

eV+ Operating System

User's Guide

Copyright Notice

The information contained herein is the property of Omron Adept Technologies, Inc., and shall not be reproduced in whole or in part without prior written approval of Omron Adept Technologies, Inc. The information herein is subject to change without notice and should not be construed as a commitment by Omron Adept Technologies, Inc. The documentation is periodically reviewed and revised.

Omron Adept Technologies, Inc., assumes no responsibility for any errors or omissions in the documentation. Critical evaluation of the documentation by the user is welcomed. Your comments assist us in preparation of future documentation. Please submit your comments to: techpubs@adept.com.

Table Of Contents

Introduction	7
What Is Described in This Manual?	8
What Systems Can I Use This Manual With?	8
What Other Manuals Might I Need?	8
What About Safety?	10
Conventions	13
Your Omron Adept System Hardware	15
Before You Begin	16
The System Controller	16
The Controller and Memory	17
Robots and Motion Devices	17
The Pendant	18
Other Input and Output Devices	19
Learning the Basics	21
Using the Controller	22
Using the Command Line	25
Using the Graphics-Based Interface	29
Using Software	30
The Operating System	31
Using Files and Directories	31
Using Files	33
Understanding Disk Files	34
Using Directories	39
Executing Programs	46
Understanding Programs	47
Removing Objects from System Memory	50
Understanding Program Modules	53
Using Permanent File Storage	54
Using Digital Storage Cards	55

Using Hard Drives	55
Customizing an Omron Adept System	56
Introduction	57
Customizing the Controller Hardware	57
Changing the Controller Configuration	57
System Software Switches	58
Command Programs	60
Introduction	61
Creating Command Programs	61
Running Command Programs	62
Controlling Command Programs	63
Example Command Program	64
System Messages	66
Error Messages	66

Introduction

The following topics are described in this chapter:

What Is Described in This Manual?	8
What Systems Can I Use This Manual With?	8
What Other Manuals Might I Need?	8
What About Safety?	10
Conventions	13

What Is Described in This Manual?

This manual describes the eV+ operating system for v2.x and later. After your Omron Adept controller (if used) and any attached devices are installed, the text in this manual describes what you need to know to start, operate, and shut down the system. If you are running a system that is already programmed and configured, this manual contains most of the information you will need.

What Systems Can I Use This Manual With?

This manual is for use with eV+ version v2.x and later.

What Other Manuals Might I Need?

If you will be modifying the system configuration, installing new equipment, or programming in the eV+ language, you may need the following manuals:

Manual	Material Covered	Recommended for:
Robot or motion device instruction handbook or user's guide	Installation of robot or motion device. Periodic maintenance of the device.	All users, service personnel, and programmers using the robot
<i>eV+ Operating System Reference Guide</i>	Detailed descriptions of all Monitor commands, some of which are described only briefly in the eV+ Operating System User's Guide.	All programmers and users installing or configuring the controller
<i>eV+ Language User's Guide</i>	A description of the structure and elements of the eV+ programming language.	All programmers

What Other Manuals Might I Need?

Manual	Material Covered	Recommended for:
<i>eV+ Language Reference Guide</i>	A complete description of the keywords used by the eV+ programming language. This manual contains a list of all system messages.	All programmers
<i>ACE Sight User's Guide</i>	A description of the structure and elements of the ACE Sight extension to the ACE software.	Vision application programmers and users
<i>Adept SmartController EX User's Guide</i>	This manual details the installation, configuration, and maintenance of your Omron Adept SmartController system. The controller must be set up and configured before control programs will execute properly.	Programmers and system installers.
AIM application module user's guides	Installing and using AIM application modules	Programmers and users of the application module

Manual	Material Covered	Recommended for:
	(VisionWare, MotionWare, AIM PCB, etc.).	
AIM module reference guides	Customizing AIM baseline modules and applications modules.	AIM customizers
<i>ACE User's Guide</i>	Configuration of the Omron Adept SmartController system through ACE software.	All programmers

What About Safety?

Safety is critical! Read the following section carefully and follow the cautions and warnings that are placed throughout this manual. There are three levels of safety notation used in Omron Adept eV+ manuals. They are:



WARNING: If the actions indicated in a warning are not complied with, injury or major equipment damage could result. A warning typically describes the potential hazard, its possible effect, and the measures that must be taken to reduce the hazard.



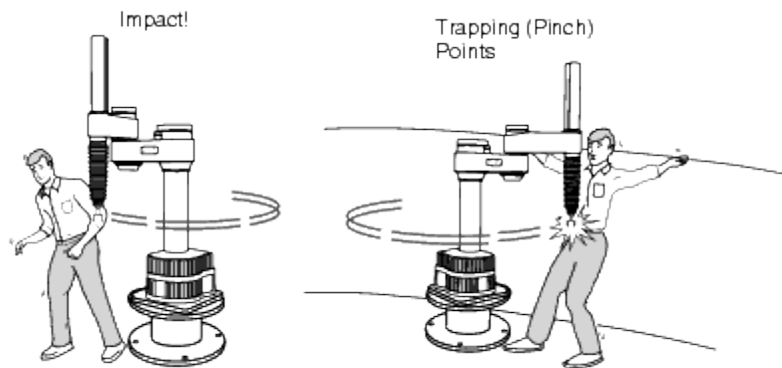
CAUTION: If the action specified in a caution is not complied with, damage to your equipment or data could result.

NOTE: A note provides supplementary information, emphasizes or supplements a point or procedure, or gives a tip for easier operation.

Reading and Training for System Users

Omron Adept robot systems include computer-controlled mechanisms that are capable of moving at high speeds and exerting considerable force. Like all robot systems and industrial equipment, they must be treated with respect by the system user.

We recommend you read the American National Standard for Industrial Robot Systems-Safety Requirements, published by the Robotic Industries Association in conjunction with the American National Standards Institute. The publication, ANSI/RIA R15.06-1992, contains guidelines for robot system installation, safeguarding, maintenance, testing, startup, and operator training. The document is available from the American National Standards Institute, 1430 Broadway, New York NY 10018. All Omron Adept robot systems sold in Europe must conform to European Certification requirements. For details, see the manual supplied with your robot.



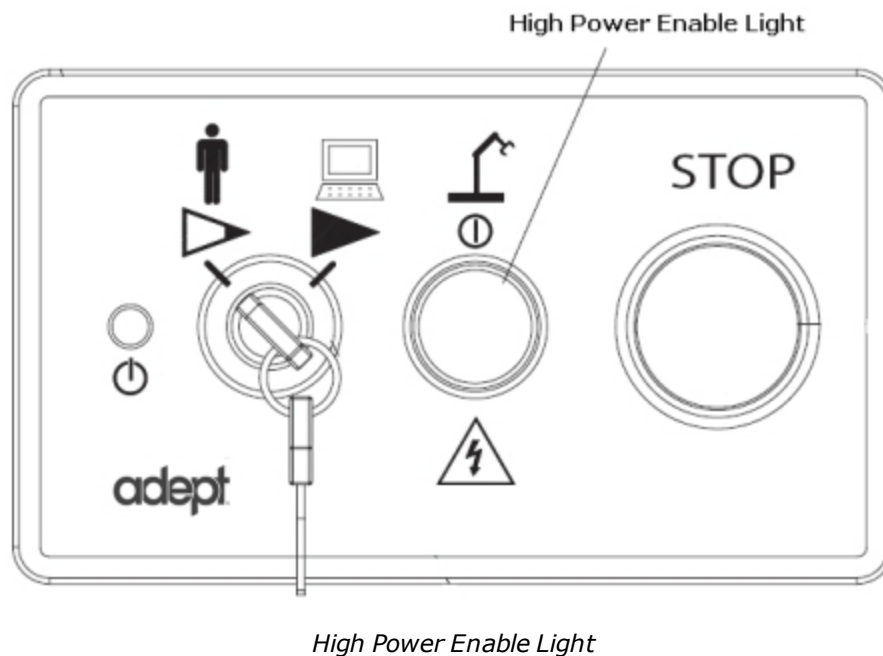
Impacts and Trapping Points

System Safeguards

Safeguards should be an integral part of robot workcell design, installation, operator training, and operating procedures. Omron Adept robot systems have various communication features to aid you in constructing system safeguards. These include remote emergency stop circuitry and digital input and output lines.

Computer-Controlled Robots

Omron Adept robots are computer controlled, and the program that is running the robot may cause it to move at times or along paths you may not anticipate. When the white HIGH POWER light¹ on the Front Panel is illuminated, do not enter the workcell because the robot may move unexpectedly.



Manually Controlled Robots

Omron Adept robots can also be controlled manually when the white HIGH POWER light on the front panel is illuminated. When this light is lit, robot motion can be initiated from the system keyboard or the manual control pendant. If you enter the workcell, set the keyswitch to manual mode, press the MAN/HALT button on the manual control pendant, and take the key with you. This prevents anyone else from initiating unexpected robot motions from the keyboard.

Other Computer-Controlled Devices

In addition, these systems can be programmed to control equipment or devices other than the robot. As with the robot, the program controlling these devices may cause them to operate at times not anticipated by personnel. Make sure that safeguards are in place to prevent personnel from entering the workcell when the white HIGH POWER light on the front of the Front Panel is illuminated.



WARNING: Omron Adept Technologies, Inc. recommends the use of additional safety features such as light curtains, safety gates, or safety floor mats to prevent entry to the workcell while HIGH POWER is enabled. These devices may be connected using the controller's remote emergency stop circuitry (see the *Adept SmartController User's Guide*).

¹White is the color of the light on the Front Panel. If a remote front panel has been installed by another party, the color may vary.

Conventions

This section discusses:

- Typographic conventions
- Keyboard conventions
- Selecting, choosing, and pressing items

Typographic Conventions

The following typographic conventions are used throughout this manual:

This	Represents
ALL CAPITALS	eV+ file names, directory names, commands, keywords, and attributes; also acronyms. The name of a <i>physical</i> key or button that you must press, such as the ENTER key and the PROGRAM START button.
<code>monospace</code>	Screen displays, code examples, nonplaceholder terms in formal syntax definitions, and case-sensitive words.
bold	When typing or entering a command or instruction, bold indicates anything that you must type exactly as it appears. For example, if you are asked to type execute 1 a.diskcopy , you type all the bold characters exactly as they are printed. What you type is shown in lowercase letters unless it must be typed in uppercase letters to work properly. You may always substitute a currently valid shortcut form when typing eV+ keywords. In order for the eV+ system to process your typing, you must conclude your entry by pressing the ENTER or RETURN key. Bold type is used for lowercase names such as subroutine names, variable names, and program names; for example, a.diskcopy . Bold type also is used for window items that you choose and window items that do not have initial capital letters in all principal words.
<i>italic</i>	Placeholders that you must provide in typed input. This font also indicates new terms and other emphasized words.
Initial	The name of an object such as a window, screen, menu, button, dialog

This	Represents
Capitals	box, or dialog box component. Examples are the Display menu and the Task Profiler window. The logical names of physical function keys or buttons use regular font and follow the interface's capitalization, which usually has initial capital letters in all principal words.

Keyboard Conventions

Key combinations appear in the following format:

Notation	Meaning
KEY1+KEY2	A plus sign (+) between keys means that you must press the keys at the same time. For example, "Press CTRL+Z" means that you press CTRL and hold it down while you press Z.

Abbreviations

The following abbreviations may appear in this manual:

Abbreviation	Meaning
CE	European Certification
DAC	Digital-to-Analog Converter
I/O	Input/output

Your Omron Adept System Hardware

The following topics are described in this chapter:

Before You Begin	16
The System Controller	16
The Controller and Memory	17
Robots and Motion Devices	17
The Pendant	18
Other Input and Output Devices	19

Before You Begin

In this part of the manual you will learn the basics of using an Omron Adept controller. You will learn to execute programs, enter Monitor commands, and customize your controller. Before you begin, you must have the Omron Adept controller installed and connected to any peripheral equipment. The minimum installation required is:

An Omron Adept SmartController EX (see the *Adept SmartController EX User's Guide*)

In addition, you may have the following equipment installed:

- A motion system, including:
 - A motion device (see the User Guide for your robot or the *Adept SmartMotion Installation Guide*)
 - or
 - Motion interface module (sMI6) (see the *Adept SmartMotion Installation Guide*)
- Conveyor belts
- A vision system, including:
 - An Omron Adept SmartVision EX vision processor (see the *Adept SmartVision EX User's Guide*)
 - ACE Sight software (see the *ACE Sight User's Guide*)

The following topics summarize the hardware components that may be part of your Omron Adept system. See Learning the Basics on page 21 for the basics about using the eV+ operating system software.

The System Controller

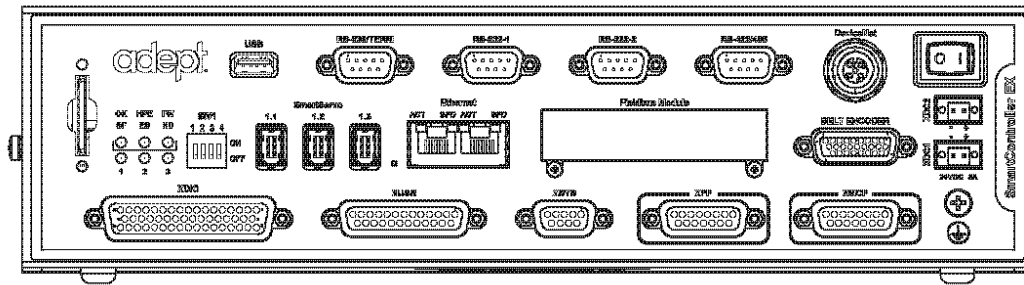
Most Omron Adept systems have an Omron Adept SmartController EX. A system will also normally have a PC as the interface device. Optionally, the Omron Adept SmartVision EX vision processor, equipped with an optional keyboard and monitor, can be used as the interface device.

Systems supporting vision will have either an Omron Adept SmartVision EX vision processor or a Windows PC, running ACE Sight software. This will be connected to the controller by Ethernet.

The *Adept SmartController User's Guide* covers the installation of the Omron Adept SmartController EX as well as the interconnection to the equipment in your automation system.

All the other hardware described in this topic is optional and may or may not be included in your system.

The following figure shows the Omron Adept SmartController EX.



Omron Adept SmartController EX

The Controller and Memory

The Omron Adept controller runs the robot-control programs. The controller contains circuits for driving the robot joints. It also contains circuits for communicating with other equipment in the workcell, such as networking equipment, sensors, feeders, etc.

The controller also contains memory chips for the system's Random Access Memory (RAM). Programs that are actively executing, and the data needed by those programs, reside in RAM. See your Omron Adept controller user's guide for more details.

Information that is not actively being used is stored on mass storage devices, such as a Secure Digital (SD) card. For more details, see the section Using Permanent File Storage on page 54.

Robots and Motion Devices

Your system may be equipped with one or more Omron Adept robots, or with the Adept SmartMotion option, which provides a means of controlling many types of motion devices. An example of an Omron Adept robot, the Omron Adept Quattro s650HS robot, is shown in the following figure.



The Quattro s650HS Robot

See your Omron Adept robot user's guide for details on the installation and configuration of the Omron Adept hardware and the interconnection between the Omron Adept controller and your Omron Adept robot. See the *Adept SmartMotion Developer's Guide* for details on configuring and tuning a motion system on the Omron Adept SmartMotion platform.

The Pendant

The Omron Adept pendant, shown in the following figure, allows you to move a robot or motion device and to teach locations used by motion control programs. The features of the pendant are covered in the *Adept T20 Pendant User's Guide*.



T20 Pendant

Other Input and Output Devices

In most cases, an automation task requires coordination between the motion device and other workcell equipment, such as part feeders, conveyor belts, and other controllers or production floor computers.

Cameras and Lighting Equipment

The *Adept SmartVision EX vision processor* incorporates a vision processor into the Omron Adept controller system. The vision system may be used with or without a motion device. *Stand-alone* vision systems perform quality and process-control inspections. *Guidance* vision systems can perform inspections as well as providing a means of guiding the motion device to pick up and place objects that are not in precise or predictable locations. The *Adept SmartController User's Guide* details the installation and hardware configuration of machine

vision equipment. The *ACE Sight User's Guide* covers the use of Omron Adept's vision software.

The Controller Disk Drives

Mass storage is used to store programs and information not actively being used by the controller. System RAM is limited, so only a certain number of programs can be resident in RAM at one time, and RAM memory is erased when power is turned off. Secure Digital (SD) cards, on the other hand, provide extensive, permanent storage capacity. For details on the system mass storage device(s), see Using Permanent File Storage on page 54.

Serial I/O

The Omron Adept SmartController has connectors for RS-232 and RS-422/RS-485 communication with other controllers or computers. See the *Adept SmartController User's Guide* for details on installing serial I/O devices. See the *eV+ Language User's Guide* for details on programming serial I/O.

Digital I/O

Omron Adept's digital I/O system behaves similarly to programmable logic controllers (PLCs), allowing input from and output to devices generating and receiving simple on/off signals. Digital I/O uses optical isolation modules to insulate the controller from noise generated by inductive devices in the workcell. See the *Adept SmartController User's Guide* for details on installing digital I/O devices. See the *eV+ Language User's Guide* for details on programming digital I/O.

Ethernet

The Omron Adept SmartVision EX and the Omron Adept SmartController EX products include shielded RJ45 Ethernet receptacles. The hardware (and required option licenses) provide TCP/IP and FTP capabilities. See the *Adept SmartController User's Guide* and the *Adept SmartVision EX User's Guide* for details on Ethernet communications on the Omron Adept SmartController EX and Omron Adept SmartVision EX products.

Remote Emergency Stop

Motion systems must be able to react immediately to interruptions and dangerous conditions in the workcell. Adept's remote E-stop circuitry monitors user-supplied series-wired sensors that detect dangerous conditions in the workcell. The controller connection for the user emergency stop is on the Adept SmartController EX XUSR port. See the *Adept SmartController User's Guide* for details on installing remote E-stop devices.

Learning the Basics

The following topics are described in this chapter:

Using the Controller	22
Using the Command Line	25
Using the Graphics-Based Interface	29
Using Software	30
The Operating System	31
Using Files and Directories	31

Using the Controller

You interact with the controller by:

- Using a pointing device to make selections. This type of interaction is used with programs written for ACE-based systems. For details on the ACE user interface basics, see the *ACE User's Guide*.
- Initiating actions and responding to system requests with the pendant. The pendant is generally used with motion systems.
 - Use of the T20 pendant is described in the *Adept T20 Pendant User's Guide*.
- Entering commands and responding to system requests by typing entries at the command line. This is the most basic type of interaction and is used by all Omron Adept controllers. For details on the basics of the command line, see *Using the eV+ Command Line*.

The following are the minimum steps required to get an Omron Adept system up and running.

Install the Equipment and Set the Hardware Configuration

All the equipment must be installed and tested. The *Adept SmartController User's Guide* covers installation and hardware configuration of the controller. As delivered from Omron Adept, all systems have their hardware configuration set to the most common defaults. You will not have to perform any hardware configuration changes if these defaults are acceptable to your installation. The controller hardware configuration options are detailed in the controller user's guide.

In addition to the controller, your system may include the following:

- An Omron Adept robot, or a motion device and the Adept SmartMotion system option.
 - Installation of the Omron Adept robot or motion device should be covered in the documentation provided with that robot or motion device.
 - Interconnection of the robot with the controller is covered in the robot user guide.
 - The motion servo hardware configuration is covered in the *Adept SmartMotion Installation Guide*. See the *Adept SmartMotion Developer's Guide* for details on software configuration for systems using the Adept SmartMotion option.
- Safety equipment.
 - In workcells with robots or motion devices, it is critical that sensors be placed that will prevent personnel from entering the workcell when the motion device may move. Make sure this equipment is installed and functioning properly

before proceeding.

- The interconnection of safety devices to the controller is covered in the *Adept SmartController User's Guide*.
- Peripheral input/output devices.
 - The *Adept SmartController User's Guide* covers the physical interconnect of I/O devices.
 - The *ACE User's Guide* describes using peripheral devices with the ACE interface and eV+.
 - The *eV+ Language User's Guide* covers eV+ programming considerations for accessing the peripheral devices.
- Conveyor Belts.
 - The *ACE User's Guide* describes using conveyor belts with the ACE interface and eV+.
 - The *eV+ Language User's Guide* describes the eV+ programming considerations for coordinating robot motions with a conveyor belt connected to one of the external encoder ports on the controller.
- Cameras and strobe lights (requires the ACE Sight option).
 - The *ACE Sight User's Guide* details installing and configuring cameras and strobe lights.

Installing or Upgrading the System

Your Omron Adept controller ships from the factory with the eV+ operating system installed and configured. The procedure for installing or upgrading the eV+ system is described in the *ACE User's Guide*.

Power Up the System

Before turning on the controller, make sure:

- All safety devices are installed and operating correctly.
- The workcell is free of obstructions and personnel.



WARNING: Do not turn on the controller unless all safety devices are in place and operating correctly. Make sure the workcell is free of obstructions and personnel.



WARNING: Do not connect any cables or make any hardware configuration changes with the controller turned on.

To Power Up the Controller

Turn on the controller by pressing the on/off switch to ON. The controller will perform a series of self-tests and load the operating system. The system power-up and boot procedures will take about a minute.

When using the ACE interface, you can connect to the controller and power-up the motion device with the ACE software. For more details, see the Getting Started chapter in the *ACE User's Guide*.

The remainder of this topic describes the operations when accessing the V+ system directly through the Monitor window.

To Power Up the Motion Device

Each time the controller is turned on, a start-up calibration procedure must be performed for the motion device. The CALIBRATE command performs this start-up procedure. Programs that send instructions to a motion device will not execute properly unless this start-up procedure is completed.

When the controller is first turned on, power is not sent to the motion device. To enable power, enter the command:

ENABLE POWER

(See the next section for complete details on entering commands.)

The system begins executing a command as soon as you press the ENTER key. (When shown in the documentation, the symbol "↵" indicates that the ENTER key should be pressed.)

The Front Panel High Power Enable button has a default 10-second timeout period. For most systems, this button will blink during that time, and you must press and release the button to enable high power. The Timeout period can be modified or disabled in the eV+ system. If disabled, you will not need to press the High Power Enable button.



WARNING: When CALIBRATE is entered, the motion device might move. Before you enter this command, make sure the workcell is free of obstructions and that all personnel are out of the workcell.

Once robot HIGH POWER has been enabled, the motion device start-up calibration must be performed (if your system is not one of those listed below). To perform start-up calibration, enter the command:

CALIBRATE

The motion device will proceed through a series of motions to verify its current location and the state of its position encoders. Once the calibration procedure is complete, robot control programs can be executed. The calibration procedure is not required for the following:

- All Viper robots, the Cobra 350 robot, and Quattro robots. The auto-calibrate bit has been set for these robots, which causes these robots to be automatically calibrated when the system is turned on. Therefore, it is not necessary to use the CALIBRATE command with these robots.
- Systems (such as stand-alone vision systems) that do not have an attached motion device.

To Reenable Power After an Emergency Shutdown

When a PANIC button is pressed or other emergency stop switch is tripped, HIGH POWER is immediately removed from the motion device. Power is also removed from the device if a servo error is reported. For example, a servo error occurs if the motion device cannot attain a desired location (for example, if the device runs into something).

Before the motion device can be used again, you must enter the command:

ENABLE POWER

You do NOT have to reissue the CALIBRATE command.

NOTE: You can also enable power by pressing the Robot Power button on the pendant.

¹This switch provides terminals for the user to connect an AC power source. If this option is not used on your system, the switch will be inactive.

Using the Command Line

There are two ways to communicate with the eV+ system: through the ACE graphical interface, and with the eV+ command line (typing commands directly at the eV+ system prompt). For more details on the eV+ system prompt, see The System Prompt and Typing Cursor on page 26.

By entering Monitor commands at the eV+ system prompt you can:

- Load program files and execute the programs
- Display system status
- Change certain characteristics of operating system behavior

NOTE: Many of the commands, formerly available only through the command line, are now more easily accessed through the ACE interface. However, you can still reach the

command line through ACE, by clicking on the Monitor Window () icon on the ACE toolbar. For more details on the ACE interface, see the *ACE User's Guide*.

The remainder of this topic describes the operations when accessing the V+ system directly through the Monitor window.

The System Prompt and Typing Cursor

The eV+ system prompt is a period, "." (sometimes referred to as the dot prompt). When the system prompt is displayed, system commands (known as Monitor commands) can be entered. Monitor commands allow you to access disk files, execute programs, and display system status. The typing cursor is a black rectangle. When the system is first started, the last line displayed will show the dot prompt followed by the typing cursor. For details, see Using the Controller.

Whenever the typing cursor appears, the system is ready to accept input from the keyboard. (At many times, you can even type before the cursor appears, and your typing will be buffered until the system is ready to accept it.)

NOTE: On ACE systems, Monitor commands cannot be entered when the Monitor window is closed. To display the Monitor window, select the Monitor Window () icon from the ACE toolbar. For more details on the ACE interface, see the *ACE User's Guide*.

The Parts of a Monitor Command

Every Monitor command has a name. Most Monitor commands have one or more parameters.

The Command Name

Every command has a unique name that tells the eV+ system what action to perform.¹ The name must be typed exactly as shown in the manual. For example, to see the system ID Information, enter the command:

```
ID ↵
```

Command Parameters

Most commands require additional information that tells the system exactly how you want the command executed. This additional information is specified as parameters on the command line. You specify unique information for each command parameter. Command parameters must be entered in the order they are listed, and they must be separated (delimited) in exactly the fashion shown. A comma is normally used to separate parameters. Occasionally, an equal sign, "=", is used. You must always type a space between a command and its list of parameters. Spaces before and after parameter separators are optional.

Required and Optional Parameters

Command parameters can be optional or required. If a parameter is required, a value must be entered on the command line or the command will not execute correctly. If a parameter is optional, its value can be omitted and the system will substitute a default value. For example, the command STATUS has one optional parameter. If the command:

```
STATUS ↵
```

is entered, status information for all the used system tasks will be displayed. If the command:

```
STATUS 1 ↵
```

is entered, status information will be displayed only for system task number 1.

If one or more parameters follow an omitted parameter, the parameter separator(s) must be typed. If all the parameters following an omitted parameter are optional, and those parameters are also omitted, the separators do not need to be typed (see below for an example).

Using the ID Command

The ID command displays identity information about the components of the system:

```
Software:      version.revision opt1-opt2
Controller:    model-serial options
Processor n:   version.revision type-options memMb
Robot n:       model-serial options module
```

A sample display from the command "ID" is:

```
Software:      16.4 87-31E0 (Edit D3, 10-May-2007, Maintenance Release)
Controller:    2000-394 105 0
Processor 1:   0.52 7-7 32MB
Robot 1:       100-0 0 8
```

For more detailed information, see the *eV+ Operating System Reference Guide*.

Monitor Command Syntax Conventions

To help present the required syntax in the clearest, simplest manner, this manual uses the following syntax conventions:

- The command name is shown in capital letters.
- Parameters shown in bold lowercase letters are required and should be replaced with values you provide. For example, **drive** should be replaced with a drive letter you choose.
- Parameters shown in non-bold lowercase letters are optional. If you omit an optional parameter, the system will use a default value.

The following sample command has one required and one optional parameter:

```
SAMPLECOMMAND param_required, param_optional
```

If both parameters are specified, the comma separating the parameters must be entered. If the optional parameter is not specified, the comma can be omitted.

Shortcuts to Typing a Command

Some command names do not have to be typed completely. You need to type only enough characters to uniquely identify the command. For example, the only command that begins with STAT is STATUS. Therefore, typing:

```
STAT ↵
```

is sufficient to execute the STATUS command. Typing:

```
STA ↵
```

will result in the error:

```
*Ambiguous name*
```

because it could refer to either the STATUS or STACK command.

To Cancel a Command

To cancel an executing command, press CTRL+C. These keystrokes will NOT halt a program that was started with the EXECUTE command. The ABORT command halts an executing program.²

Many commands output information to the monitor. This output can be temporarily halted by pressing the SCROLL LOCK key. To restart the output, press SCROLL LOCK again.

To Edit the Current Command

The  key erases characters to the left of the typing cursor.

The following keys move the typing cursor on the command line:

The  and  keys move the typing cursor left and right. Shift+ or Ctrl+ moves the typing cursor to the beginning of the line. Shift+ or Ctrl+ moves the typing cursor to the end of the line.

Ctrl+U erases all the characters from the typing cursor to the beginning of the line.

The system prompt is always in insert mode. If the typing cursor is not at the end of the line, entered characters will be inserted to the right of the cursor and the remaining characters are pushed to the right.

Regardless of the position of the typing cursor, the whole command line will be submitted when the ENTER or RETURN key is pressed.

To Retrieve Previously Entered Commands

The twenty most recently entered commands are stored in the command history buffer. The  key moves backward through the history displaying successive commands. The  key moves forward through the history. The displayed command can be re-executed by pressing . The command can also be edited as described above and then executed by pressing .

To Enter a Command While a Program Is Executing

The Adept system is a multitasking system (i.e., multiple programs can be executing concurrently in different tasks). This allows you to enter Monitor commands any time the Monitor window is displayed. While a program is executing, you can start other programs in different tasks, get status information, or abort programs before they have completed.

Once a program has begun execution in task number 0, the "." prompt will disappear. However, any time the system is not actively requesting input, you can begin typing a Monitor command. As soon as you begin typing, the asterisk "*" prompt will be displayed along with any characters you type.

The only time you cannot enter a command is when the system is waiting for input from the keyboard. For example, suppose the following message is displayed in the Monitor window:

```
Are you sure (Y/N)? [ ]
```

The typing cursor at the end of the line tells you the system is waiting for you to enter a response at the keyboard. (Entering CTRL+Z at this point will generally abort the program that displayed the prompt.)

NOTE: If you are entering a command while a program is executing, the program will wait until you press ENTER before displaying to the Monitor window any output generated by the program. Therefore, you should not leave a partially-typed command in the Monitor window, since it may cause programs with screen output to wait indefinitely.

¹This manual covers the most commonly used Monitor commands and parameters. All the Monitor commands and their complete syntax are covered in the *eV+ Operating System Reference Guide*.

² ABORT does not stop any robot motion that has already begun. Use any emergency stop button to immediately halt robot motion.

Using the Graphics-Based Interface

Working with Pointing Devices

Pointing devices (such as a mouse or trackball) control the position of the pointer on the screen. The pointing device selects windows, moves windows on the screen, opens applications, and helps you interact with application programs. The basic pointing functions described in this section use the CLICK button (the large left button on the integrated trackball, the center button on a three-button mouse, or the left button on a two-button mouse).

To Move the Pointer

You move the pointer arrow around the screen by rolling the trackball (see the following figure) or sliding the mouse in the direction you want to move the pointer. Do not press any pointer device buttons when you simply want to move the pointer to a different location on the screen.

To Click

To Click the pointing device, move the pointer until it is over the area you want to click on, then press and release the CLICK button (see the following figure). Clicking selects windows, presses buttons, and performs other actions within an application.

To Drag

To drag the pointing device, move the pointer to the starting location, press and hold the CLICK button, then move the pointer to a different location and release the CLICK button. Some of the uses for dragging are to move or resize windows and to select items in pulldown menus.

Working with the ACE Interface

A basic Omron Adept system ships with the ACE software. This graphical user interface provides a point-and-click environment for configuring and programming your Omron Adept system. Optional licenses can be purchased, which enable ACE Sight vision, PackXpert packaging application, and other options.

NOTE: The ACE software uses the basic Microsoft Windows format. Refer to the *AdeptWindows User's Guide* and the Windows Operating System online help for more information.

Using Software

In the course of using an Omron Adept system you will deal with two primary types of software: operating system software and application programs. If you are using an AIM-based application¹, you will also access special database software.

The Operating System Software

Before a computer can perform any work it must boot and make active special software known as an operating system. The eV+ operating system coordinates the activities of all the computer components and any external equipment used by the system.

When you first turn on the Omron Adept controller, this software is automatically loaded and made ready for use. You do not have to do anything special to make this software active. For details on customizing and optimizing the operating system behavior, see Customizing an Omron Adept System, which lists several options for customizing and optimizing the operating system behavior. However, as delivered from Omron Adept, the operating system will correctly run your basic system.

Application Software

Application software makes the Omron Adept system perform the tasks required by your particular automation system. Application software has many sources: Omron Adept provides standard graphical interface and optional application packages; Omron Adept system integrators provide custom software programs; or you may create your own application software. Application software must be specifically loaded from the Secure Digital card to system RAM each time the controller is turned on, or after each time the programs are removed to make room in RAM to run other programs. For details on loading and unloading the software, see Understanding eV+ Programs.

¹Omron Adept's "Assembly and Information Management" system.

The Operating System

The eV+ operating system manages the flow of information within the controller. The operating system accepts instructions from application programs, input from workcell peripheral devices, and operator input from the pendant. The tasks performed by the operating system include:

- Managing the execution of application programs
- Managing the flow of information to and from mass storage devices
- Monitoring external devices attached to the controller
- Reporting errors generated during processing

In general, unless you are programming applications, you do not have to be concerned with the internal operation of the operating system.

Using Files and Directories

Application programs and other data are stored in files on the system storage device (Secure Digital card, or USB device) when not being actively used by the controller. The eV+

operating system uses a hierarchical file system for organizing files and isolating related files.

The section Understanding Disk Files describes the file directory structure.

With large storage devices, it is particularly important that you carefully organize your file structure so you can quickly locate needed files. Carefully-organized files make common operations, such as copying, deleting, and renaming files, much easier. It is very difficult to clean up a badly disorganized file structure. Therefore, before you begin copying large quantities of files to your system storage device(s), carefully read the section Understanding Disk Files .

Using Files

The following topics are described in this chapter:

Understanding Disk Files	34
Using Directories	39

Understanding Disk Files

Information used by your Omron Adept controller is stored in files on a Secure Digital (SD) card in the Omron Adept controller. Unlike information stored in RAM, disk files provide permanent storage. However, before the data in the files can be used, it must be loaded into RAM.

NOTE: Because they are handled the same way as files stored on a hard disk drive, files stored on an SD card are referred to as disk files.

When the ACE interface is used, the Process Manager automates much of the programming and program/file management through a point-and-click interface. For those who wish to access the eV+ system directly, the Controller Development Tools are used for program/file management and direct V+ monitor prompt operations. These tools are accessed from the Controller Development Tools () icon on the ACE toolbar. For more details, see the topic Controller Development Tools in the *ACE User's Guide*.

The ACE File Explorer provides a graphical interface for copying, deleting, renaming, and storing files. It is accessed from the File Explorer () icon on the ACE toolbar. For more details, see the topic File Explorer in the *ACE User's Guide*.

The remainder of this topic describes the operations when accessing the V+ system directly through the Monitor window.

Disk Files

One of the biggest differences between the eV+ system and other operating systems is the nature of a disk file. In most operating systems, a disk file contains a single executable program, text document, or data file. With the eV+ system, a file can contain several items, the most common of which are programs and the values of global variables.

The reasons for this file strategy will become clear as later topics discuss the way the eV+ system executes programs, handles subroutines, and stores the values of program variables. The important thing to remember as you become familiar with the eV+ file structure is that when you load or store a file, you are not necessarily loading or storing a single program.

Subdirectories

In normal operations, you may wind up with large quantities of files. If you had no way of organizing and isolating related files, your disk drives would quickly become awkward and difficult to use. The eV+ system supports use of a directory structure to help organize and manage disk files. Subdirectories and the directory structure are covered in the next section.

File Names

Each file within a subdirectory must have a unique name. The eV+ system uses this name to keep track of the file and to allow you to access the file. There are also several file extensions that are used for different types of files in the eV+ system.

File Name Requirements

eV+ file name requirements are similar to MS-DOS file name requirements. File names must:

- Have a maximum of eight characters plus a zero- to three-character extension.
- Use only letters (a - z), digits (0 - 9), and the underscore (_) character. (eV+ ignores the case of letters used in file names.)
- Not contain any spaces.
- Use only one period, which marks the beginning of the file name extension.

File Name Extensions

You may use any valid file name for your files. However, you might want to be aware that Omron Adept uses several common naming schemes, including the following file name extensions:

- V+ program/module disk files use the extension **v2**.
- Various calibration programs use the extension **cal**.
- Robot specification files use the extension **xml**.
- Omron Adept's optional AIM software uses the extensions **ovr**, **ov2**, **db**, **mnu**, and **rfd**.

Contents of eV+ Disk File

Disk files may contain three different types of information, as illustrated in the following figure:

- eV+ programs
- eV+ variables and their assigned values
- User Data



Contents of a Disk File

eV+ programs contain coded instructions that tell the computer what to do and which external devices to communicate with.

User data is generated by various eV+ programs, such as camera calibration programs or the Omron Adept AIM database management system.

eV+ variables provide specific information needed by a program. For example, the location values that a robot is to move to can be stored in a disk file.

In general, unless you will be writing custom programs for the eV+ system, you do not need to be concerned about creating new data files. The files will be created already, or will be created and modified automatically by the programs that you run. You may need to copy existing files from one disk to another. Copying disk files is covered in the following sections.

Working with Disk Files

This section tells you how to view, load, copy, and delete stored files.

When the ACE system is used to access the V+ system, the File Explorer is used for most file, directory, and file-management operations. The File Explorer is accessed from the ACE toolbar.

The File Explorer can be used to list, cut, copy, paste, or rename the files on a storage device (for example, a hard drive or Secure Digital (SD) card).

For details, refer to the topic File Explorer in the *ACE User's Guide*.

NOTE: After a file has been deleted, it cannot be recovered.

The following sections describe how to access files when the ACE system is not used, or when the ACE Monitor window is used.

To List the Files on a Disk Drive

The FDIRECTORY command lists all the files in a subdirectory (see Working with Subdirectories on page 43), along with information about each file's size, type, and creation date. After you enter the command:

```
FDIRECTORY ↵
```

the output on the Monitor will look something like this:

```
Directory of D:
D      0    31-May-07  21:02:42  SYSTEM.CPY
D      0    31-May-07  21:02:42  SYSTEM
D      0    31-May-07  21:07:04  CALIB
D      0    31-May-07  21:07:14  SPECDATA
27768/41744 sectors unused/total
```

The following information is included in this display.

- The first and second columns list the file name and extension (e.g., "SYSTEM.CPY").
- The third column lists the file size in disk sectors (each sector holds 512 bytes).
- The fourth column lists special attributes of the file:
 - P** indicates that the file (and any programs it contains) is protected and cannot be modified or copied.
 - D** indicates that the entry is a subdirectory.
 - R** indicates that the file is read-only.
- The last columns list the date and time the file was created or copied.
- The final line of the display lists the total number of unused sectors on the disk and the total capacity of the disk. In this example, the disk drive has 27,768 sectors unused out of 41,744 sectors available on the disk.

Using Wildcards to List Groups of Files

You can list specific groups of files using the wildcard character (an asterisk) and portions of a file name. The command:

```
FDIRECTORY MY*.* ↵
```

will list only the files that begin with "MY". The command:

```
FDIRECTORY *.V2 ↵
```

will list all the files with the extension "V2". The command:

```
FDIRECTORY *Y*.* ↵
```

will list all the files that have a "Y" as the second character of the file name. (Note that in this case, the initial asterisk represents only one character, while the second and third asterisks represent zero or more characters.)

You can also use wildcards in certain file copying and deleting operations.

To Copy a Disk File

The FCOPY command allows you to:

- Copy files from one disk or subdirectory to another.
- Change the name of a file while copying the file (useful for making backup copies of a file).

The command:

```
FCOPY new_file = old_file ↵
```

will make a copy of `old_file`, name the copy `new_file`, and place it in the same subdirectory as `old_file`. The command:

```
FCOPY my_file.v2 = a:my_file.v2 ↵
```

will make a copy of `my_file.v2` from the A drive and place it in the current subdirectory with the name `my_file.v2`. For details on setting the current directory, see *The Current Directory* on page 42.

The FCOPY command can be used for copying a few files. However, if you are copying many files, the ACE File Explorer should be used. For more details, see the topic *File Explorer* in the *ACE User's Guide*.

To Rename a Disk File

To change a file name without changing its location, use the FRENAME command. The command:

```
FRENAME new_name = old_name ↵
```

will change the name of disk file "`old_name`" to "`new_name`". Note that any disk or directory specification must be included in the specification of "`new_name`", not in "`old_name`".

To Delete a Disk File

Disk files that are no longer used or needed should be deleted from the disk. This recovers space on the disk for other files. If you are copying or storing files to a disk and get a "disk full" message, the only way to continue with the copy or store operation is to delete some files from the disk (or specify a different disk device). The FDELETE command removes files from the disk.

NOTE: After a disk file has been deleted, it cannot be recovered.

To delete a single file, enter the command:

```
FDELETE old_file ↵
```

and the file "old_file" will be removed from the current subdirectory (after you respond to a confirmation prompt). Note that the file name "old_file" is different from "old_file.v2". You must specify the entire file name when deleting files.

To delete all the files in the current subdirectory with the extension "v2", enter the command:

```
FDELETE *.v2 ↵
```

To delete all the files in the current subdirectory, enter the command:

```
FDELETE *.* ↵
```

If you have the optional AIM software, the AIM File Manager provides a graphical interface for copying, deleting, renaming, and storing files. See the *AIM Baseline User's Guide* for details.

Using Directories

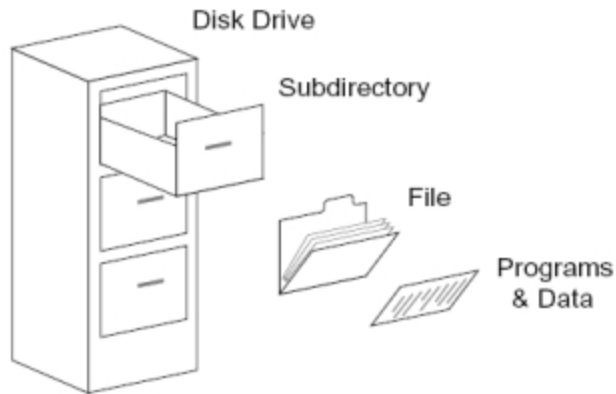
When the ACE interface is used, the Process Manager automates much of the programming and program/file management through a point-and-click interface. For those who wish to access the eV+ system directly, the Controller Development Tools are used for program/file management and direct V+ monitor prompt operations. These tools are accessed from the Controller Development Tools () icon on the ACE toolbar. For more details, see the topic Controller Development Tools in the *ACE User's Guide*.

The ACE File Explorer provides a graphical interface for copying, deleting, renaming, and storing files. It is accessed from the File Explorer () icon on the ACE toolbar. For more details, see the topic File Explorer in the *ACE User's Guide*.

The remainder of this topic describes the operations when accessing the V+ system directly through the Monitor window.

Understanding the Directory Structure

The eV+ disk file structure allows you to organize your disk files in a manner similar to a filing cabinet. In this analogy the file cabinet represents a disk drive or Secure Digital card, the individual drawers and dividers in the cabinet are referred to as directories or subdirectories, the file folders in the drawers represent eV+ disk files, and the papers in a file folder represent the programs and other objects that can be placed in a disk file. The following figure shows this file structure.



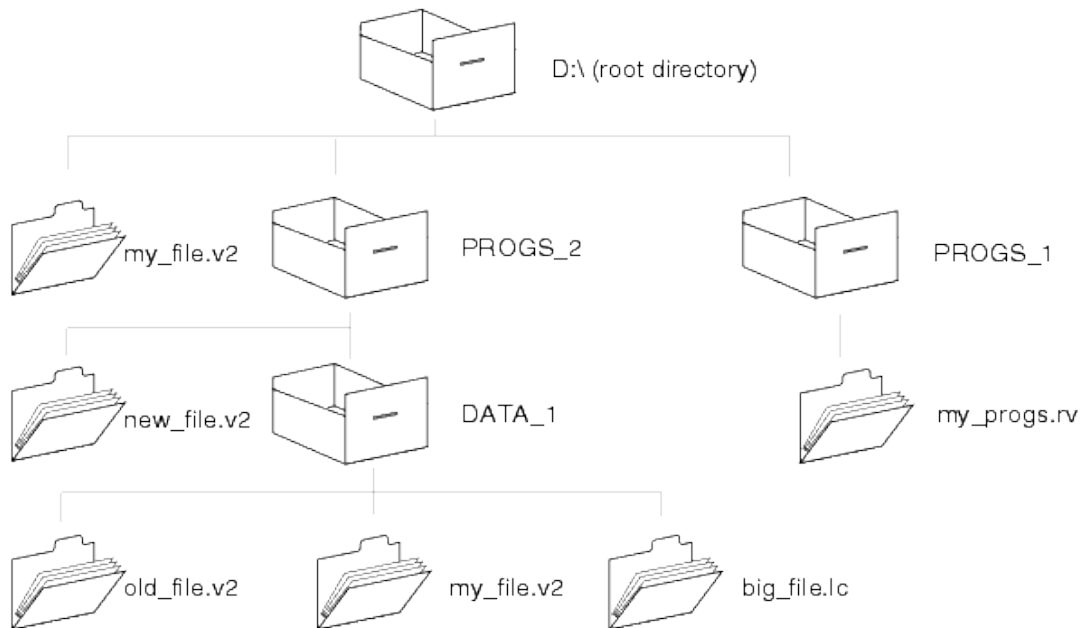
Disk File Structure

The eV+ file structure is a hierarchical file system. Files and subdirectories are organized in a multilevel tree structure. In order to get to the files stored in lower levels, you must go through all the levels starting at the top. The top-level directory is called the *root* directory (it forms the root of the inverted tree). The root directory can contain files and other directories. Each directory under the root directory can also contain other directories and files.

There are a few simple rules for using files and subdirectories:

- Directories have the same naming requirements as files, except they cannot have an extension.
- At any one level, all directory and file names must be unique.
- Each disk can have only one root directory.

The following figure shows a simple directory structure.



Directory Structure

Understanding Paths

A *path* consists of the directions to a specific file or directory within the directory structure. Paths are used with LOAD, FCOPY, FDELETE, and other file-related commands to identify a particular file. A full path always starts at the root directory and moves through each level until you reach the desired file. In the above example, the path to new_file.v2 starts at the root directory and goes through subdirectory PROGS_2 before arriving at new_file.v2.¹

To specify a path in a command:

- Start with the letter of the drive you want to access followed by a colon (":").
- Separate the elements of the path with backslashes ("\").
- Enter the names of the elements exactly as they were created.

To load the file big_file.lc, enter the command:

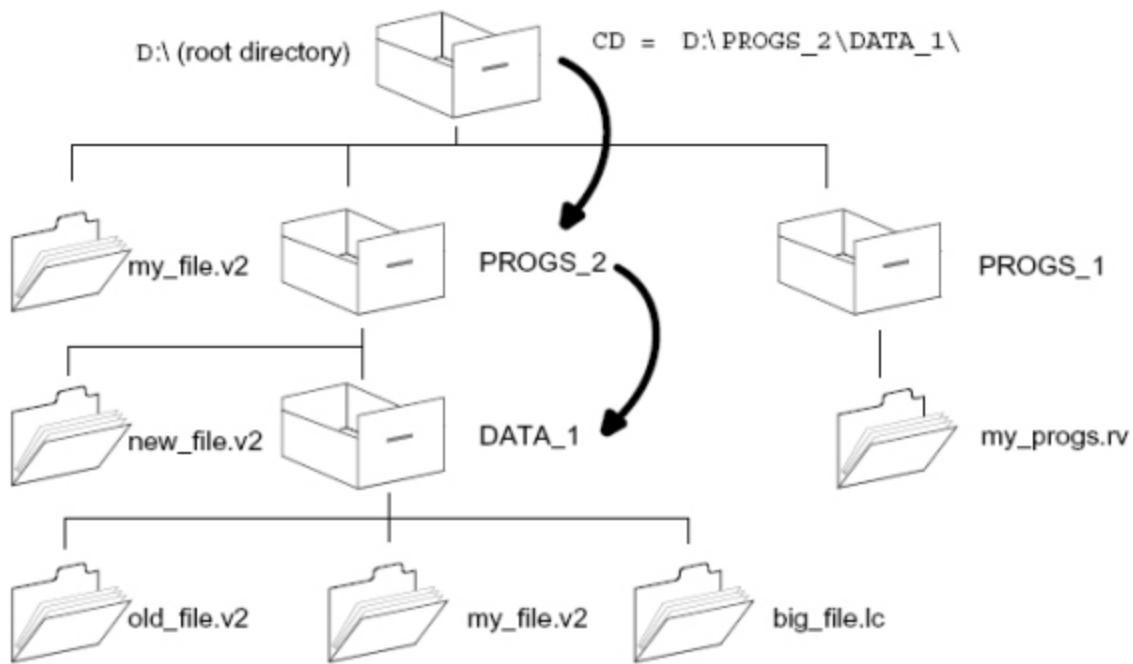
```
LOAD D:\PROGS_2\DATA_1\big_file.lc ↵
```

The Current Directory

When you are working exclusively within a single subdirectory, it is inconvenient to specify the full path for every file operation. The CD command specifies a default path to use with all subsequent commands. To make DATA_1 the default (current) directory, enter the command:

```
CD D:\PROGS_2\DATA_1 ↵
```

The following figure shows the effect of the above command.



Default Directory Specification

After the above default path has been set, the command:

```
LOAD my_file.v2 ↵
```

will load the copy of "my_file.v2" that is in the subdirectory "DATA_1" rather than the copy in the root directory.

You can also change relative to the current directory with the CD command. If the current directory is D:\PROGS_2 and you enter the command:

```
CD DATA_1 ↵
```

The current directory will change to D:\PROGS_2\DATA_1. If you now enter the command:

```
CD .. ↵
```

the current directory will become D:\PROGS_2\. The ".." tells the system to go up one level in the directory structure (note that there is a space between "CD" and "..").

Absolute Paths

The default subdirectory set with a CD command can always be overridden by using a full path specification. An *absolute* or *full* path starts with a backslash and may include the drive designation. The command:

```
LOAD D:\PROGS_1\my_progs.rv ↵
```

will load the disk file my_progs.rv in the PROGS_1 subdirectory regardless of the default set with a CD command.

Relative Paths

A partial path can be added to any default path that has been set with a CD command. The following commands will load the file old_file.v2 (from the directory structure shown in the figure Default Directory Specification):

```
CD D:\PROGS_2 ↵  
LOAD DATA_1\old_file.v2 ↵
```

To Display the Current Path

To show the current path, enter the command:

```
CD ↵
```

The current directory path will be displayed.

Path Examples

Assume you set the default path to D:\PROGS_2\. The following commands will attempt to load the indicated files (from the directory structure shown in the figure Directory Structure):

LOAD my_file.v2 ↵	Loads the disk file my_file.v2 located in D:\PROGS_2\
LOAD DATA_1\my_file.v2 ↵	Loads the disk file my_file.v2 located in D:\PROGS_2\DATA_1\
LOAD D:\PROGS_2\DATA_1\my_file.v2 ↵	Loads the disk file my_file.v2 located in D:\PROGS_2\DATA_1\

Working with Subdirectories

To View the Contents of a Directory

To view all the files in the current subdirectory, enter the command:

```
FDIRECTORY ↵
```

To view all the files in a subdirectory other than the current one, use a path. The command:

```
FDIRECTORY D:\PROGS_2\DATA_1\ ↵
```

will display the contents of the DATA_1 subdirectory.

To View Groups of Files

Use wildcards to view a group of files. The command:

```
FDIRECTORY *.v2 ↵
```

will display the files in the specified directory that have the extension "v2". The command:

```
FDIRECTORY D:\PROGS_2\file*.* ↵
```

will display the files in the current directory that have names beginning with "file".

To Create a Subdirectory

To create a new subdirectory directly below the current directory, enter the command:

```
FDIRECTORY/C dir_name ↵
```

A full path can also be used when creating a subdirectory:

```
FDIRECTORY/C D:\PROGS_2\DATA_1\dir_name ↵
```

To Delete a Subdirectory

NOTE: After a directory has been deleted, it cannot be recovered.

Before a subdirectory can be deleted, the subdirectory must be empty (no disk files and no subdirectories can be in the directory).

The command:

```
FDELETE *.* ↵
```

will delete all the files (but not subdirectories) in the current subdirectory. Since all file-related commands look only at the current directory or the directory specified at the end of a path, files in any other directories are not accessed regardless of whether they are above, below, or on the same level as the current directory. The subdirectory DATA_1 can be deleted with either of the following groups of commands (assuming there are no files in DATA_1):

```
FDELETE D:\PROGS_2\DATA_1\*.* ↵  
FDIRECTORY/D D:\PROGS_2\DATA_1 ↵
```

or:

```
CD D:\PROGS_2 ↵
```

```
FDELETE DATA_1\*. * ↵  
FDIRECTORY/D DATA_1 ↵
```

¹ Directory names and file names are shown here in capital and small letters to help illustrate the directory structure. They are normally both shown in all capital letters (and can always be typed either way).

Executing Programs

The following topics are described in this chapter:

Understanding Programs	47
Removing Objects from System Memory	50
Understanding Program Modules	53

Understanding Programs

Your Omron Adept controller will not perform any useful work until you "run" a program. This is a two-step process: First you LOAD a disk file that contains the eV+ program, then you EXECUTE or COMMAND one of the programs that is contained in the file.

When the ACE interface is used, the Process Manager automates much of the programming and program/file management through a point-and-click interface. For those who wish to access the eV+ system directly, the Controller Development Tools are used for program/file management and direct V+ monitor prompt operations. These tools are accessed from the Controller Development Tools () icon on the ACE toolbar. For more details, see the topic Controller Development Tools in the *ACE User's Guide*.

The remainder of this topic describes the operations when accessing the V+ system directly through the Monitor window.

Disk Files

A disk file may contain either or both of two different types of programs: Monitor command programs and executable eV+ programs.

NOTE: Because they are handled the same way, files stored on a Secure Digital (SD) card are referred to as disk files.

Monitor command programs perform system-level functions such as loading files, changing the default path, and starting executable eV+ programs. These programs are similar to MS-DOS batch programs or UNIX scripts. The commands detailed in the rest of this manual can be put into Monitor command programs. The section Command Programs on page 60 provides details on writing basic Monitor command programs.

Executable eV+ programs contain the logic, motion control, and vision instructions that actually control a workcell. Creating and modifying executable eV+ programs is covered in the *eV+ Language User's Guide* and *eV+ Language Reference Guide*.

The documentation for program files delivered by Omron Adept specifies which program in a disk file is the main program, and whether the main program is a Monitor command program or an executable eV+ program. The documentation for program files supplied by third parties should include this same information.

You can see what type of programs a disk file contains by entering the command:

```
FLIST file_specification ↵
```

The contents of the file will scroll up the Monitor window. Use the Scroll Lock key to pause scrolling. If a program is a Monitor command program, all the lines in the program will be preceded with "MC" (or ";", which indicates a comment). Remember, a disk file can contain many programs, including both Monitor command and executable eV+ programs.

To LOAD a Disk File

The controller cannot use the programs and information stored in disk files until they are loaded into system memory (RAM).

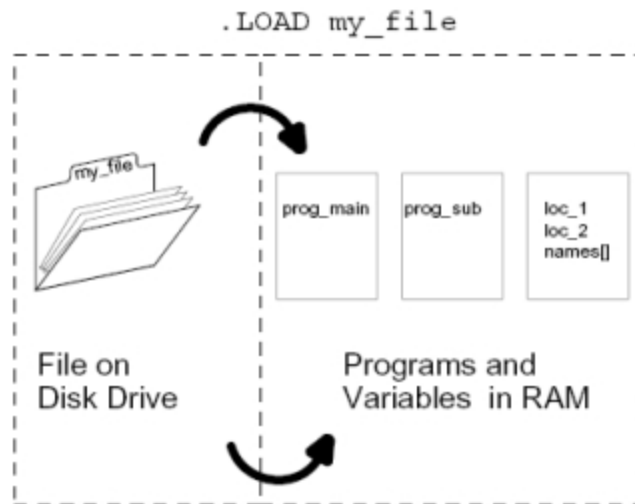
A file can be loaded through the ACE graphical interface, or with the eV+ LOAD command typed at the eV+ Monitor prompt. For information on using the ACE system, refer to the Controller Development Tools topic in the *ACE User's Guide*.

The LOAD command places a copy of the contents of a disk file in RAM. The figure below shows the effect of the command:

```
LOAD my_file ↵
```

This command places the contents of the disk file "my_file" into RAM. "my_file" contains two programs and the values of variables used by those programs.

The LOAD command does not actually start a program executing; it simply places a copy of the disk file contents into system RAM so additional commands can start the program, or modify the program, or modify the values of the program variables.



LOADing a Disk File

To see all the programs that have been loaded into memory, use the DIRECTORY command. (FDIRECTORY lists files on the SD card; DIRECTORY lists programs in RAM.)

To Start a Monitor Command Program

Once a Monitor command program is resident in system memory, you can use the COMMANDS command to start the program.

If the program "init.system" is in RAM (loaded using the ACE Program Manager) and it is a Monitor command program, the command:

```
COMMANDS init.system ↵
```

will start the program. If "init.system" is an executable eV+ program, the preceding instruction will result in the error:

```
*Can't mix MC & program instructions*
```

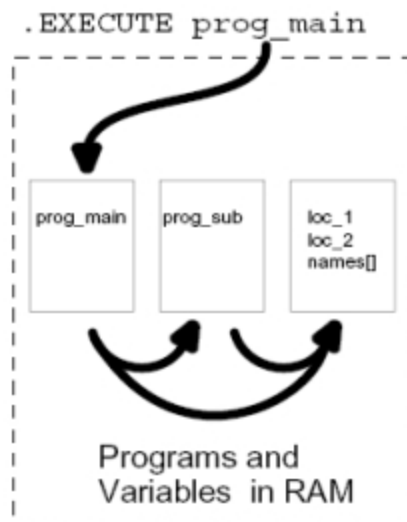
To Start an Executable V+ Program

Once an executable eV+ program (and possibly its related programs and variables) has been LOADED into RAM, the EXECUTE command initiates execution of the program.

The following figure shows the effect of the command:

```
EXECUTE prog_main ↵
```

The program "prog_main" was loaded as part of the disk file "my_file". When is pressed, the program begins executing. In the course of execution, "prog_main" makes use of "prog_sub" and the variables that were loaded with the disk file "my_file".



EXECUTing eV+ Programs

NOTE: If the executable program will control a motion device, the device must be powered up and calibrated before the program is executed. See To Power Up the Motion Device on page 24.

Which Program Should I Execute?

Normally, a disk file containing programs will have one main program that uses the other programs in the file. All program files delivered by Omron Adept follow these conventions:

- There is only one *main* program per disk file.
- The name of the main program is the name of the disk file (without any extension) preceded by "a.". For example, the disk file that holds the program could be named "MYSAMPLE.V2". The main executable program in that file is "a.mysample".

These conventions are not requirements: Programs that are not supplied by Omron Adept may follow other conventions. See the documentation provided with the files for details on which program to execute.

To Halt Execution of a Program

Emergency Stop

To immediately halt a robot or motion device and halt execution of a program, activate any remote emergency stop connected to the system (such as the red panic button on the optional remote front panel). This will immediately turn off power to the motion device and stop the execution of any program currently attached to the robot or motion device.

Early Program Termination

To halt a program in a non-emergency situation, enter the command:

```
ABORT ↵
```

ABORT can be entered even though the normal system prompt (".") is not displayed during program execution. When you type the first letter, an "*" will appear where the dot prompt would normally appear. This special prompt is displayed when commands are being entered while a program is executing.

NOTE: If a robot is in motion when the ABORT command is entered, the robot will NOT halt immediately, but will complete the current motion.

The only time you cannot enter the ABORT command is when the system is waiting for input from the keyboard. For example, suppose the following message is displayed in the Monitor window:

```
Are you sure Y/N?
```

The typing cursor at the end of the line tells you the system is waiting for you to enter a response at the keyboard. To halt program execution at this point, press CTRL+Z. Program execution will halt with an "unexpected end of file" error message.

Removing Objects from System Memory

When a program completes, it is not automatically removed from system memory (RAM). To free up the RAM for use by other programs and data, objects in memory must be specifically removed with a DELETE command. The LOAD command will not overwrite programs that reside in system memory. Therefore, if you want to load new programs with identical names, you must delete the programs currently in RAM.

When the ACE interface is used, the Process Manager automates much of the programming and program/file management through a point-and-click interface. For those who wish to access the eV+ system directly, the Controller Development Tools are used for program/file management and direct V+ monitor prompt operations. These tools are accessed from the Controller Development Tools () icon on the ACE toolbar. For more details, see the topic Controller Development Tools in the *ACE User's Guide*.

The remainder of this topic describes the operations when accessing the V+ system directly through the Monitor window.

Understanding the Delete Commands

The following table summarizes the various delete commands.

The Delete Commands

Command	Action
DELETE	Deletes specified programs and any programs or variables that are used in those programs (but are not used by other programs not being deleted).
DELETEM	Deletes a program module and all programs in the module. The variables used by the programs, and referenced programs outside the module, are not deleted. See Understanding Program Modules.
DELETEP	Deletes specified programs from memory. Other programs used by the specified programs are not deleted, and variables used by the program are not deleted.
DELETEL	Deletes specified location variables.
DELETES	Deletes specified string variables.
DELETER	Deletes numeric (real) variables.
ZERO	Deletes ALL the programs and data in system memory (the operating system is not affected).

In normal operations, you are most likely to use the DELETE, DELETEM, and ZERO commands.

Objects do not have to be removed from RAM before different disk files are loaded and other programs are executed (unless you are loading a program with the same name as one already in RAM). However, if you get an "out of memory" message while loading a disk file, the only way to complete the load operation is to delete objects from system memory.

In addition to the commands described previously, the FDELETE command removes disk files from the SD card.

NOTE: Once a disk file is deleted from the storage device, it cannot be recovered.

FDELETE does *not* remove objects from RAM (if they have been loaded). The DELETE commands remove objects from RAM. This does not delete a program that is being executed or is present on the stack for any execution task. These commands do not affect the disk files from which the objects were loaded. After objects have been removed from RAM, they can be reloaded using the LOAD command (but, of course, only if they are stored on disk).

Understanding Program Modules

When a disk file is loaded, a program "module" is created that holds the names of all programs loaded from that disk file. The name of the created module is the same as the first program loaded. A module allows you to remove all the programs loaded from a disk file without disturbing programs loaded from other disk files.

For example, if the command:

```
.LOAD my_file.v2 ↵
```

loaded the following programs:

```
a.my_file
mf.check.stuff
mf.do.stuff
mf.init
```

The MDIRECTORY command will list the modules resident in system memory:

```
.MDIRECTORY ↵
  .MODULE a.my_file
```

The DELETEM command will remove the module and all the programs in the module from system memory:

```
.DELETEM a.my_file ↵
Are you sure (Y/N)? Y ↵
.MDIRECTORY ↵
.DIRECTORY ↵
```

Using Permanent File Storage

The following topics are described in this chapter:

Using Digital Storage Cards	55
Using Hard Drives	55

Using Digital Storage Cards

In place of a traditional hard drive, the SmartController EX system uses a Secure Digital (SD) card. An SD card has no moving parts and, therefore, adds to the reliability and durability of Omron Adept's products. It is also removable and can be moved to any other SmartController EX system, should this be necessary for replacement or testing purposes.



WARNING: Use suitable measures for eliminating electrostatic discharge during removal and installation of the SD card. This includes, but is not limited to, use of a grounded wrist strap while performing this operation.

Because the eV+ system is very efficient, most, if not all, eV+ applications will fit onto the SD card.

The SD card is addressed as drive D.

Backing Up Data

SD cards are very reliable storage devices, but they can fail. Data on a failed card may not be readable and may be completely lost. Therefore, Omron Adept strongly recommends that you make periodic backups of your card. The eV+ system and user data can be backed up using the Backup utility and File Explorer tool in the ACE software. In the event of a card failure, the backup data can be used to restore the contents to a new SD card. For details on creating a system backup and using the File Explorer to copy files, see the *ACE User's Guide*.

Obtaining Formatted SD Cards

The SD card is factory-preformatted and should never need formatting, unless a failure occurs and the card must be replaced. The eV+ system requires that special formatting operations be performed before a card can be used in an Omron Adept controller.

You can obtain additional preformatted SD cards from Omron Adept. Please contact Omron Adept Customer Service for assistance.

Using Hard Drives

The ACE software allows you to use the PC hard drive, or an external hard drive that is connected to the PC, for storage of your ACE software, application programs, eV+ data and system backups, and other information. Therefore, in the ACE environment, you are not restricted to storing everything on the controller. For more information on storing information to your PC hard drive with the ACE software, see the *ACE User's Guide*.

Customizing an Omron Adept System

The following topics are described in this chapter:

Introduction	57
Customizing the Controller Hardware	57
Changing the Controller Configuration	57
System Software Switches	58

Introduction

The majority of system customization operations are made through the ACE software. A few operations require setting hardware (DIP) switches on the controller. For more details, see the Configuration Tools chapter in the *ACE User's Guide*.

Most controller configuration changes will not take effect until the controller has been restarted. The ACE software automatically prompts you when a change has been made that requires a controller restart.

The changes made with the ACE software are written to the operating system SD card. Therefore, if you make changes to the configuration and then start the system with a different SD card, a different controller configuration might be in place. You can use the ACE system to make an exact copy of the operating system and its current configuration for backup. If a card failure results in the loss of your operating system, you can load the operating system directly from the backup. You can also copy the system files from that backup to restore the copy on a replacement SD card. For more details on creating a system backup, see the *ACE User's Guide*.

Customizing the Controller Hardware

When the controller is turned on, the system first looks at the settings of the configuration DIP switches. The DIP switches are located on the front of the controller. The DIP switches determine initial options the system uses during the boot procedure. See the *Adept SmartController User's Guide* for the location of the DIP switches on the front of the controller.

DIP switch #1 controls what IP address to use. If switch #1 is ON, eV+ will use the default IP address. For details on the default IP address, see the *Adept SmartController User's Guide*. If switch #1 is OFF, eV+ will use the IP address stored on the Secure Digital (SD) card.

DIP switch #2 controls the communication interface. If switch #1 is ON, eV+ will listen on the serial port labeled TERM. If switch #2 is OFF, eV+ will use the normal connection through Ethernet.

NOTE: All the other switches should be kept on the OFF position. They are reserved for future use.

Changing the Controller Configuration

The ACE software, contains a controller configuration tool, which is accessed through the Configure button on the Controller object editor. This configuration tool allows you to access:

- the controller configuration parameters
- the controller licenses

For more details, see the Controller Editor topic in the *ACE User's Guide*. For information on configuring other aspects of your system, see the Configuration Tools chapter in the *ACE User's Guide*.

System Software Switches

Several controller options can be enabled/disabled with the ENABLE/DISABLE Monitor commands.

Most of these *system switches* should be controlled through the ACE interface, or by executing programs, and not changed from the Monitor prompt.

The remainder of this topic describes the operations when accessing the V+ system directly through the Monitor window.

Basic System Switches

The following table shows the system switches you are most likely to change from the Monitor prompt.

Basic System Switches

Switch	Use	Default
BELT	Used to turn on the conveyor tracking features of eV+ (if the option is installed). This switch must be enabled before any of the special conveyor tracking instructions can be executed. When the BELT system switch is disabled, the conveyor tracking software has minimal impact on the overall performance of the system. This will usually be set from the belt object editor in the ACE interface.	OFF
DRY.RUN	Enable/disable sending of motion commands to the robot. Enable this switch (i.e., disable motion commands) to test programs for proper logical flow and correct external communication without having to worry about the robot running into something. (Also see the TRACE switch, which is useful during program checkout.) The pendant can still be used to move the robot when DRY.RUN is enabled.	OFF
FORCE	Controls whether the stop-on-force feature of the eV+ system is active. (To be able to enable this switch, the force option must be installed on your system.)	OFF

Switch	Use	Default
POWER	Tracks the status of Robot Power; this switch is automatically enabled whenever Robot Power is turned on. This switch can also be used to turn Robot Power on or off—enabling the switch turns on Robot Power, and disabling the switch turns off Robot Power. The ACE system toolbar provides a Power () icon that can be used in place of this switch.	OFF

To enable a system switch, type `ENABLE` followed by a space and the switch name. For example:

```
ENABLE BELT
```

To disable a switch, use the `DISABLE` Monitor command. For example:

```
DISABLE BELT
```

Command Programs

The following topics are described in this chapter:

Introduction	61
Creating Command Programs	61
Running Command Programs	62
Controlling Command Programs	63
Example Command Program	64

Introduction

Command programs are special programs that consist of a series of Monitor commands. Command programs automate system startup and other frequently executed Monitor command sequences. Command programs can be started from the eV+ monitor prompt or from the manual control pendant. With the auto-start feature, a command program can be started automatically when the Omron Adept system is turned on.

Command programs are most commonly used in V+ based applications, as described in the following cases:

- At system startup, the Omron Adept controller autostart switch will launch the command program "auto" located in D:\AUTO.V2. For details, see Autostarting a Command Program on page 62.
- To save a V+ module and application variables.
Often, a command program will define the programs in a module and then perform a backup/save operation.

When the ACE software is used, this is not really needed, because the V+ Module object saves the programs in the workspace. The same can be said for V+ application variables contained in the V+ Global Variable Collection object. It is all saved in the workspace by simply clicking the Save button. For more details on the V+Module object and the V+ Global Variable Collection object, see the *ACE User's Guide*.

Creating Command Programs

Command programs can be created using the ACE system.

Either right-click in the Workspace Explorer pane, and select:

New > Programs > V+ Module

or select:

Controller Development Tools > Program Manager > Create New Program

a V+ program editor opens, which allows you to create and edit your V+ programs. For more details on using the program editor, see the *ACE User's Guide*.

Command Program Contents

Command programs can contain all the eV+ Monitor commands except the DONE, TEACH, and ZERO commands, and editor commands.

Several commands require subsequent input from the user. For example, CALIBRATE and DELETEP normally require confirmation, and POINT and HERE normally require responses to the query "Change?". When commands such as these are processed in a command program,

the commands will be performed without any prompts being issued. Make sure your Monitor command programs can execute correctly without operator response.

Every nonblank line of a Monitor command program must contain either a Monitor command or a comment.

Command Program Names

Command program names can have up to 15 characters, must begin with a character, and can contain letters, numbers, periods, and underline characters. Command programs are stored as elements in the eV+ disk file. The disk file name, not the command program name, must conform to eV+ disk file name conventions (see Using Files on page 33).

Running Command Programs

There are several ways to run command programs: from the eV+ monitor prompt, or automatically when the controller is turned on.

Starting Command Programs From the Monitor Prompt

The COMMANDS Monitor command starts a command program. To run the command program that was created in the previous section, enter the commands:

```
LOAD my_cmd.pg ↵  
COMMANDS my.cmd.program ↵
```

NOTE: The EXECUTE command expects programs that contain eV+ program instructions, not Monitor commands. If you use the EXECUTE command to run a program with lines that begin with "MC", the program will abort and you will get a message indicating that you cannot mix program instructions and Monitor commands.

Autostarting a Command Program

A Monitor command program can be loaded and started automatically when the controller is turned on, if the following conditions have been met:

1. The autostart option must be enabled in the controller NVRAM configuration settings.
2. A disk file with the name "AUTO.V2" must reside on the default disk. See The Default Disk Path for details on specifying the default disk.
3. The AUTO.V2 disk file must contain a program named "auto". The "auto" program must be a Monitor command program. (Other programs, and data, can be stored in the AUTO.V2 disk file.)

Then, when the controller is turned on, the operating system is loaded, the default disk specification is set, and the following commands are issued automatically:

```
LOAD auto.v2  
COMMANDS auto
```

NOTE: The eV+ system does not wait for HIGH POWER to be turned on, or for any other event or condition. The autostart command program must explicitly invoke any such functions, or they must be performed by programs invoked by the command program. For example, if you want to require that the operator press the PROGRAM START button on the controller when the command program is processed, you must include a WAIT.START command in the command program.

Controlling Command Programs

After a command program is initiated, it will process all the commands in the program until one of the following events occurs:

1. The end of the command program is reached.
2. A WAIT.START command is processed in the command program. (This will suspend processing of the command program until the PROGRAM START button is pressed or until a specified condition is satisfied.)
3. A CYCLE.END command is processed in the command program, and the referenced program task is executing. (That will suspend processing of the command program until the referenced program task finishes executing.)

NOTE: A command program will not be suspended when an EXECUTE command is processed. The command program will initiate execution of the specified application program and then immediately continue with the next command. If you want the command program to wait for the application program to complete before processing the next step in the command program, a CYCLE.END command must be included in the command program after the EXECUTE command.

4. The CTRL key is held down while the C key is pressed. (This will terminate processing of the command program. However, the command being processed, or a program invoked by the command program, will continue to completion.)
5. Another command program is invoked from within the active command program. (Unlike starting an executable program, control will **not** return to the first command program when the second command program completes.)
6. An error condition results when a statement is processing in the command program.

Command Programs and Language Keywords

The eV+ programming language is described in the *eV+ Language User's Guide* and the *eV+ Language Reference Guide*.

Normally, a command program contains only Monitor commands. However, language keywords can be included by using the DO Monitor command. To include a language keyword in a Monitor command program, type a line with the format:

```
MC DO keyword parameter, ...
```

eV+ control instructions (for example, GOTO and IF) are ignored in a command program. Command programs can proceed only in a top-down manner, from the first line to the last.

Example Command Program

The following Monitor command program shows a simple start-up program (the line numbers refer to the descriptions below):

```
1  .PROGRAM auto()
2
3  ; This program will perform startup procedures for the
4  ; motion device, load a program file, move the device
5  ; to a safe location, and execute the main program.
6
7      MC WAIT.START
8      MC ENABLE POWER
9      MC CALIBRATE
10     MC LOAD main.v2
11     MC DO MOVE safe
12     MC CYCLE.END
13     MC EXECUTE motion, -1
14 .END
```

The lines in this Monitor command program are described below:

1. This line is entered automatically when the program is created.
2. This blank line is ignored during execution.
3. This is a comment line. Notice that each comment line begins with a semicolon (";"). Lines beginning with a semicolon (with optional leading spaces) are ignored during execution.
4. This is another comment line.
5. Another comment line.
6. Another blank line.
7. Wait for the operator to toggle the AUTOMATIC/MANUAL switch and press the START soft key on the pendant.
8. Turn on HIGH POWER.



WARNING: Before processing a command program that turns

on HIGH POWER, the operator must make sure the robot area is clear.

9. Initiate the robot CALIBRATE sequence.
10. Load the file "main.v2" using the default path specification.
11. Move the robot to the location named "safe" (assumed to be stored in "main.v2" or already in memory).
12. CYCLE.END is used to wait for the robot to complete the motion to "safe" prior to execution of the next line.
13. Initiate continuous execution of the program "motion" (assumed to have been loaded from "main.v2" or already in memory).
14. This line is added automatically by the editor.

This program could be initiated by the Monitor command "COMMANDS auto". The program could also be initiated automatically when the system is turned on (as described in Running Command Programs on page 62).

System Messages

While the eV+ system is being used, it is possible for hardware and software errors to occur. For example, if commands or instructions are not entered in the correct way, the system rejects the input. The usual response is to output an error message to the system terminal indicating what is wrong so that the user can correct the error.

When using the ACE software to program and interface with the eV+ system, error messages are reported in that interface, usually in the form of pop-up windows. These windows typically contain the error message along with a Details button that can be selected for more information. The message and details can also be copied to the Windows clipboard and then pasted in an email or other document.

Error Messages

The *eV+ Language Reference Guide* contains descriptions of the eV+ error messages and suggests remedies to correct the problems.

NOTE: If the system has more than one robot connected and an error is associated with a specific robot, the robot number is appended to the error message in the form "(Robot #)".