;FN+UP&gt; or &lt;FN+DOWN&gt; keys.</td>
</tr>
<tr>
<td>Ax =</td>
<td>Any alpha character on the keypad (AK = &lt;K&gt; key), uppercase only.</td>
</tr>
<tr>
<td>Nx =</td>
<td>Any character on the numeric keypad (N9 = &lt;9&gt; key).</td>
</tr>
<tr>
<td>EN =</td>
<td>&lt;Enter&gt; key.</td>
</tr>
<tr>
<td>CL =</td>
<td>&lt;FN+BKSP&gt; key.</td>
</tr>
<tr>
<td>SP =</td>
<td>special &lt;*&gt; key.</td>
</tr>
<tr>
<td>BS =</td>
<td>Backspace key.</td>
</tr>
<tr>
<td>PW =</td>
<td>Power mode switch</td>
</tr>
</tbody>
</table>

**Affected System Variables**
None
Notes

1. The PI command is active during IL, IX, NI, SI and ML input commands, and the NO output command. Maximum 20 entries (including TO).

Examples

**PI | 1000 | TO | F1 | HELP | F8 | NTPAY**  
Calls the TO (time-out) routine if no input is detected during 15 seconds (1000 x .015 sec).  
Jumps to the HELP routine if the <F1> key is pressed.  
Jumps to the NTPAY routine if the <F8> key is pressed.

**PI | 0 | TO | F1 | HELP | F2 | 127 | F3 | +1**  
Disable timeout (note “TO” label must be included anyway); jump to “HELP” if the F1 key is pressed; line 127 if the F2 key is pressed, and continue with next line if the F3 key is pressed.
Parameterize Time (PT)

Function
Sets the date and time.

Syntax
PT | date_time

where:

date_time = the new time and date with format: YYYYMMDDHHMMSS.

Affected System Variables
All date/time system variables (&40 - &46, &72 - &76) are implicitly updated.

Notes
1. Valid dates are in the range January 1, 1997 to December 31, 2096.
2. The PT command is not supported on the RF scanner, since the RF scanner has no internal clock. The command does not cause a syntax error, but is simply ignored.

Example

PT | 19980526131500 Sets the time and date to 13:15:00; 26 May 1998.
Parameter eXchange (PX)

Function
Changes the contents of a System Setup Variable. (Refer to System Setup Variables on page 7-98).

Syntax
PX | parameter | value

where:

Parameter = the System Setup Variable name.
Value = the new contents of the System Setup Variable.

Affected System Variables
The addressed system variable is updated.

Notes
None.

Examples
PX | 20 | 10          Writes the value “10” into System Setup Variable 20.
PX | 25 | &A0         Writes the contents of variable A0 into System Setup Variable 25.
Parameterize Scanner (PY)

Function
Control scanner parameters using parameter number/parameter value pairs.

Syntax
PY | P | [pp]nn | vv | | [pp]nn | vv | | [pp]nn | vv | ...

where:

P = the “permanent change” flag.

Y = Make these changes permanent (write to FLASH parameter buffer).

N = Make these changes temporary (write only to RAM buffer). The previous value will be restored the next time the scanner is powered up.

nn = a single byte parameter number (value 0x00 – 0xEF).

vv = a single byte parameter value (legal values determined by the associated parameter number).

pp = an optional single byte prefix number. If a parameter number greater than 239 (0xEF) is required, then a prefix number is used as follows:

0xF0 = Add 256 to the next parameter number value
0xF1 = Add 512 to the next parameter number value
0xF2 = Add 768 to the next parameter number value

Affected System Variables
None.

Notes
1. The parameter numbers and the parameters they correspond to, along with the allowed and default values for those parameters, is shown in PY Parameter Numbers on page 7-103.

2. The reason for the prefix character is so that the vast majority of parameters may be specified in 2 bytes, thus compacting the command size, and reducing the communication time necessary to communicate these parameters over a serial interface (the historical basis of this protocol).

3. The PY command may contain up to 24 parameter number/parameter value pairs.
4. Changing the Power Down Time (Parameter Number 0x0C) does not affect the initial power down time. After a reset, the scanner will remain powered for 1 minute in order to charge the real-time clock backup capacitor. Subsequent to this 1 minute period, the Power Down Time will be in effect.

**Examples**

PY | N | 01 | 00 | 12 | 04 | F001 | 01 | F101 | 00

Parameter number 1 is assigned value 0;
Parameter number 12 is assigned value 04;
Parameter number 257 is assigned value 1;
Parameter number 513 is assigned value 0.
Quit or eXit (QX)

Function
Terminates execution of the program and gives control to the calling program, or to the operating system, depending on the type of return defined in the command.

Syntax
QX | type

where:

Type = the type of return control procedure:
1 = return control to the calling program.
2 = abort the MCI program (return to system startup).

Affected System Variables
None

Note
A QX command is mandatory to exit any program called by the RM command. If QX is omitted after execution of the RM command, System Error “9” is displayed.
The QX command can also be used to return from a SR (Skip and Return) command.

Examples

DC | 0
Clears all the lines on the scanner display.

DM | 2 | 4 | N | HELLO
Displays the message “HELLO” on the second line column 4.

IX | 2 | 10 | A1 | 1 | 1 | Y | N | Y | A | Y | N
Accepts a “one character” entry on the keyboard and stores it in A1.

QX | 2
Abort the MCL program and start over.
Run Program (RM)

Function
Calls a program and starts the execution at the beginning of the called program or from a given label.

Syntax
RM | progname [ | Label]

where:

  Progname = name of the called program.

  Label = label within the called program to execute at.

If a label is omitted, execution begins at line 1 of the called program.

Affected System Variables

| &99          | “0” if no error, “1” if the program is not found, or the execution failed. |

Notes

1. After execution, control is returned either to the calling program (at the next instruction after the call), or to the OS, depending on how the called program terminates:

2. A maximum of 6 nested program calls can be used.

3. A QX command is mandatory to exit any program called by the RM command. If QX is omitted after execution of the RM command, System Error “9” is displayed.

4. An SK can be used to jump directly to a label in the same or another program.
Examples

DC | 0  Clears all the lines on the scanner display.

RM | 2  Calls and executes the program in file 2.

DC | 0  After execution of program 2, clears all the lines on the scanner display.
     In this example, it is assumed that program 2 terminates with a QX | 1 command.
Serial Input (SI)

Function
Accepts input from a serial port and stores the data in variable.

Syntax
SI | port | variable | beginchar | endchar | maxchar | trailingchar

where:
- Port = communication port (ignored for Phaser).
- Variable = the variable receiving the input characters.
- Beginchar = the starting character or string of the data string.
- Endchar = the last character or string of the data string.
- Maxchar = maximum number of characters to receive.
- Trailingchar = number of trailing character(s) to accept after “endchar” or “maxchar” (max. 9).

Affected System Variables
If a TIMEOUT occurs (if one has been defined in the previous PI command):

| \&97 | “TO” |
| \&98 | “?” |
| \&99 | “0” |

If a KEY (defined in the previous PI command) is pressed:

| \&97 | “KB” |
| \&98 | name of the pressed key (see PI command for key names). |
| \&99 | “0” |
If a CLEAR KEY (not defined in the previous PI command) is pressed:

<table>
<thead>
<tr>
<th>&amp;97</th>
<th>“KB”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>“CL”</td>
</tr>
<tr>
<td>&amp;99</td>
<td>“1”</td>
</tr>
</tbody>
</table>

If an error occurs on the P370/P470 RF scanner:

<table>
<thead>
<tr>
<th>&amp;97</th>
<th>“?”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>error number.</td>
</tr>
<tr>
<td>&amp;99</td>
<td>“1”</td>
</tr>
</tbody>
</table>

Notes

1. Only the data included between the begin and the end characters are stored in the variable (beginchar and endchar characters excluded).
2. Serial Input terminates when a timeout occurs.
3. You can use “??” to ignore a parameter.
4. The maximum length of input data is 128 bytes.
5. Error numbers reported in &98 (in the P370/P470 RF scanner) are defined as follows:

   5   Serial protocol error. Bad characters received.
   15  RF Transmission error. Scanner is not paired properly to base or scanner is out of range.
   16  RF Address error. Base to which scanner is paired has been paired with another scanner.
   17  RF Transmission error. Scanner out of range.
   18  MCL-Link Host unavailable. The base is connected to a Synapse cable, which cannot be used to communicate to MCL-Link.
   20  Transaction error. The scanner is already busy communicating with the base.
   21  Memory allocation failure. The scanner has run out of buffers necessary to manage communication. Repeated occurrence of this error should be reported to customer service.
   24  RF Transmission error. The base is not responding. Repeated occurrence of this error should be reported to customer service.
### Examples

**SI | 1 | B1 | \02 | \03 | 127 | 1**

Accepts all data on serial port, starting with the character STX (hex. 02) and ending with the character ETX (hex. 03). When ETX is received, accepts the LRC character into the variable. Stops the input if 128 characters are received \((127 + 1 \text{ trailing char.})\).
**SKip (SK)**

**Function**
Performs an unconditional program jump to a label.

**Syntax**

```
SK | label
```

where:

- **Label** = address of the routine to execute.
  - If label is unsigned numeric: label indicates an absolute line number within the current program to jump to.
  - If label is signed numeric: label indicates a relative jump (referenced from the current line number). “+1” indicates no jump (execute the next physical line next).
  - If label is alphanumeric: label indicates a named label as defined in an LB command line.

**Affected System Variables**
None

**Note**
The scanner displays “System Error 3” if the label does not exist.

**Examples**

- **SK | +1** Jumps forward one line.
- **SK | -5** Jumps backward 5 lines.
- **SK | MAIN** Jumps to the “MAIN” routine.
- **SK | 127** Jumps to line 127.
Serial Output (SO) - P360/P460 Memory Scanner

Function
Sends a string of characters to the serial communication port.

Syntax
SO | descriptor | text

where:

Descriptor = descriptor string for the barcode to be transmitted (see command Input eXtended (IX) on page 7-45 for a discussion of the barcode descriptor.

???: Use default barcode descriptor (data transmits as Code 39).

T: User default barcode descriptor and transparent mode (keycat/keycode values are ignored – see notes below).

Other: Data transmitted using barcode descriptor provided. Value provided should be the contents of system variable 91 after the associated data was scanned.

Text = the string of characters to send.

Affected System Variables

&99 = “0” if no error, “1” if transmission fails (handshaking failure).

Notes

1. The text to be sent may include ~xyyy embedded within it. This is interpreted as KEYCAT x/KEYCODE yyy, used to insert special characters into the data stream.

2. If the text to be transmitted represents a single barcode, then the descriptor should be the descriptor which was returned in &91 when the code was scanned. If the text represents non-barcode data, or barcode data supplemented with other data, or anything other than a single barcode, the descriptor should be supplied as “??”. In this case, the host interface software will generate the necessary descriptor data.
3. If the text to be transmitted does not contain any KEYCAT/KEYCODE data, but does include the “~” character, then the descriptor must be set to ‘T’ to allow for “transparent” (non-translated) mode transmission.

4. To send a “~” character embedded in the barcode, the String Serial Output command must be used, or select transparent mode.

5. Unless the Host Override Enable barcode has been scanned, once the SO command is encountered, the user has 30 seconds to place the scanner in the cradle or to connect power. After 30 seconds, an error will be reported (\&99 = 1).

6. If the RS232 interface is set up to user an ACK/NAK, ENQ, or ENQ/ACK/NAK handshaking protocol, and the interface times out waiting for data from the host, a host error will be reported (\&99 = 1). Processing will proceed with the next MCL program statement, regardless of any Timeout branches specified in the most recent PI command.

Examples

SO | ?? | HELLO WORLD
Sends the characters HELLO WORLD to serial port using default descriptor.

SO | &91 | &A1~7013
Sends the contents of variable A1 followed by a carriage return (Category 7, value 013), using formatting defined by descriptor in system variable 91.
Serial Output (SO) - P370/P470 RF Scanner

Function
Sends a string of characters to the serial communication port.

Syntax
SO | port | text

where:
Port = communication port (ignored for Phaser).
Text = the string of characters to send.

Affected System Variables
If a CLEAR KEY is pressed:

<table>
<thead>
<tr>
<th>&amp;97</th>
<th>“KB”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>“CL”</td>
</tr>
<tr>
<td>&amp;99</td>
<td>“1”</td>
</tr>
</tbody>
</table>

If an error occurs:

<table>
<thead>
<tr>
<th>&amp;97</th>
<th>“?”</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>error number.</td>
</tr>
<tr>
<td>&amp;99</td>
<td>“1”</td>
</tr>
</tbody>
</table>
Notes

1. The maximum length of output data is 506 bytes.
2. Error numbers reported in &98 are defined as follows:

   2  Serial protocol error. Configured host does not recognize transmitted characters.
   3  Serial protocol error. Handshaking failure.
   5  Serial protocol error. Bad characters received.
   8  SYNAPSE error. Configured host does not recognize transmitted characters.
   9  SYNAPSE error. SYNAPSE transmission failed. SYNAPSE cable may not be properly installed.
  15  RF Transmission error. Scanner is not paired properly to base or scanner is out of range.
  16  RF Address error. Base to which scanner is paired has been paired with another scanner.
  17  RF Transmission error. Scanner out of range.
  18  MCL-Link Host unavailable. The base is connected to a Synapse cable, which cannot be used to communicate to MCL-Link.
  20  Transaction error. The scanner is already busy communicating with the base.
  21  Memory allocation failure. The scanner has run out of buffers necessary to manage communication. Repeated occurrence of this error should be reported to customer service.
  24  RF Transmission error. The base is not responding. Repeated occurrence of this error should be reported to customer service.

Examples

SO | 1 | HELLO WORLD          Sends the characters HELLO WORLD to serial port.
SO | 1 | &A1 WORLD \0D \0A     Sends the contents of variable A1 followed by “WORLD” and a carriage return/line feed pair to the serial port.
Skip and Return (SR)

Function
Performs an unconditional program jump to a sub-routine pointed to by a label.

Syntax
SR | label [ ] Progname

where:
- Label = address of the routine to execute.
  - If label is unsigned numeric: label indicates an absolute line number within the current program to jump to.
  - If label is signed numeric: label indicates a relative jump (referenced from the current line number). “+1” indicates no jump (execute the next physical line next).
  - If label is alphanumeric: label indicates a named label as defined in an LB command.
- Progname = (Optional) MCL Program Name

Affected System Variables
None

Notes
1. QX command will return program control to the statement immediately following the SR command.
2. The terminal will display “ERROR 3” if the label does not exist.

Examples
- SR | +1 Jump to sub-routine starting one line forward.
- SR | -5 Jump to sub-routine starting five lines backwards.
- SR | SUBR Jump to the “SUBR” sub-routine.
- SR | TEST | 2 Jump to sub-routine starting at label TEST in MCL program 2.
Variable Compare (VC)

Function
Compares the content of a variable with a pattern of characters.

Syntax
VC | source | pattern | Skip True | Skip False

where:

Source = the text to be tested.
Pattern = the test returns TRUE if the first characters of Source match Pattern, FALSE otherwise. If Pattern is longer than Source, the test fails.

A to Z or a to z - specific character.
0 to 9 - specific digit.
" - Any alphabetic character.
$ - Any numeric character.
? - ignore test on this character.

True = Skip address if comparison is true.
False = Skip address if comparison is false.

Affected System Variables
None

Notes
1. This command compares all the characters in the variable with a pattern described in the command. The variable must match all the characters of the pattern, so the variable must have at least as many characters as the pattern.
2. When an alphabetic or numeric character is placed in the pattern, the variable must perfectly match the letter or number in the pattern. Otherwise, to test if a character is alphabetic or numeric, " mask must be used, or $ for a digit.
Examples

VC | &A1| A 1 "? | +1 | VCERR  Compares variable A1 with following pattern:
first character must be ‘A’.
second character must be ‘1’.
third character must be Alphabetic (A-Z, a-z).
fourth character must be numeric ( 1-9 ).
fifth character is not tested.
If matched, go to next line ( +1) else go to label VCERR.
Variable Extract (VE)

Function
Extracts all characters of the defined field and stores them in variable. (Fields are delimited by the separator defined in the command).

Syntax
VE | variable | text | fieldnr | separator

where:

Variable = the name of the receiving variable.
Text = the text to store in variable.
Fieldnr = field number to be extracted.
Separator = character defining the beginning and end of each field. If no separator is used, all text is stored in the variable.

Affected System Variables

| &99 | “0” if no error, “1” if field not found (or field does not exist). |

Note
1. If “|” is the separator, use “\7C” to specify the “|” character.

Example

VE | A1 | JOHN + ANDREW + PETER | 2 | + Extracts the second field from the text. Fields are separated by “+” (After execution: A1 = “ANDREW”).

VE | A0 | &A1 | 2 | \7C Extract the second field from the text in variable A1, using ‘|’ as the separator character, and place the result in variable A0.
Variable Format (VF)

Function
Formats the data in a variable as a string of characters or numbers.

Syntax
VF | type | destination | origin | sign | zero | int | dec]

where:

Type = type of format.
C = characters.
D = numbers.
I = Integer.
N = Numeric.

Destination = the name of the receiving variable.

Origin = the original string to be formatted (may be a variable reference).

Only with numeric format:

Sign = add sign to the number.
S = add ‘+’ or ‘-’ before number.
B = add space or ‘-’ before number.
N = no sign.

Zero = fill the number with a specified character.
Z = fill with ‘0” before number.
B = fill with space before number.
N = remove all “0” and space before number.
Anything else = fill with specified character before number

Int = number of digits of integer part.
Dec = number of digits of decimal part.
Affected System Variables

<table>
<thead>
<tr>
<th>&amp;99</th>
<th>“0” if no error, 1 otherwise.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;98</td>
<td>“+” if result &gt;= 0 (only with numeric formatting).</td>
</tr>
<tr>
<td></td>
<td>“-” if result &lt; 0 (only with numeric formatting).</td>
</tr>
</tbody>
</table>

Notes

1. When data is defined as characters (C), all non significant zeroes and spaces on the left and right of the variable are deleted.
2. If the data contains decimal numbers, the zero just before the decimal point is not deleted.
3. When the data is defined as a number (D), the integer part is 7 digits in length and the decimal part is 4 digits in length.
4. When the data is defined as a number (I), the return value is coded in 5 characters.

Examples

**Format C (characters)**

```
VF | C | A1 | &B0
```

Format the data in variable B0 as characters and store the result in variable A1.

Example:

- Before execution: B0 = “0000125.3500”
- After execution: A1 = “125.35”

Example:

- Before execution: B0 = “0000000.3500”
- After execution: A1 = “0.35”

Example:

- Before execution: B0 = “0000100.0000”
- After execution: A1 = “100”

Example:

- Before execution: B0 = “0000000.0000”
- After execution: A1 = “0”

Example:

- Before execution: B0 = “00100”
- After execution: A1 = “100”
Format D (numbers)

VF | D | A1 | &F1 Formats the data in F1 as numbers with decimals, and stores the result in variable A1.  
Example:  
Before execution : F1 = “100”  
After execution : A1 = “0000100.0000”

VF | D | G1 | &A0 Formats the data in A0 as numbers with decimals, and stores the result in variable G1.  
Example:  
Before execution : A0 = “122.57”  
After execution : G1 = “0000122.5700”

Format I (Integers)

VF | I | A1 | 4 A1 returns 00004.

Format N (numeric)

VF | N | A0 | +123747.7489 | S | B | 9 | 6 A0 returns +···123747.748900 where the ‘·’ represents a space.

VF | N | A0 | +123747.7489 | S | B | 4 | 6 A0 returns +9999.999999 (an overflow condition).

VF | N | A0 | +123747.7489 | S | Z | 9 | 6 A0 returns +000123747.748900

VF | N | A0 | +123747.7489 | S | N | 9 | 6 A0 returns +123747.748900

VF | N | A0 | +123747.7489 | S | B | 9 | 6 A0 returns +···123747.748900

VF | N | A0 | +123747.7489 | B | B | 9 | 6 A0 returns ···123747.748900

VF | N | A0 | +123747.7489 | B | Z | 9 | 6 A0 returns 000123747.748900
**Variable Justify (VJ)**

**Function**
Justifies the data in a variable to the left or to the right, and fills gaps with characters.

**Syntax**

```
VJ | type | destination | origin | length | character
```

where:

- **Type** = type of justification.
  - L = left.
  - R = right.
- **Destination** = the name of the receiving variable.
- **Origin** = the original string to be formatted (may be a variable reference).
- **Length** = the total length of the justified data.
- **Character** = the character with which to fill the field (any ASCII character may be used).

**Affected System Variables**

| &99 | “0” if no error occurs, 1 otherwise. |

**Notes**

1. If a space is used as an ASCII character, it must be coded as “\20”.
2. If the length parameter is less than the length of the origin string, the origin will be copied to destination unchanged.
Examples

VJ | L | A1 | &A0 | 10 | *  Left-justifies variable A0 for a length of 10 characters and fill with asterisks (*)
Example:
Before execution : A0 = “HELLO”
After execution : A1 = “HELLO*****”

VJ | R | A1 | &A0 | 6 | $  Right-justifies variable A0 for a length of 6 characters and fill with “$”
Example:
Before execution : A0 = “100”
After execution : A1 = “$$100”
Variable Length (VL)

Function
 Computes the size of a text string and stores it in a variable.

Syntax
 VL | variable | text

where:
 Variable = the variable receiving the computed length.
 Text = the text to evaluate.

Affected System Variables
 None

Notes
 None

Examples

VL | A1 | HELLO_WORLD  Computes the size of “HELLO_WORLD” and stores it in variable A1

VL | A1 | &B0  Computes the size of the text in variable B0 and stores the result in
variable A1.
Example:
Before execution: B0 = “GOODBYE”
After execution: A1 = “7”.
Variable Move (VM)

Function
Replaces the current contents of a variable with text.

Syntax
VM | variable | text

where:

Variable = the name of the receiving variable.
Text = the text to store in variable.

Affected System Variables

| &99 | = 0, if no error occurs, 1 otherwise. |

Examples

VM | A1 | HELLO
Stores the word “HELLO” in variable A1.
After execution: A1 = “HELLO”

VM | A0 | &A1 WORLD
Concatenates the contents of variable A1 with the word “WORLD”, and stores the result in variable A0.
Example:
Before execution: A1 = “HELLO”.
After execution: A0 = “HELLO WORLD”
Variable Operation - Arithmetic (VO)

Function
Executes arithmetic operation on number1 and number2, and stores the result in variable.

Syntax
VO | type | variable | number1 | number2

where:

Variable = the name of the receiving variable.
Number1 = the first operand.
Number2 = the second operand.
Type = Operation to perform:
+ to add.
- to subtract.
* to multiply.
/ to divide.
% to compute Modulo.

Affected System Variables

<table>
<thead>
<tr>
<th>&amp;98</th>
<th>the sign of the result.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;97</td>
<td>remainder of divide operation.</td>
</tr>
<tr>
<td>&amp;96</td>
<td>the sign of the remainder.</td>
</tr>
</tbody>
</table>

Notes
1. When arithmetic operations with decimal positions are needed, the formatting of the result may be required.
2. Addition, Subtraction, Multiplication and Division operations work on integer (0 to 65535) and decimal numbers.
3. Modulo operation only works on integers (0 to 65535).
4. The remainder of the divide command is stored in variable &97. You can use the modulo operation to get this result directly.
5. In case of overflow, the output is equal to 99999 and the system variable 99 = 1.
6. The output format is always on 11 digits: 0000000.0000 to 9999999.9999.

Examples

**VO | + | A1 | &A1 | 1**  Increments A1 by 1.

**VO | - | A1 | &B1 | &D1**  Subtracts contents of variable D1 from variable B1 and stores the result in A1.
**Variable Substring (VS)**

**Function**
Extracts the number of characters defined by size from a text string, starting at the position defined by begin, and stores these extracted characters in a variable.

**Syntax**

```
VS | variable | text | begin | size
```

where:
- **Variable** = name of the variable receiving the characters.
- **Text** = text string from which the data must be extracted.
- **Begin** = starting position of the string to extract.
- **Size** = number of character to extract.

**Affected System Variables**

```
&99 = 0, if no error occurs, 1 otherwise.
```

**Notes**

1. If the starting position (begin) lies outside of text string, or if the number of characters to extract (size) is too large, variable will contain an empty string (empty string = “?”).
2. Use VE command if the variable contains “ | ” characters.
3. If the Size is 0 (zero), then the string will be extracted from Begin to the end of the Text.
Examples

**VS | A1 | HELLO_WORLD | 7 | 5**  
Extracts “WORLD” from the text and store it in A1.  
Beginning position = 7, size = 5 characters.  
After execution : A1 = “WORLD”

**VS | A0 | &A1 | 2 | 3**  
Extracts text from A1, store it in A0.  
Beginning position = 2, size = 3 characters.  
Example:  
Before execution: A1 = WORLD.  
After execution : A0 = “ORL”.
Variable operation - Arithmetic- eXtended (VX)

Function
Executes arithmetic operation on number1 and number2, and stores the result in a variable with a specified numeric format (see VF | N command).

Syntax
VX | type | variable | number1 | number2 | sign | zero | int | dec

where:

Variable = the name of the receiving variable.
Number1 = the first operand.
Number2 = the second operand.
Type = the operation to perform:
   + to add.
   - to subtract.
   * to multiply.
   / to divide.
   % to compute Modulo.
Sign = add sign to the number.
   S = add ‘+’ or ‘-’ before number.
   B = add space or ‘-’ before number.
   N = no sign.
Zero = fill the number with a specified character.
   Z = fill with ‘0’ before number.
   B= fill with space before number.
   N = remove all ‘0’ and space before number.
Int = number of digits of integer part.
Dec = number of digits of decimal part.
**Affected System Variables**

| &99 =  | 0, if no error occurs. |
| &98 =  | the sign of the variable. |
| &97    | remainder of divide operation. |
| &96    | the sign of the remainder. |

**Notes**

1. If the size of the decimal part is set to 0, the decimal point will be omitted.
2. If the size of the integer part is set to zero, the first character of the result will be a decimal point.
3. If both int and dec are set to 0, a syntax error (Error 4) will be declared.

**Examples**

```
VX | + | A1 | 123747.7489 | 1 | S | B | 9 | 6  | result A1 = +···123748.748900.
VX | - | A1 | 123747.7489 | 1 | S | Z | 9 | 6  | result A1 = +000123746.748900.
```
**Wait (WT)**

**Function**
Suspends execution of the current program for a defined delay period.

**Syntax**

```
WT | length [ | abort ]
```

where:

- **Length** = length of delay (in 15-millisecond increments).
- **Abort** = Allow wait to be aborted by user intervention:
  - Y = Keyboard or trigger will cause wait period to be aborted.
  - N or absent = Keyboard and trigger will have no effect.

**Affected System Variables**
None.

**Notes**
The maximum allowed wait is 60 seconds. Anything entered above 60 seconds will default to 60 seconds.

**Examples**

- **WT | 20** Current program will wait 300 milliseconds (20 x 15 ms), then continue.
- **WT | &A1 | Y** Current program will wait “n” times 15 milliseconds, then continue.
  (“n” is the value contained in variable A1). Pressing any key will abort the wait and move on to the next program step.
Programs and File Sizes

Programs

- maximum 9 programs
- maximum 512 lines per program
- maximum program size 4 KB (4095 bytes)

Files

- size of a data file is limited by the scanner's available memory.
- size of a record is limited to 255 characters (including field separator and \texttt{CR LF} characters).
- a file may not contain more than 9998 records regardless of the record size.
- total user file space = 256 KB on a 512 KB unit
  768 KB on a 1 MB unit
  3584 KB on a 4 MB unit
  7680 KB on an 8 MB unit
MCL-Code Error Messages

After the execution of a MCL-Code command, variable &99 generally reports the result of the command execution. Table 7-5 lists the standard error messages used by the MCL-Code commands. The error messages are displayed on the scanner display and the program is suspended. After pressing the ENTER key, the program returns to the operating system. Any previously stored data is retained intact, although the record entry on which the error occurs will be lost.

Table 7-5. Error Code Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
</table>
| 2          | error on File operation  
- File name is invalid  
- number of fields per records too big  
- not enough free space on scanner: check system var. 32 |
| 3          | branch error  
- jump out the scope of the current program  
- label not found, too long |
| 4          | syntax error in a MCL command  
- command code invalid  
- mandatory argument missing or invalid  
- bad number of arguments:  
  check command syntax  
  check if any variable contains “ | “ characters, if yes, use VE command to extract desired field  
  check line length (max. 512 characters) after variable substitution (variables are replaced by their contents) |
| 5          | variable error  
- variable too big (in write command)  
- variable name invalid |
| 6          | key error  
- in PI or IK command, invalid key definition  
- too many entries in PI or IK commands |
Table 7-5. Error Code Messages (Continued)

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
</table>
| 7          | serial error  
- attempt to execute SI or SO command on the Network communication port. |
| 9          | run program error  
- too many RM command without the associated QX command (max. 6 nested programs). |
System Setup Variables

Table 7-6 lists the available system variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| 02       | 4             | Country page code:  
1 = French.  
2 = Dutch.  
3 = British English.  
4 = American English. |
| 14       | 6             | Baud rate:  
2 = 600 BPS.  
3 = 1200 BPS.  
4 = 2400 BPS.  
5 = 4800 BPS.  
6 = 9600 BPS.  
7 = 19200 BPS.  
8 = 38400 BPS. |
| 15       | 0             | Default Error Beep Override:  
0: Disabled  
1: Enabled  
If the default Error Beeps are overridden, communications errors do not cause error beeps. Error detection must be coded into your application. You can then use whatever notification method (including display messages and beeps from the BP command table) is appropriate. Note, however, that if Default Beep Override is enabled, you MUST be diligent in coding error detection, or data loss may go unnoticed. |

*: Read Only variable
### Table 7-6. System Setup Variables (Continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Bar-code type mask: Y = enable, N = disable. Contains a 16 characters mask (one character per symbology).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y char. 1 = UPC A (leftmost character).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y char. 2 = UPC E0.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N char. 3 = UPC E1.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y char. 4 = EAN13.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y char. 5 = EAN 8.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N char. 6 = Standard 2/5.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y char. 7 = Interleave 2/5.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y char. 8 = Code 39.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N char. 9 = CODABAR.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y char. 10 = Code 128.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N char. 11 = Code 93.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N char. 12 = Code 11.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N char. 13 = MSI.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N char. 14 = Code 49.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N char. 15 = PDF417.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y char. 16 = EAN128 (rightmost character).</td>
<td></td>
</tr>
</tbody>
</table>
| 24       | 0070137013 | **Scan Options Control**: Variable which controls the format of data as transmitted from the default application. Note that unless the user specifically writes code to utilize this variable, the scan options control affects ONLY the default application. The format of this variable is:  

\[wxyyyyzzzz\]  
where:  

- **w** is the enable code for data prefix (0 is disabled, 1 is enabled)  
- **x** is the enable code for data suffix (0 is disabled, 1 is enabled)  
- **yyyy** is the keycat/keycode of the data prefix  
- **zzzz** is the keycat/keycode of the data suffix  

For example:  

0170137013 Specifies that data will be transmitted with a suffix of CR/LF (as specified by keycat/keycode 7013).  

* : Read Only variable
Table 7-6. System Setup Variables (Continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| 25       | 1             | Keyboard Click:  
|          |               | 0 = Off.  
|          |               | 1 = On.  |
| 27       | 4             | Display Contrast  
|          |               | 0 = Least Contrast (lightest).  
|          |               | 7 = Most Contrast (darkest).  |
| 28       | 512           | System Memory size, in KB*.  
|          |               | 512 = 512 KB  
|          |               | 1024 = 1024 KB (1 MB).  
|          |               | 4096 = 4096 KB (4 MB).  
|          |               | 8192 = 8192 KB (8 MB).  
|          |               | Note that 256KB of the memory is allocated for system code and user scripts on the 512 KB and 1 MB scanners, while 512 KB of the memory is allocated on the 4 MB and 8 MB scanners. So, on a 512 KB scanner, only 256 KB is available for data storage.  |
| 29       | 0             | Address format used by **MCL-Net**:  
|          |               | 0 = decimal addresses.  
|          |               | 1 = Hexadecimal addresses.  |
| 30       | 10            | Power off delay in seconds.  |
| 33       | 0             | Power source:*  
|          |               | 0 = batteries.  
|          |               | 1 = cradle.  
|          |               | 2 = cable.  |
| 35       |               | Batteries status: *  
|          |               | 0 = good.  
|          |               | 1 = low.  
|          |               | 2 = dead.  |
| 40       |               | Hours from internal clock system: format 00 to 23 *.  |
| 41       |               | Minutes from internal clock system: format 00 to 59 *.  |
| 42       |               | Seconds from internal clock system: format 00 to 59 *.  |

*: Read Only variable
### Table 7-6. System Setup Variables (Continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>43</td>
<td>Day from internal clock system: format 01 to 31 *.</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>Month from internal clock system: format 01 to 12 *.</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>Year from internal clock system: format 1980 to 2079 *.</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>Julian day: format 001 to 365 *.</td>
<td></td>
</tr>
</tbody>
</table>
| 52       | Date Format:  
1 = Not Used.  
2 = Not Used.  
3 = MM DD YYYY  
4 = DD MM YYYY | |
| 53       | Date Separator:  
1 = /  
2 = -  
3 = :  
4 = no separator. | |
| 54       | Hour format:  
1 = 24  
2 = 12 | |
| 55       | Decimal separator:  
0 = .  
1 = , | |
| 56       | TF_type = Supported TF (file transfer) format (scanner to host).  
0= TF Extended (Binary format support).  
3= TF Counter size in 3 digits.  
4= TF Counter size in 4 digits (Default value). | |
| 59       | ` | ` (ASCII 124) Field separator. | |
| 68       | None Program number of currently active program (0 - 8). | |
| 70       | Version information:*  
NBRVSMxx, where xx is AA, AB, AC, ... BA, BB, ... to indicate revision level of scanner code. | |

*: Read Only variable
Table 7-6. System Setup Variables (Continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>71</td>
<td>Scanner Identification number (001 to 254).</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>Date and time in format CCYYMMDDHHMMSS.</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>Date MM/DD/YY (depending on code page).</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>Time HH:MM:SS.</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>Day of week (0 = Sunday, 1 = Monday, ……).</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>AM/PM *.</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>None</td>
<td>Week number for the current date (1 - 53)</td>
</tr>
<tr>
<td>85</td>
<td>MCL Interpreter version x.x *.</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>Not supported.</td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>Version date *&lt;br&gt;<strong>format</strong>: formatted according to the Date Format and Date Separator parameters</td>
<td></td>
</tr>
<tr>
<td>90-99</td>
<td>Various return codes from MCL commands.</td>
<td></td>
</tr>
</tbody>
</table>

* : Read Only variable
## PY Parameter Numbers

The following table (Table 7-7) lists the PY Parameter Numbers.

### Table 7-7. PY Parameter Numbers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parameter Number</th>
<th>Options</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Code 39</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code 39</td>
<td>0x00</td>
<td>0 Disable</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Code 39 Length 1</td>
<td>0x12</td>
<td>Any length (len1 = len2 = 0)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range (len2 &gt; len1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Discrete lengths (len1 &gt; len2)</td>
<td>55</td>
</tr>
<tr>
<td>Verify Code 39 Check Digit</td>
<td>0x30</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Transmit Code 39 Check Digit</td>
<td>0x2B</td>
<td>0 Do Not Transmit</td>
<td>Do Not Transmit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Transmit</td>
<td></td>
</tr>
<tr>
<td>Code 39 Full ASCII</td>
<td>0x11</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Convert Code 39 to Code 32</td>
<td>0x56</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Code 32 Prefix</td>
<td>0xe7</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Trioptic Code 39</td>
<td>0x0D</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td><strong>UPC/EAN</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPC-A</td>
<td>0x01</td>
<td>0 Disable</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>UPC-E</td>
<td>0x02</td>
<td>0 Disable</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>EAN-13</td>
<td>0x03</td>
<td>0 Disable</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
</tbody>
</table>
### Table 7-7. PY Parameter Numbers (Continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parameter Number</th>
<th>Options</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>UPC/EAN - Continued</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EAN-8</td>
<td>0x04</td>
<td>0 Disable 1 Enable</td>
<td>Enable</td>
</tr>
<tr>
<td>UPC-E1</td>
<td>0x0C</td>
<td>0 Disable 1 Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>Bookland EAN</td>
<td>0x53</td>
<td>0 Disable 1 Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>Coupon</td>
<td>0x55</td>
<td>0 Disable 1 Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>Decode UPC/EAN Supplementals</td>
<td>0x10</td>
<td>0 Ignore Supps 1 Decode Supps only 2 Auto-discriminate Supps</td>
<td>Ignore Supps</td>
</tr>
<tr>
<td>Supplementals Redundancy</td>
<td>0x50</td>
<td>2-20</td>
<td>7</td>
</tr>
<tr>
<td>Transmit UPC-A Check Digit</td>
<td>0x28</td>
<td>0 Disable 1 Enable</td>
<td>Enable</td>
</tr>
<tr>
<td>Transmit UPC-E Check Digit</td>
<td>0x29</td>
<td>0 Disable 1 Enable</td>
<td>Enable</td>
</tr>
<tr>
<td>Transmit UPC-E1 Check Digit</td>
<td>0x2A</td>
<td>0 Disable 1 Enable</td>
<td>Enable</td>
</tr>
<tr>
<td>UPC-A Preamble</td>
<td>0x22</td>
<td>0 None 1 System Character 2 System Character and Country Code</td>
<td>System Character</td>
</tr>
<tr>
<td>UPC-E Preamble</td>
<td>0x23</td>
<td>0 None 1 System Character 2 System Character and Country Code</td>
<td>System Character</td>
</tr>
</tbody>
</table>
### Table 7-7. PY Parameter Numbers (Continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parameter Number</th>
<th>Options</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UPC/EAN - Continued</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPC-E1 Preamble</td>
<td>0x24</td>
<td>0 None</td>
<td>System Character</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 System Character</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 System Character and Country Code</td>
<td></td>
</tr>
<tr>
<td>Convert UPC-E to UPC-A</td>
<td>0x25</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Convert UPC-E1 to UPC-A</td>
<td>0x26</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>EAN-8 Zero Extend</td>
<td>0x27</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>EAN-8 to EAN-13 Type</td>
<td>0xE0</td>
<td>0 Type is EAN-13</td>
<td>Type is EAN-13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Type is EAN-8</td>
<td></td>
</tr>
<tr>
<td>UPC/EAN Security Level</td>
<td>0x4D</td>
<td>0-3</td>
<td>0</td>
</tr>
<tr>
<td><strong>Discrete 2 of 5</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete 2 of 5</td>
<td>0x05</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Discrete 2 of 5 Length 1</td>
<td>0x14</td>
<td>Any length (len1 = len2 = 0)</td>
<td>12</td>
</tr>
<tr>
<td>Discrete 2 of 5 Length 2</td>
<td>0x15</td>
<td>Range (len2 &gt; len1)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Discrete lengths (len2 &lt; len1)</td>
<td></td>
</tr>
<tr>
<td><strong>Interleaved 2 of 5</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interleaved 2 of 5</td>
<td>0x06</td>
<td>0 Disable</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Interleaved 2 of 5 Length 1</td>
<td>0x16</td>
<td>Any length (len1 = len2 = 0)</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range (len2 &gt; len1)</td>
<td>0</td>
</tr>
<tr>
<td>Interleaved 2 of 5 Length 2</td>
<td>0x17</td>
<td>Discrete lengths (len2 &lt; len1)</td>
<td></td>
</tr>
<tr>
<td>Verify I 2 of 5 Check Digit</td>
<td>0x31</td>
<td>0 None</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 USS Check Digit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 OPCC Check Digit</td>
<td></td>
</tr>
</tbody>
</table>
### Table 7-7. PY Parameter Numbers (Continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parameter Number</th>
<th>Options</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interleaved 2 of 5 - Continued</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmit I 2 of 5 Check Digit</td>
<td>0x2C</td>
<td>0 Do not transmit</td>
<td>Do not transmit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Transmit</td>
<td></td>
</tr>
<tr>
<td>Convert I 2 of 5 to EAN-13</td>
<td>0x52</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td><strong>Codabar</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Codabar</td>
<td>0x07</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Codabar Length 1</td>
<td>0x18</td>
<td>Any length (len1 = len2 = 0)</td>
<td>5</td>
</tr>
<tr>
<td>Codabar Length 2</td>
<td>0x19</td>
<td>Range of lengths (len2 &gt; len1)</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Discrete Lengths (len2 &lt; len1)</td>
<td></td>
</tr>
<tr>
<td>CLSI</td>
<td>0x36</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Notis</td>
<td>0x37</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td><strong>Code 128</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code 128</td>
<td>0x08</td>
<td>0 Disable</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>EAN-128</td>
<td>0x0E</td>
<td>0 Disable</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td><strong>Code 93</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code 93</td>
<td>0x09</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Code 93 Length 1</td>
<td>0x1A</td>
<td>Any length (len1 = len2 = 0)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range of lengths (len2 &gt; len1)</td>
<td></td>
</tr>
<tr>
<td>Code 93 Length 2</td>
<td>0x1B</td>
<td>Discrete Lengths (len2 &lt; len1)</td>
<td>55</td>
</tr>
</tbody>
</table>
### Table 7-7. PY Parameter Numbers (Continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parameter Number</th>
<th>Options</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Code 11</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code 11</td>
<td>0x0A</td>
<td>0 Disable&lt;br&gt;1 Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>Code 11 Length 1</td>
<td>0x1C</td>
<td>Any length (len1 = len2 = 0)</td>
<td>4</td>
</tr>
<tr>
<td>Code 11 Length 2</td>
<td>0x1D</td>
<td>Range of lengths (len2 &gt; len1)&lt;br&gt;Discrete Lengths (len2 &lt; len1)</td>
<td>55</td>
</tr>
<tr>
<td>Verify Code 11 Check Digit</td>
<td>0x34</td>
<td>0 Disable&lt;br&gt;1 Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>Transmit Code 11 Check Digit</td>
<td>0x2F</td>
<td>0 Do not transmit&lt;br&gt;1 Transmit</td>
<td>Do not transmit</td>
</tr>
<tr>
<td><strong>MSI</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSI</td>
<td>0x0B</td>
<td>0 Disable&lt;br&gt;1 Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>MSI Length 1</td>
<td>0x1E</td>
<td>Any length (len1 = len2 = 0)</td>
<td>1</td>
</tr>
<tr>
<td>MSI Length 2</td>
<td>0x1F</td>
<td>Range of lengths (len2 &gt; len1)&lt;br&gt;Discrete Lengths (len2 &lt; len1)</td>
<td>55</td>
</tr>
<tr>
<td>MSI Check Digits</td>
<td>0x32</td>
<td>0 One check digit&lt;br&gt;1 Two check digits</td>
<td>One check digit</td>
</tr>
<tr>
<td>MSI Check Digit Scheme</td>
<td>0x33</td>
<td>0 Mod-11, Mod-10&lt;br&gt;1 Mod-10, Mod-10</td>
<td>Mod-10, Mod-10</td>
</tr>
<tr>
<td>Transmit MSI Check Digit</td>
<td>0x2E</td>
<td>1 Transmit&lt;br&gt;0 Do not transmit</td>
<td>Do not transmit</td>
</tr>
<tr>
<td><strong>General Transmit Options</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmit Code ID</td>
<td>0x2D</td>
<td>0 None&lt;br&gt;1 AIM ID's&lt;br&gt;2 Symbol ID's</td>
<td>None</td>
</tr>
<tr>
<td>Transmit “No Read” Characters</td>
<td>0x5E</td>
<td>0 Disable&lt;br&gt;1 Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>Parameter</td>
<td>Parameter Number</td>
<td>Options</td>
<td>Default Value</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>------------------</td>
<td>-----------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td><strong>General</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laser On Timeout</td>
<td>0x88</td>
<td>5 – 99 (1/10 second units)</td>
<td>30 (3 seconds)</td>
</tr>
<tr>
<td><strong>General Decode Parameters</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bi-directional Redundancy</td>
<td>0x43</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td>Linear Security Level</td>
<td>0x4E</td>
<td>1-3</td>
<td>1</td>
</tr>
<tr>
<td><strong>Beeper/User Interface Parameters</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beep on Good Decode (for use by default application)</td>
<td>0x38</td>
<td>0 Disable</td>
<td>Enable</td>
</tr>
<tr>
<td>Beeper Volume</td>
<td>0x8C</td>
<td>0 High</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Medium</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 Low</td>
<td></td>
</tr>
<tr>
<td>Beeper Frequency</td>
<td>0x91</td>
<td>0 High</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Medium</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 Low</td>
<td></td>
</tr>
<tr>
<td><strong>Primary Serial Port Parameters</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response Timeout</td>
<td>0x9B</td>
<td>5 – 99 (1/10 second units)</td>
<td>20 (2 seconds)</td>
</tr>
<tr>
<td>Baud Rate</td>
<td>0x9C</td>
<td>2 = 600</td>
<td>9600</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = 1200</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 = 2400</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 = 4800</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6 = 9600</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7 = 19200</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 = 38400</td>
<td></td>
</tr>
<tr>
<td>Stop Bits</td>
<td>0x9D</td>
<td>1 = 1 stop bit</td>
<td>1 stop bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = 2 stop bits</td>
<td></td>
</tr>
</tbody>
</table>
Table 7-7. PY Parameter Numbers (Continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parameter Number</th>
<th>Options</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary Serial Port Parameters - Continued</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parity Type</td>
<td>0x9E</td>
<td>0 Odd Parity</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Even Parity</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 Mark Parity</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 Space Parity</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 No Parity</td>
<td></td>
</tr>
<tr>
<td>Software Handshaking</td>
<td>0x9F</td>
<td>0 None</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 ACK/NAK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 ENQ with ACK/NAK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 ENQ only</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 XON/XOFF</td>
<td></td>
</tr>
<tr>
<td>Hardware Handshaking</td>
<td>0xA0</td>
<td>0 None</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 RTS/CTS Standard</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 RTS/CTS Option 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 RTS/CTS Option 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 RTS/CTS Option 3</td>
<td></td>
</tr>
<tr>
<td>Number Data Bits</td>
<td>0xA2</td>
<td>7 7 bits</td>
<td>8 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 8 bits</td>
<td></td>
</tr>
<tr>
<td>RS-232 Host</td>
<td>0xA3</td>
<td>55 Standard</td>
<td>Standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>137 ICL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>139 Fujitsu</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>140 Nixdorf Mode A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>142 Nixdorf Mode B</td>
<td></td>
</tr>
<tr>
<td>Intercharacter Delay</td>
<td>0x6E</td>
<td>0 – 99 (1/1000 second units)</td>
<td>0</td>
</tr>
<tr>
<td>Beep on Bell</td>
<td>0x96</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Check Receive Errors</td>
<td>0x97</td>
<td>0 Disable</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>RTS State</td>
<td>0x9A</td>
<td>0 Low</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 High</td>
<td></td>
</tr>
</tbody>
</table>
### Table 7-7. PY Parameter Numbers (Continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parameter Number</th>
<th>Options</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MCL Parameters</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key Click</td>
<td>0xF00A</td>
<td>0 Disable</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable</td>
<td></td>
</tr>
<tr>
<td>Scanner Address</td>
<td>0xF00B</td>
<td>1 – 254</td>
<td>1</td>
</tr>
<tr>
<td>Sleep Time</td>
<td>0xF00C</td>
<td>5 – 30</td>
<td>10 secs</td>
</tr>
<tr>
<td>Date Separator</td>
<td>0xF00E</td>
<td>0 /</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 -</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 :</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 (None)</td>
<td></td>
</tr>
<tr>
<td>Hour Type</td>
<td>0xF00F</td>
<td>0 24 hour</td>
<td>12 hour</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 12 hour</td>
<td></td>
</tr>
<tr>
<td>Decimal Separator</td>
<td>0xF010</td>
<td>0 .</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 ,</td>
<td></td>
</tr>
<tr>
<td>File Separator</td>
<td>0xF011</td>
<td>Any ASCII character</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(ASCII 32 - ASCII 127)</td>
<td></td>
</tr>
<tr>
<td>Date Format</td>
<td>0xF013</td>
<td>0 MMDDYYYY</td>
<td>MMDDYYYY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 DDMMYYYY</td>
<td></td>
</tr>
<tr>
<td><strong>MCL-Net Parameters</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCL-Net Baud Rate</td>
<td>0xF01B</td>
<td>2=600</td>
<td>8 (38400 baud)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3=1200</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4=2400</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5=4800</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6=9600</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7=19200</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>8=38400</td>
<td></td>
</tr>
<tr>
<td>MCL-Net Hex Addressing</td>
<td>0xF012</td>
<td>0 Disabled</td>
<td>0 (Disabled)</td>
</tr>
<tr>
<td>Mode</td>
<td></td>
<td>1 Enabled</td>
<td></td>
</tr>
<tr>
<td>MCL-Net Max Retries</td>
<td>0xF01C</td>
<td>1-10</td>
<td>3</td>
</tr>
<tr>
<td>MCL-Net Frame Timeout</td>
<td>0xF01D</td>
<td>100-3000 ms, in 100 ms</td>
<td>500 ms (5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>increments</td>
<td></td>
</tr>
</tbody>
</table>
### Table 7-7. PY Parameter Numbers (Continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parameter Number</th>
<th>Options</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MCL-Net Parameters - Continued</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCL POS Host</td>
<td>0xF01E</td>
<td>0 MCL-Net&lt;br&gt;1 Other (RS-232 or Synapse, depending on installation)</td>
<td>1 (Other)</td>
</tr>
<tr>
<td>MCL Batch Host</td>
<td>0xF01F</td>
<td>0 MCL-Net&lt;br&gt;1 Other (RS-232 or Synapse, depending on installation)</td>
<td>0 (MCL-Net)</td>
</tr>
<tr>
<td>Power Detection Beep</td>
<td>0xF020</td>
<td>0 Disabled&lt;br&gt;1 Enabled</td>
<td>1 (Enabled)</td>
</tr>
<tr>
<td>Host Override Flag</td>
<td>0xF021</td>
<td>0 Disabled&lt;br&gt;1 Enabled</td>
<td>0 (Disabled)</td>
</tr>
<tr>
<td><strong>RF Parameters</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF Retries</td>
<td>0xF042</td>
<td>3 - 8</td>
<td>4</td>
</tr>
<tr>
<td>Scanner Decode Beep Type</td>
<td>0x95</td>
<td>0 Beep Type 1&lt;br&gt;1 Beep Type 2&lt;br&gt;2 Beep Type 3&lt;br&gt;3 Beep Type 4&lt;br&gt;4 Beep Type 5</td>
<td>0</td>
</tr>
</tbody>
</table>
# Index

## Numerics

<table>
<thead>
<tr>
<th>Numeral</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14686</td>
<td>Symbol Support Centers</td>
</tr>
</tbody>
</table>

## A

- Adding a Process: 2-17
- adding a process: 4-28
- adding text to the menu: 3-1
- adding variables to the menu: 3-4
- ampersand (&): 7-2
  - use of: 7-3
- argument separator: 7-2
- arguments
  - optional: 7-2
- arithmetic: 4-38
- assign variable: 4-30

## B

- Backlight command: 7-10
- bar code input: 5-3
- Barcode Input: 2-13
- Barcode Input - Branch Tab: 4-17
- barcode input dialog box: 4-13
- beep: 4-79
- Beep command: 7-11
- branch group: 4-40
- break: 5-3

## C

- case & branch: 4-43
- changing the printer setup: 4-84
- clear screen: 4-82
- Code 39 Tab: 1-14
- combine: 4-31
- commands
  - Backlight: 7-10
  - Beep: 7-11
  - Display Clear: 7-19
  - Display Message: 7-20
  - File Check: 7-23
  - File Delete: 7-21
  - File Erase: 7-22
  - File Open: 7-24
  - File Read: 7-28
  - File Seek: 7-30
  - file size: 7-34
  - File Update: 7-32
  - File Write: 7-33
- If: 7-39
- input cash: 7-36
- Input Date: 7-38
- Input Extended: 7-45
- Input Hour: 7-40
- Input Key: 7-41
- Input List: 7-43
- Label: 7-48
- MCL-Link Start: 7-59
- menu list: 7-50
- Modem: 7-49
- Network Input: 7-52
- Network Output: 7-55, 7-57
- Parameterize Exchange: 7-63
- Parameterize Input: 7-60
- parameterize scanner: 7-64
- Parameterize Time: 7-62
| Index-2 |
|-----------------|-----------------|
| MCL-Designer for Phaser Series User’s Guide |
| **Quit or Exit** | 7-66 |
| **Run Program** | 7-67 |
| **Serial Input** | 7-69 |
| **Serial Output** | 7-73, 7-75 |
| **Skip** | 7-72 |
| **Skip and Return** | 7-77 |
| **Variable Compare** | 7-78 |
| **Variable Extract** | 7-80 |
| **Variable Format** | 7-81 |
| **Variable Justify** | 7-84 |
| **Variable Length** | 7-86 |
| **Variable Move** | 7-87 |
| **Variable Operation - Arithmetic** | 7-88 |
| **Variable Operation - Arithmetic - Extended** | 7-92 |
| **Variable Substring** | 7-90 |
| **Wait** | 7-94 |
| **comment** | 4-83 |
| **control & branch** | 4-44 |
| **copying a screen** | 4-23 |
| **copying fields** | 4-21 |
| **CP CRC** | 7-13 |
| **CP CVT** | 7-16 |
| **CP XOR** | 7-15 |
| **D** |
| **delete a record** | 4-53 |
| **deleting a screen** | 4-24 |
| **deleting all screens** | 4-25 |
| **designing a program** | 2-10 |
| **designing with quick tour** | 2-1 |
| **direct files** | 7-6 |
| **Display Clear command** | 7-19 |
| **Display Message command** | 7-20 |
| **display text** | 4-6 |
| **display variable** | 4-7 |
| **downloading a project** | 6-1 |
| **E** |
| **editing fields on a screen** | 4-20 |
| **editing your design** | 2-23 |
| **erase file** | 4-57 |
| **Error Code Messages** | 7-96 |
| **error messages** | 7-96 |
| **extract** | 4-35 |
| **F** |
| **features** | 1-2 |
| **File Check command** | 7-23 |
| **File Delete command** | 7-21 |
| **File Erase command** | 7-22 |
| **File Open command** | 7-24 |
| **File Read command** | 7-28 |
| **file seek** | 4-58 |
| **File Seek command** | 7-30 |
| **File Size** | 7-34 |
| **File Size command** | 7-34 |
| **file structure** | 7-6 |
| **File Update command** | 7-32 |
| **file viewer** | 5-5 |
| **File Write command** | 7-33 |
| **files and records** | 7-5 |
| **files group** | 4-49 |
| **files window** | 4-87 |
| **format** | 4-33 |
| **G** |
| **global view of variables** | 4-93 |
| **gosub window** | 4-47 |
| **Goto** | 4-40 |
| **H** |
| **host look-up** | 4-66 |
| **how to create a main menu** | 3-1 |
| **I** |
| **IC** | 7-36 |
| **If command** | 7-39 |
| **information, service** | 1-x |
| **Input Cash command** | 7-36 |
| **Input Date command** | 7-38 |
| **Input Extended command** | 7-45 |
| **Input Hour command** | 7-40 |
| **Input Key command** | 7-41 |
| **Input List command** | 7-43 |
Index-3

inserting a blank screen .................. 4-22
Installation ............................... 1-1
installation ................................ 1-1

K
key & branch ................................. 4-41
Keyboard Input ............................. 2-16
keyboard input ............................. 4-8
keyboard key symbols ...................... 7-2

L
Label command ............................... 7-48
Large Variables ............................. 7-4
length ..................................... 4-37
local file check ............................. 4-59

M
Macro Command Language ................. 7-1, 7-3
main program window ...................... 4-1
MCL Collection ............................. 1-1
MCL commands
  9 logical groups ........................ 7-3
MCL Designer - main tab .................. 3-2
MCL Designer - menu tab .................. 3-2
MCL kernel ................................ 7-1
MCL-Code .................................. 4-83
MCL-Code display ........................... 5-5
MCL-Code viewer ........................... 5-5
MCL-Designer ............................... 1-21, 2-1
MCL-Designer setup ........................ 1-21
MCL-Link communications group ........ 4-61
MCL-Link start ............................. 4-67
MCL-Link Start command ................... 7-59
Memory Mapping ............................ 1-8
Menu List command ........................ 7-50
Menu Options ............................... 2-9
menu setup ................................ 2-4
message .................................... 4-81
Modem command ............................ 7-49
modem command ............................ 4-71
moving a screen ............................ 4-24
moving fields ............................... 4-20

N
Network Input command ..................... 7-52
Network Output command ................... 7-55, 7-57
NO - P360/P460 Memory Scanner ........ 7-55
NO - P370/P470 RF Scanner ............... 7-57
notational conventions ..................... x

O
others group ................................. 4-78
Others Tab ................................. 1-15

P
Parameter Numbers ......................... 7-103
Parameterize Exchange command .......... 7-63
Parameterize Input command .............. 7-60
parameterize scanner command ........... 7-64
Parameterize Time command ............... 7-62
pause ..................................... 4-78
Phaser RS-232 Setup ........................ 1-17
Phaser RS-232 setup ........................ 1-17
Phaser system setup ........................ 1-7
placing fields on a screen ............... 4-19
print file .................................. 4-77
printing a screen ........................... 4-84
printing options ........................... 4-84
printing reports ........................... 4-84
process
  beep ..................................... 4-79
  comment .................................. 4-83
  file delete ............................... 4-53
  file erase ............................... 4-57
  file read ................................ 4-51
  file seek ............................... 4-58
  file write ................................ 4-49
  host look-up ............................. 4-66
  key & branch ............................. 4-41
  local file check ........................ 4-59
  MCL-Code ................................. 4-83
  modem command ........................... 4-71
  pause .................................... 4-78
  print file ............................... 4-77
  receive data ............................. 4-75
  receive file ............................. 4-65
Index-4

receive packet .................................. 4-62
remote file check .................................. 4-68
scanner serial output ............................... 4-74
send file .............................................. 4-63
send packet to host .................................. 4-61
serial input ............................................ 4-75
start EXE program on host ........................... 4-69
string serial output ................................. 4-73
variable arithmetic .................................. 4-38
variable assign ....................................... 4-30
variable extract ...................................... 4-35
variable format ....................................... 4-33
variable length ....................................... 4-37
variable substring .................................... 4-34
process list ............................................ 4-28
Programs and File Sizes ............................. 7-95
project downloading .................................. 6-1
Project Setup ......................................... 2-1

Q
quick view of all screens ............................. 4-25
Quit or Exit command .................................. 7-66

R
read a record .......................................... 4-51
receive data .......................................... 4-75
receive file ............................................. 4-65
receive packet ......................................... 4-62
remote file check ...................................... 4-68
review File ............................................. 4-54
Run Program command ................................ 7-67

S
saving your design .................................... 2-20
Scanner Setup .......................................... 1-10
scanner setup .......................................... 1-5, 1-10
screen description .................................... 4-5
send file ............................................... 4-63
send packet ............................................. 4-61
sequential files ........................................ 7-6
Serial Input command ................................ 7-69
Serial Output command ................................ 7-73, 7-75
service information ................................... x
set label ............................................... 4-46
Set up the System ...................................... 1-7
setting menu options .................................. 3-6
setting up a new project ............................... 1-3
SI ......................................................... 7-69
simulating and testing your design .................. 2-20
simulation .............................................. 5-1
Skip and Return command ............................. 7-77
Skip command .......................................... 7-72
SO - P360/P460 Memory Scanner .................... 7-73
SO - P370/P470 RF Scanner ........................... 7-75
start EXE program on host ............................ 4-69
Starting MCL Designer ................................ 1-3
starting MCL-Designer ................................ 1-3
string serial output .................................... 4-73
substring ................................................ 4-34
symbol support center ................................ xi
system setup variables ................................. 7-98
System Variables ....................................... 7-4

test & branch .......................................... 4-42
testing and simulating ................................ 5-1
typographic conventions ............................... 7-2

U
UPC/EAN Tab ............................................ 1-12
User-Defined Variables ................................. 7-4

V
Variable Compare command ........................... 7-78
Variable Extract command ............................. 7-80
Variable Format command ............................. 7-81
variable group ........................................... 4-30
Variable Justify command ............................. 7-84
Variable Length command .............................. 7-86
variable list ............................................. 5-4
Variable Move command ............................... 7-87
Variable Operation - Arithmetic - Extended command ........................................ 7-92
Variable Operation - Arithmetic command  7-88
Variable Substring command  ............... 7-90
variables  ..................................... 7-3
types of  ..................................... 7-4

W
Wait command  ......................... 7-94
warning message  ......................... 4-81

working items of the simulator ........... 5-2
working with fields  ........................ 4-19
working with files  .......................... 4-87
working with processes .................... 4-26
working with screens  ..................... 4-22
working with variables .................... 4-92
write a record  ............................. 4-49
Tell Us What You Think...

We’d like to know what you think about this Manual. Please take a moment to fill out this questionnaire and fax this form to: (631) 738-3318, or mail to:

Symbol Technologies, Inc.
One Symbol Plaza M/S B-4
Holtsville, NY 11742-1300
Attention: Technical Publications Manager

IMPORTANT: If you need product support, please call the appropriate customer support number provided. Unfortunately, we cannot provide customer support at the fax number above.

User’s Manual Title: _____________________________________________
(please include revision level)

How familiar were you with this product before using this manual?

☐ Very familiar  ☐ Slightly familiar  ☐ Not at all familiar

Did this manual meet your needs? If not, please explain.

________________________________________________________________________
________________________________________________________________________

What topics need to be added to the index, if applicable?

________________________________________________________________________
________________________________________________________________________

What topics do you feel need to be better discussed? Please be specific.

________________________________________________________________________
________________________________________________________________________

What can we do to further improve our manuals?

________________________________________________________________________
________________________________________________________________________

Thank you for your input—We value your comments.