UniSite xpi

Programming System User Manual

Also Covering Legacy UniSite Programmers

981-0014

Data I/O

UniSite-*xpi*™

Programming System User Manual

Also Covering Legacy UniSite Programmers

981-0014-016

June 2001 981-0014-016

Data I/O has made every attempt to ensure that the information in this document is accurate and complete. Data I/O assumes no liability for errors or for any incidental, consequential, indirect, or special damages, including, without limitation, loss of use, loss or alteration of data, delays, or lost profits or savings, arising from the use of this document or the product which it accompanies.

Data I/O Corporation
10525 Willows Road N.E., P.O. Box 97046
Redmond, Washington 98073-9746 USA
(425) 881-6444
http://www.dataio.com

Acknowledgments:

Data I/O is a registered trademark and UniSite-*xpi*, UniSite, Site40, Site48, ChipSite, PinSite, SetSite, Keep Current, MatchBook, and HiTerm Terminal Emulator are trademarks of Data I/O Corporation.

Data I/O Corporation acknowledges the trademarks of other organizations for their respective products or services mentioned in this document.

UniSite is protected under U.S. Patent numbers 4,837,653; 4,840,576; 5,033,084; 5,176,525; and 5,289,118. Other U.S. and Foreign Patents Pending.

Table of Contents

2. Setup and Installation

3. Getting Started

4. Commands

5. Computer Remote Control

6. Translation Formats

7. Messages

List of Figures

Safety Summary

General safety information for operating personnel is contained in this summary. In addition, specific WARNINGS and CAUTIONS appear throughout this manual where they apply and are not included in this summary.

Antistatic Wrist Strap

To avoid electric shock, the antistatic wrist strap must contain a 1 MΩ (minimum) to 10 MΩ (maximum) isolating resistor.

Definitions

WARNING statements identify conditions or practices that could result in personal injury or loss of life.

CAUTION statements identify conditions or practices that could result in damage to equipment or other property.

Legacy UniSite

Information about legacy UniSite products is presented in grayed out text in this manual.

Fuse Replacement

For continued protection against the possibility of fire, replace the fuse only with a fuse of the specified voltage, current, and type ratings.

Grounding the Product

The product is grounded through the grounding conductor of the power cord. To avoid electric shock, plug the power cord into a properly wired and grounded receptacle only. Grounding this equipment is essential for its safe operation.

Power Cord

Use only the power cord specified for your equipment.

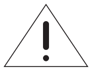
Power Source

To avoid damage, operate the equipment only within the specified line (ac) voltage range.

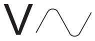
Servicing

To reduce the risk of electric shock, perform only the servicing described in this manual.

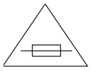
Symbol



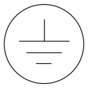
This symbol indicates that the user should consult the manual for further detail.



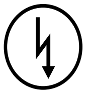
This symbol stands for Volts ac, for example: 120 V∿ = 120 Vac.



This symbol denotes a fuse rating for a user-replaceable fuse.



This symbol denotes earth ground. An antistatic wrist strap with impedance of 1 MΩ (minimum) to 10 MΩ (maximum) can be attached to terminals designated for that function and marked with this symbol.



This symbol denotes dangerous, high voltage is present and precautions should be taken to prevent injury from electrical shock.

Preface

The Preface describes how to contact Data I/O for technical assistance, repair and warranty services, and Keep Current subscription service. It also describes how to access Data I/O's Web site on the World Wide Web.

Data I/O Customer Support

United States

For technical assistance, repair or warranty service, or Keep Current subscription service, contact:

Data I/O Channel and Customer Support
Telephone: 1-800-3-DATAIO (then follow the prompts)
Fax: 425-867-6898
E-mail: techhelp@dataio.com

Data I/O Corporation
10525 Willows Road NE
P.O. Box 97046
Redmond, WA 98073-9746

Canada

For technical assistance, repair or warranty service, or Keep Current subscription service, contact:

Data I/O Customer Resource Center
Telephone: 905-678-0761
Fax: 905-678-7306

Data I/O Canada
6725 Airport Road, Suite 102
Mississauga, Ontario, L4V 1V2
Telephone: 905-678-0761
Fax: 905-678-7306

Japan

For technical assistance, repair or warranty service, or Keep Current subscription service, contact:

Data I/O Japan
Saisho Building 7F
8-1-14, Nishigotanda
Shinagawa-ku
Tokyo 141-0031 Japan
Telephone: 3-3779-2151
Fax: 3-3779-2203

Germany

For technical assistance, repair or warranty service, or Keep Current subscription service, contact:

Data I/O GmbH
Lochhamer Schlag 5
82166 Gräfelfing
Telephone: 89-858-580
Fax: 89-858-5810
E-mail: servicegmbh@data-io.de

Other Countries

For technical assistance, repair or warranty service, or Keep Current subscription service, contact your local Data I/O representative. To find your local Data I/O representative, go to the Data I/O Web site (www.dataio.com) and use the Representative Search feature.

Contacting Data I/O

You can contact Data I/O for technical assistance by calling, sending a fax or sending electronic mail (e-mail).

To help us give you quick and accurate assistance, please provide the following information:

- Product version number
- Product serial number if available
- Detailed description of the problem you are experiencing
- Error messages (if any)
- Device manufacturer and part number (if device related)

Telephone

Call the appropriate Data I/O Customer Support number listed in this Preface. When you call, please be at your programmer or computer, have the product User Manual nearby, and be ready to provide the information listed above.

Fax

Fax the information listed above with your name, phone number, and address to the appropriate Data I/O Customer Support fax number listed in this Preface.

E-mail

To reach Data I/O via e-mail, send a message including your name, telephone number, e-mail address, and the information listed above to the following address:

`techhelp@dataio.com`

World Wide Web (www.dataio.com)

The Data I/O Web site includes links to online information about technical products, general information about Data I/O, a list of sales offices, a representative search feature, and technical user information such as application notes and device lists.

To access the Data I/O Web site, you need an Internet account with Web access, and a Web browser. The address of Data I/O's Web site is:

http://www.dataio.com

Warranty Information

Data I/O Corporation warrants this product against defects in materials and workmanship at the time of delivery and thereafter for a period of one (1) year. The foregoing warranty and the manufacturers' warranties, if any, are in lieu of all other warranties, expressed, implied, or arising under law, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Data I/O maintains customer service offices throughout the world, each staffed with factory-trained technicians to provide prompt, quality service. For warranty service, contact Data I/O Customer Support at the numbers listed in this Preface.

Keep Current Subscription Service

To keep your product up-to-date with the latest features and device support, Data I/O offers the Keep Current™ Subscription Service, a one-year renewable subscription that incorporates manufacturer-recommended changes to existing device support to maintain optimum yields, throughput, and long-term reliability.

For more information or to order Keep Current Subscription Service, contact Data I/O Customer Support.

Repair Service

After the warranty period expires, repair services are available at Data I/O Service Centers on a time-and-materials basis, and through a fixed price annual agreement that covers all parts and labor needed to correct normal malfunctions. The annual agreement includes semiannual performance certification.

For more information or to order a Repair Service Contract, call Data I/O Customer Support.

End User or Address Change

If the end user for this product or your address has changed since the Registration Card was mailed, please notify Data I/O Customer Support at the numbers listed at the front of the Preface. This ensures that you receive information about product enhancements. Be sure to include the product serial number, if available.

1. Introduction

Product Description

The UniSite-*xpi*™ Universal Programmer is a tool for programming, verifying, and testing all programmable device technologies and packaging. The many facets of UniSite-*xpi* allow it to address a variety of needs in both engineering and manufacturing environments.

The UniSite-*xpi* features all the benefits of the legacy UniSite universal device programmer with the additional ability to quickly and easily handle devices that require large data files, as well as other enhancements that make the programmer easier to use in production or engineering environments.

Standard Configuration

Standard configuration of the UniSite-*xpi* System includes 9 MB of RAM, a Mass Storage Module (MSM) for non-volatile internal bulk data storage, pin driver boards for 68+ pin devices, and a Site48HS Module for support of DIP devices up to 48 pins.

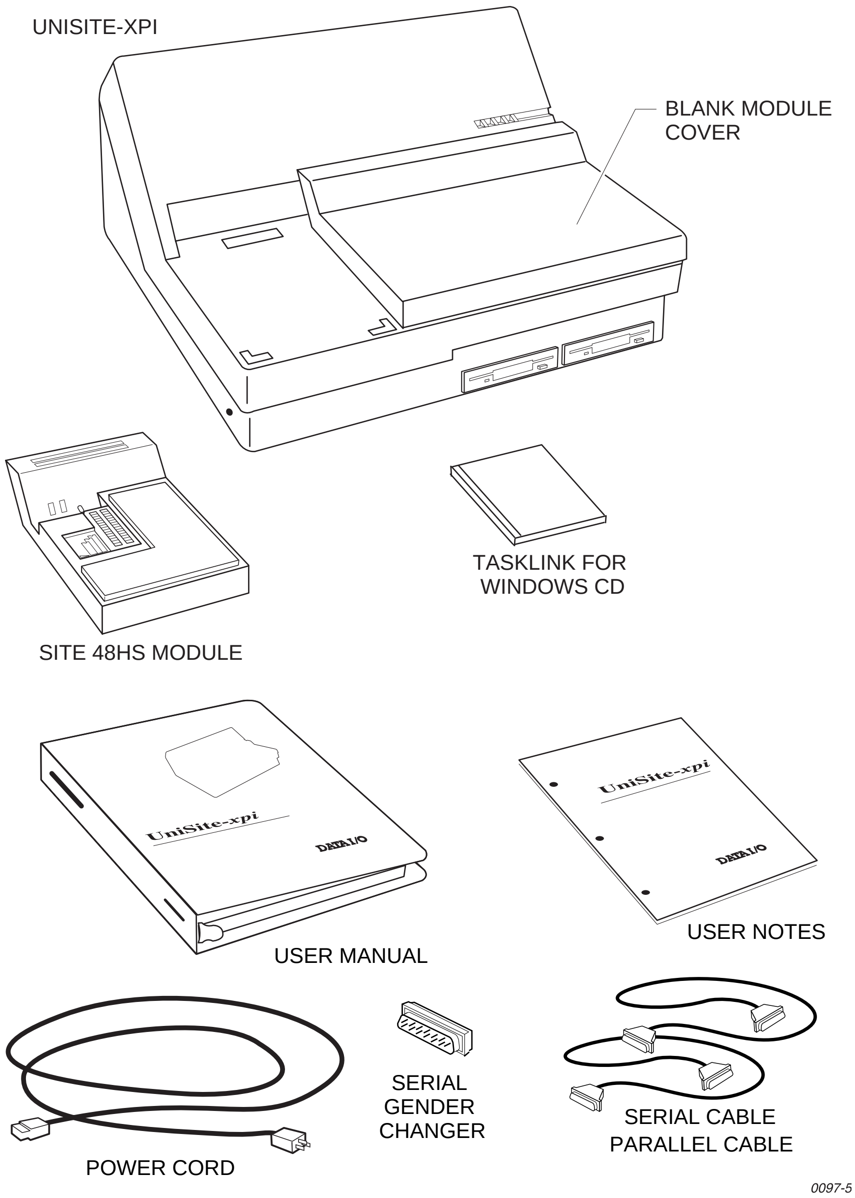
Options

An available option for the UniSite-*xpi* is the PinSite Module. To accompany the PinSite Module, you can purchase either Standard Bases (for support of devices up to 84 pins in packages such as SOIC, PLCC, LCC, and PGA) or the PPI Base (for support of devices greater than 84 pins in packages such as DIP, PLCC, LCC, QFP, PGA, and SOIC).

Contents of Package

Figure 1-1 illustrates the contents of the UniSite-*xpi* system. Check the contents of your UniSite-*xpi* against the items shown in Figure 1-1.

Figure 1-1
Contents of the UniSite-xpi Universal Programmer



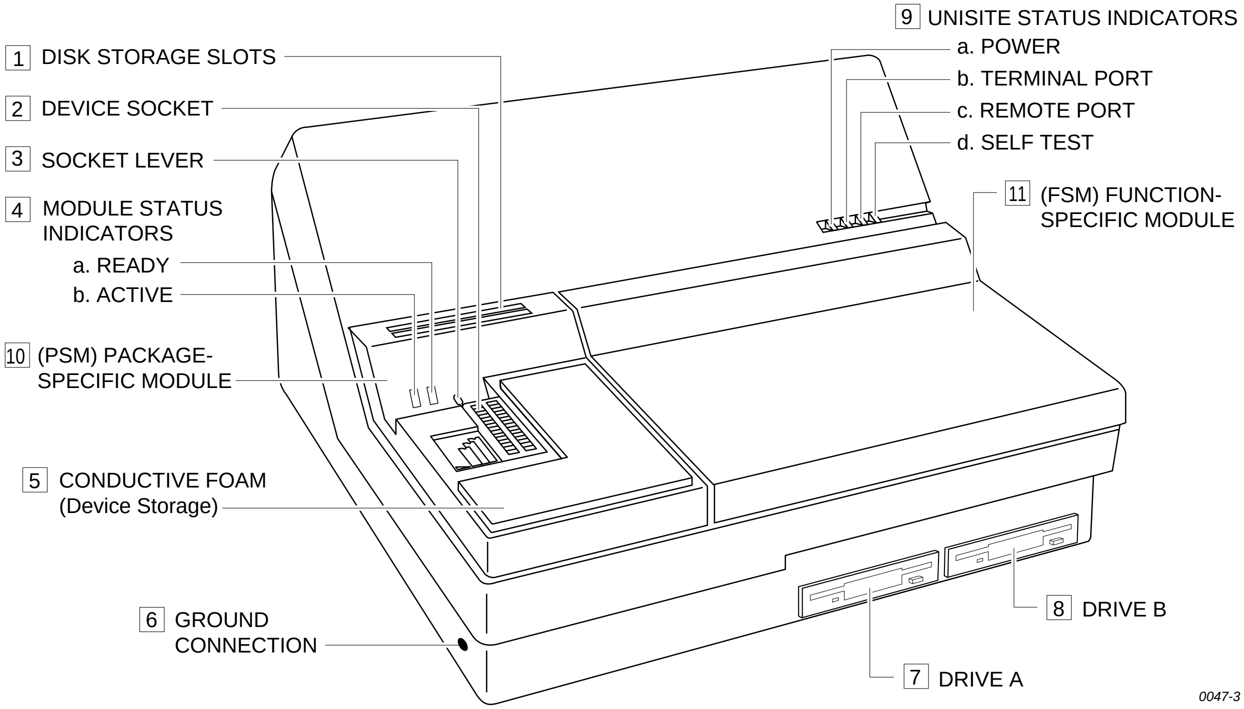
Note: You will not receive a blank module cover if you ordered a PinSite Module when you ordered your UniSite-xpi.

UniSite-*xpi* External Features

The Front Panel

The front panel features of UniSite-*xpi* are shown in Figure 1-2.

Figure 1-2
Front Panel Features



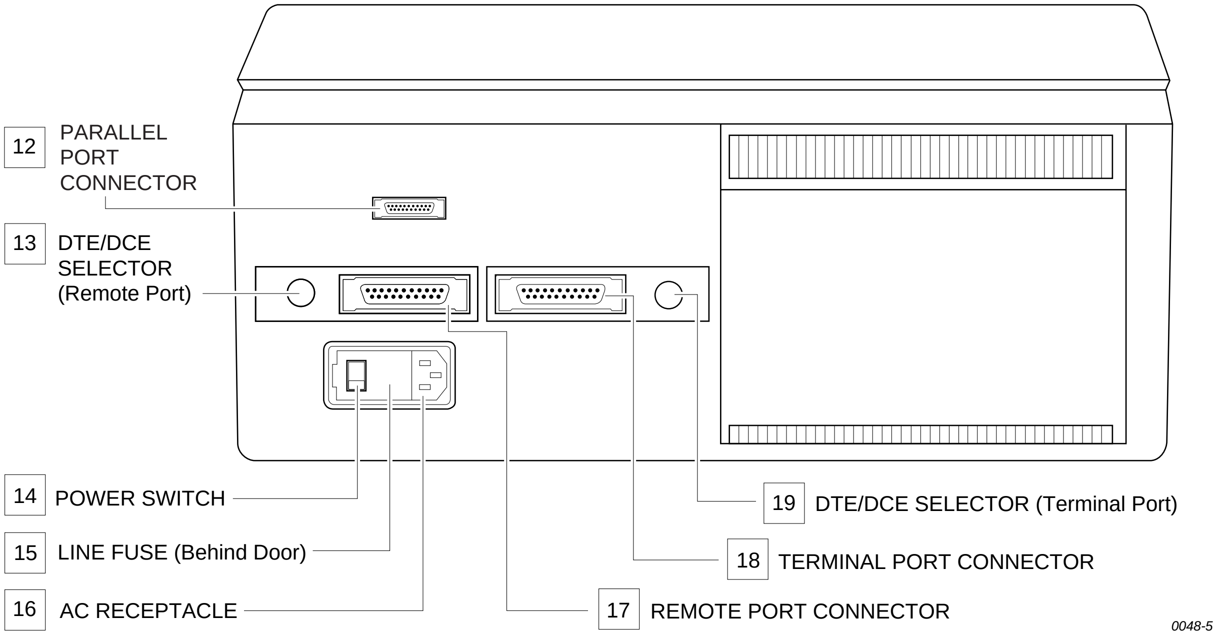
1. **Disk Storage Slots**—A convenient place to store two data disks.
2. **Device Socket**—Holds the device to be programmed or the master device to be read.
3. **Socket Lever**—Locks the device into the device socket.
4. **Module Status Indicators**—Provide information about the operational status of the module.
   a) **ACTIVE Indicator**—This lamp is lit when a device-related operation is in progress.
   b) **READY Indicator**—This lamp is lit when the device socket is ready to accept a device.
5. **Conductive Foam**—Provides a convenient, safe place to store static-sensitive devices while using UniSite-*xpi*.
6. **Ground Connection**—Connect an antistatic wrist strap here.
7. **Drive A**—The main disk drive. Insert the System disk here.
8. **Drive B**—The auxiliary disk drive. Insert an Algorithm disk here.

9. **UniSite-*xpi* Status Indicators—**These indicators provide information about UniSite-*xpi*'s operational status:
    a) **Power Indicator**—Lights when the power is on.
    b) **Terminal Port Indicator—**Lights when equipment is connected properly to the Terminal port.
    c) **Remote Port Indicator—**Lights when equipment is connected properly to the Remote port.
    d) **Self-test Indicator—**Lights when UniSite-*xpi* is running a self-test.
10. **Package Specific Module (PSM)**—A module that supports a specific package style.
11. **Function Specific Module (FSM)**—A module that supports a special type of programming operation (e.g. set or gang programming).

The Back Panel

The back panel features of UniSite-*xpi* are shown in Figure 1-3.

Figure 1-3
Back Panel Features



12. **Parallel Port Connector**—Connects UniSite-*xpi* to PCs.
13. **Remote Port DTE/DCE Selector**—Configures the Remote port as Data Terminal Equipment or Data Communication Equipment.
14. **Power Switch**—Applies ac power to UniSite-*xpi*.
15. **Line Fuse**—Houses the line fuse.
16. **AC Receptacle**—Connects UniSite-*xpi* to ac power.

Note: On older units, the positions of the power switch and ac receptacle are reversed. Older units also have a line voltage selector.

17. **Remote Port**—Connects UniSite-*xpi* to PCs, workstations, terminals, etc.

18. **Terminal Port Connector**—Connects UniSite-*xpi* to PCs, workstations, terminals, file servers, etc.
19. **Terminal Port DTE/DCE Selector**—Configures the Terminal port as Data Terminal Equipment or Data Communication Equipment.

Disks

Programmer Disks

The Boot Files disk and the Algorithm / System disks are inserted in the programmer's disk drive. After setting up the programmer, make backup copies of the Boot disk and Algorithm / System disks using disks formatted on the programmer. Use the copies during daily operation.

Algorithm/System Disks

Algorithm / System disks contain the system software and programming algorithms for the currently supported devices. The UniSite-*xpi* is shipped with the algorithm files already stored on the Mass Storage Module (MSM).

(Legacy) For legacy UniSite, your algorithm files are not stored on the Mass Storage Module, and one of the Algorithm / System disks must be installed in the programmer disk drive each time you select a device. You will be prompted for the specific disk to insert.

Boot Files Disk

The Boot Files disk contains system software and configuration files used to boot up the programmer.

PC Disk

The Utility Disk is inserted in the floppy disk drive on your PC.

Utility Disk

The Utility Disk contains HiTerm software for your PC. For more information, see the **manual.doc** file on the Utility Disk.

Specifications

Functional	**RAM**	9 MB standard
	Disk Format	Double-sided, double-density 3.5-inch disk with 135 tracks per inch. 720 KB formatted
	Controller	Motorola 68000 16-bit microprocessor
	Terminal Support	Interfaces with ANSI 3.64 compatible terminals, IBM PCs and compatibles running a terminal emulator program, and many popular ASCII terminals
	Parallel Port Communication	IEEE 1284A Type C
	Serial Port Communication	RS-232C, with data transfer rate 110 to 19.2 kilobaud (up to 115.2 kilobaud using HiTerm)
Power Requirements	**Operating Voltages**	100 to 120V ac ±10% or 220-240V ac ±10%
	Frequency Range	48 to 63 Hz
	Power Consumption	500 VA maximum
	Input Current	5A maximum
	Fuse Ratings	250V / 6A / F (Fast Blow)
Physical and Environmental	**Dimensions**	18.06h x 43.48w x 36.20d cm 7.11h x 17.12w x 14.25d in.
	Weight	9.1 kg (20 lbs.)
	Shipping Weight	11.4 kg (25 lbs.)
	Temperature	Operating: +10° to 40°C (+50° to 105°F) Storage: +4° to +50°C (+40° to +122°F) Transportation: -40° to +55°C (-40° to +130°F)
	Relative Humidity	Operating: 20 to 80% noncondensing Storage: 10 to 90% noncondensing
	Altitude	Operating: To 5,000 meters Storage: To 15,000 meters

Safety

The UniSite-*xpi* and all legacy programmers are certified by UL to comply with the following safety standards:

Figure 1-4
UL Symbol

Underwriters Laboratories —
UL 60950 Third Edition and/or
EN 60950 Third Edition

Electrostatic Discharge (ESD)

IEC 801-2 (± 8 kV)

Electromagnetic Emissions

UniSite-*xpi* (part number 991-0014-058 and earlier versions) is certified to meet VDE 0871 Limit B.

Certificate of RFI/EMI Compliance

Data I/O certifies that the UniSite-*xpi* and all legacy programmers comply with the Radio Frequency Interference (RFI) and Electromagnetic Interference (EMI) requirements of EN55022 Class A and EN50082-1 as called out in 89/336/EEC, the EMC Directive for the European Community.

Figure 1-5
CE Symbol

EC Conformity Mark

WARNING: This equipment is a class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

Performance Verification

UniSite-*xpi* verifies internal voltages every time it is powered up and every time a complete self-test is run. The voltage verification is performed by software and is compared to a laser-trimmed voltage reference.

Data I/O recommends that you cycle power AND run a complete self-test cycle at least every three months.

To ensure that your UniSite-*xpi* continues to meet product performance specifications, Data I/O recommends that your programmer be returned to an authorized Data I/O Service Center every twelve months for a complete performance evaluation.

Options

The items listed below complement the UniSite-*xpi* Universal Programmer. For more information, or to order an item below, contact Data I/O Customer Support as listed in the Preface.

Keep Current Subscription Service

Data I/O offers a one-year subscription to keep your programmer and documentation up-to-date with the latest features and device support. This subscription also incorporates manufacturer-recommended changes to existing device support to maintain optimum yields, throughput, and long-term reliability.

In addition, you receive immediate access to new and updated programming algorithms via our Keep Current Service —before the algorithms are available in the update kit. Keep Current algorithms are available on the Data I/O Home Page on the World Wide Web at **http://www.dataio.com**.

For more information, see the Keep Current documentation located behind the Keep Current tab, or contact Data I/O Customer Support.

TaskLink for Windows

PC/Programmer Interface Software designed for use with UniSite-*xpi*. TaskLink for Windows runs on an IBM PC (or compatible) and allows you to control UniSite-*xpi* from a personal computer for streamlined and enhanced programming operations. TaskLink for Windows features automatic programming file configuration, full-screen editing, and error-logging. TaskLink for Windows also features a new windowed interface, extensive online context-sensitive help, and full mouse support.

PinSite Module

Expands support for devices up to 84 pins on Standard Bases and for devices with more than 84 pins on PPI Base and Adapters.

Standard Bases (for PinSite Module)

Works with PinSite Module to support several different package types, including PGA, PLCC, LCC, and SOIC. Uses MatchBook™ Device Carriers to improve throughput and reduce damage that may be caused by traditional sockets for surface-mount devices.

PPI Base and Adapters (for PinSite Module)

Works with PinSite Module to support devices with more than 84 pins and in additional package styles, such as SDIP, QFP, and memory cards.

Maintenance Manual

Contains information on the following: disassembly/reassembly, circuit description, maintenance, and troubleshooting.

Cable Set

Includes an additional parallel cable, serial (RS-232C) cable and gender changer.

Carrying Case

Protects UniSite-*xpi* from damage during travel. The custom-designed, soft-sided carrying case holds UniSite-*xpi*, the Algorithm disk, the System disk, and this manual. The Carrying Case is ideal if you will be using UniSite-*xpi* in the field.

RAM Upgrades (Legacy)

Expand system RAM on legacy UniSites. Contact Data I/O Customer Support for more information on RAM upgrades.

Site48HS (Legacy)
Replaces Site40 and Site48 on legacy UniSites and increases support for DIP devices to 48 pins.

Pin Driver (Legacy)
Provides additional support for devices with higher pin counts. Each pin driver board provides additional support for 4 pins. For example, if your UniSite has 10 pin drivers, you can program devices with up to 40 pins. If you want to program 48-pin devices, you will need to add two pin driver boards to your UniSite.

How Many Pin Driver Boards Do You Need (Legacy)?

The following table shows the minimum number of pin driver boards needed to program a device with a given number of pins. Each pin driver board can drive four device pins. Depending on how the pins on the device are mapped out, the number of pin driver boards required is sometimes greater than the number of device pins divided by four.

Note: The information in the following table applies to legacy UniSites running system software version 3.60 and later.

Pin driver boards are easily installed and can be ordered by contacting your nearest Data I/O representative as listed in the Preface.

Number of Device Pins	Site 40 or HI-QA-DIP	Site 48HS	SetSite	ChipSite or HI-QA-SMT	PLCC, LCC, SOIC (PinSite)	PGA (PinSite)
8	7	7	10	7	7	n/a
14	7	7	10	7	7	n/a
16	7	7	10	7	7	n/a
18	7	7	10	7	7	n/a
20	7	7	10	7	7	n/a
24	7	7	10	7	7	n/a
28	7	7	10	7	7	17*
32	9	8	10	9	8	n/a
40	10	10	10	n/a	10	n/a
44	n/a	11	n/a	12	11	17
48	n/a	12	n/a	n/a	12	n/a
52	n/a	n/a	n/a	15	13	n/a
68	n/a	n/a	n/a	17	17	17
84	n/a	n/a	n/a	n/a	17	17
>84	n/a	n/a	n/a	n/a	n/a	17

* *Requires the PGA Base, PSBASE-0402. Contact Data I/O Customer Support as listed in the Preface for more information.*

When you program devices with a PPI Base, we recommend that you install 17 boards, although the requirements vary among adapters. For the exact number of boards you need with a specific PPI Base, call Data I/O Customer Support as listed in the Preface.

2. Setup and Installation

This chapter describes how to set up UniSite-*xpi* and get it working with your equipment. Before you read this chapter, make sure you have read Chapter 1, “Introduction.”

This chapter guides you through configuring the hardware and installing the system software. The process is divided as follows:

1. Install a Module 2-3
2. Check the Line Voltage (on older units) 2-5
3. Choose Your Configuration 2-5
4. Connect Your Equipment... 2-6
5. Install the Base 2-15
6. Power Up the Programmer.................................... 2-16
7. The Power-on Screen 2-22
8. Insert the Algorithm Disk(s).................................. 2-24

The following steps are optional, depending on your programmer configuration.

9. Update System Software 2-24
10. Set Up High Speed Download 2-26
11. Install and Update the MSM 2-29

When you have finished with steps 1 through 8 and any optional steps 9 through 11, do the following:

- Refer to step 12, “Install Devices,” on page 2-30 for instructions on how to install devices.
- Refer to the “What to Do Next Time” section on page 2-41 for instructions on what to do next time you powerup your programmer.

The following maintenance information is also included:

- Changing the Line Voltage (on older units)......... 2-39
- Replacing the Line Fuse.. 2-40

Before You Begin

Before you begin the setup and installation, make sure you have read and understand the terms of the Software License Agreement, which is printed on the outside of the envelope containing the Algorithm / System disk(s).

Licensing Agreement

IMPORTANT: If you do not agree with the terms of the licensing agreement, do not open the software package. Opening the package indicates that you have accepted the terms of the agreement.

Definitions

Some important terms are described below.

Algorithm Disks

Algorithm / System disks contain the system software and programming algorithms for the currently supported devices.

UniSite-*xpi* is shipped with the algorithm files stored on the Mass Storage Module (MSM). For updates, you will need to install new algorithm sets onto the MSM. See “Update System Software” on 2-24 for details.

Legacy

For legacy UniSite, algorithm disks are inserted into the **programmer’s** disk drive and contain the system software and all the programming algorithms for the devices currently supported by the programmer. Your programmer package includes at least one Algorithm disk (the number of disks your programmer package contains can be found on the disk labels). Unless you have the RAM Device Selection feature enabled or have your algorithm files stored on a Mass Storage Module (MSM), one of the Algorithm disks must be installed each time you select a device (the programmer will prompt you for which disk to insert).

Boot Files Disk

The Boot Files disk is inserted into the **programmer’s** disk drive and contains the system software and the configuration files used when you boot the programmer, which you must do when no MSM is installed.

Host

A workstation, such as a Sun, Digital, or HP system. You can use the workstation to control the programmer and for remote storage of data files.

PC

A DOS- or Windows-based personal computer, such as an IBM PC or compatible, or other microcomputer. The PC can be used to control the programmer with either TaskLink™ for Windows®, TaskLink™ for DOS, or terminal emulation software. The PC can be used for remote storage of data files.

Programmer

A generic name for UniSite-*xpi* and other device programmers.

Terminal

A stand-alone dumb terminal, such as the DEC VT-100, Qume VT-101, and the Wyse WY-30 / 40 / 70 family of terminals.

Utility Disk

The Utility disk is inserted into a **DOS-based PC’s** disk drive and contains HiTerm terminal emulation software for your PC. For more information, see the **manual.doc** file, which is located on the Utility disk.

1. Install a Module

This section tells you how to insert a module into UniSite-*xpi*. Although this might seem like a simple procedure, we strongly recommend you read this section.

About Modules

A module serves as the interface between the device and the programmer. The modules available at the time of the printing of this manual are listed in the “Options” section of Chapter 1. For a current list of all available modules, or for more information, contact Data I/O Customer Support as listed in the Preface.

For more information on using a specific module, see Appendix A, “Using Modules.”

Removing the Blank Module Cover

When you receive UniSite-*xpi*, a blank module cover might be installed in place of a large module. To remove the blank module cover, snap it off.

If you choose, you can leave the blank module cover in place until you want to install a large module, such as the PinSite Module.

Installing a Module

UniSite-*xpi* is designed to accept two modules: a small one on the left and a large one on the right. Each module allows you to support a specific package style or support special types of programming operations such as set or gang programming.

Follow the instructions below to install a small module and a large module.

Installing a Small Module

To install a small module, such as Site48HS, follow the steps below:

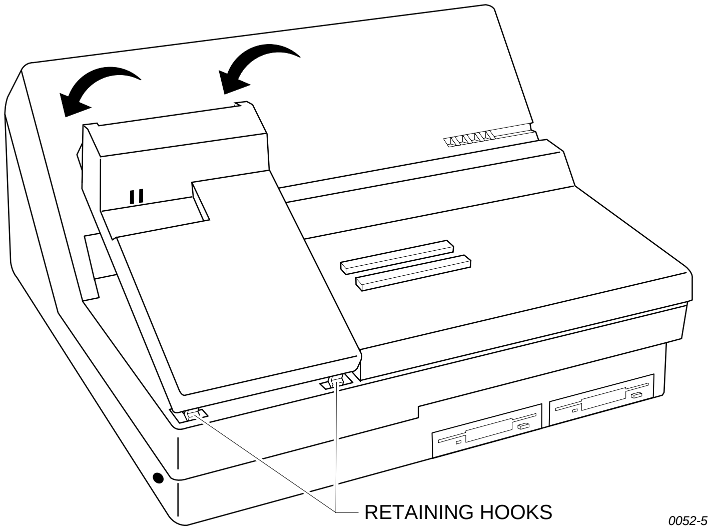
CAUTION: Remove any devices in a module before you install the module in UniSite-xpi.

1. Insert the two retaining hooks on the bottom of the small module into the two slots on the top left side of UniSite-*xpi*. Figure 2-1 shows the location of the retaining hooks on the module and also shows the location of the slots for the small module on UniSite-*xpi*.
2. Carefully lower the back of the small module until the module connector touches its mating connector on UniSite-*xpi*.
3. To ensure complete contact, firmly press down on the rear of the small module.

   You do not need to use excessive force when pressing on the module.

CAUTION: You can damage UniSite-xpi by exerting too much force on the module.

Figure 2-1
Installing a Small Module



Installing a Large Module

To install a large module, follow the steps below:

1. Insert the two retaining hooks on the bottom of the large module into the two slots on the top right side of UniSite-*xpi*. Figure 2-1 shows a large module already inserted into UniSite-*xpi*.
2. Carefully lower the back of the large module until the module connector touches its mating connector on UniSite-*xpi*.
3. To ensure complete contact, firmly press down on the rear of the large module.

   You do not need to use excessive force when pressing on the module.

CAUTION: You can damage UniSite-xpi by exerting too much force on the module.

Removing a Module

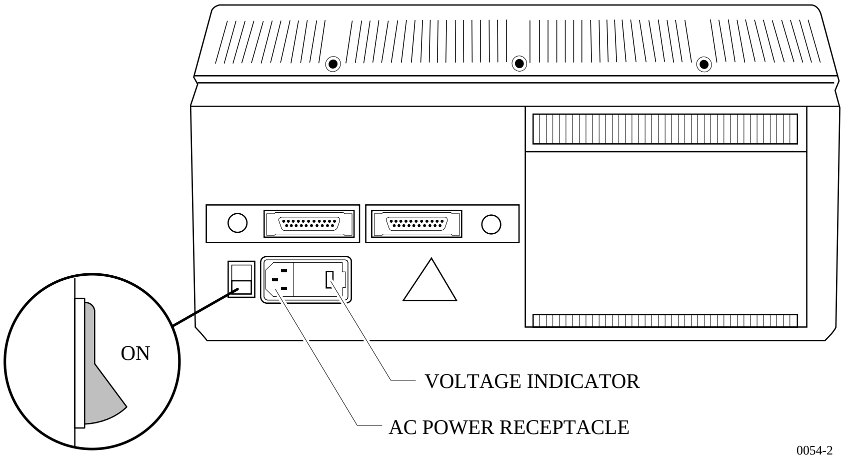
To remove a large module or a small module, follow the steps below:

1. Make sure you are not performing any device operations, such as loading, programming, or verifying.
2. Make sure the Active LED on the module is not lit.
3. Remove any devices in the module.
4. Lift the rear of the module until the retaining hooks on the front of the module are disengaged from the slots on the top of UniSite-*xpi*.
5. Set the module aside.

2. Check the Line Voltage (on older units)

(Legacy) On older units, verify that the line voltage is correct by checking the line voltage indicator on the rear panel. Figure 2-2 shows the location of the voltage indicator.

Figure 2-2
UniSite's Rear Panel



CAUTION: Damage to the equipment may occur if the instrument is operated with the wrong voltage.

If the line voltage is correct, continue with the next section.

If the line voltage is not correct, skip to the section titled "Changing the Line Voltage," which is located at the end of this chapter. Continue with the next section after you have changed the line voltage.

3. Choose Your Configuration

Hardware Choices

Review the equipment you have available and then decide which of the following hardware configurations you will use to control UniSite-*xpi*.

- Connecting to a PC — Support for serial port and parallel port operations.
- Connecting to a Host — Support for serial port operations only.
- Connecting to a Terminal — Support for serial port operations only.

Software Choices

If you connect to a PC you have three software choices:

- TaskLink for Windows — **Support for both serial and parallel port operations.** TaskLink for Windows is shipped with UniSite-*xpi*.

- TaskLink for DOS — **Support for serial port operations only.** TaskLink for DOS can be downloaded free from the Data I/O Web site.
- HiTerm (terminal emulation software) — **Support for serial port operations only.** HiTerm is available on the Utility Disk shipped with the UniSite-*xpi*, on the TaskLink for Windows CD, or it can be downloaded free from the Data I/O Web site.

Once you have decided the hardware configuration you will use, skip to that section. For example, if you decide to connect to a terminal, skip to the section titled “Connecting to a Terminal.”

4. Connect Your Equipment

This section is divided into three subsections:

- Connecting to a PC
- Connecting to a Host
- Connecting to a Terminal

Skip to the section that applies to your configuration.

Connecting to a PC

Connecting UniSite-*xpi* to a PC allows you to take advantage of all UniSite-*xpi*'s capabilities. For this configuration, you can choose to use either TaskLink for Windows or TaskLink for DOS to control UniSite-*xpi*. You must also use terminal emulation software, which allows UniSite-*xpi* and the PC to communicate.

What You Need

To connect the programmer to a PC, you need the following:

- An unused parallel port on the PC.
- An unused RS-232C serial port on the PC. (Serial ports on a PC are usually labeled COM1 or COM2.)
- TaskLink for Windows or TaskLink for DOS software. TaskLink for Windows is our recommended interface solution. It allows you to download files to the programmer 600% faster than other interfaces.
- Terminal emulation software (supplied with the programmer) to allow the programmer and the PC to communicate through the serial port connection. When using terminal emulation interface from within TaskLink, we recommend that you use HiTerm™.
- IEEE 1284C parallel port cable.
- A 25-pin serial cable. To build your own serial cable, see Appendix B, “Making Your Own Cable.”

Making the Connections

To connect UniSite-*xpi* to a PC, follow the steps below.

1. Connect one end of the parallel port cable to the **Parallel** port on the programmer back panel and then connect the other end to the **Parallel** port on the PC.

2. Connect one end of the serial cable to the **Remote** serial port on the programmer back panel, and then connect the other end to the **COM1** or **COM2** serial port on the PC.

Note: While PC serial ports are usually labeled COM1, COM2, etc., some PCs may label serial ports differently. Consult the documentation that came with the PC for more information.

3. Plug the power cord into the programmer back panel and into the wall socket.
4. Install either TaskLink for Windows or TaskLink for DOS.

Note: Make sure nothing is connected to the programmer's Terminal port.

Installing TaskLink for Windows (serial and parallel modes)

1. Insert the TaskLink for Windows disk into the disk drive of your PC.

Note: If you have previously installed TaskLink for Windows, this new installation remembers your old settings and uses them as the defaults for your new installation. There is no need to un-install. New installations, however, do not overwrite existing Task or Configuration files.

2. For a first installation, open **Setup.exe.**
3. Follow the on screen prompts, which will include checking the box for **Unisite Family Programmers.**
4. Select the **Program** folder.
5. Start TaskLink by clicking on the newly created icon.
6. Select the programmer that you want to work in and set the settings.
7. Configure the PC port.
8. Configure the programmer.
9. Verify connection and resolve issues.
10. Update device list.

Note: See TaskLink online help for step-by-step instructions.

Installing TaskLink for DOS (serial mode only)

1. Insert the TaskLink for DOS disk into the disk drive of your PC.
2. Type ***drive*****:install** (for instance, **a:install**) to begin the installation program. If you accept the default settings, TaskLink is installed in a **c:\tl** directory.
3. Start TaskLink. From the DOS prompt, type **cd tl** to change to the TaskLink directory.
4. Type **tl a** to open TaskLink in Administrative mode. In Windows, you can create program items or shortcuts to run TaskLink. The command to run TaskLink in Administrative mode is **tl.exe a**.

*Note: A DOS mouse driver must be installed if you plan to control TaskLink with a mouse. To access TaskLink commands using the keyboard, press **Alt**, then press the highlighted letter. Use the arrow and Tab keys to move around.*

5. Select programmer and set up options.
6. From TaskLink's main screen, select **Options (Alt+O)**, press **T,** select **UniSite-*xpi*** then press **OK**.
7. Press **H** (Handler Type), select **No handler used**, then press **OK**.
8. Press **P** (Programmer Port). Select **COM1** (or COM2), **9600**, **None**, **8**, and **1**. Select **None** for Host Port and Port options. Press **OK**
9. After the Self Test LED goes out, from TaskLink's main screen select **Utilities**, select **VT100 on Programmer Port (Alt+P)**, then press **SHIFT+Z+ENTER**.
10. Press **Ctrl+R**. Select **N**, then press **Enter** at the `Do you want to select new terminal type?` prompt. A screen with a T-bar menu appears, indicating that the PC is in Terminal mode.
11. Press **M** (More Commands), **C** (Configure System), **E** (Edit), then **C** (Communication).
12. Select **User Menu Port**. Press the **space bar** to change the T to an **R**, then press **Enter**.
13. Move the cable from Terminal on the programmer's back panel to **Remote.**
14. Press **F2**. If the cursor moves back to **Communication** on the left side of the T-bar menu, the operation was successful.
15. Press **I** (Interface), then **Y** to change the Power on CRC Mode to **Yes**.
16. Press **F2**, **F2**, **S**, then **Enter** to save the parameters.
17. Press **F1**, **M**, then **R** to return the programmer to Remote mode.
18. Press **Alt+F1** to return to TaskLink, then press **Ctrl+F1** to confirm that the programmer is communicating with the PC.
19. From the TaskLink main menu, select **Utilities**, then select **Device List Update** to update the TaskLink algorithm database so it matches the current version of the programmer algorithms.

HiTerm (Terminal Emulation Software — serial mode only)

Written especially for UniSite, HiTerm terminal emulation software is a VT-100 terminal emulator that is required to perform several specialty operations. This interface also allows you to download files to UniSite-*xpi* at up to 115.2K baud on the serial communication port. HiTerm is included on the TaskLink for Windows CD and is installed when TaskLink for Windows is installed on your PC. HiTerm is also provided on the Utility disk.

(Legacy) Installing HiTerm

You can run HiTerm from a floppy disk or hard disk. These instructions tell you how to install HiTerm on a hard disk. If you want to run HiTerm from a floppy disk, consult the *HiTerm User Manual*.

These instructions assume you have a working knowledge of DOS. If you have questions, consult your DOS manual or the HiTerm documentation.

As mentioned above, terminal emulation software is required to control and communicate with UniSite in several specialty modes of operation when working from the PC.

Follow the steps below to install HiTerm on your hard disk drive.

1. Copy all the files from the HiTerm disk to a directory on the PC's hard disk, such as **C:\hiterm**.
2. Make sure the PATH statement in the **autoexec.bat** file includes the directory where the HiTerm files reside.
3. Edit the **prg9600.cfg** file so it reflects the setup of your PC. This configuration file, specifying the operating mode, communication parameters, and PC type, is read when HiTerm is invoked.

   If HiTerm cannot read the configuration file, HiTerm will use the following default settings:

   Mode: Programmer
   Baud rate: 9600
   Parity: None
   Data bits: 8
   Stop bits: 1
   COM port: 1
   PC type: Auto detect

Note: The PC type parameter applies only to HiTerm version 3.10 and later. If you received an earlier version of HiTerm set the configuration file as described above but do not include the PC type parameter.

   When you edit the **prg9600.cfg** configuration file, be sure each line conforms to the specifications below.

   - **First line—mode**. Enter either General (G) or Programmer (P). Specify Programmer mode for use with UniSite. Only the first character of the line is significant.
   - **Second line—baud rate**. The complete number is required (for example, 9600, not 96). See the *HiTerm User Manual* for a list of baud rates supported by HiTerm.
   - **Third line—parity**. Specify None, Odd, or Even (N,O,E). Only the first character of the line is significant.
   - **Fourth line—data bits**. Specify 7 or 8.
   - **Fifth line—stop bits**. Specify 1 or 2.
   - **Sixth line—COM port**. Specify 1 or 2.
   - **Seventh line—PC Type**. Specify IBM, NEC, or Autodetect (I,N,A). Select **I** for IBM-compatible PCs, **N** for NEC's PC-9800 family of PCs, **I** for IBM-compatible PCs, or **A** for Autodetect if unsure.

4. Edit the **program.bat** file to reflect the location of the configuration files. Edit the two lines that invoke HiTerm so they point to the drive and subdirectory containing the HiTerm files. An example of the **program.bat** file modified to reflect HiTerm's installation in the **c:\util\hiterm** directory is shown below.

Note: In the following example, bold print indicates items you should modify if your configuration is different.

```
echo off
Rem: HITERM will use the configuration filename
Rem: from command line if present.
If not (%1) == () HITERM c:\util\hiterm\%1

Rem: HITERM will use PRG9600.CFG if no
Rem: configuration file is specified.
If (%1) == () HITERM c:\util\hiterm\prg9600.cfg
```

5. Installation of HiTerm is now complete. Reboot your PC.

The Next Step

UniSite-*xpi* is now connected to your PC. Skip to the section titled “Install the Base” to continue with the setup and installation of UniSite-*xpi*.

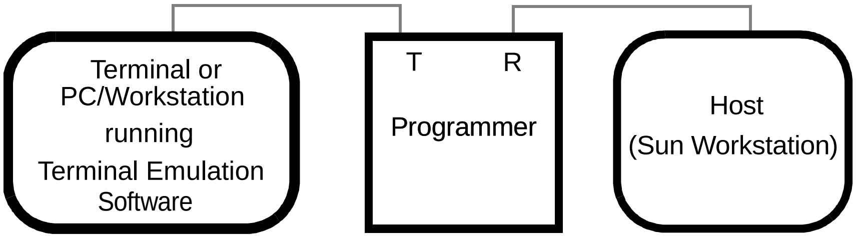
Connecting to a Host

Connecting the programmer to a host allows you to use the host for remote file storage.

Transparent Mode

Transparent mode, shown in Figure 2-3, is a feature of UniSite-*xpi* that allows the programmer to be inline between your host computer and a terminal. This eliminates the need for a switch box or a second link to the host and enables you to communicate with the host and download directly from the host to the programmer. The host could be a networked file server, such as a VAX or a Sun. When set up properly, the terminal connected to the programmer can control both the programmer and the remote host.

Figure 2-3
Transparent Mode


Terminal or PC/Workstation running Terminal Emulation Software
T
R
Programmer
Host (Sun Workstation)

In Transparent mode, the programmer passes all characters through its Terminal and Remote ports as if it weren’t there. The two serial ports on the programmer can even operate at different baud rates. While operating the programmer from the terminal, press **Esc, Ctrl** + **T** to toggle the programmer between terminal mode and transparent mode. The programmer remains in transparent mode until it receives another **Esc, Ctrl** + **T** command, at which time it switches back to terminal mode.

What You Need

To connect the programmer to a host, you need the following:

- An unused RS-232C serial port on the host.
- A 25-pin serial cable. If you need more information about which type of cable will work with the programmer, see the section later in this chapter titled “More About Cables.”

Making the Connections

To connect the programmer to a host, follow the steps below.

1. Connect one end of the RS-232C serial cable to the serial port connector on the host.

   On Sun workstations the serial ports are usually labeled Serial Port A, Serial Port B, etc. Other brands of workstations may label serial ports differently. Consult the documentation that came with the workstation for more information.

CAUTION: To minimize electromagnetic interference, use only properly shielded and terminated cable.

2. Connect the other end of the serial cable to the Remote port on the back of the programmer.

Checking the Communication Parameters

With the programmer connected to the host, adjust the communication parameters of the serial port connected to the programmer as follows:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit
- Full duplex
- CTS / DTR handshaking

Note: CTS/DTR (Hardware Handshake) is enabled as the default; however, if those signals aren't connected, the programmer will sense this and still communicate properly using XON/XOFF (Software Handshake). The programmer always uses XON/XOFF regardless of whether CTS/DTR handshake is enabled.

Later, once you have established communication and the programmer is operating, you can change the communication parameters to suit your needs. Consult the operator's manual supplied with the host if you need to change the host's communication parameters.

The Next Step

The programmer is now connected to your host. Skip to the section titled “Install the Base” to continue with the setup and installation of the programmer.

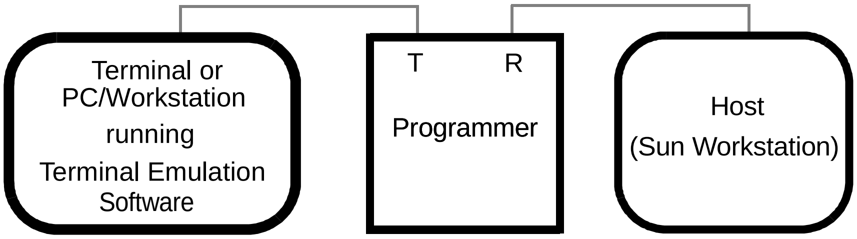
Connecting to a Terminal

Connecting the programmer to a terminal is the simplest configuration. By connecting the programmer to a terminal, you can take advantage of all the programmer's serial port capabilities.

Transparent Mode

Transparent mode, shown in Figure 2-4, is a feature of the programmer that allows the programmer to be inline between your terminal and host computer. This eliminates the need for a switch box or a second link to the host, and enables you to download directly from the host to the programmer. The host could be a networked file server such as a VAX or a Sun. When set up properly, the terminal connected to the programmer can control both the programmer and the remote host.

Figure 2-4
Transparent Mode



In transparent mode, the programmer passes all characters through its Terminal and Remote ports as if it weren't there. The two serial ports on the programmer can even operate at different baud rates. While operating the programmer from the terminal, press **ESC, CTRL** + **T** to toggle the programmer between terminal mode and transparent mode. The programmer remains in transparent mode until it receives another **ESC, CTRL** + **T** command, at which time it switches back to terminal mode.

What You Need

To connect the programmer to a terminal, you need the following:

- An unused RS-232C serial port on the terminal.
- 25-pin serial cable. If you need more information about which type of cable will work with the programmer, or if you need to build your own cable, see the section later in this chapter titled "More About Cables."
- If you are going to operate the programmer in transparent mode, you will also need the following:
- An unused RS-232C serial port on the host. The host can be a VAX, a Sun, etc.
- 25-pin serial cable.

Approved Terminals

Before you connect your terminal to the programmer, make sure your terminal is compatible with the programmer. If your terminal type is listed below, then your terminal is compatible with the programmer.

- ANSI 3.64 compatible terminals
- DEC VT-100 compatible terminals
- Qume QVT-101 compatible terminals
- TELEVIDEO TVI-910 compatible terminals
- Wyse WY-30 compatible terminals

If your terminal is not included in the above list, check the documentation that came with your terminal to see if the terminal can emulate one of the terminal types listed above. If your terminal has programmable function keys, the following table lists the expected codes for the four function keys:

VT-100 Key	Expected Code	Wyse-30 Key	Expected Code
PF1	Esc O P	F1	SOH @ CR
PF2	Esc O Q	F2	SOH A CR
PF3	Esc O R	F3	SOH B CR
PF4	Esc O S	F4	SOH C CR

Making the Connections

To connect the programmer to a terminal, follow the steps below.

1. Connect one end of an RS-232C serial cable to the serial port connector on the back of the terminal.

   On some terminals the serial port might be labeled Modem; on others the serial port might be labeled EIA. Consult the documentation that came with the terminal for more information.

CAUTION: To minimize electromagnetic interference, use only properly shielded and terminated cables.

2. Connect the other end of the cable to the Terminal port on the back of the programmer.

   If you will not be using transparent mode, skip to the section titled “Checking the Communication Parameters.” Follow the steps below if you will be using transparent mode.

3. Connect one end of an RS-232C serial cable to the serial port connector on the host. If the host is not available locally, i.e., the host is a networked VAX, connect the serial cable to the appropriate serial port.

CAUTION: To minimize electromagnetic interference, use only properly shielded and terminated cables.

4. Connect the other end of the cable to the Remote port on the back of the programmer.

Checking the Communication Parameters

Adjust the communication parameters of the equipment connected to the programmer as follows:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit
- Full duplex
- CTS / DTR handshaking

Note: CTS/DTR (Hardware Handshake) is enabled as the default; however, if those signals aren’t connected, the programmer will sense this and still communicate properly using XON/XOFF (Software Handshake). The programmer always uses XON/XOFF regardless of whether CTS/DTR handshake is enabled.

Once you have established communication and the programmer is operating, you can change the communication parameters to suit your needs. Consult the manual supplied with the terminal if you need to change the terminal’s communication parameters.

If you are using transparent mode, set the communication parameters on the serial port on the host as described above. After you have powered up the programmer, you can change the parameters on the programmer's Remote port to match the communication parameters of the host.

The programmer is now connected to your terminal. Skip to the section titled "Install the Base."

More About Cables

You will need one 25-pin RS-232C serial cable for each piece of equipment you will connect to the programmer.

Normally, when you connect equipment to a programmer, you must match Data Terminal Equipment (DTE) to Data Communications Equipment (DCE).

The programmer is compatible with both types of equipment and automatically configures the Terminal and Remote ports to be compatible with the equipment connected to it.

For more information, see Appendix B, "Making Your Own Cable."

5. Install the Base

A Base is installed in a module that requires a Base (not all modules require a Base). The Base serves as the interface between the device and the programmer. Some Bases (such as the PLCC Base) are designed to be used with MatchBook device carriers. The MatchBook device carrier holds a device in place on the Base (MatchBooks are described on page 2-31). Some bases (such as the PPI Base) use device adapters.

Follow the instructions in this section if the module installed on UniSite-*xpi* requires a base. For additional information, refer to Appendix A, "Using Modules," and to "Install Devices" on page 2-30.

Installing a Base

Follow the procedure below to install a Base in the programmer.

Note: You can install and remove a Base with the power on as long as you are not performing a device operation.

1. Position the programmer so that the front points toward you.
2. Examine the module opening for the base. It has a sliding handle on the front half and notches on the back half. Pull the sliding handle toward the front.

CAUTION: Poking a foreign object into the Base opening will damage the programmer.

3. Align the notches on the Base with the notches in the opening.
4. Insert the Base into the opening, making sure the guide pins in the Base opening line up with the holes in the Base.
5. Finally, squeeze the handles together, locking the Base in place. Do not use excessive force when compressing the handles.

CAUTION: You can damage the programmer by squeezing too hard on the handles.

Removing a Base

When removing a Base from the programmer, be sure to apply even pressure while moving the handles apart. If you exert uneven pressure on the handles, you could damage the sliding handle.

To remove a Base, follow the steps described below.

1. To remove the Base, separate the handles with your thumbs and fingers. Push back on the Base handle with both thumbs while pushing forward on the sliding handle with your forefingers.
2. Lift the Base up and out of the module. Store the Base in a safe place.

CAUTION: You must apply even pressure while moving the handles apart. If you exert uneven pressure on the handles, they may jam in the tracks. Apply an even force to realign the handles.

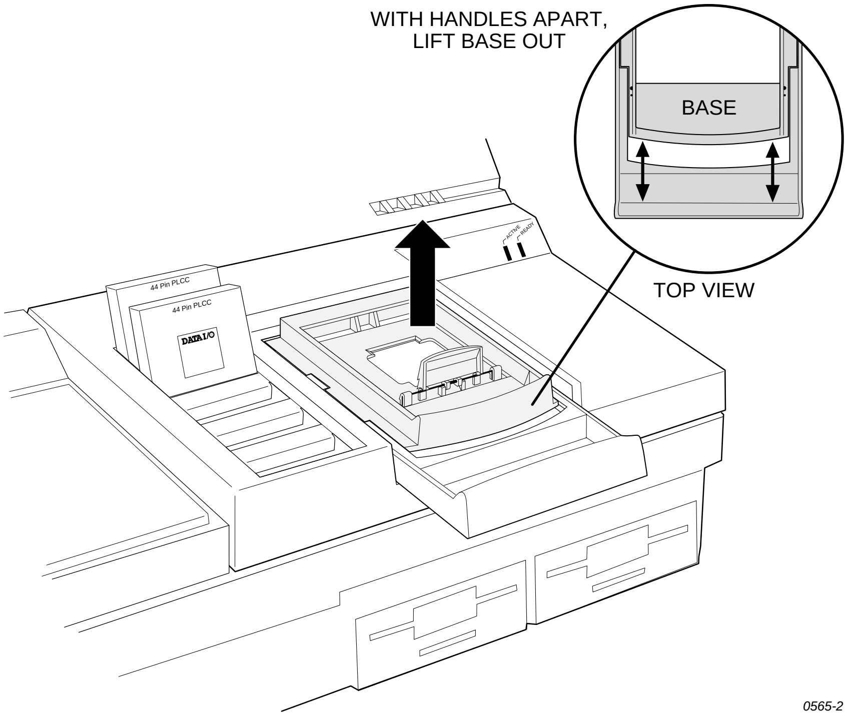


Figure 2-5
Removing a Base from the PinSite Module

6. Power Up the Programmer

To power up the programmer, follow the steps below.

1. Make sure the programmer is positioned so the fan on the bottom will not be obstructed.

2. If you will be using an antistatic wrist strap, put on the wrist strap and connect it to the programmer. The ground connector for the antistatic wrist strap is on the side panel of the programmer. Refer to Figure 1-2 for the location of the ground connector on the programmer.

WARNING: ***To help prevent electric shock, the antistatic wrist strap must contain a 1 MΩ (minimum) to 10 MΩ (maximum) isolating resistor.***

Connecting the Power Cord

Connect one end of the ac line cord to the ac receptacle on the rear panel of the programmer and the other end to a properly grounded ac outlet.

WARNING: ***To ensure proper grounding, and to avoid hazard of electrical shock, connect the programmer ONLY to a properly grounded ac outlet.***

The programmer contains a switching power supply that configures itself to operate on the proper voltage. The power supply accepts voltages ranging from 90 to 264 Vac and frequencies ranging from 48 to 63 Hz.

Make sure that a module is installed in the programmer. Also, make sure the device socket in the Base is empty.

CAUTION: ***Leaving a device in the socket during powerup will cause power-up self-test failures and could damage the device.***

Power up the PC, workstation, or terminal. If you will be controlling the programmer from a PC or workstation, make sure that the terminal emulation software (such as HiTerm) is running. If you will be controlling the programmer from a terminal, make sure that the terminal is in the proper emulation mode (such as VT-100 mode). See the “Approved Terminals” section earlier in this chapter for more information.

Powering Up

Power up the programmer. The power switch is located on the back panel. When you turn the power switch on, the Power LED should light. If it doesn’t, turn the programmer off, check the power connections, and turn the programmer on again.

Note: Do not remove the Boot Files disk while either the Self Test or disk drive LED is lit. If you remove the Boot Files disk during powerup, you will need to reboot the programmer.

Did the Programmer Pass Self-test?

The programmer always performs a self-test upon powerup. The four front panel LEDs illuminate in different patterns, depending on the results of the self-test. The programmer has completed powerup when the Self Test LED and disk drive LED are off. If the Self Test LED does go off, then go to the section titled “Are the Right LEDs Lit?”

If the powerup self-test detected anything wrong, the four LEDs illuminate in different patterns, telling you what the self-test found. In general, if one or more front panel indicators is blinking after the self-test, there may be a faulty circuit board in the programmer. Contact your nearest Data I/O Customer Support office for more information. The different combinations of blinking LEDs are explained below.

--------------------Indicator--------------------				
Power	**Terminal**	**Remote**	**Self-test**	**Description**
On	Off	Off	On	Self-test in progress.
Off	X	X	On	Bad power supply
On	Blinking	Blinking	Blinking	Power supply problem; check voltage selector.
On	Blinking	Off	On	Bad CPU or EPROM
On	Off	Blinking	On	Bad system RAM
On	Blinking	Blinking	On	Bad serial port DUART
On	On	Off	Off	Terminal port connected; self-test finished. No errors.
On	Off	On	Off	Remote port connected; self-test finished. No errors.
On	On	On	Off	Remote & terminal ports connected; self-test finished. No errors.

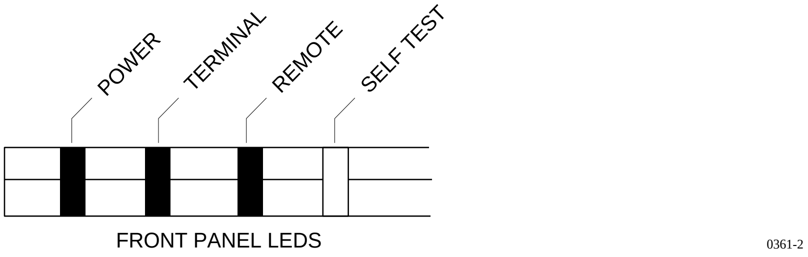
Note: X = don't care condition.

"Terminal port OK" means that the programmer has detected an RS-232 serial device connected to the Terminal port and that a valid connection has been established with that device.

Are the Right LEDs Lit?

If the programmer completed the self-test successfully, the Power LED will be lit. Also, if you have equipment connected to the Terminal and Remote ports, the corresponding LEDs should be lit. Figure 2-6 shows the front panel indicators.

Figure 2-6
Front Panel Indicators



If all the right LEDs are lit, go to the section titled "Is the Power-on Screen Displayed?"

If the Remote LED and / or Terminal LED should be lit and they are not, check the connections between the programmer and the connected equipment. If all connections appear correct, refer to "Configuring the Ports" below.

Configuring the Ports

Before communication between UniSite-*xpi* and a device connected to one of UniSite-*xpi*'s serial ports can be established, three requirements must be met:

- UniSite-*xpi*'s port must be configured using the DCE / DTE button so that it is compatible with the equipment it is connected to.
- UniSite-*xpi*'s serial I / O parameters must match those of the equipment it is connected to.
- UniSite-*xpi* must recognize the type of terminal that you are using. (Selecting the correct terminal type is described in the next section, "The Power-on Screen for Terminal Mode.")

The steps below tell you how to configure the Terminal port. Use the same procedure for configuring the Remote port.

1. After you have powered up UniSite-*xpi* and the Self-Test indicator has gone out, look at the Terminal LED on the front panel.
2. If the LED is lit, UniSite-*xpi*'s Terminal port is configured correctly. Skip to the section titled "Is the Power-on Screen Displayed?"
3. If the LED is not lit, press the button next to the Terminal port on the rear panel of UniSite-*xpi* once. (Remember, if you are configuring the Remote port, press the button next to the Remote port.) When the switch is pressed in, UniSite-*xpi* acts as Data Terminal Equipment (DTE); when in the out position UniSite-*xpi* acts as Data Communications Equipment (DCE).

   After you press the button, the LED should light. If it does light, skip to the section titled "Is the Power-On Screen Displayed?"

   If the LED does not light, continue with the section titled "Checking the Connections."

Checking the Connections

Sometimes problems are caused by unconnected cables. Turn the programmer off and check all of the following:

- **Power cords**—Are they all plugged into a live outlet and into the equipment?
- **Cables**—Is each cable between the programmer and a peripheral connected properly? Is each cable connected to the proper port?
- **Terminal**—Is the terminal plugged in and turned on? Are the display controls adjusted to allow viewing? Is the terminal an approved terminal type? (See the list of approved terminals for more information.) Are the communication parameters set correctly? Is the cable connected to the proper port?

- **PC/Host**—Is the PC / host plugged in and turned on? Is the terminal emulation software installed and configured properly? Is the terminal emulation software running properly? Is the TaskLink for Windows or TaskLink for DOS software installed and configured properly? (See “Software Choices” on page 2-5). Is the software running properly? Are the display controls on the monitor adjusted to allow viewing?
- **Disk**—Is the System disk inserted properly?
- **Module**—Is there a Module inserted into the programmer? Is it inserted properly?
- **Base**—Is there a Base inserted into the Module? Is it inserted properly? (A Base should be installed if the Module requires one.)
- **Devices**—Is the device socket(s) empty? (Sockets must be empty for the programmer to boot up properly.)

Now, after checking everything described above, reboot the programmer and go to the beginning of this section, “Power Up the Programmer.”

Rebooting the Programmer

If you need to reboot the programmer, you can do it in either one of the following two ways:

- Turn the programmer off, wait a few seconds, and then turn the programmer on again.
- Press **ESC, CTRL** + **W**.

The programmer always performs a powerup self-test whether you do a cold boot (cycling power) or a warm boot (pressing **ESC**, **CTRL** + **W**).

Is the Power-on Screen Displayed?

The Start-up screen is displayed after the programmer completes its powerup self-test successfully. The version and configuration information for the programmer along with the current and default terminal types are displayed.

Configuring the Programmer Parallel Port (TaskLink for Windows Only)

1. Connect the programmer to the parallel port on your PC using the parallel port cable supplied with your programmer.

Note: If your PC has been more than one parallel port, note which port is used.

2. Connect the programmer to the serial port on your PC using the serial port cable supplied with your programmer.

Note: Because some operations are not supported on the parallel port, it is necessary to connect the programmer and PC via their serial port as well. TaskLink uses the parallel port if ***Programmer Parallel Port*** *is enabled (see Step 4 or Step 7 below). If the operation is not supported on the parallel port, TaskLink switches to the serial port to complete the operation.*

3. Start TaskLink and select **Options** from the System menu.

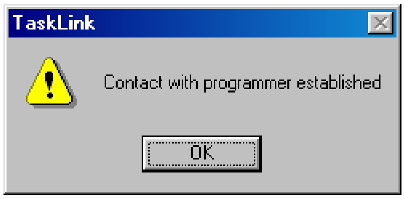
If running Windows 9x/ME:

4. On the Port Settings tab, enter a check to enable **Programmer Parallel Port.** From the drop-down list, select the parallel port that matches the port used on the PC.
5. Enter the correct Port Address. (Most computer systems have default settings of LPT1 and address 378.)
6. Go to Step 10.

If running Windows NT/2000:

7. On the Port Settings tab, enter a check to enable **Programmer Parallel Port.** Click **Configure**.
8. On the UniSystem Parallel Port Setup dialog, from the drop-down list, select the parallel port that matches the port used on the PC. Click **Configure**.
9. On the Port Settings tab, click **OK** to return to the TaskLink main screen.
10. On the TaskLink main screen, click the **Establish Contact** icon in the upper left corner.
11. If the programmer is properly connected via the parallel port, the message shown in Figure 2-7 is displayed.

Figure 2-7
TaskLink Connection Established Message



TaskLink

Contact with programmer established

If the connection establish message (shown in Figure 2-7) is displayed, you have successfully booted up the programmer. Go to the next step, “The Power-on Screen for Terminal Mode.”

Power-on Screen for Terminal Mode

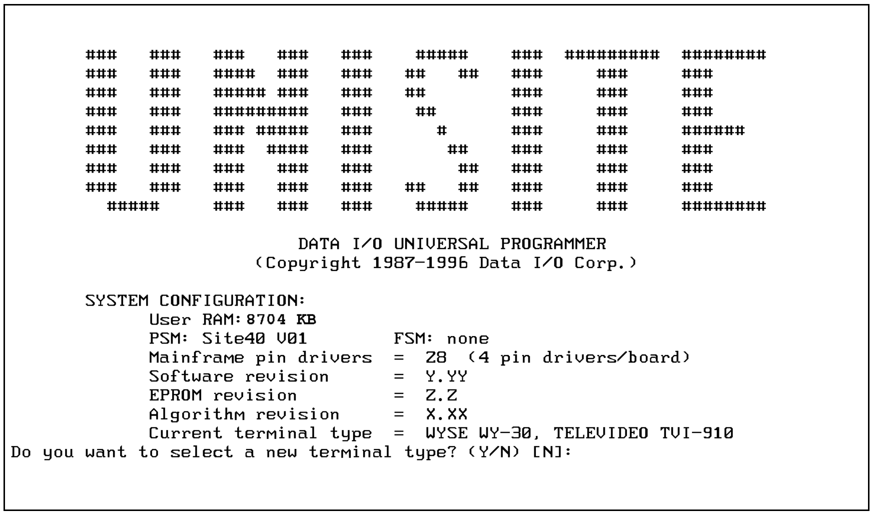
If the power-on screen (shown in Figure 2-8) is displayed, you have successfully booted up the programmer in Terminal mode. Go to the next step, “The Power-on Screen.”

If you do not see the Power-on screen, or if you see random characters, the programmer is not communicating properly with your controlling terminal/workstation. One possible cause of the random characters is that the baud rates of the programmer and the controlling equipment do not match. In this case, follow the procedure in the next section to execute the AutoBaud function to “sync up” the baud rates.

7. The Power-on Screen for Terminal Mode

The Power-on screen, shown in Figure 2-8, appears after the programmer completes its powerup self-test successfully. The Power-on screen lists the version and configuration information for the programmer and the current and default terminal types.

Figure 2-8
Typical Power On Screen



```
        ###   ###   ###   ###   ###    #####    ###  #########  ########
        ###   ###   ####  ###   ###   ##   ##   ###     ###     ###
        ###   ###   ##### ###   ###   ##        ###     ###     ###
        ###   ###   #########   ###    ##       ###     ###     ###
        ###   ###   ### #####   ###      #      ###     ###     ######
        ###   ###   ###  ####   ###       ##    ###     ###     ###
        ###   ###   ###   ###   ###        ##   ###     ###     ###
        ###   ###   ###   ###   ###   ##   ##   ###     ###     ###
          #####     ###   ###   ###    #####    ###     ###     ########

                             DATA I/O UNIVERSAL PROGRAMMER
                         (Copyright 1987-1996 Data I/O Corp.)

        SYSTEM CONFIGURATION:
              User RAM:8704 KB
              PSM: Site40 V01        FSM: none
              Mainframe pin drivers  =  28  (4 pin drivers/board)
              Software revision      =  Y.YY
              EPROM revision         =  Z.Z
              Algorithm revision     =  X.XX
              Current terminal type  =  WYSE WY-30, TELEVIDEO TVI-910
Do you want to select a new terminal type? (Y/N) [N]:
```

Note: The power-on screen your programmer displays might be slightly different from the screen shown above.

At this point, you are asked if you want to choose a new terminal type. If the current terminal type is correct, press **ENTER** and go to the section titled “Setting Up High Speed Download.”

To change the current terminal type and / or the default terminal type, refer to the appropriate section below, “Selecting a New Terminal Type” or “Changing the Default Terminal.” If you will be using the current configuration for a while, we suggest you change the default terminal type to match the terminal type you will be using to control the programmer. For information on changing the default terminal type, go to the section titled “Changing the Default Terminal.”

Selecting a New Terminal Type

The following steps describe how to change the current terminal type. The change in terminal type will remain in effect until you reboot the programmer or change the terminal type.

1. At the bottom of the Power-on screen, the programmer displays the following prompt:

   ```
   Do you want to select a new terminal type? (Y/N) [N]:
   ```

   Press **Y**, **ENTER** to select a new terminal type.

2. The programmer then displays the default and current terminal types and a list of the available terminal types. To select a new terminal type, enter the number corresponding to that terminal type and press **ENTER**.

Note: If the screen is blank or if random characters appear after you press Enter, the Terminal type configuration is incorrect. Switch the power off and then power up the programmer. Then return to the beginning of this section and select the proper terminal type.

If you do not see your terminal listed on the screen and do not know what terminal type(s) it can emulate, refer to the Compatible Terminals list shown previously in this chapter.

After you have changed the terminal type for this current session, the programmer responds with the following prompt:

```
Save terminal type as power on default? (Y/N) [N]
```

3. Press **N**, **ENTER** to go to the Main Menu without changing the default terminal type.

When the Main Menu appears, the programmer is ready for operation. Go to the section titled “Insert Algorithm Disk.”

Changing the Default Terminal

Follow the steps below to change the default terminal type.

1. At the bottom of the Power-on screen, the programmer displays the following prompt:

```
Do you want to select a new terminal type? (Y/N) [N]:
```

Press **Y**, **ENTER** to select a new terminal type.

2. The programmer then displays the default and current terminal types and a list of the available terminal types. To select a new terminal type, enter the number corresponding to that terminal type and press **ENTER**.

Note: If the screen is blank or if random characters appear after you press Enter, the Terminal type configuration is incorrect. Switch the power off and then power up the programmer. Then return to the beginning of this section and select the proper terminal type.

If you do not see your terminal listed on the screen and do not know what terminal type(s) it can emulate, refer to the Approved Terminals list shown previously in this chapter.

After you have changed the terminal type for this current session, the programmer will respond with the following prompt:

```
Save terminal type as power on default? (Y/N) [N]
```

3. Press **Y**, **ENTER** to change the default terminal type. After the programmer saves the currently selected terminal type as the power-on default, the Main Menu appears, indicating that the programmer is ready for operation. Go to the section titled “Insert Algorithm Disk.”

8. Insert Algorithm Disk

UniSite (Legacy)

If you do not have an MSM installed in your UniSite, you will have to access the algorithm files from the Algorithm disks. Once UniSite has finished booting (the self test is finished and the drive LED is off), insert the Algorithm disk that contains the programming algorithms for the devices you plan on using into drive B (the drive on the right). (If you are unsure which Algorithm disk to insert, insert the Algorithm disk that contains Algorithm Set 1.)

At various times, such as when you select a device, UniSite accesses the Algorithm disk. You might be prompted to insert a different Algorithm disk than the one currently in the programmer. If so, insert the disk for which you are prompted. For instance, if UniSite prompts you to `Insert Algorithm Set 3 Disk`, insert the Algorithm disk that contains Algorithm Set 3. If UniSite prompts you to `Insert Algorithm Disk`, insert any one of the Algorithm disks.

Once you have inserted the Algorithm Disk, complete steps 9 through 12 as necessary. Refer to “What to Do Next” on page 2-41 for instructions on what to do the next time you power up your programmer.

9. Update System Software

UniSite-*xpi*

When you have a new version of system software you want to use with UniSite-*xpi*, perform the steps in the update instructions that accompany the new software.

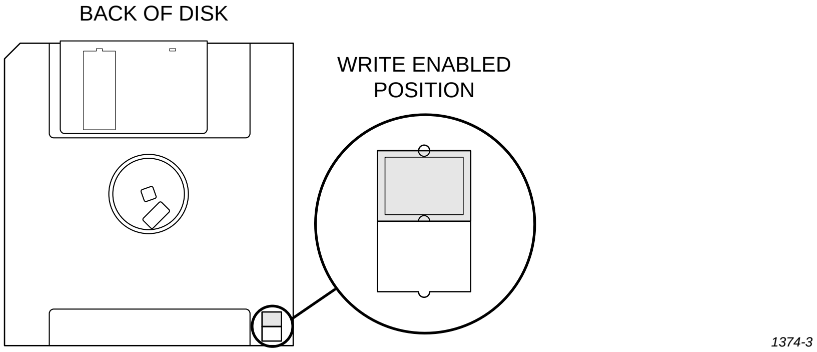
UniSite (Legacy)

If your legacy UniSite does not have an MSM, you will need to boot the UniSite using the Boot Files disk each time you turn on the programmer. Follow the steps below to insert the Boot Files disk into UniSite.

1. Your UniSite software package contains the programmer disks. The first time you open the software package, remove the programmer disks from the software envelope. (You should receive several disks: the **Boot Files** disk, **Algorithm** disks, and the **Device List** disk.)

2. Make sure both the Boot Files disk and the Algorithm disks are write enabled. See Figure 2-9.

Figure 2-9
Write-enabled Disk



3. Insert the Boot Files disk into drive A.
4. Insert an Algorithm disk into drive B (the drive on the right). For more information, see “Insert Algorithm Disk” on page 2-24.

Note: Once you get the programmer set up, make a backup copy of the Boot Files disk and the Algorithm disk(s).

Software Version Compatibility

While the UniSite-*xpi* Boot Files disk and Algorithm disks that are shipped with the system have the same version number, the version numbers of the disks do not have to be identical for the disks to work together. UniSite-*xpi* checks the version numbers of the disks and informs you if they are not compatible with each other.

A Boot Files disk and the Algorithm disks are compatible if they meet the following rules:

- All digits to the left of the decimal point are the same.
- The first digit to the right of the decimal point is the same. (Any additional digits, if present, do not have to match.)

  For example:

System Version	**Algorithm Version**	**Compatible?**
6.50	6.50	Yes
6.51	6.50	Yes
6.50	6.51	Yes
6.50	6.40	No

This version control system allows the Algorithm disks to be updated without requiring a new Boot Files disk.

10. Set Up High Speed Download

This section explains how to set up the programmer so you can download files to the programmer at 115.2K baud.

To take advantage of the High Speed Download feature, you must be controlling the programmer from a PC, the PC must be connected to the programmer's Remote port, and you must be using HiTerm as your terminal emulation software. See the section titled "Installing HiTerm" earlier in this chapter.

Skip to the section titled "What to Do Next Time" if you are not using HiTerm on a PC.

Why Should I Use High Speed Download?

High Speed Download can reduce the transfer time for a large data file by as much as 92%.

As an example, downloading a formatted data file for a 1MB EPROM takes 6 minutes and 38 seconds at 9600 baud. With High Speed Download, it only takes 32 seconds to download the same data file from a 25 MHz 386-based PC.

Note: To obtain the best results, we suggest you run HiTerm on at least a 286-based PC.

What Happens During High Speed Download?

When you begin a download to the programmer, the programmer and HiTerm communicate with each other to coordinate the transfer. HiTerm translates the data file on the PC into a special binary format. The compression of the data file reduces the size of the data file by up to 64%.

Note: HiTerm compresses a temporary copy of the data file; your original data file is not changed.

HiTerm then switches the baud rates on the PC and the programmer to 115.2K baud and downloads the data file to the programmer. After the download, the baud rates on the PC and programmer are restored to their original values.

Configuring the Remote Port

For High Speed Download to work, you must have your PC connected to the programmer's Remote port and the programmer must be configured to run from the Remote port. Follow the steps below to configure the programmer to run in Terminal mode from the Remote port.

1. Start HiTerm. Power up the programmer. If the programmer is already powered up, reboot it by pressing **ESC, CTRL** +**W**. Complete booting up as normal.
2. From the Main Menu, press **M**, **C**, **E**, **S** to get to the Serial Port Configuration screen.

3. The Mode: field for the Terminal port and the Remote port should match. If the two Mode: fields match, skip to step 4.

   If the Mode: fields are different, toggle the DTE/DCE switch for the Remote port on UniSite-*xpi*'s back panel. Figure 1-3 shows the location of the DTE/DCE switch for the Remote port. Go back to the beginning of this step after you have toggled the DTE/DCE switch on the Remote port.

4. From the Main Menu, press **1**, **ENTER** to get to the Communication Parameters screen.

5. Move the cursor to the High Speed Download field. Press **Y** to enable High Speed Download. When you press **Y**, UniSite-*xpi* sets the High Speed Download parameter to **Y** and displays the following message in the message bar:

   ```
   Hit return to switch user menu port, ^Z to abort
   ```

   UniSite-*xpi* also sets the User Menu Port parameter to **R**, which configures UniSite-*xpi* to send its user interface data to the Remote port.

6. Press **ENTER**. You do not see any response on the screen.

7. Look at the back panel of UniSite-*xpi*. Remove the cable connecting UniSite-*xpi* and the PC from the Terminal port and connect the cable to the Remote port.

   When you switch the cable from the Terminal port to the Remote port, the Terminal LED should go out and the Remote LED should light. (See "Configuring the Ports" on page 2-19.) If the Remote LED does not light, power down UniSite-*xpi*, reconnect the PC cable to the Terminal port, and go back to step 1.

8. After you have connected the cable to the Remote port, press **CTRL + R** to redisplay the Communication Parameters screen. If you cannot redisplay the Communication Parameters screen, power down UniSite-*xpi*, reconnect the PC cable to the Terminal port, and go back to step 1.

9. Press **F1** to display the Main Menu. If the Main Menu appears, continue with the next step. If you do not see the Main Menu, power down UniSite-*xpi*, reconnect the PC cable to the Terminal port, and go back to step 1.

High Speed Download is now enabled.

About the User Menu Port

In the previous procedure, you configured UniSite-*xpi* for High Speed Download. As part of the configuration, you switched the User Menu Port from the Terminal port to the Remote port. The User Menu Port is the port through which the user interface data for UniSite-*xpi* is sent. User interface data includes screens, menu information, and online help. When shipped from the factory, the Terminal port is the User Menu Port.

Changing the Powerup Defaults

The previous procedure told you how to change the User Menu Port for the current session. The following procedure tells you how to make that change part of your default working environment on the programmer.

If you do not save the changes you made in the previous procedure, you can still use High Speed Download for the current session; but the next time you power up the programmer, you will have to repeat the procedure to enable High Speed Download.

About Powerup Defaults

As part of the powerup process, the programmer reads a configuration file that contains the settings to use for over 50 system parameters. Collectively, the settings of these parameters are known as the Powerup Defaults. By comparison, the Powerup Defaults are similar to the **autoexec.bat** and **config.sys** files on a PC.

The parameters that comprise the Powerup parameters are listed and described in Chapter 4, “Commands.”

Follow the steps below to make High Speed Download part of your powerup defaults.

1. From the Main Menu, press **M**, **C**, **S** to display the Save System Parameters screen. The programmer displays up to ten configuration files.

   If you want to save High Speed Download as part of your powerup defaults, press **1**, **ENTER**. The programmer displays the following message in the message bar:

   ```
   Parameter Entered
   ```

2. Press **ENTER** to save the current settings as the Powerup Defaults. (Make sure you have your Boot disk handy.) The action symbol rotates while the programmer is saving the current parameter settings as the Powerup Defaults. When finished, the programmer displays the following message in the message bar:

   ```
   System parameters saved
   ```

With High Speed Download saved as part of your Powerup Defaults, you will be able to use High Speed Download for the current session and for any future time you power up the programmer.

Using High Speed Download

With the programmer properly configured for High Speed Download, you can use HiTerm to download files from the PC to the programmer at 115.2K baud.

For an example of downloading files to the programmer using High Speed Download, see “Session 17: Loading Data from a PC Using HiTerm” on page 3-46.

11. Using the Mass Storage Module (Hard Drive)

The Mass Storage Module (MSM) is an internal hard drive installed in UniSite-*xpi*. The Mass Storage Module allows you to store system and algorithm files in the programmer, which can speed up the start-up and disk access routines.

Booting from the MSM

After installing the system software on the MSM (see "Update System Software" on page 2-24), you can boot directly from the MSM by doing the following:

1. Make sure that drive A is empty.
2. Reboot or powerup UniSite-*xpi*.

When UniSite-*xpi* boots up and does not find a disk in drive A, it boots using the system software located on your MSM.

Storage Capacity

The MSM is partitioned into 4 logical drives, with specifications as follows:

Drive	Storage Space	Maximum No. of Files	Data Type
C	31 MB	512	User data
D	31 MB	512	User data
H	7 MB	320	System data
I	10 MB	320	System data

Storage Suggestions

Drives C and D are reserved for user data and drives H and I are reserved for system use. Despite the fact that drives H and I can be written to and read from, we STRONGLY suggest you use only C and D to store your data.

Limitations

Other than the file operations listed below, drives C and D on the MSM can be used for all file operations that can be done with a floppy drive:

File Operation	**Comments**
Format Disk	Format only C and D. Formatting H or I will render your programmer inoperative until you restore the system software from floppy.
Duplicate Disk	Drives C, D, H, and I cannot be duplicated using the Duplicate Disk command.

Backing up the MSM

We STRONGLY suggest that you periodically make copies of the files on the C and D partitions of the MSM as backups. If the MSM were to "crash," you could restore your data files to C and D from your backups, and you could restore the system files from the Boot Files and Algorithm disks.

Installing Hardware (Legacy)

For legacy UniSite programmers without a Mass Storage Module, follow the installation instructions found in the Installation Guide that is shipped with the Mass Storage Module.

Installing Software (Legacy)

After installing the Mass Storage Module, you can set up your programmer to boot system software and read algorithm files directly from the MSM.

To set up your programmer to boot from the MSM, install the system software on the MSM using the More Commands / Configure System / Mass Storage command.

To set up your programmer to read algorithm files from the MSM, install the algorithm files on the MSM using the More Commands / Configure System / Mass Storage command.

See "Update System Software" on page 2-24 of this manual for more information.

12. Working with Devices

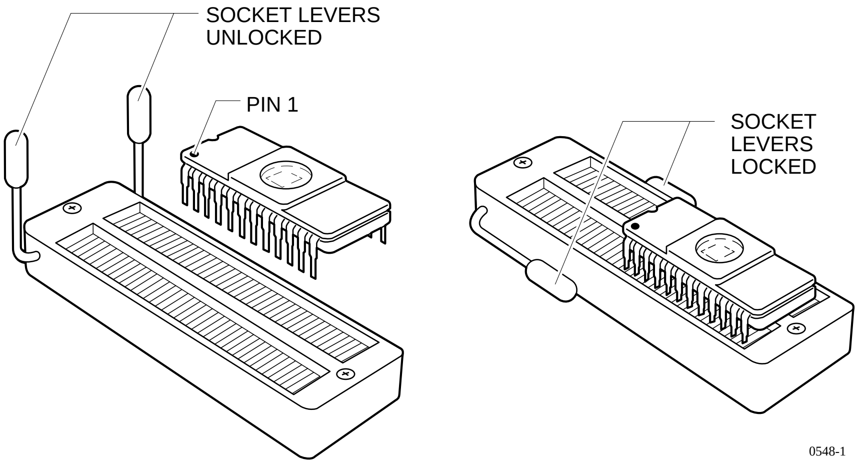
Follow the instructions below to install devices into the programmer's Module or Base sockets (see page 2-15 for a description of Bases). If the device type you wish to install is not listed in this section, refer to the documentation that is included with your Module or Base, or to Appendix A, for installation instructions.

Inserting a DIP Device

To insert a DIP device into the DIP socket, follow the steps below:

1. If a DIP Module or Base is not installed, install the Module or Base.
2. Unlock the socket by pulling up on the socket lever.
3. Insert the DIP device into the socket. Make sure that the device is bottom justified, as shown in Figure 2-10. If the device is not bottom justified, the programmer will not read or program the device.

Figure 2-10
Inserting a DIP Device into the DIP Base



4. Lock the device into place by pressing the socket lever down.

Installing a MatchBook

The MatchBook device carrier holds a device in place on the Base. When the device is locked into place, the conductive pad in the Base forms a conductive path between the pin drivers in the programmer and the device in the MatchBook.

MatchBooks do away with clumsy and expensive sockets and adapters, and make inserting and removing surface mount devices easier and faster. MatchBooks eliminate the guesswork involved when you insert a device into a socket. All you have to do is align pin 1, set the device in the MatchBook, and close the lid.

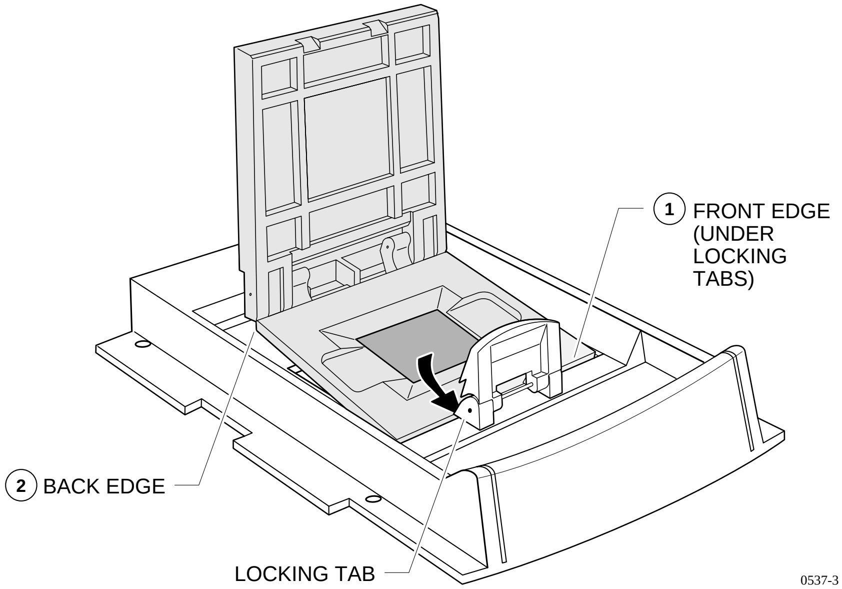
The instructions below explain how to use a MatchBook and how to insert and remove a device from a MatchBook.

Note: The information in this section covers all device types that are supported by MatchBook device carriers. For more information, see the section on the particular package type you are using.

1. Insert the Base into the programmer. Lock the Base into place.
2. Select the MatchBook for the device you are going to use.
3. With the MatchBook open 90 degrees, insert it into the Base by setting the front edge onto the Base under the two locking tabs at the front edge of the Base. Then lower the back edge of the MatchBook into place on the Base. See Figure 2-11.

4. Insert the device into the MatchBook as described later in this section.

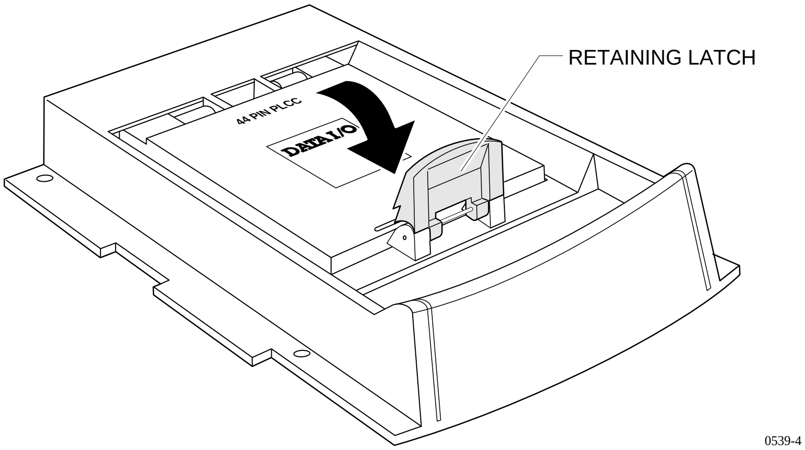
Figure 2-11
Inserting a MatchBook into the Base



5. Finally, close the MatchBook and press the retaining latch forward with your thumb until the latch snaps into place as shown in Figure 2-12.

CAUTION: Do not place excessive force on the top of the MatchBook, as this may cause premature wear on the conductive pad.

Figure 2-12
Closing the MatchBook

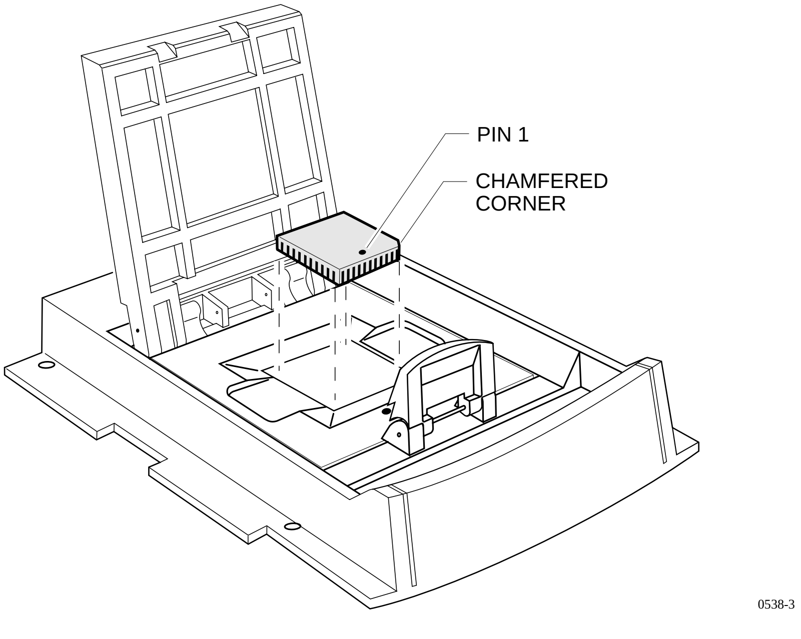


Inserting a PLCC or LCC Device into a MatchBook

Use the following procedure to insert a PLCC or LCC device into a MatchBook.

1. Select the appropriate MatchBook and Base. For example, if you are using a 44-pin PLCC device, you would select the 44-pin PLCC MatchBook and the PLCC Base.
2. Insert the appropriate Base into the programmer.
3. Insert the appropriate MatchBook into the Base.
4. Position the PLCC or LCC device so that pin 1 is near the handle on the Base. There is a small dot molded into each MatchBook to help you align your device. Each PLCC MatchBook and LCC MatchBook also has a beveled corner to help you align devices with a chamfered corner to indicate pin 1. Figure 2-13 illustrates the proper positioning of a PLCC or LCC device.

Figure 2-13
Inserting a Device into the PLCC or LCC Base



5. Insert the device into the open MatchBook.
6. Finally, close the MatchBook and press the retaining latch forward with your thumb until the latch snaps into place, as shown in Figure 2-12.

CAUTION: Do not place excessive force on the top of the MatchBook, as this may cause premature wear on the conductive pad.

Inserting an SOIC Device into a MatchBook

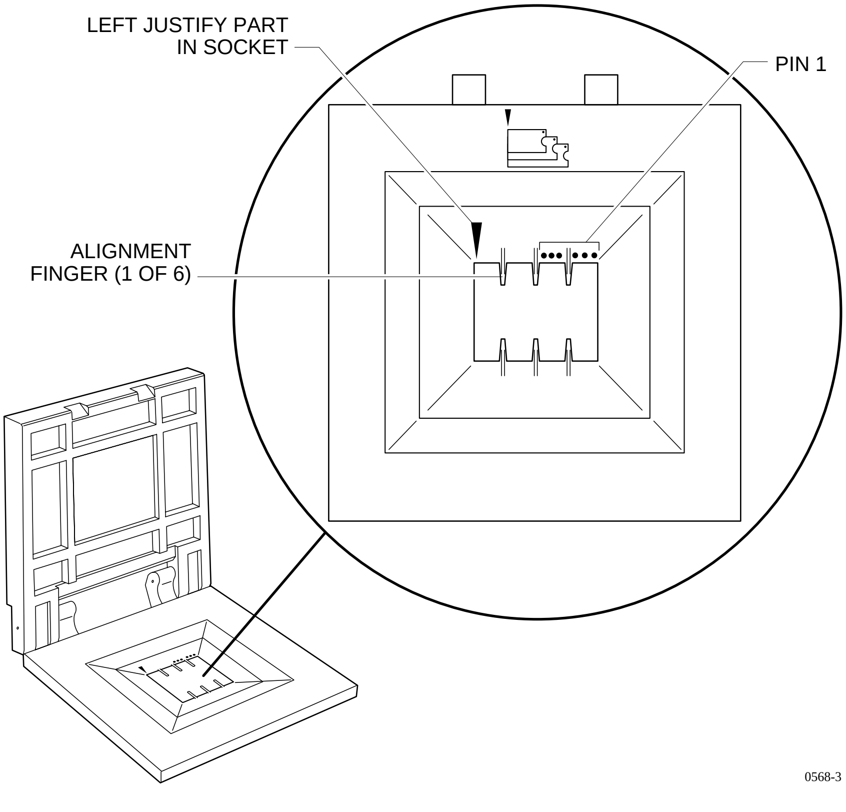
Use the following procedure to insert an SOIC device into the SOIC MatchBook. Before you go any further, make sure you have the SOIC Base properly installed in the programmer.

1. Select the appropriate MatchBook and Base. For SOIC devices, select the MatchBook according to package width: 0.150 inches, 0.300 inches, etc.
2. Insert the SOIC Base into the programmer. Insert the appropriate MatchBook into the SOIC Base.
3. Position the SOIC device so that pin 1 is up and to the right as you view it from the top.
4. Insert the SOIC device into the open MatchBook. Make sure the SOIC device is flush against the left-hand side of the MatchBook. Also make sure that the device is positioned between the six alignment fingers and not on top of them.

Note: The device should be left justified. The unused portion of the socket will be on the right as you view it from the top.

The small, round dots along the top of the opening, as shown in Figure 2-14, indicate the location of pin 1 for the various sizes of SOIC devices the SOIC MatchBook will accept.

Figure 2-14
Inserting an SOIC Device



5. Finally, close the MatchBook and press the retaining latch forward with your thumb until the latch snaps into place, as shown in Figure 2-12.

Removing a Device From a MatchBook

To remove a device from the MatchBook, unsnap the retaining latch, open the MatchBook, and lift out the device.

Special Note About the Conductive Pad

After a number of insertions you may notice an indentation in the middle of the conductive pad. (The conductive pad is the material the MatchBook rests on.) The indentation is normal and does not degrade the contact resistance. Nor does it degrade the performance of the MatchBook.

The life of the pad is dependent on proper care as well as the pin count and package type of the device being used. Not all devices have the same tolerances, and each device type may result in different life cycles for the pad.

If you experience an increase in device insertion errors or continuity errors, or if you experience a sudden drop in programming yields, it may be an indication that the pad needs to be replaced. The Base has been designed to allow you to replace the pads quickly and easily and to minimize downtime. Replacement pads are available from Data I/O.

Pad Care

Each pad should be inspected and cleaned as needed; we recommend you do this approximately every 1000 insertions or once a month. It is normal for the pad to show signs of discoloration as it is used.

Clean the pad by blowing air over the pad. If you use compressed air, direct the air stream from the front or back of the Base. Make sure the compressed air does not contain any petroleum based products as these will cause premature deterioration of the pad material.

CAUTION: Blowing air from the side of the pad could lift the pad off the circuit board.

To further clean the pad, apply a small amount of isopropyl alcohol on a cotton swab and, with a rolling motion, gently wipe off the pad. Make sure the pad is clear of any cotton filaments left over from the cleaning process.

CAUTION: Do not use any petroleum or freon based products to clean the pad. These substances will cause premature deterioration of the pad material.

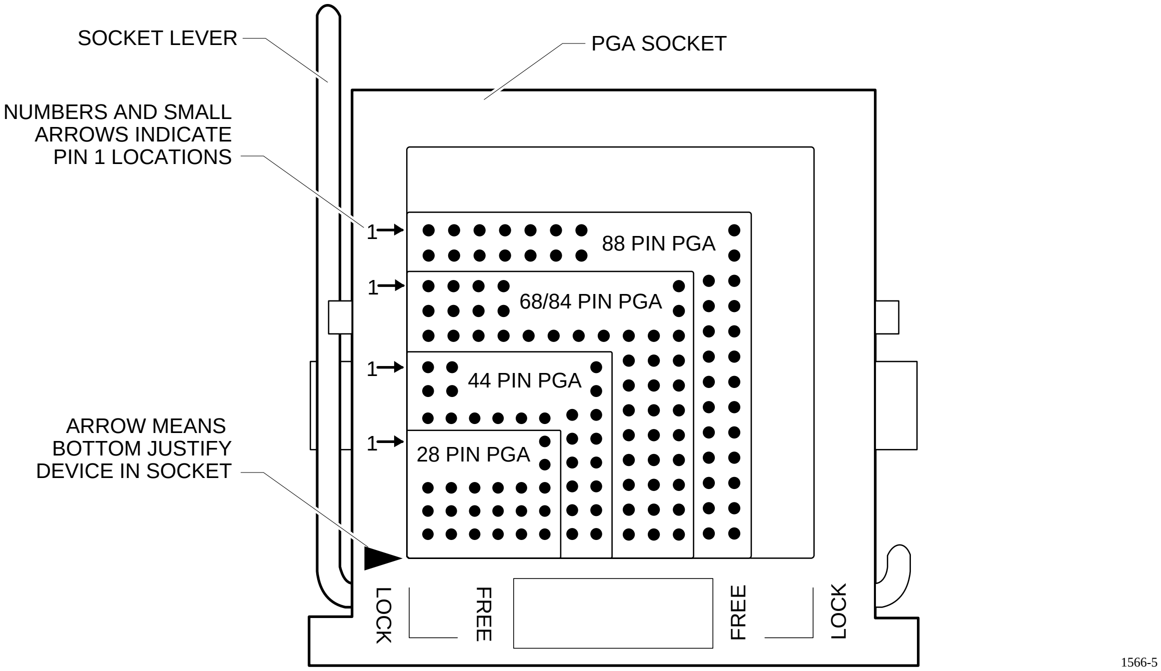
Inserting a PGA Device in a PGA Base

The PGA Base can accommodate PGA packages with up to 15 x 15 pin arrays. Use the following procedure to insert a PGA device into the PGA Base.

Note: When you insert a PGA device into the PGA Base, pay particular attention to the orientation and positioning of the device.

1. Install the PGA Base into the programmer.
2. Unlock the PGA socket by lifting up on the socket lever.
3. Insert the device into the PGA socket. Make sure the PGA device is bottom justified and that pin 1 of the device is against the left side of the socket. Figure 2-15 shows the proper alignment of a PGA device.
4. Push the socket lever down to lock the PGA device into the PGA socket.

Figure 2-15
Orienting a PGA Device in the PGA Base


SOCKET LEVER
PGA SOCKET
NUMBERS AND SMALL ARROWS INDICATE PIN 1 LOCATIONS
ARROW MEANS BOTTOM JUSTIFY DEVICE IN SOCKET
1
1
1
1
88 PIN PGA
68/84 PIN PGA
44 PIN PGA
28 PIN PGA
LOCK
FREE
FREE
LOCK
1566-5

Note: Figure 2-15 shows the socket lever attached at the lower left. On some bases, the socket lever may be attached at a different point.

Inserting Devices into a PPI Adapter

The PPI system consists of one Base for each programmer, and exchangeable adapters. These PPI adapters accept a variety of device package types and pin configurations. The following figures show three common device packages and the proper device orientation in the adapter. Each adapter has an ICON showing the proper device orientation.

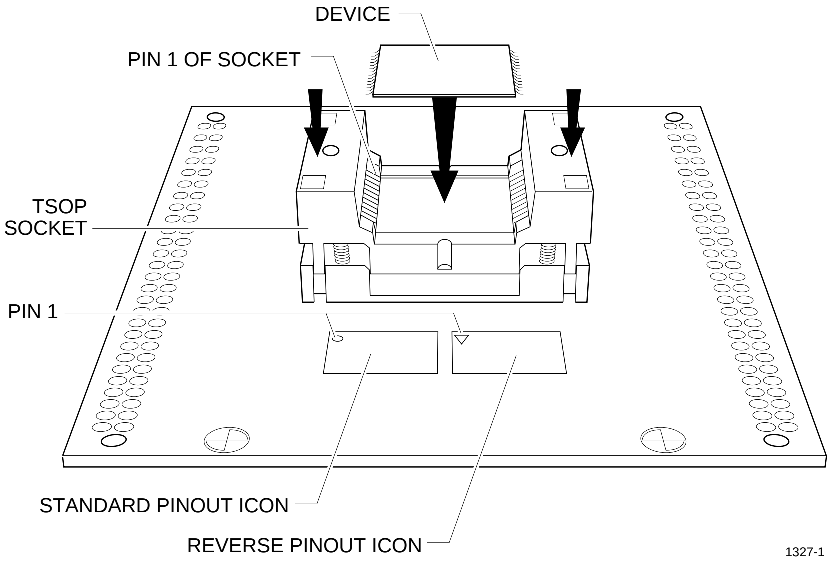
TSOP Device Orientation

There are two different pinouts for the TSOP device as shown in Figure 2-16. The standard pinout, shown by the standard pinout ICON on the left in Figure 2-16, shows Pin 1 indicated by a circle in the upper-left corner of the device. The reverse pinout, shown by the reverse pinout ICON on the right in Figure 2-16, shows Pin 1 indicated by a triangle in the upper-left corner of the device.

1. Press down on the outer edges of the socket.
2. Drop the TSOP device into the socket while still holding the socket edges down.
3. Release the socket edges. The device should be secured.

To remove the device, press down on the socket edges and lift the device out.

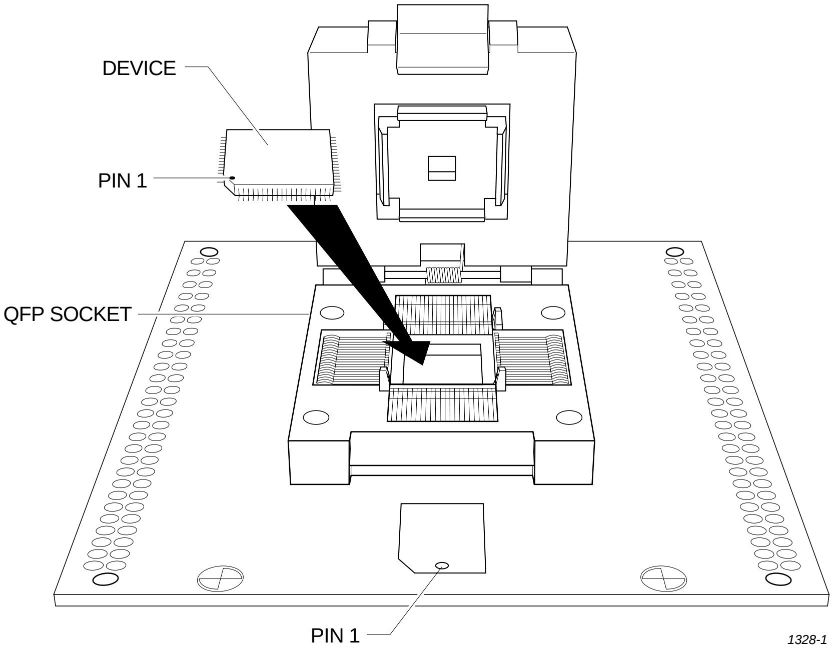
Figure 2-16
TSOP Device Orientation



Quad Flat Pack Device Orientation

Pin 1 on the QFP device is usually at the front left of the device. Orient the QFP device according to the icon or indicator on the adapter. See Figure 2-17.

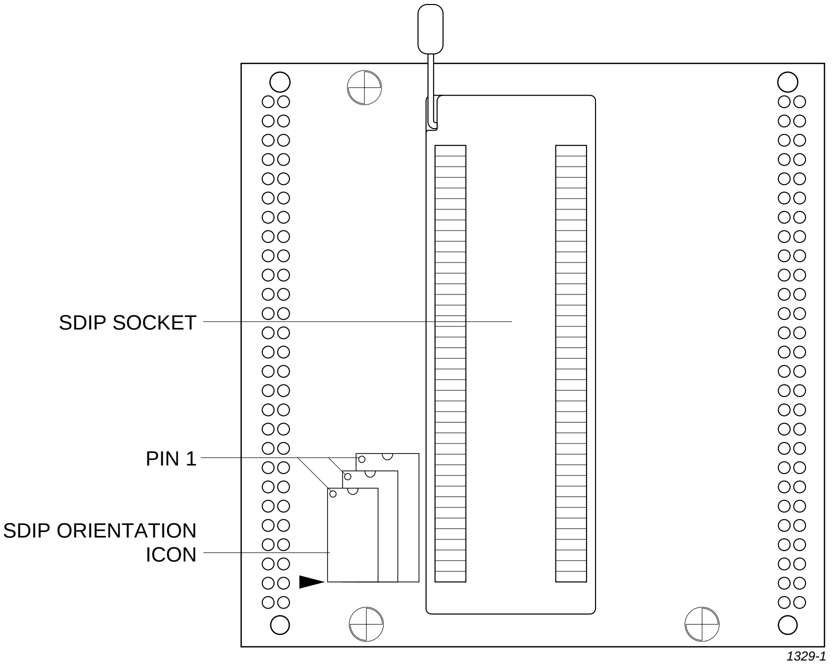
Figure 2-17
QFP Device Orientation



Shrink Dip Device

Orient the SDIP device in the socket with pin 1 at the top left and bottom justified as shown in Figure 2-18.

Figure 2-18
Shrink Dip Device Orientation



Maintenance

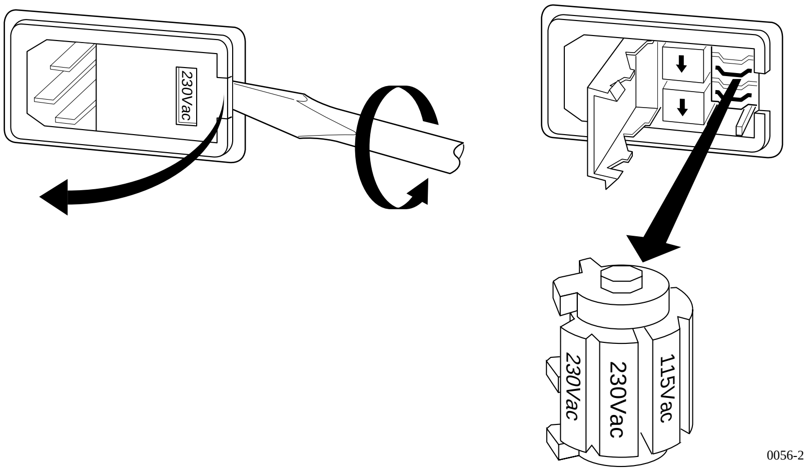
Changing the Line Voltage (Legacy)

Data I/O has configured UniSite to operate on 115 Vac unless specified otherwise. Older units have a line voltage indicator that is visible through the window in the back panel door covering the voltage selector wheel, shown in Figure 2-19. The ac line voltage that will be used to operate UniSite must match the number indicated in the window. If the voltage you need to use is NOT the same as the number in the window, follow the steps below to change the selected voltage.

CAUTION: You could damage UniSite if you operate it with the wrong line voltage.

1. Disconnect the power cord.
2. Using a flat-tipped screwdriver, gently pry open the door that covers the voltage selector wheel.

Figure 2-19
Removing the Voltage Selector Wheel



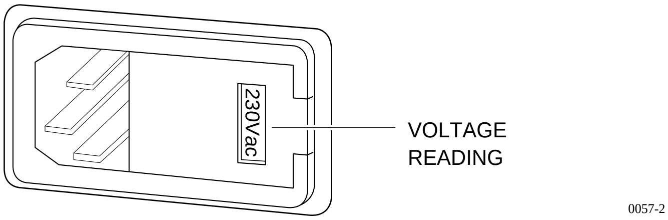
3. As shown in Figure 2-19, pull the voltage selector wheel out of its slot.
4. Rotate the selector until the correct line voltage points away from UniSite’s rear panel. Insert the selector back into its slot.

Note: The voltage wheel has two positions: 115 Vac and 230 Vac. These are nominal voltages—each voltage has high and low limits. The limits for 115 Vac is 90 Vac to 132 Vac and the limits for the 230 Vac position are 180 Vac to 264 Vac.

5. Snap the door closed. The correct voltage now appears in the window.

6. Verify that the line voltage is correct by checking the line voltage indicator. Figure 2-20 shows the location of the voltage indicator.

Figure 2-20
Voltage Reading

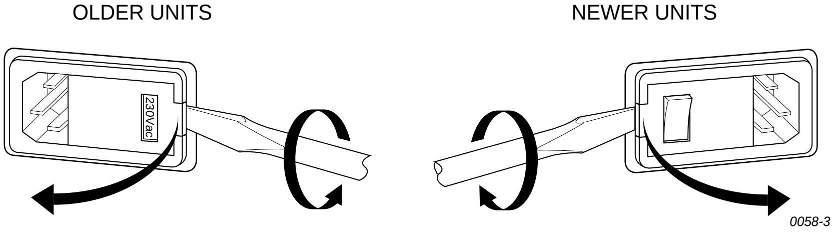


Replacing the Line Fuse

The line fuse is located behind the same door that covers the voltage selector wheel. Perform the following procedure to replace the line fuse. If the fuse is blown, replace it with one of the same size and rating.

1. Gently pry open the door that covers the fuse holder using a flat-tipped screwdriver.

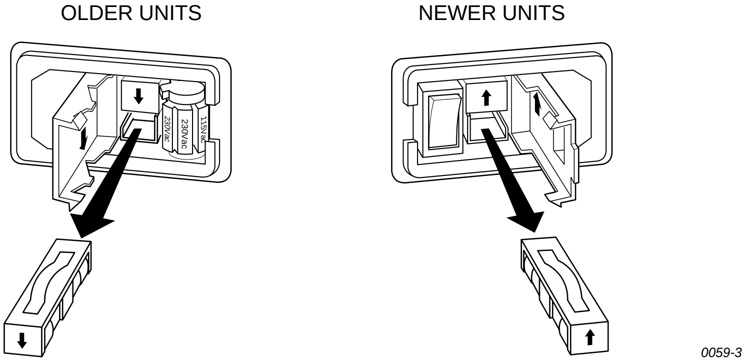
Figure 2-21
Opening the Fuse Holder



Note: The power entry module will accept two fuse cartridges. One cartridge holds USA standard size fuses (1/4” x 1 1/4”) and the other holds international standard size fuses (5 mm x 20 mm). Only the bottom receptacle is connected to the programmer’s circuitry.

2. Pull the bottom fuse holder out of its slot.

Figure 2-22
Removing the Fuse Holder

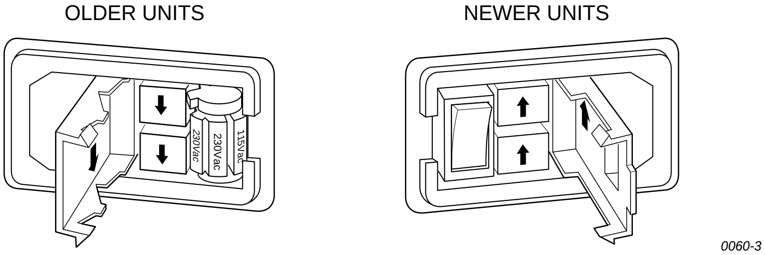


3. Determine whether the fuse is intact. If it is intact, proceed to Step 4. If the fuse is blown, install a new fuse.

CAUTION: For continued protection against the possibility of fire, replace only with a fuse of the correct voltage, current and type ratings.

4. Insert the fuse holder into its slot so that the arrow points in the same direction as the arrows on the door of the fuse holder.
5. Snap the door closed.

Figure 2-23
Removing the Fuse Holder



What To Do Next Time

Next time you power up the programmer, you probably do not need to follow all the steps outlined in this chapter. Listed below are the normal steps for preparing for another session on the programmer.

Note: *If you have not used the programmer for awhile, or if you suspect the programmer might have been moved from one area to another, follow the procedure below before you use the programmer.*

To prepare the programmer for another session, follow the procedure below:

1. Check the power cords and cables between the programmer and the connected equipment.
2. If you are controlling the programmer from a PC or workstation, make sure it is on and that the terminal emulation software (such as HiTerm) is running.

   If you are controlling the programmer from a terminal, make sure it is on.

   If you are using terminal emulation software, select **VT100 on Programmer Port** from the **Utilities** menu.
3. (Legacy) If you do not have an MSM installed, insert the Boot Files disk into UniSite’s disk drive A, and insert an Algorithm disk into disk drive B.
4. Select and insert a Module (and Base if necessary) into the programmer. Make sure the Module and Base are locked in place.
5. Power up the programmer.
6. Verify the terminal type when the Power-on screen appears.

You are now ready to begin a new session on the programmer.

Refer to step 12, “Install Devices,” on page 2-30 for instructions on how to install devices. Refer to Chapter 3 “Getting Started” for tutorials on how to use your programmer.

3. Getting Started

Before You Begin

Before you start the Sessions in this chapter, read Chapter 1, “Introduction,” which introduces you to UniSite-*xpi*. Next, read Chapter 2, “Setup and Installation,” which tells you how to set up UniSite-*xpi* and install the software.

Should I Read This Chapter?

You should read and follow the Sessions in this chapter if you are unfamiliar with UniSite-*xpi* Universal Programmer or if you are unfamiliar with device programmers in general. By following the Sessions, you will learn how to select a device, load device data, edit data, program the device, and verify that it programmed correctly.

The Sessions will not teach you everything about UniSite-*xpi*; instead, they will give you a working knowledge of the programmer. If you are using TaskLink for Windows or TaskLink for DOS, see the online Help for more detailed information. If you are using Terminal Mode, see Chapter 4, “Commands,” for more detailed information.

What Will I Learn?

The Sessions are meant to be read sequentially and are divided into Sessions for TaskLink for Windows mode, TaskLink for DOS mode, and HiTerm (Terminal) mode. The most basic steps are taught first and lead to more complex procedures.

The Sessions are organized as follows:

TaskLink for Windows Mode (serial and parallel)

Session 1: Selecting a Device

Session 2: Setting Programmer Properties

Session 3: Loading Data

Session 4: Selecting Processes

Session 5: Programming a Device

TaskLink for DOS Mode (serial only)

Session 6: Selecting a Device

Session 7: Setting Programmer Properties

Session 8: Loading Data

Session 9: Selecting Processes

Session 10: Programming a Device

HiTerm (Terminal) Mode (serial only)

Session 11: Navigating Through the Menus

Session 12: Selecting a Device

Session 13: Selecting a Keep Current Algorithm

Session 14: Loading Data from a Device

Session 15: Loading Data from Disk

Session 16: Selecting a Translation Format

Session 17: Loading Data from a PC Using HiTerm

Session 18: Loading Data from a Host

Session 19: Editing Data

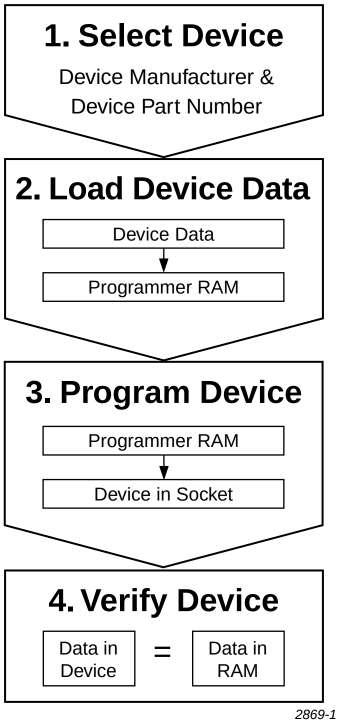
Session 20: Programming a Memory Device

Session 21: Verifying a Device

Outline of the Programming Operation

Figure 3-1 shows an outline of the programming operation. The programming operation is basically the same whether you use TaskLink for Windows, TaskLink for DOS, or HiTerm as your user interface.

Figure 3-1
Programming Operation



1. The **Select Device** operation selects the manufacturer and part number of the device you will be using so the programmer can perform device operations with the appropriate programming algorithm.
2. The **Load Device Data** operation moves device data from a master device, hard drive, floppy disk drive, or network into programmer RAM.
3. The **Program Device** operation transfers the device data in programmer RAM into the device in the socket using the device's programming algorithm.
4. The **Verify Device** operation (included with programming) compares the data in a programmed device to the data in programmer RAM. Additional verify operations provide information about programming errors. Logic device verification can include functional testing.

Tutorials for TaskLink for Windows Mode

Sessions 1-5 take you through the basic process of programming a device using TaskLink for Windows as your user interface.

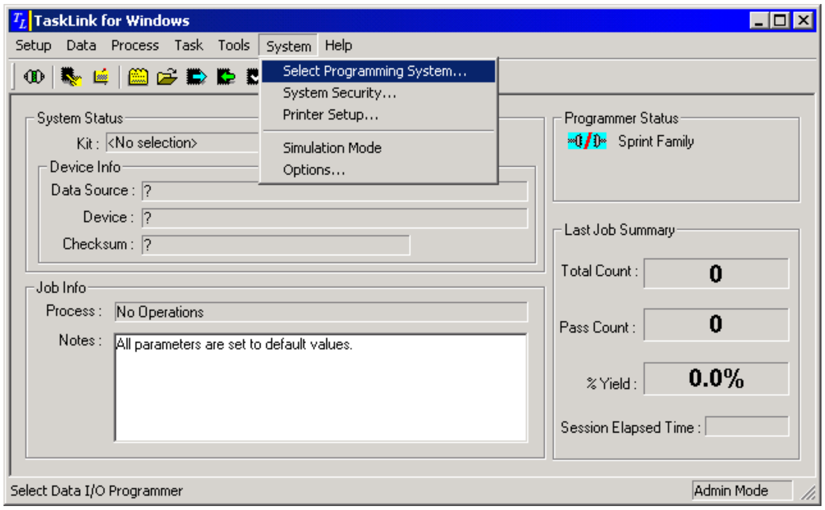
Before you go any further in this chapter, make sure you have read and completed the instructions in Chapter 2, “Setup and Installation.” You should also have read Chapter 1, “Introduction.”

For these Sessions, you will use TaskLink for Windows in simulation mode, and you do not need to have your programmer connected.

Session 1: Selecting a Device

On the TaskLink main screen, under the *System* menu select **Select Programming System.**

Figure 3-2
System Menu: Select Programming System



Kit : <No selection>

Data Source : ?

Device : ?

Checksum : ?

Process : No Operations

Notes : All parameters are set to default values.

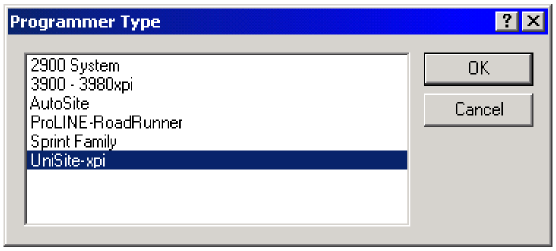
Total Count : 0

Pass Count : 0

% Yield : 0.0%

From the Programmer Type list, highlight UniSite-*xpi* and click **OK**.

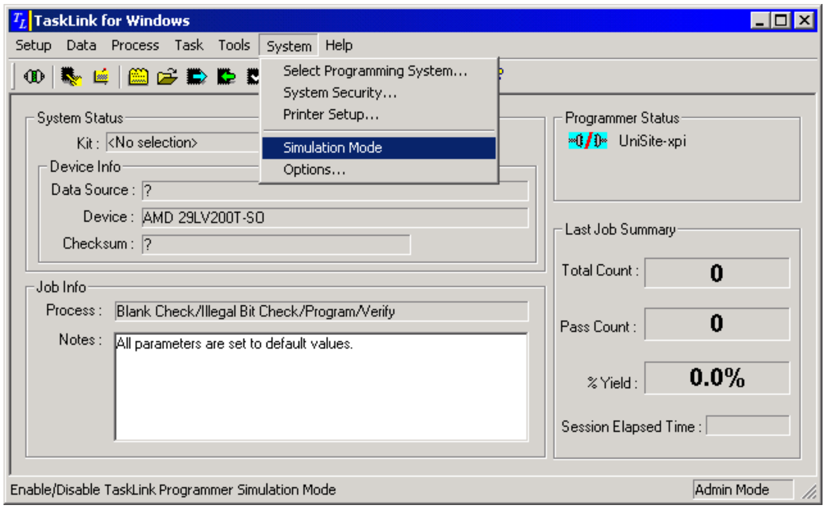
Figure 3-3
Programmer Type



2900 System
3900 - 3980xpi
AutoSite
ProLINE-RoadRunner
Sprint Family
UniSite-xpi

On the TaskLink main screen, under the *System* menu select **Simulation Mode**.

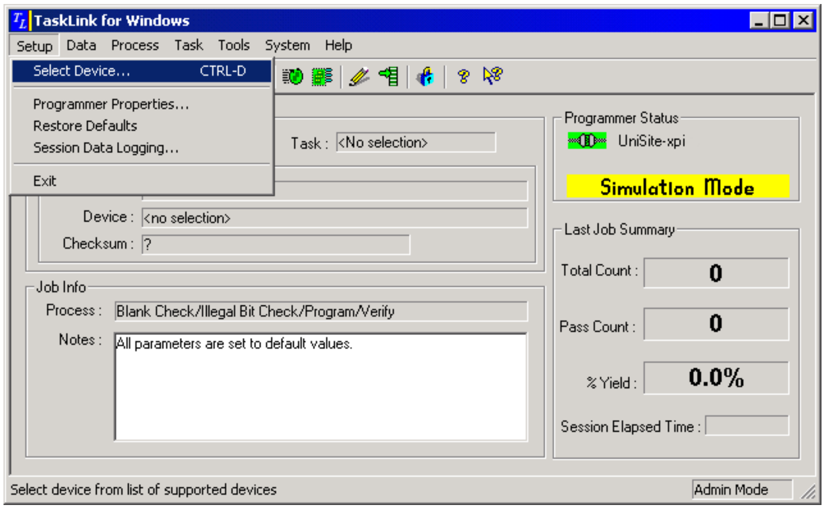
Figure 3-4
System Menu: Simulation Mode



Kit : <No selection>
Data Source : ?
Device : AMD 29LV200T-SO
Checksum : ?
Process : Blank Check/Illegal Bit Check/Program/Verify
Notes : All parameters are set to default values.
UniSite-xpi
Total Count : 0
Pass Count : 0
% Yield : 0.0%
Session Elapsed Time :
Enable/Disable TaskLink Programmer Simulation Mode

Next specify the device to be programmed. From the TaskLink *Setup* menu, select **Select Device**.

Figure 3-5
Setup Menu: Select Device

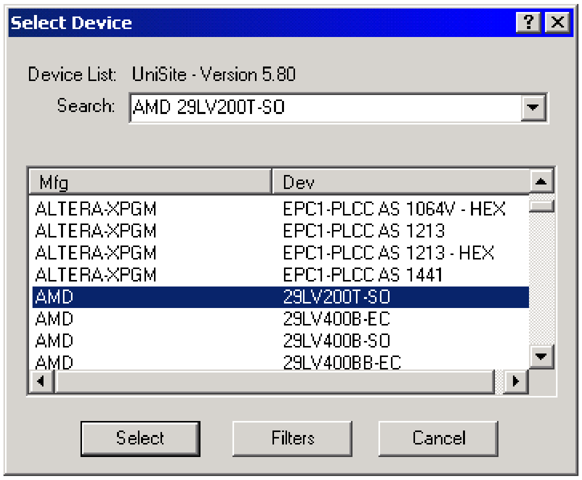


Task : <No selection>
Device : <no selection>
Checksum : ?
Process : Blank Check/Illegal Bit Check/Program/Verify
Notes : All parameters are set to default values.
UniSite-xpi
Total Count : 0
Pass Count : 0
% Yield : 0.0%
Session Elapsed Time :
Select device from list of supported devices

On the **Select Device** dialog, scroll through the list of device part numbers until the one you want is highlighted.

Click **Select**.

Figure 3-6
Select Device Dialog



Device List: UniSite - Version 5.80

Search: AMD 29LV200T-SO

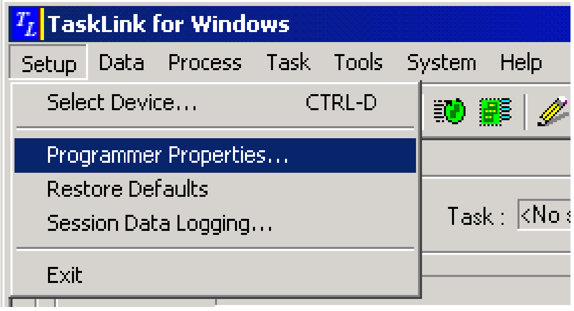
Mfg	Dev
ALTERA-XPGM	EPC1-PLCC AS 1064V - HEX
ALTERA-XPGM	EPC1-PLCC AS 1213
ALTERA-XPGM	EPC1-PLCC AS 1213 - HEX
ALTERA-XPGM	EPC1-PLCC AS 1441
AMD	29LV200T-SO
AMD	29LV400B-EC
AMD	29LV400B-SO
AMD	29LV400BB-EC

Session 2: Setting Programmer Properties

After specifying the device to be programmed, the next step is setting the programmer properties.

From the *Setup* menu, select **Programmer Properties**.

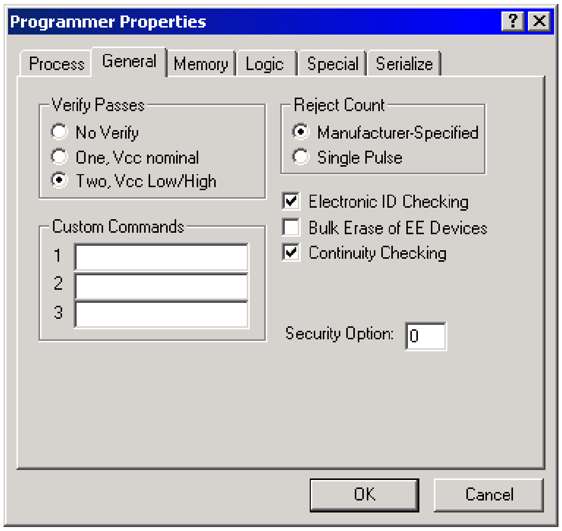
Figure 3-7
Setup Menu: Programmer Properties



Select the **General** tab and enter the parameters that you wish to use. These parameters determine what device operations are performed before, during and after programming. For more detailed information about each parameter, see TaskLink context-sensitive Help.

Click **OK**.

Figure 3-8
General Tab



Verify Passes
○ No Verify
○ One, Vcc nominal
● Two, Vcc Low/High

Custom Commands
1
2
3

Reject Count
● Manufacturer-Specified
○ Single Pulse

☑ Electronic ID Checking
□ Bulk Erase of EE Devices
☑ Continuity Checking

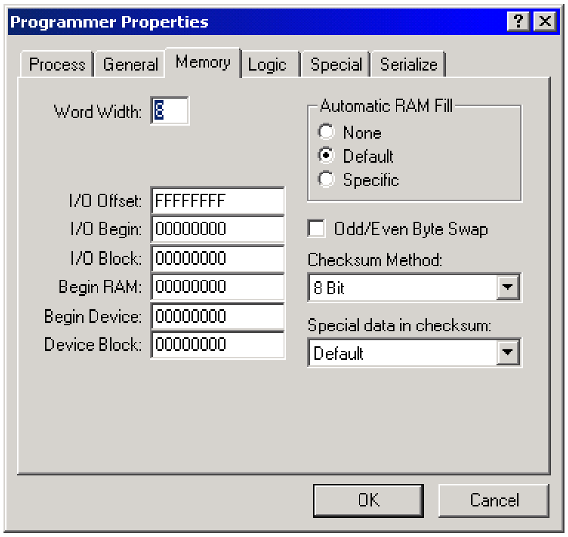
Security Option: 0

Next, select the **Memory** tab. Enter the parameters to specify how and where the data is loaded in programmer RAM. For more detailed information about each parameter, see TaskLink context-sensitive Help.

To ensure consistent checksum, you can preload all address locations into RAM with a known hexadecimal value (many data files do not contain data for every memory address). For more information, see TaskLink online Help

Click **OK**.

Figure 3-9
Memory Tab



Word Width: 8

I/O Offset: FFFFFFFF
I/O Begin: 00000000
I/O Block: 00000000
Begin RAM: 00000000
Begin Device: 00000000
Device Block: 00000000

Automatic RAM Fill
○ None
● Default
○ Specific

□ Odd/Even Byte Swap

Checksum Method: 8 Bit

Special data in checksum: Default

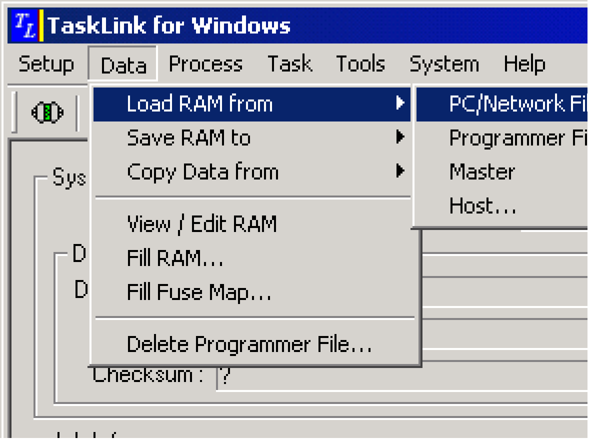
Session 3: Loading Data

After setting the programmer properties, the next step is to load programmer RAM with the data to be programmed into the device.

Data can be loaded from four sources: a PC / Network file, a Programmer file, a Master device, or a Host. For this Session, we will load data from a PC / Network file.

From the *Data* menu, select **Load RAM from PC/Network File**.

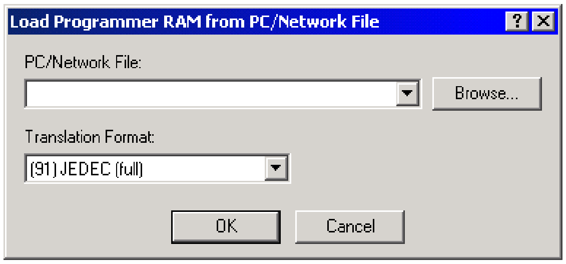
Figure 3-10
Data Menu: Load RAM from PC/Network File



In the **Load Programmer RAM from PC/Network File** dialog, there are two fields to fill.

In the **PC/Network File** field, enter the name of the file, or click **Browse** for help in locating the file that contains the data you want to load into programmer RAM.

Figure 3-11
Browsing for Data File


Load Programmer RAM from PC/Network File

PC/Network File:

Translation Format: (91) JEDEC (full)

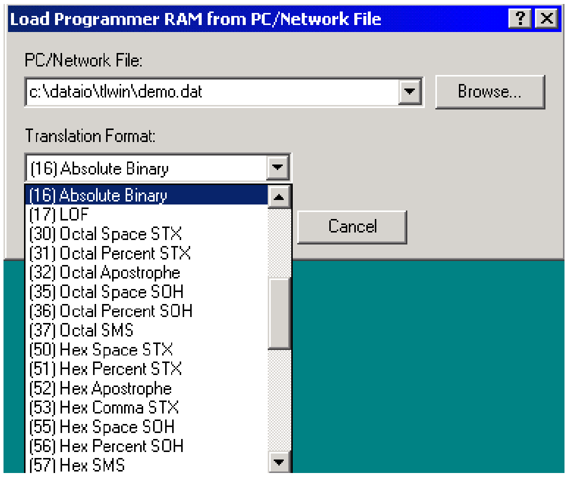
The other field to fill is **Translation Format**.

When a data file is created, it is stored in a specific data translation format. When the data file is transferred to a programmer, the programmer must be set to handle the appropriate translation format. For more information on translation formats, see Chapter 6 “Translation Formats.”

Select the appropriate translation format from the drop-down list.

Click **OK**.

Figure 3-12
Choosing a Translation Format



Load Programmer RAM from PC/Network File

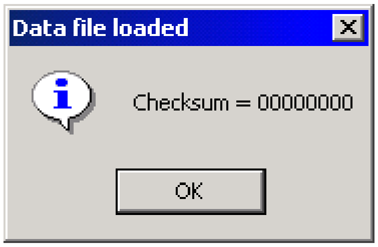
PC/Network File: c:\dataio\tlwin\demo.dat

Translation Format: (16) Absolute Binary

When the data file is successfully loaded into programmer RAM, a message box displays the RAM checksum.

Click **OK**.

Figure 3-13
Data File Loaded



Data file loaded

Checksum = 00000000

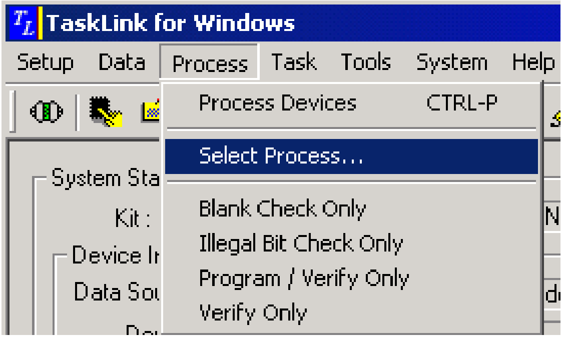
Note: Because TaskLink for Windows is running in Simulation Mode, the checksum displayed is 00000000. When connected to a programmer and loading real data, the checksum will display actual values.

Session 4: Selecting Processes

After loading data into programmer RAM, the next step is determining which processes the programmer will perform on the device.

From the *Process* menu, select **Select Process**.

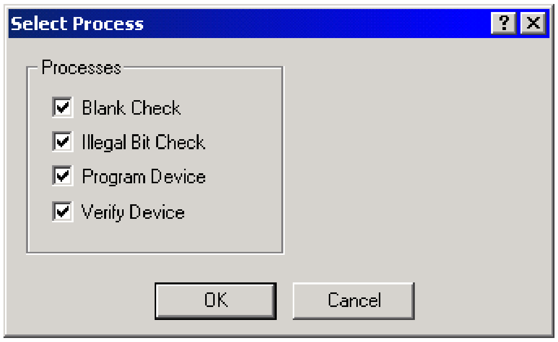
Figure 3-14
Process Menu: Select Process



In the **Select Process** dialog, enter a check for each process you want performed on the device to be programmed.

Click **OK**.

Figure 3-15
Select Process

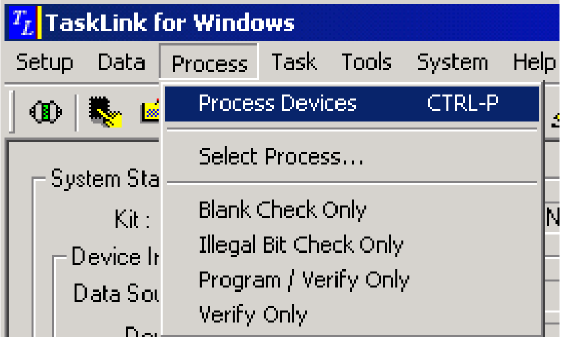


Session 5: Programming a Device

Now it is time to program a device.

From the *Process* menu, select **Process Devices**.

Figure 3-16
Process Menu: Process Devices



TaskLink for Windows

In the **Process Devices** dialog, enter a Pass Limit. This number sets an upper limit on the number of devices processed in this programming session. If you want to process an unlimited number of devices, set the value at “0.”

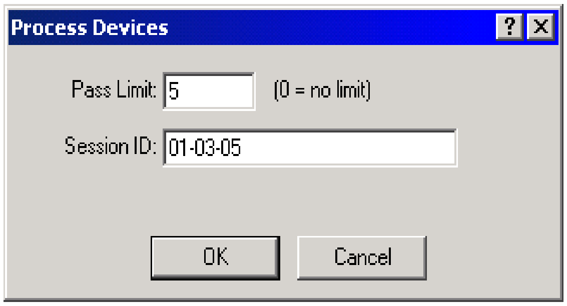
In this example, we’ll enter a Pass Limit of 5.

You may also enter a **Session ID** name or number. A Session ID is optional and helps when tracking statistics.

In this example, we’ll enter a Session ID of 01-03-05.

Click **OK**.

Figure 3-17
Process Devices



Process Devices

Pass Limit: 5 (0 = no limit)

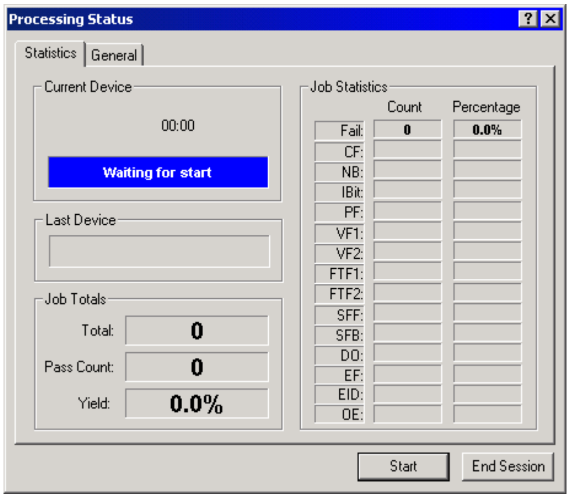
Session ID: 01-03-05

After a brief pause while the programmer sets up, a Device Footnote is displayed. The programmer is now ready to process a device.

The **Processing Status** window is displayed with “Waiting to Start” in the **Current Device** field.

Click **Start**.

Figure 3-18
Waiting for Start



Current Device

00:00

Waiting for start

Last Device

Job Totals

Total: 0

Pass Count: 0

Yield: 0.0%

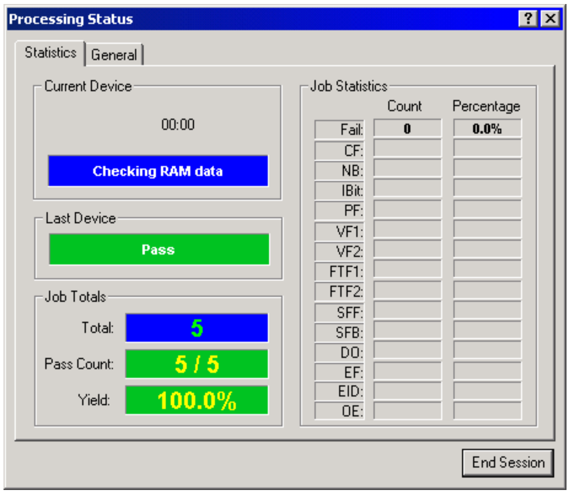
Job Statistics

	Count	Percentage
Fail:	0	0.0%
CF:		
NB:		
IBit:		
PF:		
VF1:		
VF2:		
FTF1:		
FTF2:		
SFF:		
SFB:		
DO:		
EF:		
EID:		
OE:		

After reaching the Pass Limit, or programming the desired number of devices, the **Processing Status** window displays the results of the programming session.

In this example, we processed five devices and all passed.

Figure 3-19
Results of Programming Session



Current Device

00:00

Checking RAM data

Last Device

Pass

Job Totals

Total: 5

Pass Count: 5 / 5

Yield: 100.0%

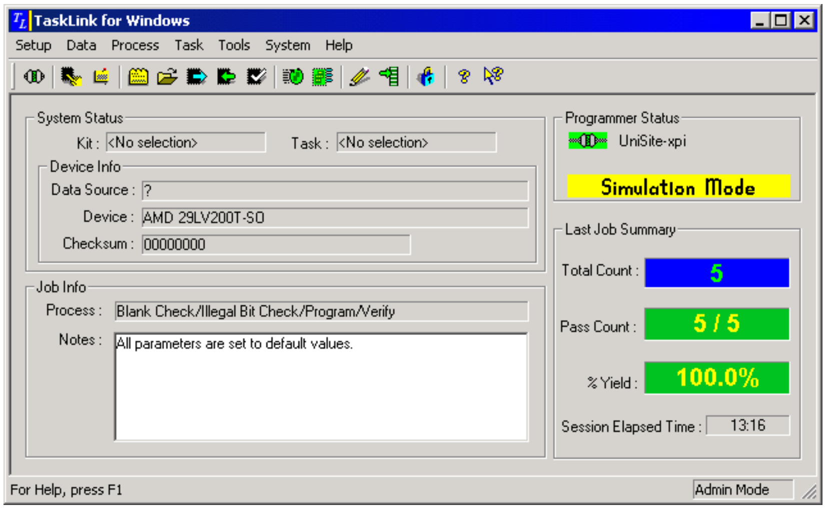
Job Statistics

	Count	Percentage
Fail:	0	0.0%
CF:		
NB:		
IBit:		
PF:		
VF1:		
VF2:		
FTF1:		
FTF2:		
SFF:		
SFB:		
DO:		
EF:		
EID:		
OE:		

To finish, click **End Session**.

This returns you to the TaskLink main screen which now displays a **Last Job Summary** about the most recently completed programming session.

Figure 3-20
TaskLink Main Screen with Last Job Summary



System Status

Kit : <No selection> Task : <No selection>

Device Info

Data Source : ?

Device : AMD 29LV200T-SO

Checksum : 00000000

Job Info

Process : Blank Check/Illegal Bit Check/Program/Verify

Notes : All parameters are set to default values.

Programmer Status

UniSite-xpi

Simulation Mode

Last Job Summary

Total Count : 5

Pass Count : 5 / 5

% Yield : 100.0%

Session Elapsed Time : 13:16

For Help, press F1

Admin Mode

This completes the process of programming a device.

Tutorials for TaskLink for DOS Mode

Sessions 6-10 take you through the basic process of programming a device using TaskLink for DOS as your user interface.

Note: TaskLink for DOS supports only serial mode operations.

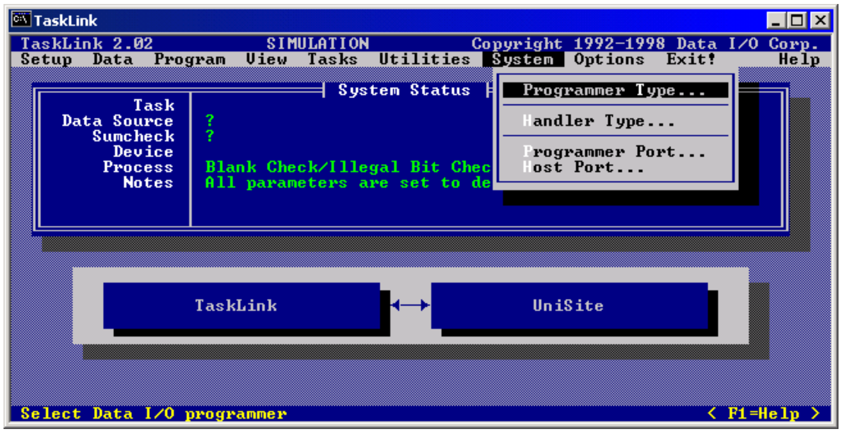
Before you go any further in this chapter, make sure you have read and completed the instructions in Chapter 2, “Setup and Installation.” You should also have read Chapter 1, “Introduction.”

Session 6: Selecting a Device

From the TaskLink directory, type **tl as** and then press **Enter**. This opens TaskLink in simulation mode. (To run TaskLink when you want to actually program a device, type **tl a**, then press **Enter**.)

From the *System* menu, select **Programmer Type**.

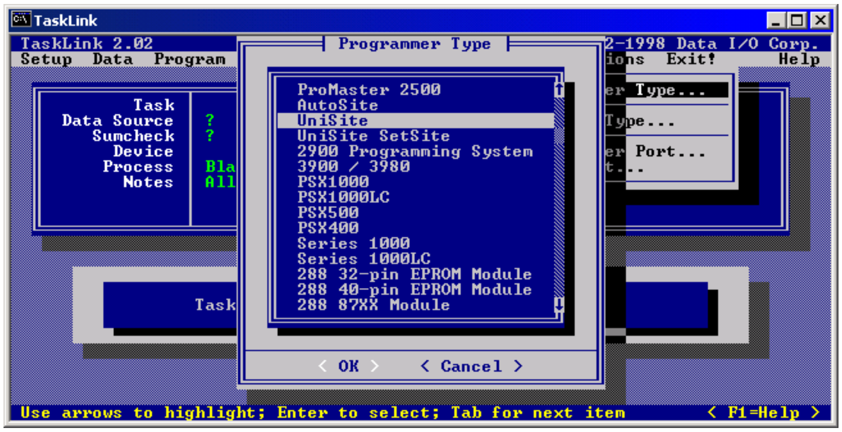
Figure 3-21
System Menu: Programmer Type



```
TaskLink 2.02          SIMULATION          Copyright 1992-1998 Data I/O Corp.
              Task
       Data Source   ?
          Sumcheck   ?
            Device
           Process   Blank Check/Illegal Bit Chec
             Notes   All parameters are set to de
Select Data I/O programmer                                         < F1=Help >
```

On the Programmer Type screen, select UniSite and click **OK**.

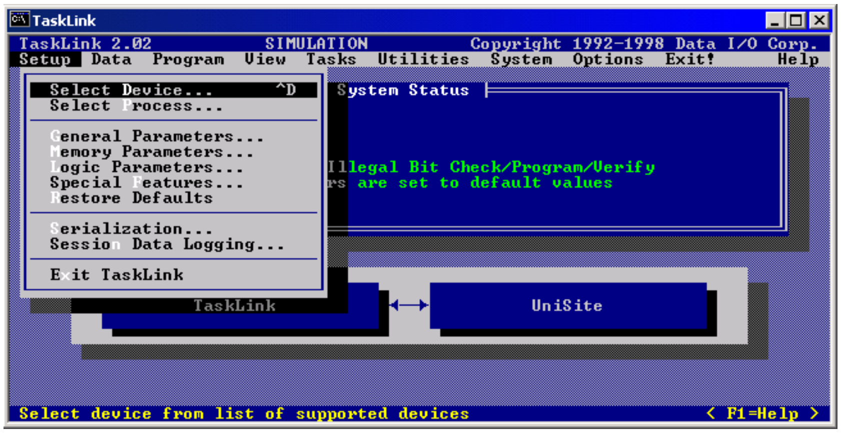
Figure 3-22
Programmer Type



```
TaskLink 2.02                                    2-1998 Data I/O Corp.
ProMaster 2500
AutoSite
UniSite
UniSite SetSite
2900 Programming System
3900 / 3980
PSX1000
PSX1000LC
PSX500
PSX400
Series 1000
Series 1000LC
288 32-pin EPROM Module
288 40-pin EPROM Module
288 87XX Module
Use arrows to highlight; Enter to select; Tab for next item      < F1=Help >
```

From the *Setup* menu, select **Select Device**.

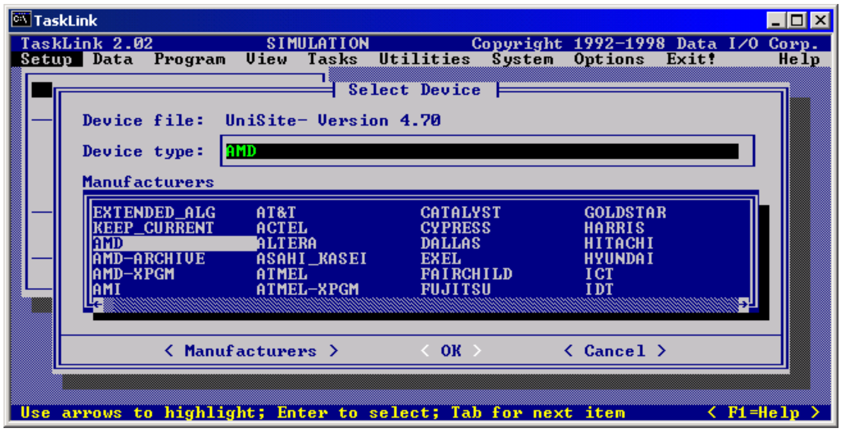
Figure 3-23
Setup Menu: Select Device



```
TaskLink
TaskLink 2.02             SIMULATION            Copyright 1992-1998 Data I/O Corp.
Setup  Data  Program  View  Tasks  Utilities  System  Options  Exit!        Help

  Select Device...      ^D   System Status
  Select Process...
  ---------------------------
  General Parameters...
  Memory Parameters...
  Logic Parameters...        Illegal Bit Check/Program/Verify
  Special Features...        rs are set to default values
  Restore Defaults
  ---------------------------
  Serialization...
  Session Data Logging...
  ---------------------------
  Exit TaskLink

            TaskLink        <-->        UniSite

Select device from list of supported devices                         < F1=Help >
```

Scroll through the list until the manufacturer of the device you want to use is highlighted, then click **OK**.

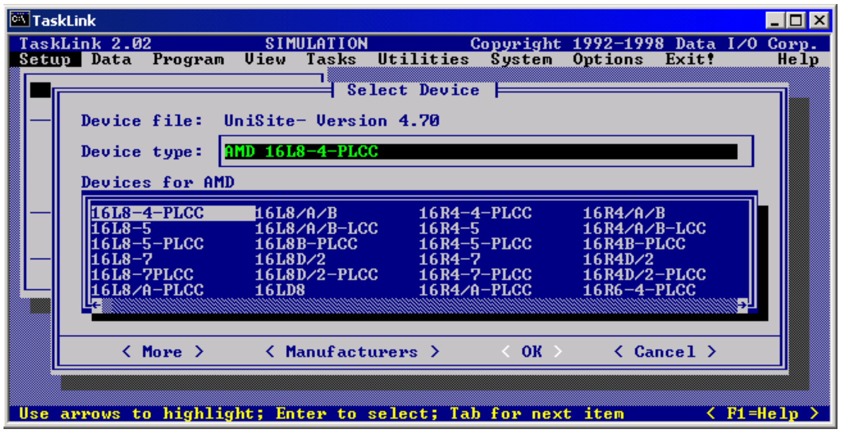
Figure 3-24
Select Manufacturer



```
TaskLink
TaskLink 2.02             SIMULATION            Copyright 1992-1998 Data I/O Corp.
Setup  Data  Program  View  Tasks  Utilities  System  Options  Exit!        Help

                      Select Device

   Device file:  UniSite- Version 4.70

   Device type:  AMD

   Manufacturers
    EXTENDED_ALG     AT&T          CATALYST      GOLDSTAR
    KEEP_CURRENT     ACTEL         CYPRESS       HARRIS
    AMD              ALTERA        DALLAS        HITACHI
    AMD-ARCHIVE      ASAHI_KASEI   EXEL          HYUNDAI
    AMD-XPGM         ATMEL         FAIRCHILD     ICT
    AMI              ATMEL-XPGM    FUJITSU       IDT

          < Manufacturers >        < OK >        < Cancel >

Use arrows to highlight; Enter to select; Tab for next item           < F1=Help >
```

Scroll through the list of device part numbers until the one you want is highlighted, then click **OK**.

Figure 3-25
Select Device



TaskLink
TaskLink 2.02 SIMULATION Copyright 1992-1998 Data I/O Corp.
Setup Data Program View Tasks Utilities System Options Exit! Help
Select Device
< More > < Manufacturers > < OK > < Cancel >
< F1=Help >

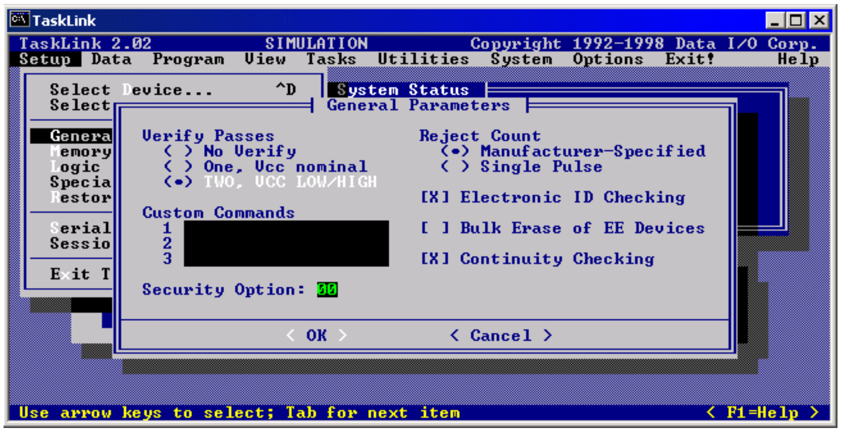
```
Device file:  UniSite- Version 4.70

Device type:  AMD 16L8-4-PLCC

Devices for AMD

16L8-4-PLCC   16L8/A/B        16R4-4-PLCC   16R4/A/B
16L8-5        16L8/A/B-LCC    16R4-5        16R4/A/B-LCC
16L8-5-PLCC   16L8B-PLCC      16R4-5-PLCC   16R4B-PLCC
16L8-7        16L8D/2         16R4-7        16R4D/2
16L8-7PLCC    16L8D/2-PLCC    16R4-7-PLCC   16R4D/2-PLCC
16L8/A-PLCC   16LD8           16R4/A-PLCC   16R6-4-PLCC

Use arrows to highlight; Enter to select; Tab for next item
```

Session 7: Setting Programmer Properties

After specifying the device to be programmed, the next step is setting the programmer properties.

From the *Setup* menu, select **General Parameters**. These parameters determine what device operations are performed before, during and after programming. To learn the parameter's function, highlight it and then press **F1**. The Help topic is displayed.

Figure 3-26
Setup Menu: General Parameters



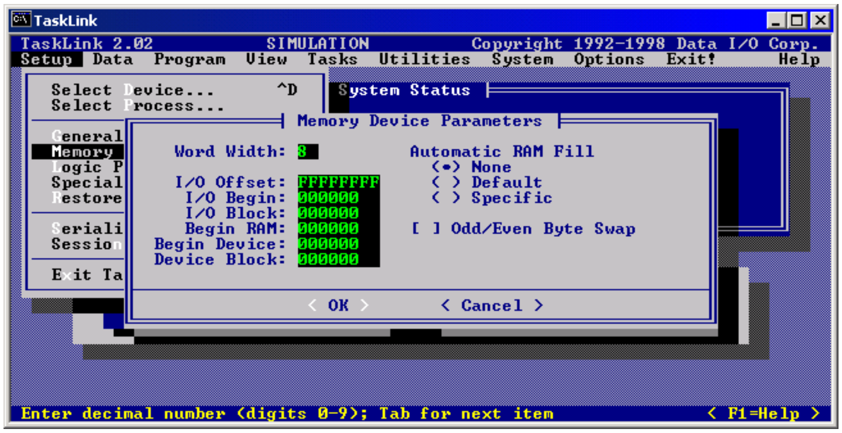
```
Verify Passes                    Reject Count
  ( ) No Verify                    (•) Manufacturer-Specified
  ( ) One, Vcc nominal             ( ) Single Pulse
  (•) Two, Vcc Low/High
                                 [X] Electronic ID Checking
Custom Commands
  1                              [ ] Bulk Erase of EE Devices
  2
  3                              [X] Continuity Checking

Security Option: 00

Use arrow keys to select; Tab for next item
```

From the *Setup* menu, select **Memory Parameters** to specify how and where data is loaded in programmer RAM.

To ensure a consistent sumcheck, you can preload all address locations into RAM with a known hexadecimal value (many data files do not contain data for every memory address). To do this, select **Specific** under **Automatic RAM Fill** and enter a value (usually **00** or **FF**) to be placed in RAM before a file is loaded.

To learn about **I/O Offset**, highlight it, and then press **F1** to display Help.

Figure 3-27
Memory Device Parameters



Memory Device Parameters

```
   Word Width: 8                 Automatic RAM Fill
                                  (•) None
   I/O Offset: FFFFFFFF           ( ) Default
    I/O Begin: 000000             ( ) Specific
    I/O Block: 000000
    Begin RAM: 000000            [ ] Odd/Even Byte Swap
 Begin Device: 000000
 Device Block: 000000
```

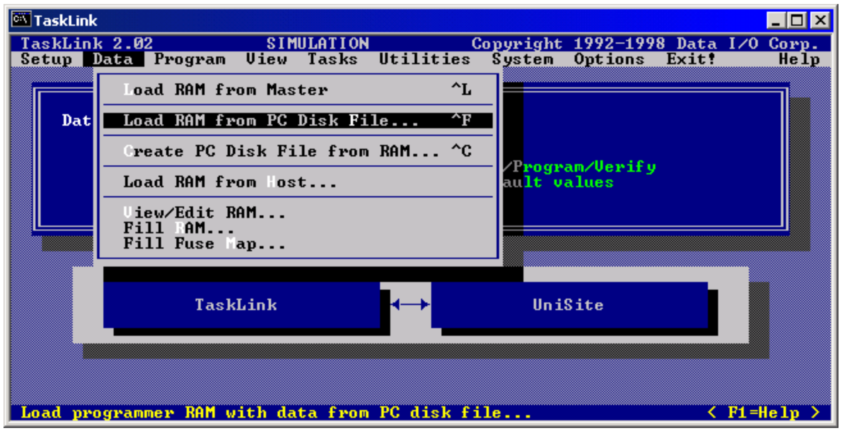
Enter decimal number (digits 0-9); Tab for next item

Session 8: Loading Data

After setting the programmer properties, the next step is to load programmer RAM with the data to be programmed in the device. In this example, the data is stored in a file on a disk.

From the *Data* menu, select **Load RAM from PC Disk File**.

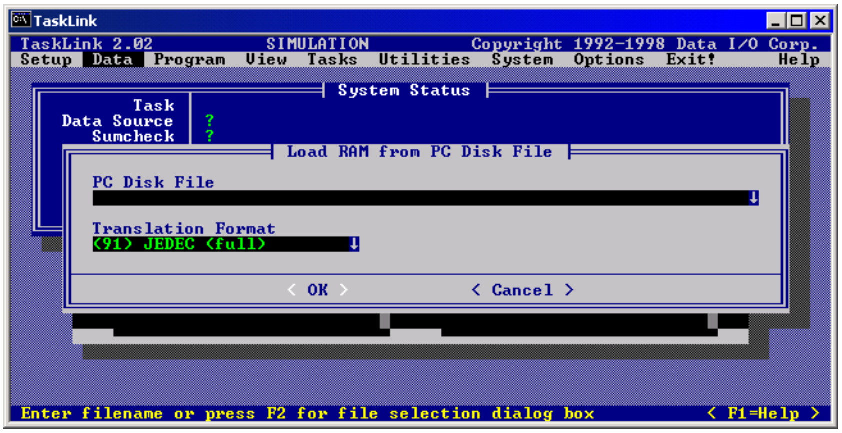
Figure 3-28
Data Menu: Load RAM



Load programmer RAM with data from PC disk file...

Type the name of the data file in the PC Disk File box or press **F2** to display the **Data File** selection box.

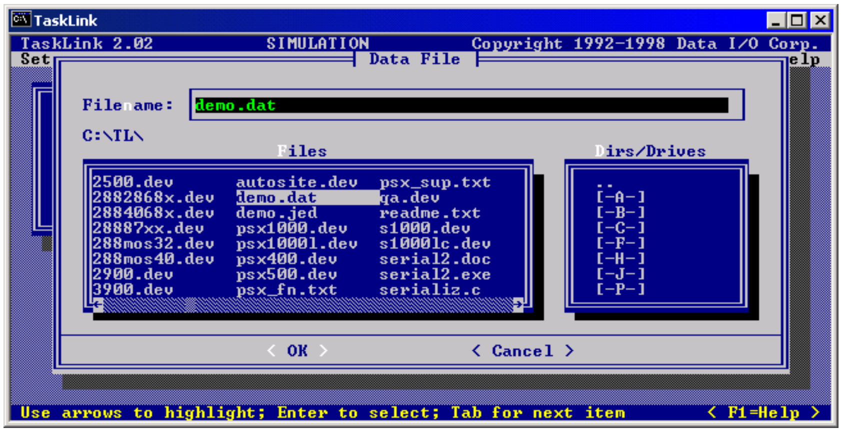
Figure 3-29
PC Disk File



```
TaskLink
TaskLink 2.02           SIMULATION          Copyright 1992-1998 Data I/O Corp.
Setup  Data  Program  View  Tasks  Utilities  System  Options  Exit!     Help
                         System Status
          Task
   Data Source   ?
      Sumcheck   ?
                      Load RAM from PC Disk File
    PC Disk File

    Translation Format
    (91) JEDEC (full)
                  < OK >                 < Cancel >
Enter filename or press F2 for file selection dialog box           < F1=Help >
```

In the Data File list box, search for and select a file from any drive accessible from your PC.

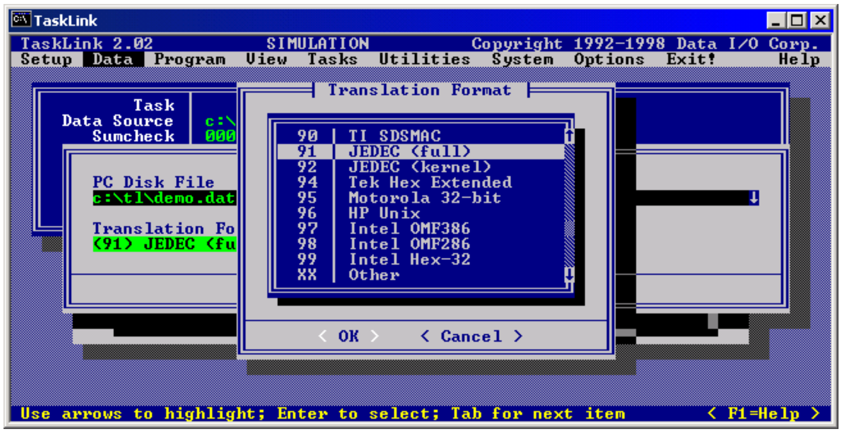
Figure 3-30
Select Data File



```
TaskLink
TaskLink 2.02           SIMULATION          Copyright 1992-1998 Data I/O Corp.
Set                          Data File                                    elp
   Filename:  demo.dat
   C:\TL\
                       Files                               Dirs/Drives
    2500.dev       autosite.dev   psx_sup.txt               ..
    2882868x.dev   demo.dat       qa.dev                    [-A-]
    2884068x.dev   demo.jed       readme.txt                [-B-]
    28887xx.dev    psx1000.dev    s1000.dev                 [-C-]
    288mos32.dev   psx10001.dev   s1000lc.dev               [-F-]
    288mos40.dev   psx400.dev     serial2.doc               [-H-]
    2900.dev       psx500.dev     serial2.exe               [-J-]
    3900.dev       psx_fn.txt     serializ.c                [-P-]
                  < OK >                 < Cancel >
Use arrows to highlight; Enter to select; Tab for next item        < F1=Help >
```

From the *Data* menu, select **Translation Format**. Highlight the format of the file you will load into RAM, then click **OK**. For more information about translation formats, see Chapter 6, “Translation Formats.”

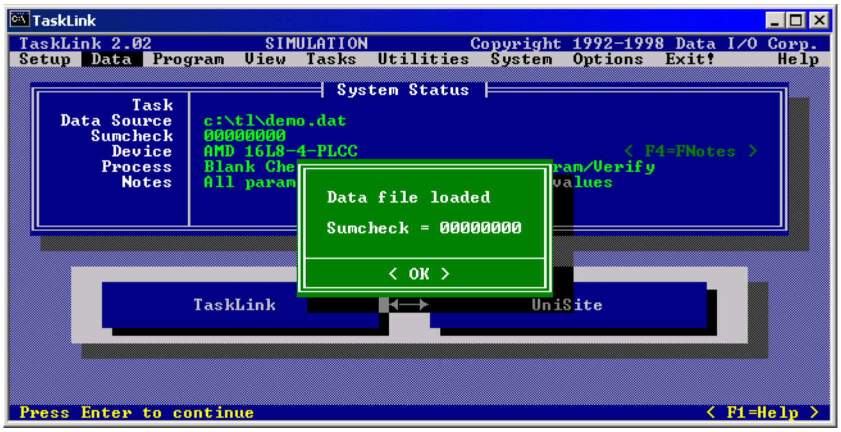
Figure 3-31
Translation Format



```
TaskLink 2.02              SIMULATION          Copyright 1992-1998 Data I/O Corp.

        Task
 Data Source   c:\
    Sumcheck   000

   PC Disk File
   c:\tl\demo.dat

   Translation Fo
   (91) JEDEC (fu

                    Translation Format

                 90 | TI SDSMAC
                 91 | JEDEC (full)
                 92 | JEDEC (kernel)
                 94 | Tek Hex Extended
                 95 | Motorola 32-bit
                 96 | HP Unix
                 97 | Intel OMF386
                 98 | Intel OMF286
                 99 | Intel Hex-32
                 XX | Other

Use arrows to highlight; Enter to select; Tab for next item        < F1=Help >
```

When the file is successfully loaded into programmer RAM, a message box displays the RAM sumcheck.

Figure 3-32
Sumcheck



```
TaskLink 2.02              SIMULATION          Copyright 1992-1998 Data I/O Corp.

                          System Status
        Task
 Data Source   c:\tl\demo.dat
    Sumcheck   00000000
      Device   AMD 16L8-4-PLCC                              < F4=FNotes >
     Process   Blank Che                  ram/Verify
       Notes   All param                  values

                   Data file loaded

                   Sumcheck = 00000000

Press Enter to continue                                            < F1=Help >
```

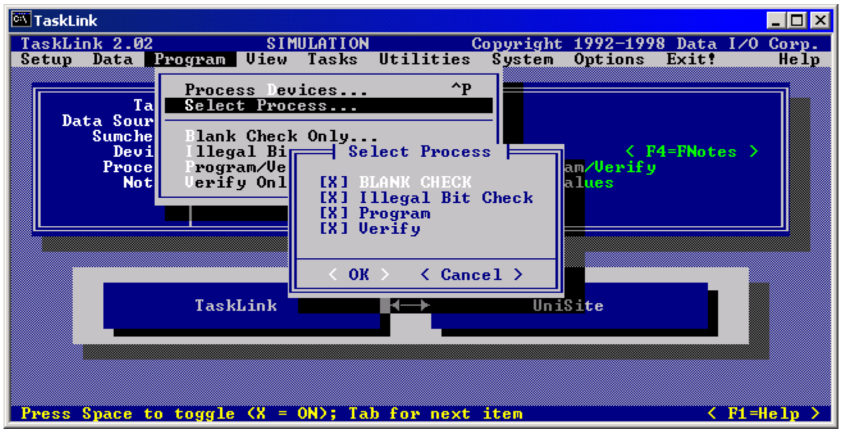
Note: Because TaskLink for DOS is running in Simulation Mode, the checksum displayed is 00000000. When connected to a programmer and loading real data, the checksum will display actual values.

Session 9: Selecting Processes

After loading data into programmer RAM, the next step is determining which processes the programmer will perform on the device.

From the *Program* menu, select **Select Process**. An **X** in the selection box indicates that process will be performed. When the appropriate processes are selected, click **OK**.

Figure 3-33
Selecting Processes



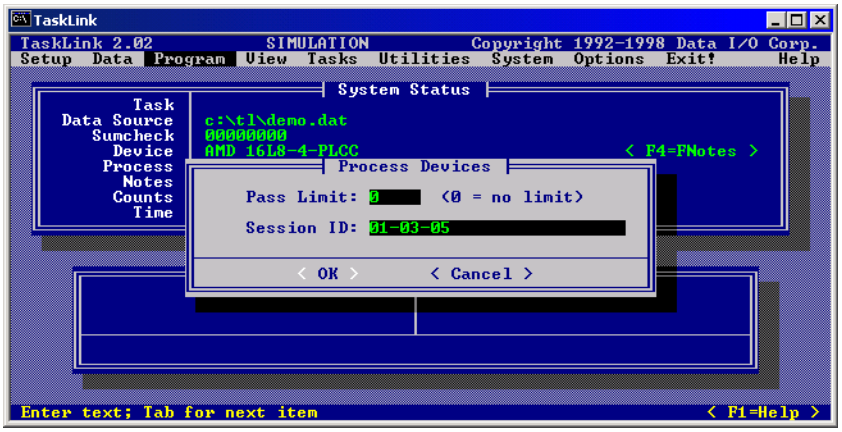
```
TaskLink
TaskLink 2.02              SIMULATION          Copyright 1992-1998 Data I/O Corp.
Setup  Data  Program  View  Tasks  Utilities  System  Options  Exit!      Help

            Process Devices...       ^P
            Select Process...

            Blank Check Only...
            Illegal Bi
            Program/Ve
            Verify Onl

                  ┤ Select Process ├
                  [X] BLANK CHECK
                  [X] Illegal Bit Check
                  [X] Program
                  [X] Verify

                   < OK >   < Cancel >

                               < F4=FNotes >
         TaskLink        ←→        UniSite

Press Space to toggle (X = ON); Tab for next item              < F1=Help >
```

Session 10: Programming a Device

From the *Program* menu, select **Process Devices**. In the **Pass Limit** field, enter the number of devices you want to program. To program an unlimited number of devices, set the value at “0.”

You may also enter a Session ID name or number. A Session ID is optional and helps when tracking statistics.

Figure 3-34
Process Devices

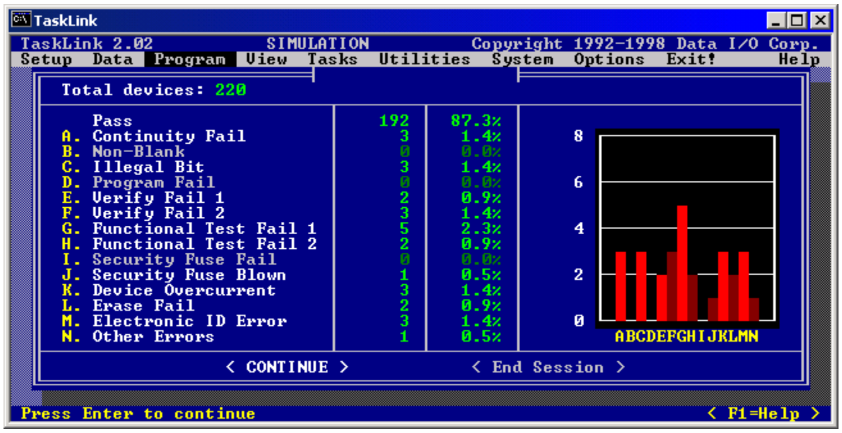


```
TaskLink
TaskLink 2.02              SIMULATION          Copyright 1992-1998 Data I/O Corp.
Setup  Data  Program  View  Tasks  Utilities  System  Options  Exit!      Help

                         ┤ System Status ├
          Task
   Data Source   c:\tl\demo.dat
      Sumcheck   00000000
        Device   AMD 16L8-4-PLCC                        < F4=FNotes >
       Process           ┤ Process Devices ├
         Notes
        Counts      Pass Limit: 0       (0 = no limit)
          Time
                    Session ID: 01-03-05

                       < OK >        < Cancel >

Enter text; Tab for next item                                  < F1=Help >
```

When actually programming a device, you will be prompted to insert the device to be programmed. For information about inserting devices, see “Install Devices” on page 2-30.

During processing, the number of successfully programmed devices and total number of attempts are displayed. The **Last Device** box displays “Pass” on a green background or a message that describes a failure on a red background.

Figure 3-35
Statistics



TaskLink
TaskLink 2.02 SIMULATION Copyright 1992-1998 Data I/O Corp.
Setup Data Program View Tasks Utilities System Options Exit! Help
8
6
4
2
0
ABCDEFGHIJKLMN
< CONTINUE >
< End Session >
Press Enter to continue
< F1=Help >

```
Total devices: 220

   Pass                        192   87.3%
A. Continuity Fail               3    1.4%
B. Non-Blank                     0    0.0%
C. Illegal Bit                   3    1.4%
D. Program Fail                  0    0.0%
E. Verify Fail 1                 2    0.9%
F. Verify Fail 2                 3    1.4%
G. Functional Test Fail 1        5    2.3%
H. Functional Test Fail 2        2    0.9%
I. Security Fuse Fail            0    0.0%
J. Security Fuse Blown           1    0.5%
K. Device Overcurrent            3    1.4%
L. Erase Fail                    2    0.9%
M. Electronic ID Error           3    1.4%
N. Other Errors                  1    0.5%
```

This completes the process of programming a device.

Tutorials for HiTerm (Terminal) Mode

Sessions 11-21 take you through the basic process of programming a device using HiTerm (terminal mode) as your user interface. Be sure to read “General Information About HiTerm” for help in understanding the main screen, moving the cursor, selecting commands, and accessing online help, for example.

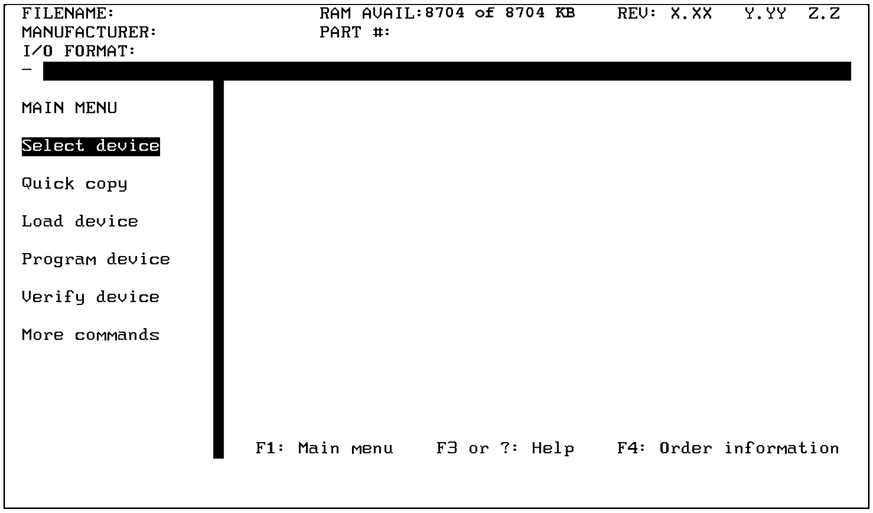
Note: HiTerm supports only serial mode operations.

Before you go any further in this chapter, make sure you have read and completed the instructions in Chapter 2, “Setup and Installation.” You should also have read Chapter 1, “Introduction.”

General Information About HiTerm

UniSite-*xpi* should be powered up and you should be looking at the Main Menu, shown in Figure 3-36.

Figure 3-36
The Main Menu

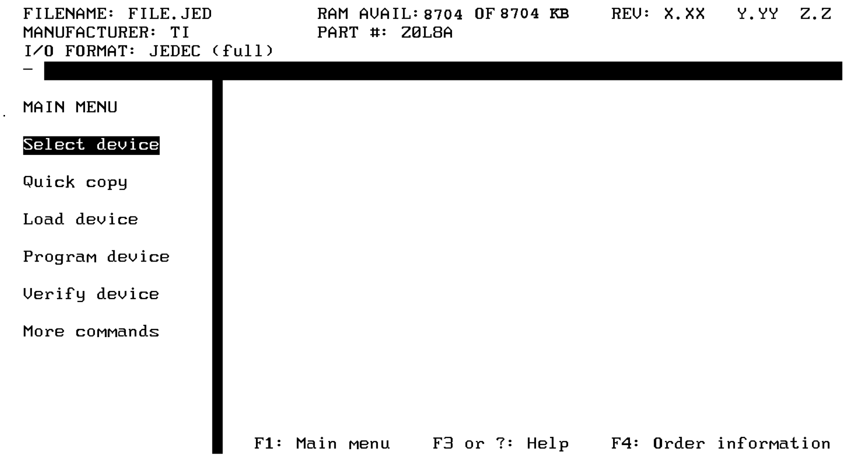


```
FILENAME:                    RAM AVAIL:8704 of 8704 KB    REV: X.XX   Y.YY  Z.Z
MANUFACTURER:                PART #:
I/O FORMAT:
-

MAIN MENU

Select device

Quick copy

Load device

Program device

Verify device

More commands


                      F1: Main menu    F3 or ?: Help    F4: Order information
```

The UniSite-*xpi* Screen

You will see a consistent format to the screens as you use UniSite-*xpi*. Most screens, such as the screen shown in Figure 2-23, are broken into five areas: the status window, the message bar, the command window, the dialog window, and the reminder bar.

Figure 3-37
Areas of the UniSite-xpi Screen



```
FILENAME: FILE.JED          RAM AVAIL:8704 OF 8704 KB    REV: X.XX   Y.YY  Z.Z
MANUFACTURER: TI            PART #: Z0L8A
I/O FORMAT: JEDEC (full)
-

MAIN MENU

Select device

Quick copy

Load device

Program device

Verify device

More commands


                  F1: Main menu    F3 or ?: Help    F4: Order information
```

Status Window

Occupying the top three lines of the screen, the status window displays important system information. The following are included in the status window:

- Name of the data file (FILE.JED in Figure 3-37)
- Amount of user RAM (8704 of 8704KB)
- Version numbers of the algorithm disk (X.XX), the system disk (Y.YY), and the system EPROM (Z.Z).
- Device manufacturer (TI) and part number (20L8A)
- Data translation format (JEDEC full)

Message Bar

Located just below the status window, the message bar displays system and error messages. The action symbol is also located in the message bar. The action symbol rotates while an operation is taking place, indicating that UniSite-*xpi* is busy.

Command Window

Occupying the left side of the screen is the command window. At the top of the window is the menu name, displayed in uppercase letters. Below the menu name, the available commands are displayed in upper- and lowercase letters.

Dialog Window

Occupying the largest area of the screen, the dialog window displays different information and system parameters, depending on the selected command.

Reminder Bar

Located at the bottom of the dialog window, the reminder bar tells you what function keys are available and what they will do if pressed.

Moving the Cursor

Pressing the arrow keys moves the cursor in the direction indicated on the key. The cursor will wrap around when it has reached the edge (top, bottom, left, right) of a window.

Selecting a Command

You can select a command using one of two methods: you can either press the first letter of the command, or move the cursor to the menu item and press **ENTER**.

Accessing Online Help

Online Help screens are available throughout UniSite-*xpi* and provide both general help and context-sensitive help. Context-sensitive help gives you help text that is specific to a particular field on the screen. For example, if you are in a parameter selection screen, each parameter on the screen has a different piece of help text associated with it.

Getting Help

To access help, move the cursor to the item you want help on and press either **F3** or **?**. The Help screen is divided into four sections: the key listing, the general help, the specific help, and the reminder bar.

See Figure 3-38 for a sample help screen.

Key Listing

Displayed at the top of the help screen, the key listing provides a quick summary of some of the most often used key commands. Key combinations displayed with a dash between them, such as

```
CTRL-P
```

indicate you should press and hold the first key; then press the second key. Key combinations displayed with a space between them, such as

```
ESC CTRL-T
```

indicate you should press and release the first key, then press the key combination.

General Help

Displayed in the middle of the screen, the general help text tells you about the next higher command or menu. Although the general help is not as specific as the context-sensitive help, it provides you with a good reference point.

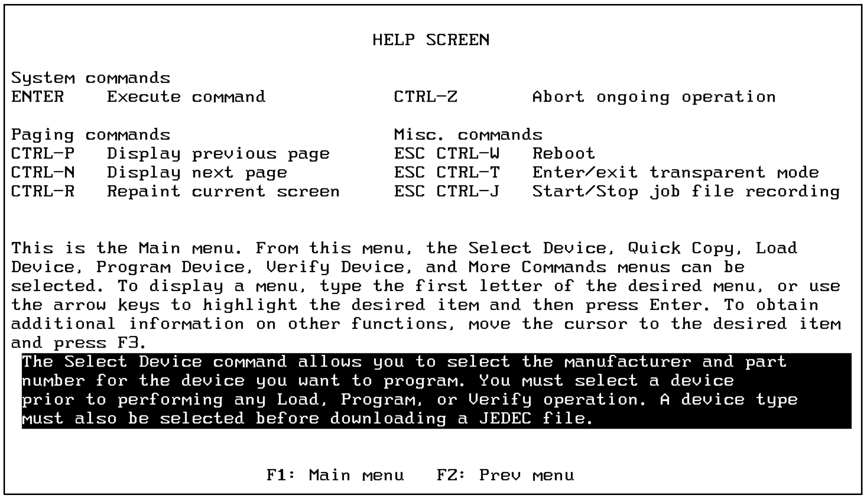
Context-sensitive Help

Displayed in reverse video, the context-sensitive help provides you with information about the specific item the cursor was on when you called the help function. The information in this field changes every time you move the cursor to a different location on the screen.

Reminder Bar

Displayed on the bottom line of the screen, the reminder bar shows which function keys you can use to exit Help.

Figure 3-38
Areas of the Help Screen



```
                             HELP SCREEN

System commands
ENTER    Execute command            CTRL-Z       Abort ongoing operation

Paging commands                     Misc. commands
CTRL-P   Display previous page      ESC CTRL-W   Reboot
CTRL-N   Display next page          ESC CTRL-T   Enter/exit transparent mode
CTRL-R   Repaint current screen     ESC CTRL-J   Start/Stop job file recording


This is the Main menu. From this menu, the Select Device, Quick Copy, Load
Device, Program Device, Verify Device, and More Commands menus can be
selected. To display a menu, type the first letter of the desired menu, or use
the arrow keys to highlight the desired item and then press Enter. To obtain
additional information on other functions, move the cursor to the desired item
and press F3.
 The Select Device command allows you to select the manufacturer and part
 number for the device you want to program. You must select a device
 prior to performing any Load, Program, or Verify operation. A device type
 must also be selected before downloading a JEDEC file.


                    F1: Main menu   F2: Prev menu
```

Accessing Online Help for System Messages

Online help is available for non-fatal system messages, which result from situations that do not interrupt UniSite-*xpi*'s operation. Fatal messages, which result from situations that do interrupt UniSite-*xpi*'s operation, are listed and described in Chapter 7, "Messages."

Non-fatal error messages are generally displayed in the message bar. To access the online help for the message, press **F3** or **?**. UniSite-*xpi* displays the online help for the message. Exit the message help screen as you would any help screen.

Accessing Device-Specific Online Information

After selecting a device, you will see the following message in the message bar if there is online device-specific information for the selected device.

```
Hit F3 or ? to view device specific message.
```

Press **F3** or **?** to display the device-specific information. If there is more than one screen of device-specific information, press **CTRL** + **N** to view the next screen of information. Exit the screen as you would any help screen.

Exiting Help

To exit any Help screen, press either **F1** or **F2**. **F1** returns you to the Main Menu and **F2** displays the previous screen from which the Help function was invoked.

Using Key Functions

Some of UniSite-*xpi*'s functions may be performed by pressing a key or a combination of keys. When using the CTRL key, hold it down and then momentarily press the second key. The key functions are listed below with their corresponding keystroke sequence.

Keystrokes	Description
F1	Return to the Main Menu
F2	Go to the previous menu
F3 or **?**	Display online help for the current menu and cursor position
F4	Display the Optional Parameters screen
ENTER	Execute highlighted command
SPACE	Toggle a parameter
CTRL + **N**	Display next page
CTRL + **P**	Display previous page
CTRL + **R**	Repaint screen
CTRL + **Z**	Halt current operation
ESC CTRL + **T**	Enter / exit transparent mode with host computer
ESC CTRL + **J**	Start / stop job file recording
ESC CTRL + **W**	Restart UniSite-*xpi* (warm boot)

Review of Programming Concepts

Below is a brief description of the basic concepts involved in programming a device.

Select Device

Before attempting most operations, you must first tell UniSite-*xpi* what device you will be working with. UniSite-*xpi* then selects the appropriate programming algorithm based on the manufacturer and device part number selected. Once you have selected a device, you can move on to other device operations, such as loading, programming, and verifying.

Load Data

Load operations determine how you move device data into UniSite-*xpi*. You can load data from a device, from one of UniSite-*xpi*'s internal disk drives, or from one of the two serial ports on UniSite-*xpi* (for example, from the Remote port).

Program Device

Programming transfers the device data into the device inserted in UniSite-*xpi*. The programming is done according to the programming algorithm selected by the Select Device command. The programming operation also includes a verify operation, which is described below.

Verify Device

A verify operation compares the data in a programmed device against the data in UniSite-*xpi*'s RAM or against the data in a disk file. In the case of logic devices, verifying can also include functional testing. Verify is normally an automatic part of the program operation, but additional verify operations can provide useful information about many programming errors.

Session 11: Navigating Through the Menus

This Session describes the organization of UniSite-*xpi*'s menus and gives you an opportunity to explore the menu tree and the online help.

Select a Menu Item

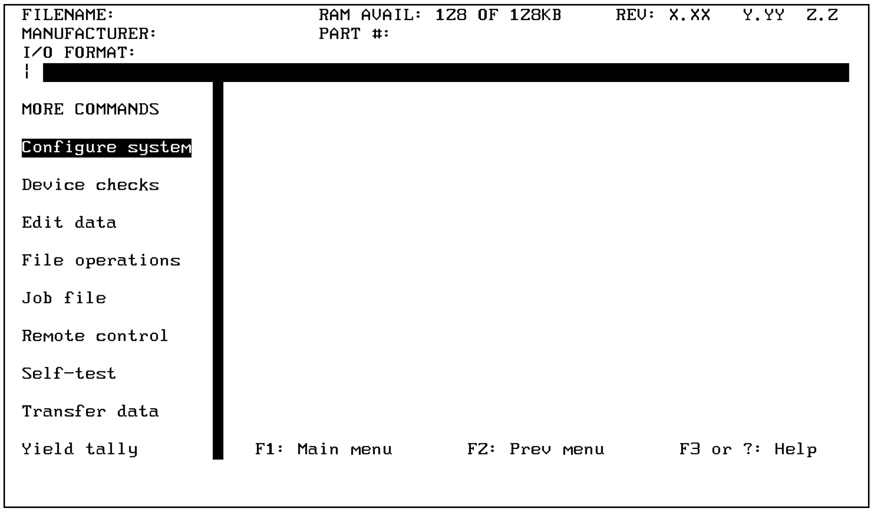
First, let's go over how you select an entry from the command window. (Remember, the command window is on the left side of the vertical bar.)

To select a command, either move the cursor to the menu item and press **ENTER**, or press the first letter of the menu item.

For example, if you want to test your UniSite-*xpi*, select the Self-test command on the More Commands menu. To access the More Commands menu, move the cursor to the More Commands menu item and press **ENTER**. Or, as mentioned above, simply press **M**.

The More Commands menu should appear. It should look like the screen shown in Figure 3-39.

Figure 3-39
The More Commands Menu Screen



```
FILENAME:                    RAM AVAIL: 128 OF 128KB     REV: X.XX   Y.YY  Z.Z
MANUFACTURER:                PART #:
I/O FORMAT:
¦

MORE COMMANDS

Configure system

Device checks

Edit data

File operations

Job file

Remote control

Self-test

Transfer data

Yield tally         F1: Main menu       F2: Prev menu       F3 or ?: Help
```

If a different menu appears, don't worry. Press **F1** and try again.

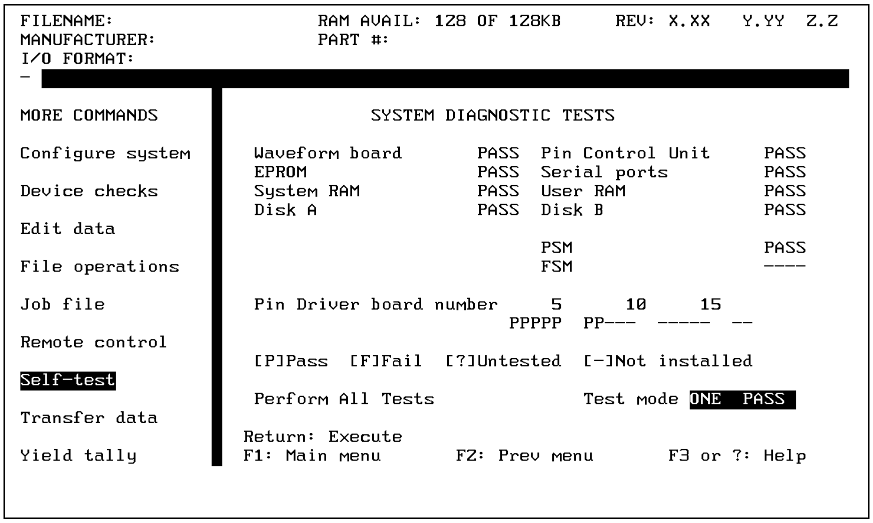
Scan the menu, which is arranged alphabetically, and find the Self-test menu item. To access the Self-test, press ↓ to move the cursor to the Self-test menu item and then press **ENTER** to execute the command. (Remember, you could also have pressed the first letter of the menu item to select that command. In this case, you would press **S**.)

The dialog window should fill with the Self-test screen, and the top of the window should read

```
SYSTEM DIAGNOSTIC TESTS
```

You have reached the Self-test screen, which is shown in Figure 3-40.

Figure 3-40
The Self-test Screen

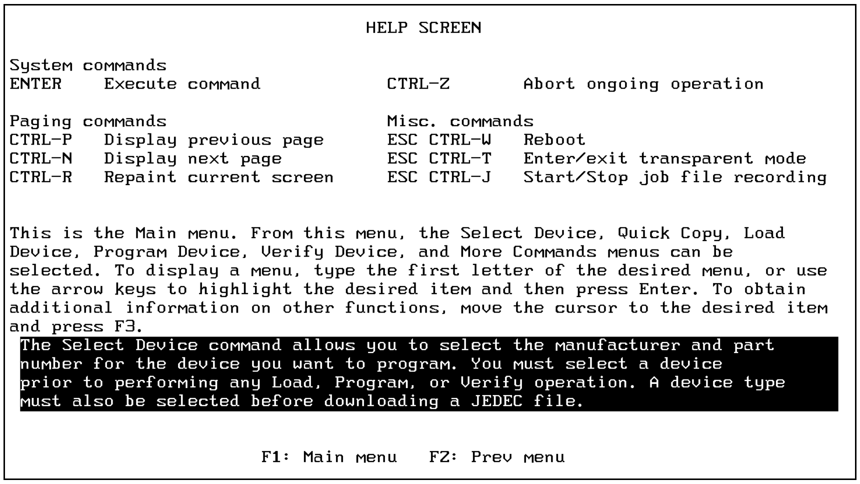


```
FILENAME:                    RAM AVAIL: 128 OF 128KB     REV: X.XX   Y.YY  Z.Z
MANUFACTURER:                PART #:
I/O FORMAT:
-

MORE COMMANDS                         SYSTEM DIAGNOSTIC TESTS

Configure system      Waveform board       PASS  Pin Control Unit     PASS
                      EPROM                PASS  Serial ports         PASS
Device checks         System RAM           PASS  User RAM             PASS
                      Disk A               PASS  Disk B               PASS
Edit data
                                                 PSM                  PASS
File operations                                  FSM                  ----

Job file              Pin Driver board number     5      10     15
                                               PPPPP  PP---  -----  --
Remote control
                      [P]Pass  [F]Fail  [?]Untested  [-]Not installed
Self-test
                      Perform All Tests             Test mode ONE  PASS
Transfer data
                     Return: Execute
Yield tally          F1: Main menu       F2: Prev menu       F3 or ?: Help
```

Select Online Help

To access the context-sensitive online help for the item the cursor is highlighting, press either **F3** or **?**. The entire screen clears and fills with the Help screen. The top third of the Help screen is a list of commonly used keys, the next third is general help, and the last third is the context-sensitive help.

Figure 3-41
Areas of the Help Screen



```
                                  HELP SCREEN

System commands
ENTER    Execute command             CTRL-Z      Abort ongoing operation

Paging commands                      Misc. commands
CTRL-P   Display previous page       ESC CTRL-W   Reboot
CTRL-N   Display next page           ESC CTRL-T   Enter/exit transparent mode
CTRL-R   Repaint current screen      ESC CTRL-J   Start/Stop job file recording


This is the Main menu. From this menu, the Select Device, Quick Copy, Load
Device, Program Device, Verify Device, and More Commands menus can be
selected. To display a menu, type the first letter of the desired menu, or use
the arrow keys to highlight the desired item and then press Enter. To obtain
additional information on other functions, move the cursor to the desired item
and press F3.
 The Select Device command allows you to select the manufacturer and part
 number for the device you want to program. You must select a device
 prior to performing any Load, Program, or Verify operation. A device type
 must also be selected before downloading a JEDEC file.


                         F1: Main menu   F2: Prev menu
```

Context-sensitive Help

Context-sensitive help means that the text displayed on the screen changes every time you move the cursor to a different field. If the cursor is in the Disk field, you get information on the Disk; if the cursor is in the RAM field, you get information on the RAM.

To leave the help screen and return to the Self-test screen, press **F2**. (Remember, if you forget what key to press, look at the reminder bar on the bottom of the screen for a quick reminder.) You should now be looking at the Self-test screen.

To illustrate that the context-sensitive help changes every time you change the cursor position, get help for two different fields on the Self-test screen. For example, move the cursor to the RAM field and access online help. (Remember, press **F3** or **?** to get help and **F2** to exit the help screen.)

Next, move the cursor to the Disk field and access the online help. (Remember, press **F3** or **?**.)

Notice that the context-sensitive help (the text in reverse video on the lower third of the screen) changed. The general help (the text in normal video on the middle third of the screen) didn't change because you are still on the Self-test screen. The general help changes when you move to another menu level. For example, leaving the Self-test screen and returning to the More Commands menu causes the general help to change.

To change the general help, press **F2** until the flashing cursor returns to the command window. Then access the online help. Notice that the general help has changed.

Exit the Online Help

In the previous section, you pressed **F2** to return to the Self-test screen from the help screen. Pressing **F2** takes you one step closer to the Main Menu.

If you had wanted to return to the Main Menu quickly, you could have pressed **F1** instead.

Let's try it. If you are not already at the Main Menu, press **F1** to return to the Main Menu. Press **M** and then **S** to access the Self-test screen. From here, you can either press **F2** twice to return to the Main Menu, or you can press **F1** once to return to the Main Menu. Step through the menus until you are comfortable. (Access the online help if you want to.)

When finished, return to the Main Menu.

This completes your tour of UniSite-*xpi*'s user interface and online help system.

Review

In this Session you learned how to navigate through the UniSite-*xpi* interface.

Select menu items by either pressing the first letter of the command or moving the cursor to the command and pressing **ENTER**.

To access previous screens, press **F2**. To return to the Main Menu, press **F1**.

Finally, you learned how to access online help. To access the online help screens, press either **F3** or **?**.

Session 12: Selecting a Device

This Session describes how to tell UniSite-*xpi* the manufacturer and part number of the device you are using. The device selection process is a two-step process: first you select the device manufacturer, and then you select the device part number.

Read this Session if you are going to select a device that is supported by an algorithm included on one of the UniSite-*xpi* Algorithm disks. The next Session, “Selecting a Keep Current Algorithm,” covers how to select a device that is supported by a Keep Current algorithm.

For more information on the Keep Current Subscription Service, see the documentation behind the Keep Current tab at the back of this binder.

Before You Begin

You should have completed Session 1, which introduces you to the UniSite-*xpi* interface. Also, make sure that you are at the Main Menu before you start. (Press **F1** to return to the Main Menu.)

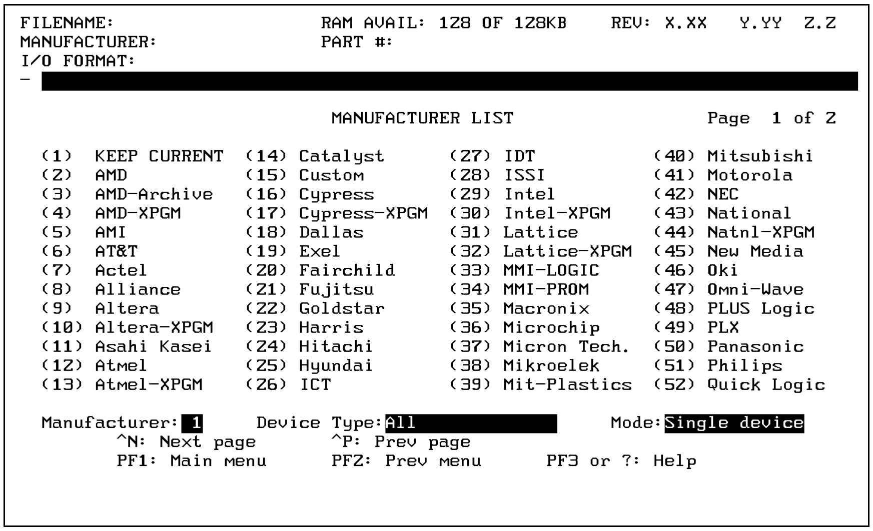
Can I Use Another Device?

If you do not have an AMD 27256 (the device we are using for this Tutorial), substitute the manufacturer and part number of the device you want to use. Keep in mind that the device you are using might not have the same capabilities as the AMD 27256. For example, the AMD 27256 supports Electronic ID while the Hitachi 27256 does not. Also, be sure to insert the correct algorithm disk.

Select a Manufacturer

Insert Algorithm Disk 2 (for memory or Emicro devices) into drive B.

Choose the Select Device command from the Main Menu. (You can either press **S** or move the cursor to the Select Device menu item and press **ENTER**.) The Manufacturer List screen appears and should look like the screen shown in Figure 3-42.



```
FILENAME:                    RAM AVAIL: 128 OF 128KB    REV: X.XX   Y.YY  Z.Z
MANUFACTURER:                PART #:
I/O FORMAT:
-

                              MANUFACTURER LIST                 Page  1 of 2

  (1)  KEEP CURRENT (14) Catalyst     (27) IDT          (40) Mitsubishi
  (2)  AMD          (15) Custom       (28) ISSI         (41) Motorola
  (3)  AMD-Archive  (16) Cypress      (29) Intel        (42) NEC
  (4)  AMD-XPGM     (17) Cypress-XPGM (30) Intel-XPGM   (43) National
  (5)  AMI          (18) Dallas       (31) Lattice      (44) Natnl-XPGM
  (6)  AT&T         (19) Exel         (32) Lattice-XPGM (45) New Media
  (7)  Actel        (20) Fairchild    (33) MMI-LOGIC    (46) Oki
  (8)  Alliance     (21) Fujitsu      (34) MMI-PROM     (47) Omni-Wave
  (9)  Altera       (22) Goldstar     (35) Macronix     (48) PLUS Logic
  (10) Altera-XPGM  (23) Harris       (36) Microchip    (49) PLX
  (11) Asahi Kasei  (24) Hitachi      (37) Micron Tech. (50) Panasonic
  (12) Atmel        (25) Hyundai      (38) Mikroelek    (51) Philips
  (13) Atmel-XPGM   (26) ICT          (39) Mit-Plastics (52) Quick Logic

  Manufacturer: 1     Device Type:All                Mode:Single device
         ^N: Next page       ^P: Prev page
         PF1: Main menu      PF2: Prev menu      PF3 or ?: Help
```

Figure 3-42
The Device Manufacturer Selection Screen

Examine the list of manufacturers and notice that they are listed alphabetically. Also, notice that some manufacturers are listed by their commonly used abbreviations. For example, Advanced Micro Devices is listed as AMD and Texas Instruments is listed as TI.

Expanding Your Options

Look at the manufacturer screen again. Specifically, look at the upper-right corner of the screen and notice the text that looks similar to the following:

```
Page 1 of 2
```

This tells you how many screens (pages) of manufacturers there are and what page you are on. To go to the previous page of manufacturers, press **CTRL** + **P**. To go to the next page of manufacturers, press **CTRL** + **N**. When the screen repaints, you see a new list of manufacturers, and the page counter now reads

```
Page 2 of 2
```

Keep paging through the screens until you find the manufacturer you are looking for. For this Session, we will be using a device from AMD. Page through the manufacturer listing until you see the entry for AMD.

The Device Type Filter

The Device Type filter allows you to select which device types you want displayed. Press **SPACE** to cycle through the three settings: All, Memory & Emicros, or Logic Only. When you select a manufacturer, UniSite-*xpi* displays only those devices that fit the filter you selected.

For this Session, we will be using a 27256, which is an EPROM. Move the cursor to the Device Type field, press **SPACE** to cycle through the device types until Memory & Emicros appears in the field. Notice how the display changes when you cycle from one filter to another.

The Programming Mode Filter

The Mode filter toggles UniSite-*xpi* between Single Device programming and Gang / Set programming. Gang programming is the programming of a single data file into multiple devices at once. Set programming is the partitioning and programming of a single large data file into multiple devices.

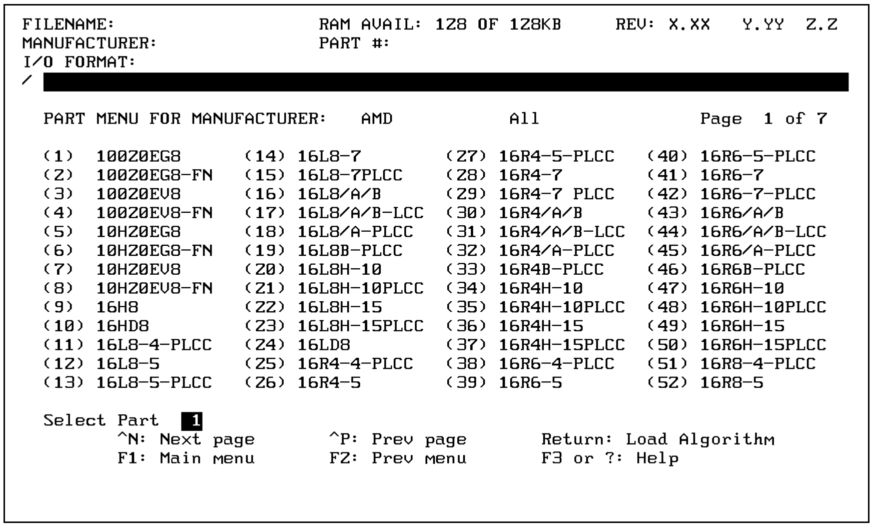
Move the cursor to the Mode field and press **SPACE**. Watch what happens to the Manufacturer List when you toggle from Single to Gang / Set: UniSite-*xpi* displays fewer manufacturers. This is because some of the manufacturers displayed in Single Mode do not make devices that can be programmed in Gang or Set Mode. As with the Device Type field, UniSite-*xpi* only displays the manufacturers that make devices that fit the currently selected filters.

For this Session, we will be programming a single 27256 EPROM; so, set the Mode filter to Single. If we wanted to make several copies of the EPROM, you would set the Mode filter to Gang / Set, but for this Session, set the Mode filter to Single.

Selecting the Manufacturer

For this Session, we will be using an AMD 27256 DIP EPROM. To select AMD from the Manufacturer List, move the cursor to the Manufacturer field, enter the number shown to the left of the name, and press **ENTER**.

After selecting the manufacturer, the Part Menu should appear and should look like the screen shown in Figure 3-43.



```
FILENAME:                    RAM AVAIL: 128 OF 128KB     REV: X.XX   Y.YY  Z.Z
MANUFACTURER:                PART #:
I/O FORMAT:
/

  PART MENU FOR MANUFACTURER:   AMD           All              Page  1 of 7

  (1)  10020EG8      (14) 16L8-7       (27) 16R4-5-PLCC   (40) 16R6-5-PLCC
  (2)  10020EG8-FN   (15) 16L8-7PLCC   (28) 16R4-7        (41) 16R6-7
  (3)  10020EV8      (16) 16L8/A/B     (29) 16R4-7 PLCC   (42) 16R6-7-PLCC
  (4)  10020EV8-FN   (17) 16L8/A/B-LCC (30) 16R4/A/B      (43) 16R6/A/B
  (5)  10H20EG8      (18) 16L8/A-PLCC  (31) 16R4/A/B-LCC  (44) 16R6/A/B-LCC
  (6)  10H20EG8-FN   (19) 16L8B-PLCC   (32) 16R4/A-PLCC   (45) 16R6/A-PLCC
  (7)  10H20EV8      (20) 16L8H-10     (33) 16R4B-PLCC    (46) 16R6B-PLCC
  (8)  10H20EV8-FN   (21) 16L8H-10PLCC (34) 16R4H-10      (47) 16R6H-10
  (9)  16H8          (22) 16L8H-15     (35) 16R4H-10PLCC  (48) 16R6H-10PLCC
  (10) 16HD8         (23) 16L8H-15PLCC (36) 16R4H-15      (49) 16R6H-15
  (11) 16L8-4-PLCC   (24) 16LD8        (37) 16R4H-15PLCC  (50) 16R6H-15PLCC
  (12) 16L8-5        (25) 16R4-4-PLCC  (38) 16R6-4-PLCC   (51) 16R8-4-PLCC
  (13) 16L8-5-PLCC   (26) 16R4-5       (39) 16R6-5        (52) 16R8-5

  Select Part  1
         ^N: Next page       ^P: Prev page       Return: Load Algorithm
         F1: Main menu       F2: Prev menu       F3 or ?: Help
```

Figure 3-43
The Part Number Selection Screen

Select a Device Part Number

Selecting a device part number is the same as selecting a device manufacturer: find the item in the menu and enter the number beside the item. In this case we are looking for a 27256.

Where Is It?

Notice that the screen is full of part numbers, and that the part numbers are arranged alphanumerically. If you examine the screen, you might find that the 27256 is not listed. This is because all the supported devices for this manufacturer could not be shown on one screen.

Expanding Your Options

In the upper-right corner of the screen you will see text that looks similar to the following:

```
Page 1 of 5
```

This tells you how many screens (pages) of devices there are and what page you are on. To go to the next page, press **CTRL** + **N**. When the screen repaints, you see a new list of devices, and the page counter now reads

```
Page 2 of 5
```

Keep paging through the screens until you find the 27256 part number.

Reviewing Your Options

If you scroll too far forward, and want to look at a previous screen, press **CTRL** + **P** until you are back to where you want to be.

Found It

After finding the 27256 on the part menu, type the number enclosed in parenthesis. For example, in Figure 3-43, you would type **7** and press **ENTER**.

UniSite-*xpi* loads the algorithm for the device you selected, in this case an AMD 27256. While UniSite-*xpi* is loading the algorithm, the action symbol rotates. (Remember, the action symbol is located at the left end of the message bar at the top of the screen.)

Done

When UniSite-*xpi* has loaded the programming algorithm for the device you have selected, UniSite-*xpi* will return to the Main Menu. The manufacturer and part number of the device you selected will appear in the status window. At this point, you will see only screens related to the type of device you selected. For example, if you choose a logic device, you will see only screens that are required to load, program, edit, and verify a logic device.

Accessing Device-specific Online Information

Some devices have special information relating to their use and programming. The special information is listed in the Device List as footnotes and is also available online. If you select a device that has footnotes or other device-specific information, UniSite-*xpi* displays the footnote after you select the device, as long as the Display Device Footnote parameter is set to Yes (default).

For this Session, the device you have selected does not require any online information. However, if you had selected a Lattice 22V10, you would have seen a device footnote for that device.

Review

Selecting a device is a two-step process.

- First, select the manufacturer of the device.
- Second, select the part number of the device.

Some devices have special information related to their use and programming. The information is available online and can be viewed after a device is selected.

Session 13: Selecting a Keep Current Algorithm

This Session describes how to select a device that is supported by a Keep Current algorithm you downloaded from the Data I/O Web site. The device selection process is a two-step process: first UniSite-*xpi* displays a list of the available Keep Current algorithms, and then you select the algorithm.

Read this Session if you are going to select a device that is supported by a Keep Current algorithm. The previous Session, “Selecting a Device,” covers how to select a device that is supported by an algorithm that is included on the UniSite-*xpi* Algorithm disk.

Before You Begin

You should have completed Session 1, which introduces the UniSite-*xpi* interface. Make sure that you are at the Main Menu before you start. (Press **F1** to return to the Main Menu.)

You should have placed your Keep Current algorithm(s) on a 3.5-inch disk formatted in your UniSite-*xpi*. See the Keep Current documentation for more information.

Can I Use Another Device?

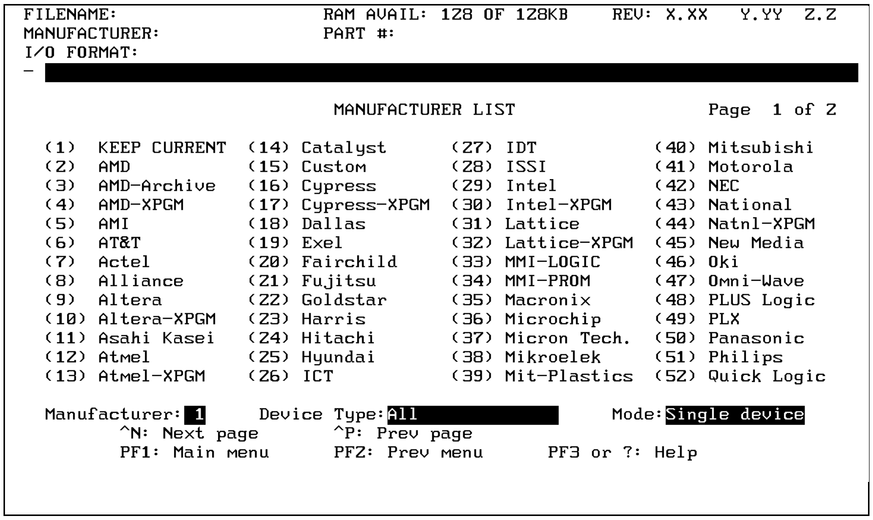
If you have not downloaded a Keep Current algorithm for your UniSite-*xpi*, you will not be able to complete this Session. However, you can still read the remainder of the Session to learn more about how to select a Keep Current algorithm. Or, you can skip this Session and continue with Session 4.

If you select a device supported by a Keep Current algorithm, keep in mind that the device you are using might not have the same capabilities as the device we are using for this Tutorial.

Select the Keep Current Option

First, choose the Select Device command from the Main Menu. (Either press **S** or move the cursor to the Select Device menu item and press **ENTER**.) The Manufacturer List screen appears, as shown in Figure 3-44.

Figure 3-44
The Device Manufacturer Selection Screen



```
FILENAME:                       RAM AVAIL: 128 OF 128KB    REV: X.XX   Y.YY  Z.Z
MANUFACTURER:                   PART #:
I/O FORMAT:
-

                               MANUFACTURER LIST                  Page  1 of 2

  (1)  KEEP CURRENT (14) Catalyst      (27) IDT           (40) Mitsubishi
  (2)  AMD          (15) Custom        (28) ISSI          (41) Motorola
  (3)  AMD-Archive  (16) Cypress       (29) Intel         (42) NEC
  (4)  AMD-XPGM     (17) Cypress-XPGM  (30) Intel-XPGM    (43) National
  (5)  AMI          (18) Dallas        (31) Lattice       (44) Natnl-XPGM
  (6)  AT&T         (19) Exel          (32) Lattice-XPGM  (45) New Media
  (7)  Actel        (20) Fairchild     (33) MMI-LOGIC     (46) Oki
  (8)  Alliance     (21) Fujitsu       (34) MMI-PROM      (47) Omni-Wave
  (9)  Altera       (22) Goldstar      (35) Macronix      (48) PLUS Logic
  (10) Altera-XPGM  (23) Harris        (36) Microchip     (49) PLX
  (11) Asahi Kasei  (24) Hitachi       (37) Micron Tech.  (50) Panasonic
  (12) Atmel        (25) Hyundai       (38) Mikroelek     (51) Philips
  (13) Atmel-XPGM   (26) ICT           (39) Mit-Plastics  (52) Quick Logic

  Manufacturer: 1     Device Type:All                  Mode:Single device
         ^N: Next page       ^P: Prev page
         PF1: Main menu      PF2: Prev menu      PF3 or ?: Help
```

Examine the list of manufacturers and locate the KEEP CURRENT entry. All the Keep Current algorithms are located under the KEEP CURRENT menu.

Expanding Your Options

In the upper-right corner of the Manufacturer List screen you will see text that looks similar to the following:

```
Page 1 of 2
```

This tells you how many screens (pages) of manufacturers there are and what page you are on. If you do not see the KEEP CURRENT entry, press **CTRL** + **N** to display the next page of manufacturers. (Press **CTRL** + **P** to display the previous page of manufacturers.) When the screen repaints, you will see a new list of manufacturers and the page counter will read

```
Page 2 of 2
```

Page through the screens until you find the KEEP CURRENT entry.

The Device Type Filter

The Device Type filter has no effect on the devices displayed when you select Keep Current devices.

The Programming Mode Filter

The Mode filter has no effect on the devices displayed when you select Keep Current devices. However, setting the Mode filter to Gang/Set will enable gang/set programming for devices that support that programming mode. Press **SPACE** to toggle between Single Device operations and Gang/Set operations.

Insert the Keep Current Algorithm Disk

Before you select the KEEP CURRENT entry, make sure the 3.5-inch disk containing the Keep Current algorithm(s) is inserted in drive B.

Selecting the Manufacturer

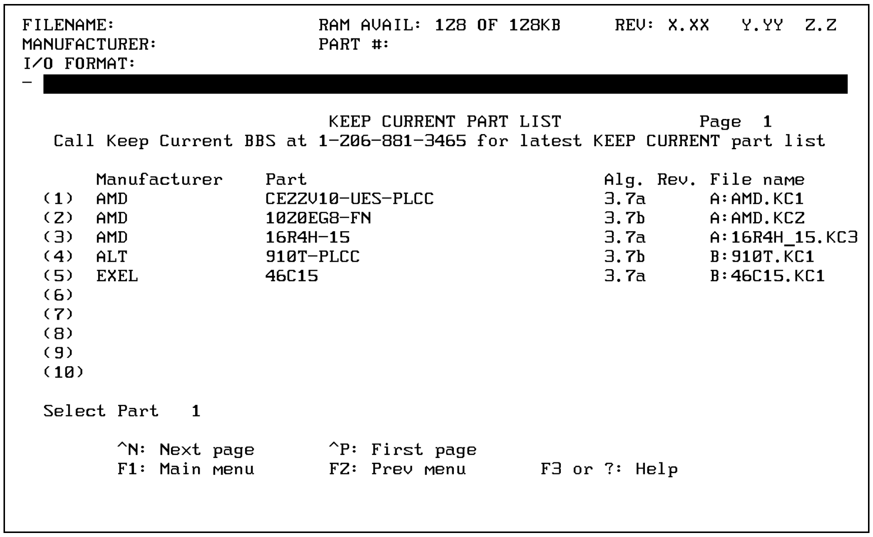
For this Session, we will be selecting a Keep Current algorithm that you have downloaded from the Data I/O Web site. Regardless of manufacturer, all Keep Current algorithms are located under the KEEP CURRENT entry.

To select KEEP CURRENT from the Manufacturer List, move the cursor to the Manufacturer field, type the number shown to the left of the entry and press **ENTER**. For example, in Figure 3-44 you would type **2** and press **ENTER**.

What Do You See?

After you select the KEEP CURRENT entry, the Keep Current Part List screen will appear, as shown in Figure 3-45.

Figure 3-45
The Keep Current Part List



```
FILENAME:                     RAM AVAIL: 128 OF 128KB     REV: X.XX   Y.YY  Z.Z
MANUFACTURER:                 PART #:
I/O FORMAT:
-

                              KEEP CURRENT PART LIST             Page  1
   Call Keep Current BBS at 1-206-881-3465 for latest KEEP CURRENT part list

       Manufacturer    Part                            Alg. Rev. File name
  (1)  AMD             CE22V10-UES-PLCC                3.7a      A:AMD.KC1
  (2)  AMD             1020EG8-FN                      3.7b      A:AMD.KC2
  (3)  AMD             16R4H-15                        3.7a      A:16R4H_15.KC3
  (4)  ALT             910T-PLCC                       3.7b      B:910T.KC1
  (5)  EXEL            46C15                           3.7a      B:46C15.KC1
  (6)
  (7)
  (8)
  (9)
  (10)

  Select Part   1

         ^N: Next page       ^P: First page
         F1: Main menu       F2: Prev menu       F3 or ?: Help
```

If you see the Keep Current Part List screen, continue with the section titled “Select the Keep Current Algorithm.”

If you do not see the Keep Current Part List screen, you are probably looking at a screen that describes the Keep Current service. UniSite-*xpi* will not display the Keep Current Part List screen if you do not have any Keep Current algorithms.

If you have downloaded some Keep Current algorithms, make sure the disk with the algorithms is in drive B. After you insert the disk with the Keep Current algorithms, press **F1** and restart this Session.

Select the Keep Current Algorithm

At this point, you should be looking at the Keep Current Part List screen, which is shown in Figure 3-45.

Selecting a Keep Current algorithm is the same as selecting a standard device algorithm: find the item in the menu and enter the number corresponding to the item. In this case you are looking for a Keep Current algorithm you downloaded from the Data I/O Web site.

Where Is the Algorithm?

Notice that the Keep Current Part List screen lists both the device manufacturer and the device part number. Also, notice that the devices are listed in no particular order: UniSite-*xpi* lists the devices in the order that they are found on the disk.

If you do not see the algorithm you are looking for, make sure the disk with the Keep Current algorithm is inserted in UniSite-*xpi*. If you have more than 10 algorithms on one disk, the algorithm you are looking for could be on the next screen of algorithms. Press **CTRL** + **N** to display the next page of algorithms.

Keep paging through the screens until you find the algorithm you are looking for.

Reviewing Your Options

If you page too far, press **CTRL** + **P** to return to the previous page of Keep Current algorithms.

Found It

After finding the algorithm, type the number enclosed in parentheses and press **ENTER**.

UniSite-*xpi* will load the algorithm for the device you selected. The action symbol rotates while UniSite-*xpi* is loading the algorithm.

After loading the algorithm, UniSite-*xpi* returns to the Main Menu and updates the status window to show the device you have selected.

Keep Current Algorithms and Software Updates

Each Keep Current algorithm is designed to work with a particular version of system software. When UniSite-*xpi* displays the available Keep Current algorithm(s) on the Keep Current Part List screen, it filters out the Keep Current algorithms that are invalid and incompatible with the installed version of system software.

A Keep Current algorithm and a version of UniSite-*xpi* system software are compatible when the numbers to the left and immediate right of the decimal point match. For example,

Algorithm Version	System Software Version	Compatible?
5.51	5.5	Yes
5.7	5.7	Yes
6.1	6.2	No

Keep Current algorithms are valid for one major release of software because the Keep Current algorithms are included with the next release of system software.

The following example illustrates a typical Keep Current scenario:

- In May, you update your system software to version X.4. At the same time, you enroll in the Keep Current Subscription Service.
- In June,   Technologies announces a new device, the Cruft 1263.
- A week later, Data I/O announces support for the Cruft 1263 and places a Keep Current algorithm for the Cruft 1263 on the Data I/O Web site.
- The next day, you go to the Data I/O Web site and download the new algorithm for the Cruft 1263.
- In August, Data I/O releases version X.5 system software, complete with the new algorithm for the Cruft 1263.
- You update your UniSite-*xpi* to version X.5 system software. The algorithm for the Cruft 1263 is part of the system software.

With Keep Current algorithms, you get immediate device support, rather than having to wait for the next release of system software.

Special Note

The next five Sessions cover how to load programming data into UniSite-*xpi*. Because each method applies to a different case, choose the method that most resembles your situation. The three methods are listed below and are explained briefly.

Session 14:
Loading from a Device

In this Session, you load device data from a previously programmed device (a master device). If you do not have a master device and you still want to follow the steps in the Session, you can use a blank device. (Although the data loaded into UniSite-*xpi* will be blank, you will have a chance to follow the procedures.)

Session 15:
Loading Data from Disk

In this Session, you load device data from UniSite-*xpi*'s internal disk drive. A sample data file has been provided on the UniSite-*xpi* System disk for you to use.

Sessions 16, 17, & 18:
Loading Data from a Port

In these Sessions, you load device data into UniSite-*xpi* from a host, such as a PC or workstation connected to one of the serial ports on UniSite-*xpi*. If you are using a PC, you need a terminal emulation package (like Data I/O's HiTerm). If you are using a workstation, you need to know how to transfer a file out of the workstation through the workstation's serial port.

Session 14: Loading Data from a Device

This Session illustrates how to load data into UniSite-*xpi* from a master device. (A master device is a device that is already programmed.) The device used for this session is an AMD 27256 DIP EPROM. If you have a different memory device, you can still follow along, substituting your device manufacturer and part number in place of the AMD 27256.

If you do not have a master device, we suggest you go to the next Session, which shows you how to load data from UniSite-*xpi*'s disk drive. When you have completed Session 4, skip the next three Sessions and continue with Session 9.

Before You Begin

Before starting this Session, you should have completed the first two Sessions. Specifically, you should be familiar with UniSite-*xpi*'s interface and should know how to select a device.

You should also have Site48 installed in UniSite-*xpi*. If Site48 is not installed in UniSite-*xpi*, or if you would like to review the instructions for installing / removing or Site48, see Chapter 2, "Setup and Installation." Read the section titled "Installing a Module" for more information.

In any event, you should not go any further until you are familiar with installing / removing a module or inserting / removing a device.

Can I Use Another Device?

If you do not have an AMD 27256 (the device we are going to use for this Session), then you should go to the Select Device screen and select the device you are going to program. Keep in mind that the device you select might not have the same capabilities as the AMD 27256. For example, the AMD 27256 supports Electronic ID while the Hitachi 27256 does not.

Select a Device

Before loading data from a device, you must first tell UniSite-*xpi* what type of device you are using.

Choose Select Device from the Main Menu. Select the manufacturer and part number of the master device.

Remember, if you don't see the part number on the first screen of part numbers, press **CTRL** + **N** to display the next screen of part numbers. If you don't see the device part number you are looking for, go back to the Manufacturer List and look at the Device Type field. This field allows you to filter out certain device types. Press **SPACE** to cycle through the three settings: All, Memory & Emicros, or Logic Only. When you select a manufacturer, UniSite-*xpi* displays only devices that fit the filter you selected.

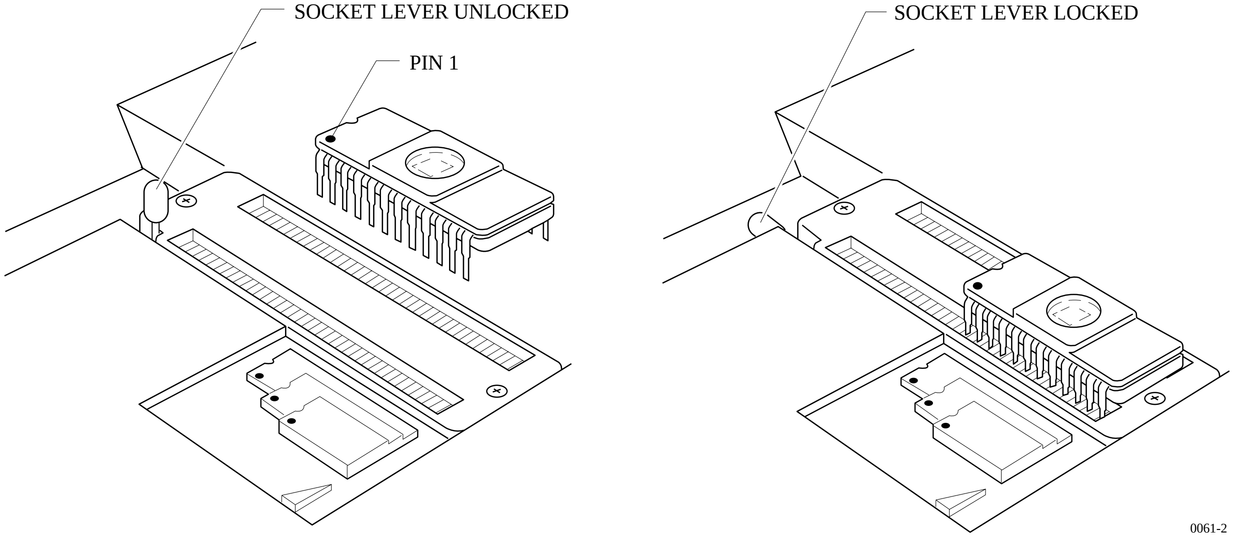
When the programming algorithm has finished loading, the status window will display the selected device, and you will be returned to the Main Menu. UniSite-*xpi* will set the load parameters to match the size of the selected device.

Insert the Device

Make sure Site48 is properly installed in UniSite-*xpi*. If the device socket is locked, unlock it by pulling up the socket lever. Insert the device into the device socket, making sure that the device is bottom-justified and that pin 1 is in the upper-left corner.

Lock the device into place by pressing the socket lever down. Figure 3-46 shows examples of locked and unlocked sockets and also shows the proper orientation for a DIP device.

Figure 3-46
Inserting a DIP Device



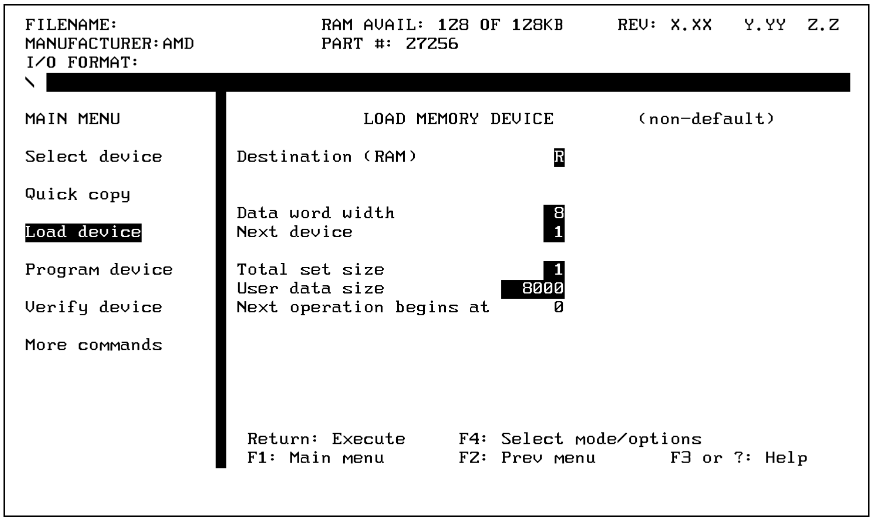
Note: Insert DIP devices into UniSite-xpi AFTER you have a module installed in UniSite-xpi.

Load the Data

With the device type selected and the master device in the socket, the next step is to load the data into UniSite-*xpi*’s RAM.

Choose Load Device from the Main Menu. The Load Memory Device parameter screen, shown in Figure 3-47, is displayed.

Figure 3-47
The Load Memory Device Parameter Screen



```
FILENAME:                     RAM AVAIL: 128 OF 128KB     REV: X.XX   Y.YY  Z.Z
MANUFACTURER:AMD              PART #: 27256
I/O FORMAT:
\

MAIN MENU                  LOAD MEMORY DEVICE        (non-default)

Select device     Destination (RAM)             R

Quick copy
                  Data word width               8
Load device       Next device                   1

Program device    Total set size                1
                  User data size             8000
Verify device     Next operation begins at      0

More commands




                   Return: Execute     F4: Select mode/options
                   F1: Main menu       F2: Prev menu       F3 or ?: Help
```

About Parameters

Parameters are user-definable fields that either qualify or quantify UniSite-*xpi*'s actions. Qualifying parameters, such as Illegal Bit Check, control whether or not UniSite-*xpi* performs an operation. Quantifying parameters, such as Block Size or I/O Translation Format, give UniSite-*xpi* a range or variable to use in an operation.

Six different parameters are shown in Figure 3-47, including Destination and User Data Size.

Parameter Screens

Look at the Load Memory Device screen. At the top of the dialog window, you see one of the following:

```
LOAD MEMORY DEVICE (all parameters)
LOAD MEMORY DEVICE (non-default)
```

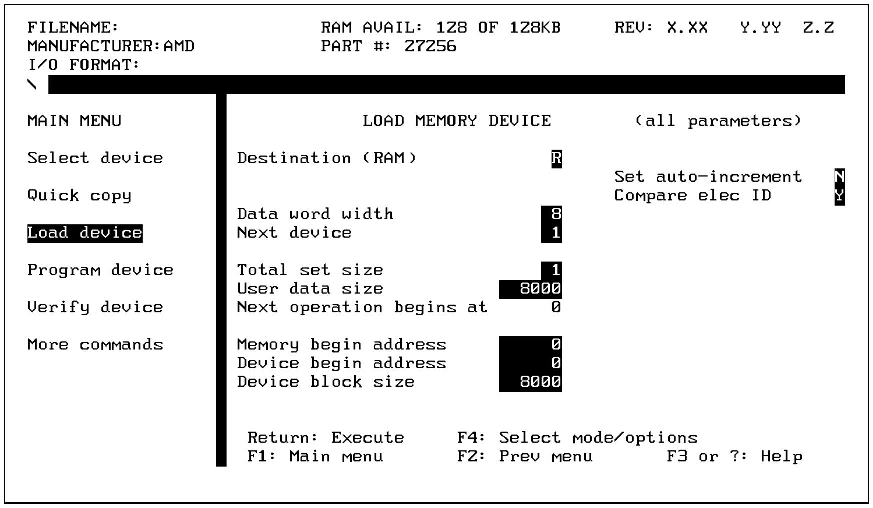
There are two types of parameter screens: simple and complex. UniSite-*xpi* defaults to displaying the simple, Non-default parameters screen.

What Is the Difference?

Both screens let you change load parameters. The All Parameters screen contains all the load parameters supported by the selected device, while the Non-default parameters screen contains a simplified subset of the load parameters supported by the selected device.

To see the difference between the two screens, press **F4**. If you were looking at the Non-default screen, you are now be looking at the All Parameters screen. (Likewise, if you were looking at the All Parameters screen you are now looking at the Non-default screen.) As you can see, there is roughly a 2:1 difference in the number of parameters between the two screens. The All Parameters screen is shown in Figure 3-48. Compare the difference between the two screens.

Figure 3-48
The All Parameters Load Memory Screen



```
FILENAME:                     RAM AVAIL: 128 OF 128KB     REV: X.XX   Y.YY  Z.Z
MANUFACTURER:AMD              PART #: 27256
I/O FORMAT:
\

MAIN MENU                         LOAD MEMORY DEVICE         (all parameters)

Select device      Destination (RAM)             R
                                                      Set auto-increment   N
Quick copy                                            Compare elec ID      Y
                   Data word width               8
Load device        Next device                   1

Program device     Total set size                1
                   User data size             8000
Verify device      Next operation begins at      0

More commands      Memory begin address          0
                   Device begin address          0
                   Device block size          8000

                    Return: Execute     F4: Select mode/options
                    F1: Main menu       F2: Prev menu      F3 or ?: Help
```

Which Should I Use?

During the course of normal device operations, you will usually need only the parameters on the Non-default screen. But if you want to do more complicated operations, such as loading only part of the data from a device, you need to change some of the parameters on the All Parameters screen.

You will not be doing any complicated load operations for this Session, so you should be looking at the Non-default screen. If the All Parameters screen is displayed, press **F4** to switch to the Non-default screen.

Setting the Parameters

Look again at the parameters on the screen. With these parameters, you can change the destination, word width, memory begin address, and block size of the data to be loaded. For this Session, make sure that the parameters are set to the values shown in Figure 3-47.

Changing a Parameter

To change the value of a parameter entry, move the cursor to the desired field and type the new value. Then either press **ENTER** or use the arrow keys to move the cursor to a new field. If you try to enter an incorrect parameter, UniSite-*xpi* beeps and the message line reads

```
Illegal parameter value
```

If you enter a valid parameter, the message line reads

```
Parameter Entered
```

For more information on any of the parameters, see Chapter 4, “Commands.” (Or you could use online help to get information on each of the parameters on the Load Memory Device screen.)

Loading the Data

Now that you have set the parameters, you are ready to load the data from the device.

To begin the load, press **ENTER**. When UniSite-*xpi* has loaded the data, it displays the following message in the message bar:

```
OPERATION COMPLETE: Sumcheck = xxxxxxxx
```

where *xxxxxxxx* represents the 8-digit sumcheck of the data loaded from the device.

Note: 8-bit is the default value for the sumcheck. You can change this to 16-bit by setting the Checksum Calculation parameter to **D**.

Review

When you select Load Device from the Main Menu, you see one of two parameter-entry screens: the simple Non-default screen or the complex All Parameters screen. Press **F4** to toggle between the two parameter-entry screens.

When you have entered the parameters, press **ENTER** to begin the loading.

When UniSite-*xpi* is finished loading the data from the device, it displays an 8-digit sumcheck of the data loaded. (UniSite-*xpi* displays a 4-digit, 8-bit sumcheck when you load data from a logic device.)

Session 15: Loading Data from Disk

This Session illustrates how to load binary device data into UniSite-*xpi* from one of UniSite-*xpi*'s internal disk drives. When you have completed this Session, skip the next two Sessions and continue with Session 8. If you have a data file you want to download to UniSite-*xpi*, you should skip this Session and complete the next two Sessions.

Before You Begin

Before starting this Session, you should have completed the first two Sessions. Specifically, you should be familiar with UniSite-*xpi*'s user interface and should know how to select a device.

Getting There

Go to the More Commands menu and notice the following two menu items:

```
File Operations
```

```
Transfer Data
```

They both sound like logical places to put a command for loading a data file. In fact, both menus contain commands for loading data files, but the two commands manipulate data in different ways.

What's the Difference?

The Transfer Data command calls up the Transfer Menu. On this menu, you will find commands used for transferring formatted data files between UniSite-*xpi* and another machine (like a PC). You will also find commands for transferring **formatted** data files (not binary files) between UniSite-*xpi* and a disk in one of UniSite-*xpi*'s disk drives. This menu does not contain the command you are looking for.

The File Operations command calls up the File Menu. On this menu you will find DOS-like commands, such as Load, Rename, Delete, Format Disk, and Copy File. The command you are looking for (Load File) is on the File Menu. The Load File command simply loads the binary data file directly into memory.

When Would I Use this Command?

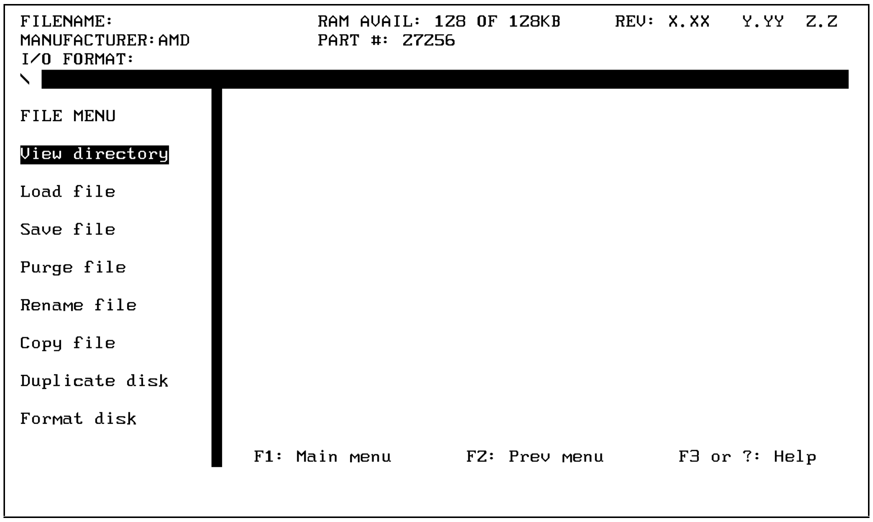
Use the Load File command if the data file you want to load contains a binary image of the data you want to program into a device. The Save File command creates a binary image file by copying the data in UniSite-*xpi*'s RAM directly to a disk file.

Note: Do not use the Load File and Save File commands with formatted data files. Use the Input from Disk and Output to Disk commands to load and save formatted data files. See Chapter 4, "Commands," for more information.

The File Menu

To get to the File menu, select File Operations from the More Commands menu. The File menu appears and should look like the screen shown in Figure 3-49.

Figure 3-49
The File Menu



```
FILENAME:                    RAM AVAIL: 128 OF 128KB    REV: X.XX   Y.YY  Z.Z
MANUFACTURER:AMD             PART #: 27256
I/O FORMAT:
\
FILE MENU

View directory

Load file

Save file

Purge file

Rename file

Copy file

Duplicate disk

Format disk

                 F1: Main menu      F2: Prev menu      F3 or ?: Help
```

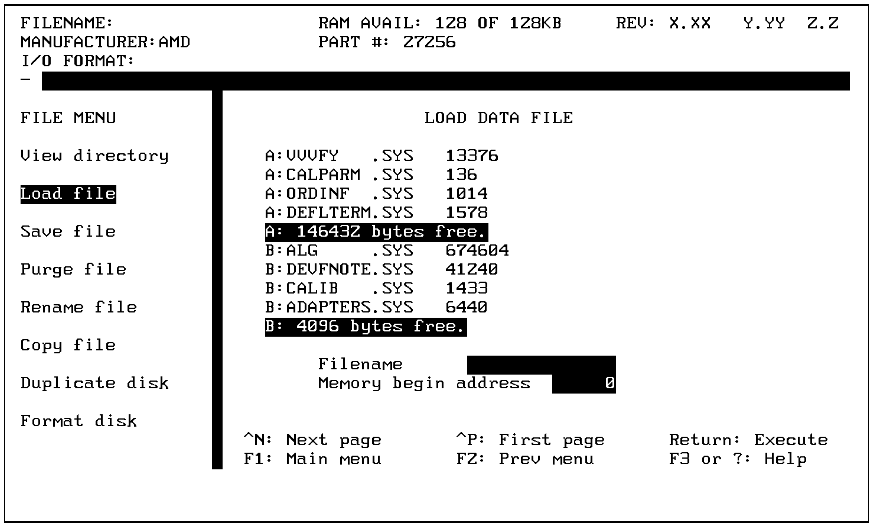
Viewing a Directory

The View Directory command displays a directory of the files stored on the disks in both disk drives. The files on the disk in the lefthand drive (drive A) are prefixed with A:, and the files on the disk in the righthand drive (drive B) are prefixed with B:.

Loading the File

Create a binary file on a disk and place the disk in drive A. Select **Load File** from the File menu. The dialog window displays a directory of the files on the disks in UniSite-*xpi*'s two disk drives. (Notice that the Load File command displays a directory similar to the View Directory command.) A screen similar to what you will see is shown in Figure 3-50.

Figure 3-50
The Load File Dialog Screen



```
FILENAME:                    RAM AVAIL: 128 OF 128KB    REV: X.XX   Y.YY  Z.Z
MANUFACTURER:AMD             PART #: 27256
I/O FORMAT:
-
FILE MENU                              LOAD DATA FILE

View directory          A:VVVFY   .SYS   13376
                        A:CALPARM .SYS   136
Load file               A:ORDINF  .SYS   1014
                        A:DEFLTERM.SYS   1578
Save file               A: 146432 bytes free.
                        B:ALG     .SYS   674604
Purge file              B:DEVFNOTE.SYS   41240
                        B:CALIB   .SYS   1433
Rename file             B:ADAPTERS.SYS   6440
                        B: 4096 bytes free.
Copy file
                             Filename
Duplicate disk               Memory begin address        0

Format disk
                      ^N: Next page      ^P: First page     Return: Execute
                      F1: Main menu      F2: Prev menu      F3 or ?: Help
```

To load the binary data file, follow the steps below:

1. Move the cursor to the Filename field and type **a:*filename*.bin** (where *filename* represents the name you have given your binary file) and press **ENTER**. UniSite-*xpi* displays the following message in the message bar:

   ```
   Parameter Entered
   ```

2. Next, move the cursor to the Memory Begin Address field; type **0** and press **ENTER**.
3. Press **ENTER** to begin loading the data file. The action symbol rotates while UniSite-*xpi* loads the data file, and displays the following message in the message bar:

   ```
   Loading data from file.
   ```

   If the data file loads successfully, UniSite-*xpi* displays the following message in the message bar:

   ```
   Done.
   ```

Did the File Load Successfully?

If the binary data file did not load successfully, return to the beginning of this Session and try it again.

If the binary data file loaded successfully, you will be returned to the Load File dialog screen. Skip to Session 8. In that Session you will find out how to download data from a host. In Session 9 you will learn how to edit a data file.

Review

You can load device data into UniSite-*xpi* by loading a data file from one of UniSite-*xpi*'s internal disk drives. Use the commands on the File Operations menu (such as the Load File command) if your data file is stored in binary image format. Use the commands on the Transfer Data menu (such as the Input from Disk command) if your data file is stored in a specific data translation format, such as Intel Hex or JEDEC. There is a list of supported formats at the beginning of Chapter 6, "Translation Formats."

When you select the Load File command, UniSite-*xpi* displays a directory of the files on the disks in UniSite-*xpi*'s two drives. Enter the drive name and filename of the file you want to load.

If you do not see the file you want to load, press **CTRL** + **N** to display the next page of files, or press **CTRL** + **P** to display the previous page of files. If you are loading data for a memory device, you should also enter a memory begin address.

Finally, press **ENTER** to begin the loading. During the load operation, the action symbol rotates. If the load operation completes successfully, UniSite-*xpi* displays the following message in the message bar:

```
Done.
```

Session 16: Selecting a Translation Format

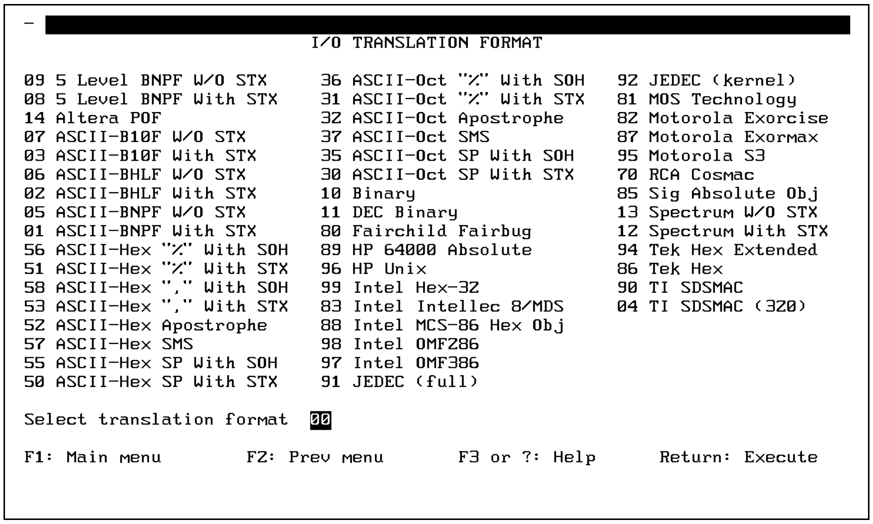
This Session and the following Session are companion Sessions. This Session introduces you to translation formats and shows you how to select a translation format. The next Session shows you how to load a data file through UniSite-*xpi*'s Terminal port.

About Translation Formats

Translation formats represent different ways of representing the device data in a data file. A data file could contain the fuse pattern and test vectors for a logic device or the data for a memory device.

Getting There

Starting at the Main Menu, press **M**, **T**, **F** to get to the Translation Format screen, which should look like Figure 3-51. If you get lost, return to the Main Menu and start over. (Remember, press **F1** to get to the Main Menu.)



```
-
                            I/O TRANSLATION FORMAT

09 5 Level BNPF W/O STX     36 ASCII-Oct "%" With SOH   92 JEDEC (kernel)
08 5 Level BNPF With STX    31 ASCII-Oct "%" With STX   81 MOS Technology
14 Altera POF               32 ASCII-Oct Apostrophe     82 Motorola Exorcise
07 ASCII-B10F W/O STX       37 ASCII-Oct SMS            87 Motorola Exormax
03 ASCII-B10F With STX      35 ASCII-Oct SP With SOH    95 Motorola S3
06 ASCII-BHLF W/O STX       30 ASCII-Oct SP With STX    70 RCA Cosmac
02 ASCII-BHLF With STX      10 Binary                   85 Sig Absolute Obj
05 ASCII-BNPF W/O STX       11 DEC Binary               13 Spectrum W/O STX
01 ASCII-BNPF With STX      80 Fairchild Fairbug        12 Spectrum With STX
56 ASCII-Hex "%" With SOH   89 HP 64000 Absolute        94 Tek Hex Extended
51 ASCII-Hex "%" With STX   96 HP Unix                  86 Tek Hex
58 ASCII-Hex "," With SOH   99 Intel Hex-32             90 TI SDSMAC
53 ASCII-Hex "," With STX   83 Intel Intellec 8/MDS     04 TI SDSMAC (320)
52 ASCII-Hex Apostrophe     88 Intel MCS-86 Hex Obj
57 ASCII-Hex SMS            98 Intel OMF286
55 ASCII-Hex SP With SOH    97 Intel OMF386
50 ASCII-Hex SP With STX    91 JEDEC (full)

Select translation format  00

F1: Main menu         F2: Prev menu        F3 or ?: Help       Return: Execute
```

Figure 3-51
The Translation Format Selection Screen

Making a Choice

Now that you are at the I/O Translation Format selection screen, you need to choose a translation format to use. Normally, you would select the same format as your data file. For this Session, select the Intel 8-bit Hex translation format.

Select the translation format the same way you selected a device manufacturer: find what you are looking for and type the number beside it. So, for the format selection screen shown in Figure 3-51, you would type **8**, **3** and press **ENTER** to select the Intel 8-bit Hex (Intel Intellec 8/MDS) translation format. When you press **ENTER**, UniSite-*xpi* configures itself for the selected translation format and returns you to the Transfer Menu. Notice that the I/O Format line in the status window has changed and now reads

```
I/O FORMAT: Intel Intellec 8/MDS
```

Review

Use the Format Select command on the Transfer Data menu to select a new translation format. After locating the desired format, type the number to the left of the format.

Session 17: Loading Data from a PC Using HiTerm

In this Session, you will learn how to use HiTerm to download data from a PC to UniSite-*xpi* through one of the serial ports on UniSite-*xpi*.

Note: If you are using a DOS-based PC, we recommend you use HiTerm as your terminal emulation software. Although you might already have other terminal emulation software on your PC, HiTerm was written to work with UniSite-xpi and can download files to UniSite-xpi at 115.2K baud.

Why HiTerm?

At this point, you might be wondering why we recommend you use HiTerm and not another terminal emulator. First and foremost, only HiTerm supports 115.2K baud downloading. Second, HiTerm supports formatted binary transfer, while some other terminal emulators do not. Third, HiTerm was written by Data I/O with UniSite in mind. HiTerm does all the opening and closing of files automatically; with other terminal emulators you must do that yourself.

Before You Begin

This Session assumes you have a DOS-based PC connected to UniSite-*xpi*. See Chapter 2, “Setup and Installation,” for information on connecting a PC to UniSite-*xpi*.

You must have HiTerm properly installed on the PC connected to UniSite-*xpi*. See Chapter 2, “Setup and Installation,” for quick installation instructions for HiTerm. See the *HiTerm User Manual* behind the Utilities tab in this binder if you need more information about installing HiTerm.

Finally, we suggest you have UniSite-*xpi* configured for High Speed Download, which allows you to download a data file to UniSite-*xpi* at 115.2K baud. See “Set Up High Speed Download” on page 2-26 for more information about configuring UniSite-*xpi* for High Speed Download.

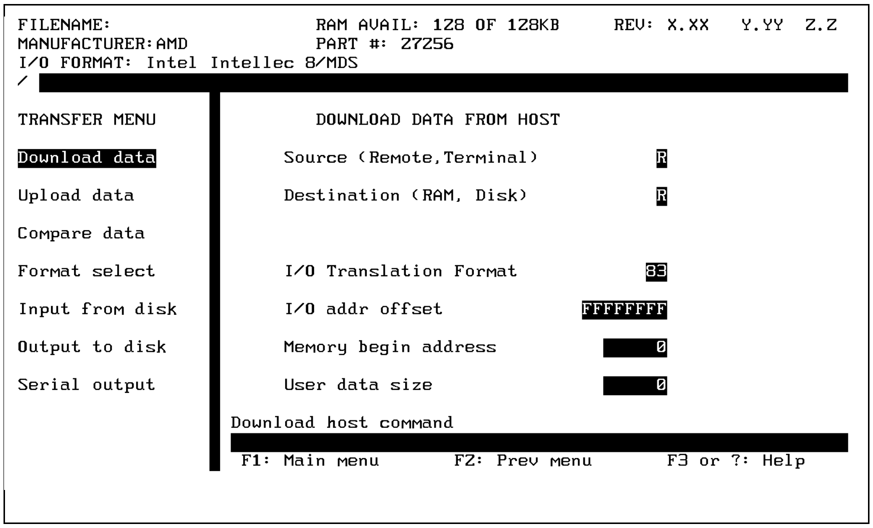
What to Transfer?

We suggest you use **sample.dat**, a sample data file supplied on the HiTerm disk. If you want to transfer a different data file, substitute its filename for **sample.dat**. If your data file is in a format other than Intel Intellec 8/MDS, use the Format Select command to select that format. See Session 6 for information on selecting a translation format.

Preparing UniSite-*xpi*

From the Main Menu, press **M**, **T**, **D** to get to the Download Data from Host screen, which is shown in Figure 3-52.

Figure 3-52
The Download Data from Host Screen



```
FILENAME:                    RAM AVAIL: 128 OF 128KB    REV: X.XX   Y.YY  Z.Z
MANUFACTURER:AMD             PART #: 27256
I/O FORMAT: Intel Intellec 8/MDS
/

TRANSFER MENU                  DOWNLOAD DATA FROM HOST

Download data              Source (Remote,Terminal)          R

Upload data                Destination (RAM, Disk)           R

Compare data

Format select              I/O Translation Format           83

Input from disk            I/O addr offset            FFFFFFFF

Output to disk             Memory begin address              0

Serial output              User data size                    0

                         Download host command

                           F1: Main menu      F2: Prev menu      F3 or ?: Help
```

Look at the parameters and make sure they reflect your system configuration. If you are running HiTerm on a PC connected to the Remote port on UniSite-*xpi*, use the following settings:

- Source—Remote port
- Destination—RAM
- I/O Translation Format—83 (Intel Hex)
- I/O Address Offset—FFFFFFFF
- Memory Begin Address—0
- User Data Size—0

If your configuration is different, change the parameters to match your configuration. For example, select translation format 91 if the data file you will download is a JEDEC file.

For now, leave the Download Host Command blank; you will fill that in later. If the parameters are correct, skip ahead to the section titled “Downloading the File.”

Note: See Chapter 4, “Commands,” if you want to find out more about the parameters on the Download Data from Host screen.

About Parameters

Parameters are user-definable fields that determine what UniSite-*xpi* does. Parameters either qualify or quantify UniSite-*xpi*’s actions. Qualifying parameters, such as Source and Destination, control the type of operation to perform. Quantifying parameters, such as Block Size or I/O Translation Format, give UniSite-*xpi* a range or variable to use in an operation.

Seven parameters are shown in Figure 3-52, including Source, Memory Begin Address, and Download Host Command.

Changing a Parameter

Follow the steps below if you need to change a parameter.

1. Move the cursor to the field you need to change.
2. Type the new value and move the cursor to another field. If the parameter is acceptable, UniSite-*xpi* displays

   ```
   Parameter Entered
   ```

   in the message bar. (Remember, the message bar is located below the status window.)

If you enter an incorrect parameter, UniSite-*xpi* beeps and displays an error message. Continue until the displayed parameters reflect your configuration.

Downloading the File

Locate the Data File on the PC

Change to the directory containing the data file you are going to download. (Remember, if you are using the sample data file supplied with HiTerm, you are looking for the **sample.dat** file.)

If you are in DOS, use the CD and DIR commands to change to the directory containing the data file to download.

If you are running HiTerm, press **ALT** + **F6** then **ENTER** to view the current directory. If **sample.dat** is in the current directory, press **ENTER** to return to the terminal emulation.

Note: The function key commands in this Session are for IBM-compatible PCs. However, if you are using an NEC-9800 PC, the HiTerm commands will be slightly different. Consult the HiTerm documentation for more information.

If **sample.dat** is not in the current directory, press **ALT** + **F5** to bring up HiTerm's change directory command. Type the drive and pathname of the directory containing **sample.dat** and press **ENTER** to change to that directory. Press **ALT** + **F6** then **ENTER** to view the current directory. If **sample.dat** is in the current directory, press **ENTER** to return to the terminal emulation. If the data file is not in the current directory, repeat the procedure until you find the data file you want to download.

About the Download Host Command

Earlier in this Session, you entered the parameters for the download, but you left the Download Host Command parameter blank. The Download Host Command is a command that UniSite-*xpi* sends to the PC to initiate the download. Because you are running HiTerm, you can specify a special Download Host Command, the `TRANSFER` command.

When UniSite-*xpi* sends the `TRANSFER` command (or TR for short) it tells HiTerm to download a specified file to UniSite-*xpi*.

If the sample data file is in your current directory, move the cursor to the Download Host Command field and enter the following Download Host Command

tr sample.dat ↵

Remember to substitute the appropriate filename if you are using a different file. When you enter the Download Host command, the message bar displays

```
Parameter Entered
```

Download the File

Press **ENTER** to begin the download. The action symbol rotates while the data is being downloaded. When the download is finished the message bar displays

```
Data transfer complete. Data sum = xxxxxxxx (8-bit)
```

Go to the section titled “After the Download” when the download is complete.

After the Download

By this point, you should have successfully downloaded a data file into UniSite-*xpi*.

About the Sumcheck

When the download was completed, the message bar displayed

```
Data transfer complete. Data sum = xxxxxxxx (8-bit)
```

This message tells you two things: first, it tells you that the data transfer completed successfully, and second, it tells you what the sumcheck of the data transferred is. The sumcheck is shown above as *xxxxxxxx*.

The sumcheck for memory devices is an eight-digit hexadecimal summation of the data downloaded. If you change one byte of information in the data file, the sumcheck will also change. The sumcheck is a good method of verifying that the data you downloaded matches the data on your host.

Later, when you program this data into a device, UniSite-*xpi* will generate another sumcheck. If the two are the same, then the data programmed is the same as the data downloaded. If the two sumchecks are different, then the data programmed is not the same as the data downloaded.

Review

In this Session, you learned how to download data to UniSite-*xpi* from a host connected to UniSite-*xpi*’s Terminal port. The steps were

- Select the translation format that matched the format of your data file.
- Go to the download screen by pressing **M**, **T**, **D**.
- Set the parameters on the Download screen to match your setup.
- Enter the host Download Host Command. For example, to transfer the file **filename.dat** you would type **tr filename.dat** for a PC using HiTerm.

Session 18: Loading Data from a Host

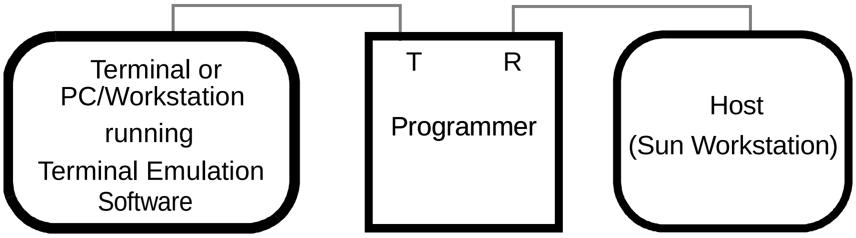
In this Session, you will learn how to download data from a host to UniSite-*xpi* through one of the serial ports on UniSite-*xpi*. The procedures in this Session apply to many types of hosts, including VAXes, UNIX-based workstations, and DOS-based PCs.

Note: *If you are using a DOS-based PC, we recommend that you use HiTerm as your terminal emulation software. Downloading a file with HiTerm is covered in the previous Session.*

Before You Begin

This Session assumes that you have a host connected to one of the serial ports on UniSite-*xpi* and a terminal connected to the other serial port on UniSite-*xpi*. This type of configuration is called Transparent mode and is shown in Figure 3-53.

Figure 3-53
Transparent Mode


Terminal or PC/Workstation running Terminal Emulation Software
T
R
Programmer
Host
(Sun Workstation)

See Chapter 2, “Setup and Installation,” for information on connecting a host and terminal to UniSite-*xpi*.

About Transparent Mode

Transparent mode, shown in Figure 3-53, is a feature of UniSite-*xpi* that allows the programmer to be inline between your terminal and host computer. This eliminates the need for a switch box or a second link to the host, and enables you to download directly from the host to UniSite-*xpi*. The host could be a networked file server such as a VAX or a Sun. When set up properly, the terminal connected to UniSite-*xpi* can control both UniSite-*xpi* and the remote host.

In Transparent mode, UniSite-*xpi* passes all characters through its Terminal and Remote ports as if it weren't there. The two serial ports on UniSite-*xpi* can even operate at different baud rates. While operating UniSite-*xpi* from the terminal, press **ESC**, **CTRL** + **T** to toggle UniSite-*xpi* between terminal mode and transparent mode. UniSite-*xpi* remains in transparent mode until it receives another **ESC**, **CTRL** + **T** command, at which time it switches back to terminal mode.

What to Transfer?

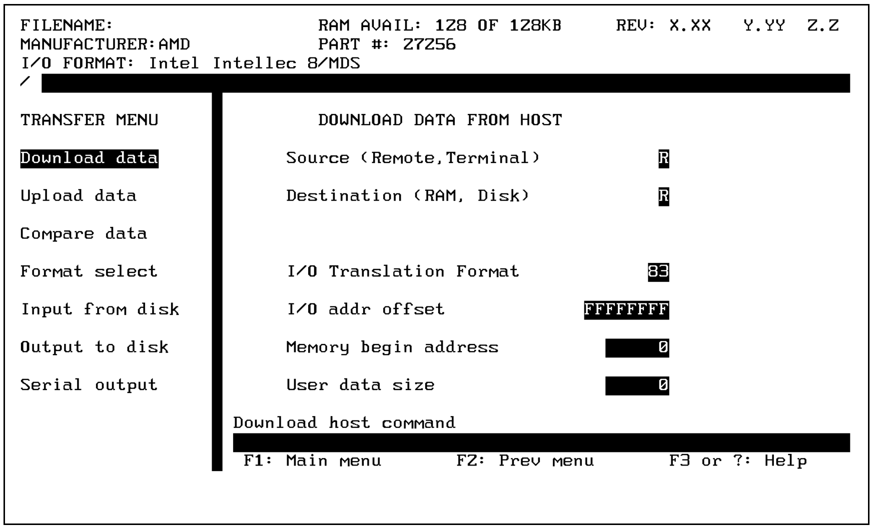
We suggest you use **sample.dat**, a sample data file supplied on the HiTerm disk. If you want to transfer a different data file, substitute its filename for **sample.dat**. If your data file is in a format other than Intel Intellec 8/MDS, use the Format Select command to select that format. See Session 6 for information on selecting a translation format.

The rest of this Session assumes that your sample data file is stored in the Intel Intellec 8/MDS data translation format. If your data file is in a different translation format, use the Format Select command to select that format. See Session 6 for information on selecting a data translation format.

Preparing UniSite-*xpi*

From the Main Menu, press **M**, **T**, **D** to get to the Download Data From Host screen, which is shown in Figure 3-54.

Figure 3-54
The Download Data from Host Screen



```
FILENAME:                    RAM AVAIL: 128 OF 128KB     REV: X.XX   Y.YY  Z.Z
MANUFACTURER:AMD             PART #: 27256
I/O FORMAT: Intel Intellec 8/MDS
/

TRANSFER MENU                   DOWNLOAD DATA FROM HOST

Download data               Source (Remote,Terminal)           R

Upload data                 Destination (RAM, Disk)            R

Compare data

Format select               I/O Translation Format            83

Input from disk             I/O addr offset             FFFFFFFF

Output to disk              Memory begin address               0

Serial output               User data size                     0

                     Download host command

                      F1: Main menu       F2: Prev menu       F3 or ?: Help
```

Look at the parameters and make sure they reflect your system configuration. Use the following settings if your host is connected to the Remote port on UniSite-*xpi*:

- Source—Remote port
- Destination—RAM
- I/O Translation Format—83 (Intel Hex)
- I/O Address Offset—FFFFFFFF
- Memory Begin Address—0
- User Data Size—0

If your configuration is different, change the parameters to match your configuration. For example, select translation format 91 if the data file you will download is a JEDEC file.

For now, leave the Download Host Command blank; you will fill that in later. If the parameters are correct, skip ahead to the section titled "Downloading the File."

About Parameters

Parameters are user-definable fields that determine what UniSite-*xpi* does. Parameters either qualify or quantify UniSite-*xpi*'s actions. Qualifying parameters, such as Source and Destination, control the type of operation to perform. Quantifying parameters, such as Block Size or I/O Translation Format, give UniSite-*xpi* a range or variable to use in an operation.

Seven parameters are shown in Figure 3-54, including Source, Memory Begin Address, and Download Host Command.

Changing a Parameter

Follow the steps below if you need to change a parameter:

1. Move the cursor to the field you need to change.
2. Type the new value and move the cursor to another field. If the parameter is acceptable, UniSite-*xpi* displays

   ```
   Parameter Entered
   ```

   in the message bar. (Remember, the message bar is located below the status window.)

If you enter an incorrect parameter, UniSite-*xpi* beeps and displays an error message. Continue until the displayed parameters reflect your configuration.

Downloading the File

Locate the Data File on the Host

Change to the directory containing the file you are going to download. Before you change directories on the host, press **ESC**, **CTRL** + **T** to enter Transparent mode.

If you are using a UNIX-based host, use the CD command to change to the directory containing the data file to download.

Note: The commands in this Session are for UNIX-based hosts. Substitute the appropriate commands if you are using a different type of host, such as a VMS-based machine. If you need more information, consult the system's documentation or your system administrator.

Once you have changed to the directory containing the data file you are going to download, press **ESC**, **CTRL** + **T** again to leave Transparent mode and return to UniSite-*xpi*'s full screen interface. You might have to press **CTRL** + **R** to redraw the screen.

About the Download Host Command

Earlier in this Session, you entered the parameters for the download, but you left the Download Host Command parameter blank. The Download Host Command is a command that UniSite-*xpi* sends to the host to initiate the download. Because you are on a UNIX-based host, use the CAT command as the Download Host Command.

If the sample data file is in your current directory, enter the following Download Host Command

cat sample.dat ↵

Remember to substitute the appropriate filename if you are using a different file. When you enter the Download Host command, the message bar displays

```
Parameter Entered
```

Download the File

Press **ENTER** to begin the download. The action symbol rotates while the data is being downloaded. When the download is finished the message bar displays

```
Data transfer complete. Data sum = xxxxxxxx (8-bit)
```

After the Download

By this point, you should have successfully downloaded a data file into UniSite-*xpi*.

About the Sumcheck

When the download was completed, the message bar displayed

```
Data transfer complete. Data sum = xxxxxxxx (8-bit)
```

This message tells you two things: first, it tells you that the data transfer completed successfully, and second, it tells you what the sumcheck of the transferred data is. The sumcheck is 8-bit and is shown above as *xxxxxxxx*.

The sumcheck for memory devices is an eight-digit hexadecimal summation of the downloaded data. If you change one byte of information in the data file, the sumcheck will also change. The sumcheck is a good method of verifying that the data you downloaded matches the data on your host.

Later, when you program this data into a device, UniSite-*xpi* will generate another sumcheck. If the two are the same, then the data programmed is the same as the downloaded data. If the two sumchecks are different, then the data programmed is not the same as the downloaded data.

Review

In this Session, you learned how to download data to UniSite-*xpi* from a host connected to the Remote port on UniSite-*xpi*. The steps were

- Select the translation format that matched the format of your data file.
- From the Main Menu, press **M**, **T**, **D** to go to the download screen.
- Set the parameters on the Download screen to match your setup.
- Enter the host Download Host Command. For example, to transfer the file **filename.dat** you would enter **cat filename.dat** for a UNIX-based host. If you are using a VMS-based host, you would enter **type filename.dat**.

Session 19: Editing Data

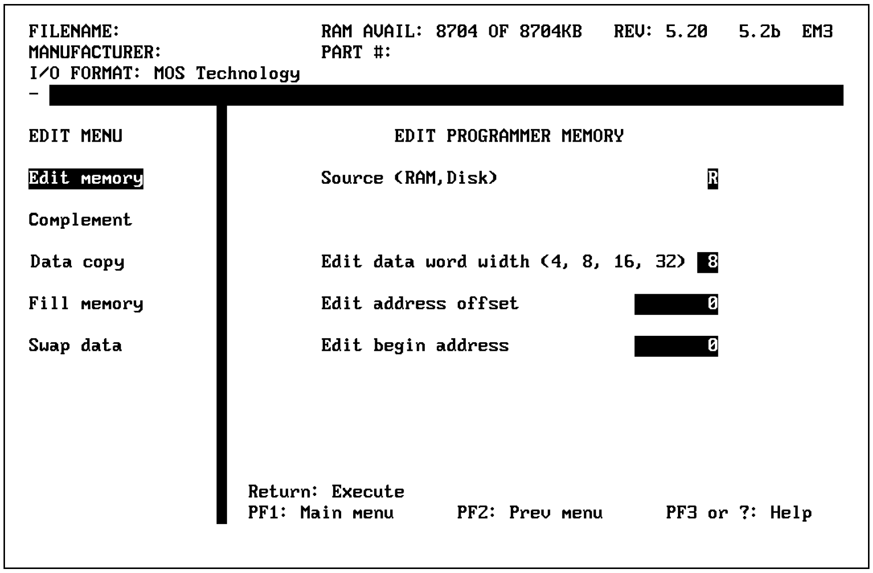
In the previous Session, you loaded a data file into UniSite-*xpi*. In this Session, you will learn how to edit that data file.

Before You Begin

Return to the Main Menu before you begin this Session. You should also select a memory device before you begin this Session. See Session 2 for a quick explanation of how to select a device.

Getting There

The Edit Data command is located on the More Commands menu. Select More Commands / Edit Data from the Main Menu. Finally, select Edit Memory to bring up the Edit Programmer Memory dialog window, which is shown in Figure 3-55.

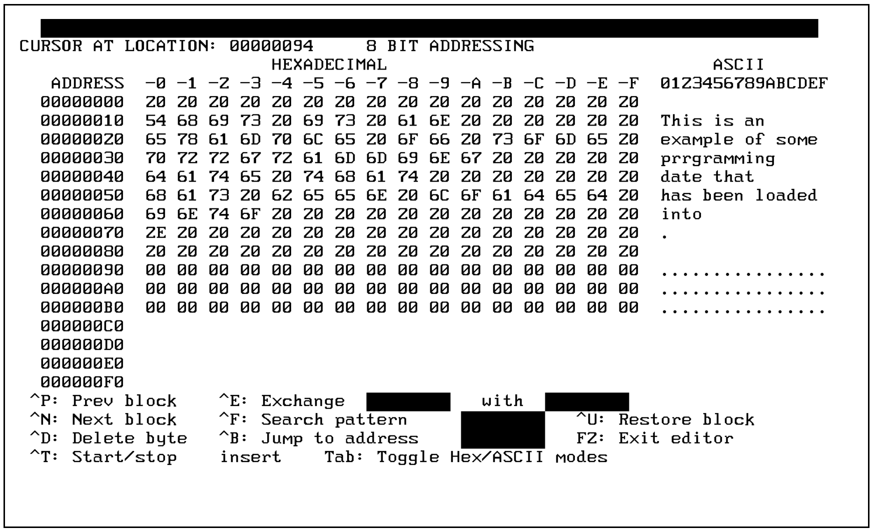


```
FILENAME:                   RAM AVAIL: 8704 OF 8704KB   REV: 5.20   5.2b  EM3
MANUFACTURER:               PART #:
I/O FORMAT: MOS Technology
-

EDIT MENU                        EDIT PROGRAMMER MEMORY

Edit memory                 Source (RAM,Disk)                  R

Complement

Data copy                   Edit data word width (4, 8, 16, 32) 8

Fill memory                 Edit address offset                0

Swap data                   Edit begin address                 0




                  Return: Execute
                  PF1: Main menu      PF2: Prev menu      PF3 or ?: Help
```

Figure 3-55
The Edit Programmer Memory Screen

Make sure the parameters are set as they are shown in Figure 3-55, then press **ENTER**. (For an explanation of how to change parameters, refer to the “Changing a Parameter” section of Session 7.) The entire screen clears and the edit screen appears, as shown in Figure 3-56.

Figure 3-56
The Edit Screen



```
CURSOR AT LOCATION: 00000094     8 BIT ADDRESSING
                        HEXADECIMAL                                ASCII
  ADDRESS   -0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -A -B -C -D -E -F  0123456789ABCDEF
 00000000   20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
 00000010   54 68 69 73 20 69 73 20 61 6E 20 20 20 20 20 20  This is an
 00000020   65 78 61 6D 70 6C 65 20 6F 66 20 73 6F 6D 65 20  example of some
 00000030   70 72 72 67 72 61 6D 6D 69 6E 67 20 20 20 20 20  prrgramming
 00000040   64 61 74 65 20 74 68 61 74 20 20 20 20 20 20 20  date that
 00000050   68 61 73 20 62 65 65 6E 20 6C 6F 61 64 65 64 20  has been loaded
 00000060   69 6E 74 6F 20 20 20 20 20 20 20 20 20 20 20 20  into
 00000070   2E 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .
 00000080   20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
 00000090   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
 000000A0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
 000000B0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
 000000C0
 000000D0
 000000E0
 000000F0
^P: Prev block    ^E: Exchange           with
^N: Next block    ^F: Search pattern                   ^U: Restore block
^D: Delete byte   ^B: Jump to address                  FZ: Exit editor
^T: Start/stop    insert    Tab: Toggle Hex/ASCII modes
```

What Does It Say?

The edit screen displays 256 bytes at a time. The four-line reminder bar at the bottom of the screen displays the available commands (no online help is available for this screen). The right third of the screen displays the ASCII translation of the hexadecimal bytes on the left portion of the screen. The top line of the screen displays the address location of the cursor (in hexadecimal) and the addressing mode.

If you haven't already, read the message in the sample data file. As you can see there are some errors that need correcting.

Moving Around

The cursor starts in the upper-left corner of the hex display. Move the cursor around and watch what happens to the cursor location counter (on the top line of the screen). As you move the cursor, the counter changes to reflect your current location. If you wanted to edit hex data, you would do it in the hex column; however, during this Session, you will be editing text, so you should move the cursor to the ASCII side of the screen.

To move to the ASCII side of the screen, press **TAB**. Move the cursor around. Notice that the cursor location counter still changes when you move from one address to another.

Making Changes

Overtyping

With the cursor in the ASCII field, move the cursor to the second **r** in **prrgramming**. Type **o** to correct the spelling mistake. Next, move the cursor to **date** and change it to read **data**.

The previous two corrections were done in overtype mode: the typed characters replaced the previous characters. The next correction will be done in insert mode.

Inserting

To turn on insert mode, press **CTRL** + **T**. This will light up the insert indicator on the bottom of the screen. With this indicator lit, every character typed will be inserted at the cursor and all following characters will be pushed to the right.

The third and last correction is adding the word **UniSite-***xpi* before the period below the word **loaded**. Move the cursor on top of the period and type

UniSite-*xpi*

Restoring Your File

If you are not satisfied with your edits and you want to restore the current screen of data, press **CTRL** + **U**. UniSite-*xpi* restores the current screen of data to the state it was in after the last save.

Note: When you move to another block, all edits are saved.

Saving Your Edits

Save your edits if you are satisfied with them. Press **F2** to save your edits and return to the previous menu.

Note: When you exit the Edit screen or move to another block, all edits are saved. If you are saving to RAM, there is no way to recall the original data file once you save an edited data file.

Review

In this Session, you learned how to edit a data file stored in RAM. To access UniSite-*xpi*'s built-in editor, press **M**, **E**, **E** from the Main Menu. Then, from the Edit Programmer Memory screen, select **R** to edit RAM. Once you are in the editor, you can edit data in either its hex representation or its ASCII representation. Press **TAB** to toggle between the two modes.

The editor defaults to overtype mode, but it can also operate in insert mode. Press **CTRL** + **T** to toggle between the two modes.

After making edits to a page of data, you can restore the page to its original condition by pressing **CTRL** + **U**.

Session 20: Programming a Memory Device

This Session shows you how to use UniSite-*xpi* to program a memory device.

Before You Begin

Make sure you have completed Session 1, an introduction to the UniSite-*xpi* interface.

Can I Use Another Device?

If you do not have an AMD 27256 (the device we are going to use for this Session), go to the Select Device screen and select the device you are going to program. Keep in mind that the device you select might not have the same capabilities as the AMD 27256. For example, the AMD 27256 supports Electronic ID while the Hitachi 27256 does not.

Load the Data File

Before you can program the device, you must perform the following steps to load your data into RAM.

1. Create a binary file on a disk.
2. Place the disk in drive A.
3. From the More Commands/File Operations menu, use the Load File command to load the data file into RAM. This operation is described in detail in Session 5.

Parameter Screens

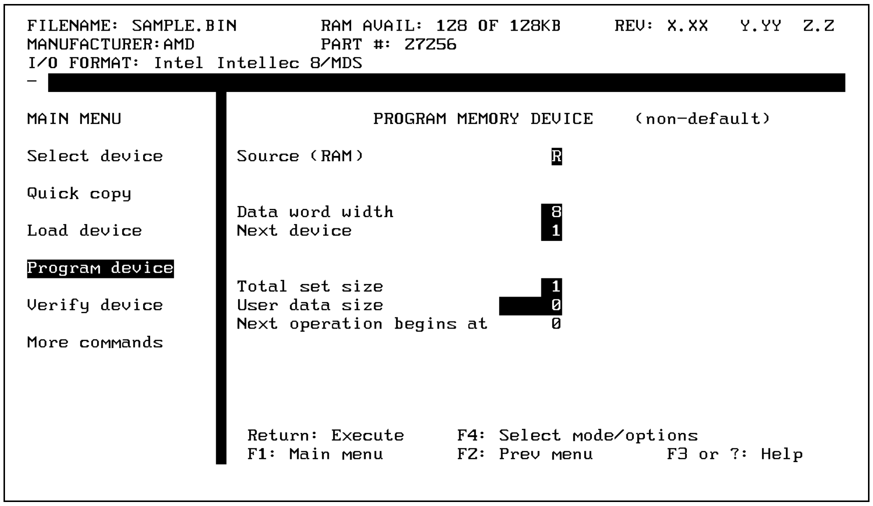
After the data is loaded into RAM, select Program Device from the Main Menu. The dialog window displays the Program Memory Device screen, shown in Figure 3-57.

Look at the Program Memory Device screen. At the top of the dialog window, you see one of the following:

```
PROGRAM MEMORY DEVICE (all parameters)
PROGRAM MEMORY DEVICE (non-default)
```

There are two types of programming parameter screens: simple and complex. UniSite-*xpi* defaults to displaying the simple, Non-default parameters screen.

Figure 3-57
The Program Memory Device Screen



```
FILENAME: SAMPLE.BIN        RAM AVAIL: 128 OF 128KB     REV: X.XX   Y.YY  Z.Z
MANUFACTURER:AMD            PART #: 27256
I/O FORMAT: Intel Intellec 8/MDS
-
MAIN MENU                         PROGRAM MEMORY DEVICE    (non-default)

Select device     Source (RAM)                    R

Quick copy
                  Data word width                 8
Load device       Next device                     1

Program device
                  Total set size                  1
Verify device     User data size                  0
                  Next operation begins at        0
More commands




                   Return: Execute     F4: Select mode/options
                   F1: Main menu       F2: Prev menu       F3 or ?: Help
```

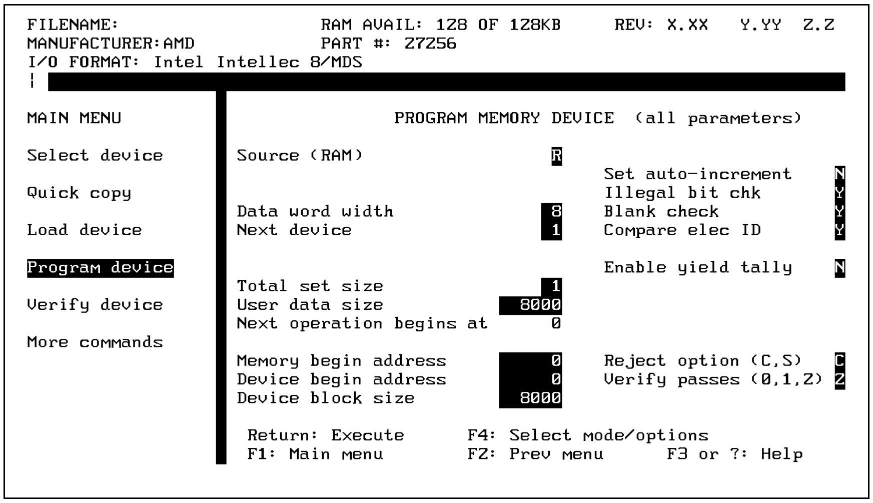
What Is the Difference?

Both screens let you change programming parameters. The All Parameters screen contains all the programming parameters supported by the selected device. The Non-default parameters screen contains a simplified subset of the programming parameters supported by the selected device.

To move between the two screens, press **F4**. If you were looking at the Non-default screen, you are now looking at the All Parameters screen. (Likewise, if you were looking at the All Parameters screen, you are now looking at the Non-default screen.) As you can see, there is roughly a 2:1 difference in the number of parameters between the two screens.

The All Parameters screen is shown in Figure 3-58.

Figure 3-58
The All Parameters Screen



```
FILENAME:                    RAM AVAIL: 128 OF 128KB     REV: X.XX   Y.YY  Z.Z
MANUFACTURER:AMD             PART #: 27256
I/O FORMAT: Intel Intellec 8/MDS
|

MAIN MENU                          PROGRAM MEMORY DEVICE  (all parameters)

Select device      Source (RAM)                R
                                                      Set auto-increment     N
Quick copy                                            Illegal bit chk        Y
                   Data word width             8      Blank check            Y
Load device        Next device                 1      Compare elec ID        Y

Program device                                        Enable yield tally     N
                   Total set size              1
Verify device      User data size           8000
                   Next operation begins at    0
More commands
                   Memory begin address        0      Reject option (C,S)    C
                   Device begin address        0      Verify passes (0,1,2)  2
                   Device block size        8000

                    Return: Execute      F4: Select mode/options
                    F1: Main menu        F2: Prev menu      F3 or ?: Help
```

About Parameters

Parameters are user-definable fields that determine what UniSite-*xpi* does. Parameters either qualify or quantify UniSite-*xpi*'s actions. Qualifying parameters, such as Illegal Bit Check, control whether or not UniSite-*xpi* performs an operation. Quantifying parameters, such as Block Size, give UniSite-*xpi* a range or variable to use in an operation.

Which Screen Should I Use?

During the course of normal programming, you usually only need the parameters on the Non-default screen. But if you want to do some more complicated programming operations, you need to change some of the parameters on the All Parameters screen. A complicated programming operation might be if you wanted to program only part of a device.

For this Session, view the Non-default screen. If the All Parameters screen is displayed, press **F4** to switch to the Non-default screen.

Programming the Device

For this session, you do not need to set any programming parameters, so you are ready to program the data file into the device.

To begin programming, press **Enter**. After UniSite-*xpi* has programmed the device, the following message appears in the message bar:

```
OPERATION COMPLETE: Sumcheck = xxxxxxxx (8-bit)
```

where *xxxxxxxx* represents the 8-digit, 8-bit sumcheck of the data programmed into the device in the socket.

Note: For 16-bit devices, the checksum can be calculated and displayed as either 8-bit or 16-bit. See Chapter 4, "Commands," for more information on setting the checksum parameter.

Review

Load the data into RAM, then select Program Device from the Main Menu. You will see one of two parameter-entry screens: the simple Non-default screen or the complex All Parameters screen. Press **F4** to toggle between the two parameter-entry screens. Change the parameters if necessary. Press **ENTER** to begin the programming.

When UniSite-*xpi* is finished programming, it displays an 8-digit sumcheck of the data programmed into the device. (UniSite-*xpi* displays a 4-digit sumcheck when you program a logic device.)

Session 21: Verifying a Device

Once you have programmed a device, you can perform a number of different device checks and programming tests on the device. For the rest of this Session we will refer to device checks and programming tests as verify operations.

During this Session, you will verify a device with a data file stored in RAM.

Before You Begin

Make sure you have completed Session 1, an introduction to the UniSite-*xpi* interface. Also make sure you have completed Session 9, which covers programming a device.

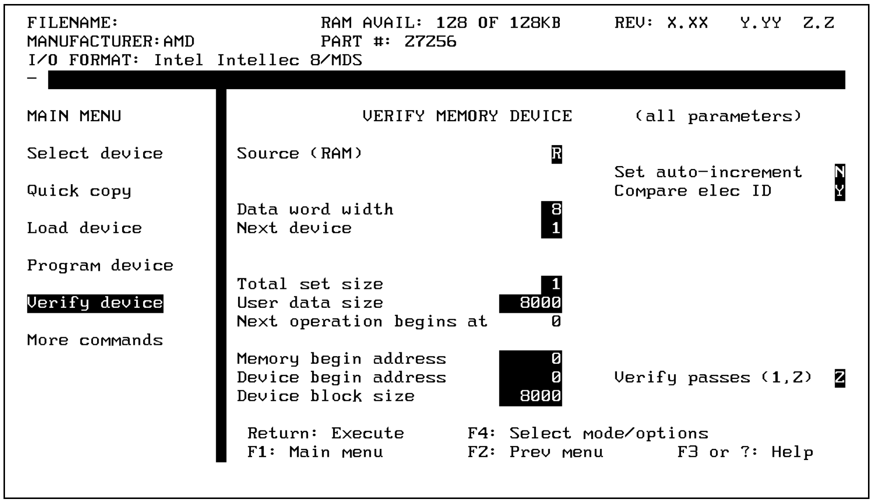
Can I Use Another Device?

If you do not have an AMD 27256 (the device we are going to use for this Session), then go to the Select Device screen and select the device you are going to program. Keep in mind that the device you select might not have the same capabilities as the AMD 27256. For example, the AMD 27256 supports Electronic ID while the Hitachi 27256 does not.

Parameter Screens

Select Verify Device from the Main Menu. The dialog window displays the Verify Memory Device screen, shown in Figure 3-59.

Figure 3-59
The All Parameters Verify Screen



```
FILENAME:                     RAM AVAIL: 128 OF 128KB     REV: X.XX   Y.YY  Z.Z
MANUFACTURER:AMD              PART #: 27256
I/O FORMAT: Intel Intellec 8/MDS
-

MAIN MENU                    VERIFY MEMORY DEVICE      (all parameters)

Select device      Source (RAM)                  R
                                                    Set auto-increment   N
Quick copy                                          Compare elec ID      Y
                   Data word width               8
Load device        Next device                   1

Program device
                   Total set size                1
Verify device      User data size             8000
                   Next operation begins at      0
More commands
                   Memory begin address          0
                   Device begin address          0  Verify passes (1,2)  2
                   Device block size          8000

                    Return: Execute      F4: Select mode/options
                    F1: Main menu        F2: Prev menu       F3 or ?: Help
```

Look at the Verify Memory Device screen. At the top of the dialog window, you see one of the following:

```
VERIFY MEMORY DEVICE (all parameters)

VERIFY MEMORY DEVICE (non-default)
```

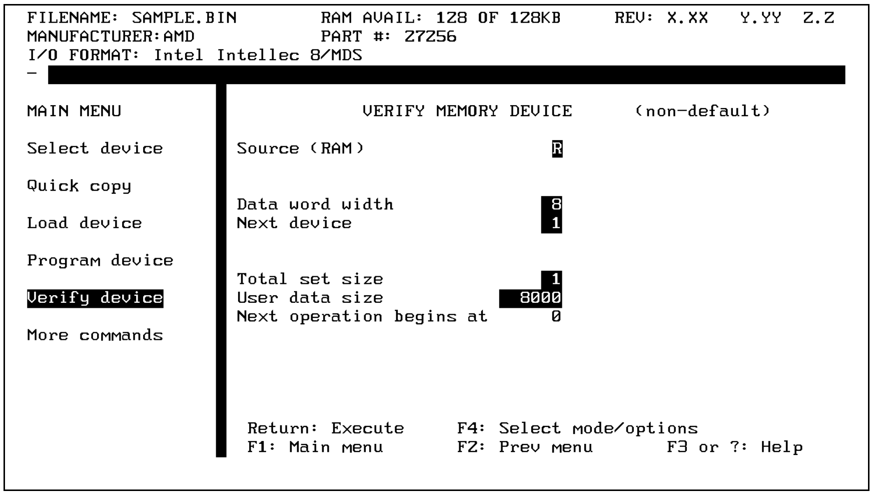
As with the programming parameters screens, the parameter screens let you change programming parameters. The All Parameters screen contains all the verify parameters supported by the selected device. The Non-default parameters screen contains a simplified subset of the verify parameters supported by the selected device.

Which Should I Use?

During the course of verifying a device, you will usually need only the parameters on the Non-default screen. But if you want to do some more complicated verify operations, you will need to change some of the parameters on the All Parameters screen. A complicated verify operation might involve, for instance, verifying only part of a device.

For this Session, you should be looking at the Non-default screen. If the All Parameters screen is displayed, press **F4** to switch to the Non-default screen, which is shown in Figure 3-60.

Figure 3-60
The Non-default Verify Parameters Screen



```
FILENAME: SAMPLE.BIN        RAM AVAIL: 128 OF 128KB     REV: X.XX   Y.YY  Z.Z
MANUFACTURER:AMD            PART #: 27256
I/O FORMAT: Intel Intellec 8/MDS
-
MAIN MENU                        VERIFY MEMORY DEVICE       (non-default)

Select device       Source (RAM)                   R

Quick copy
                    Data word width                8
Load device         Next device                    1

Program device
                    Total set size                 1
Verify device       User data size              8000
                    Next operation begins at       0
More commands




                     Return: Execute     F4: Select mode/options
                     F1: Main menu       F2: Prev menu       F3 or ?: Help
```

What Happens When I Change Parameters?

Normally, User data size defaults to the size of the selected device. But if you have performed a previous device operation, UniSite-*xpi* sets this value, and other values, to the parameters specified in the previous device operation. Other parameters affected are Block Size and Begin Address.

Setting the Verify Parameters

If you just completed the previous Session, Program a Memory Device, and then continued immediately with this Session, you probably do not need to set any parameters. But, to be sure, check the parameters shown in Figure 3-60 against the parameters you see on your UniSite-*xpi*.

Note: Remember that the displays may look different if you are using a device other than an AMD 27256.

Verifying the Device

Now that you have set the verify parameters, you are ready to verify the data programmed into the device.

To begin the verify, press **ENTER**. If the verify operation completes successfully, the following message is displayed in the message bar:

```
OPERATION COMPLETE: Sumcheck = xxxxxxxx (8-bit)
```

where *xxxxxxxx* represents the 8-digit, 8-bit sumcheck of the data in the device.

Review

When you select Verify Device from the Main Menu, you see one of two parameter-entry screens: the simple Non-default screen or the complex All Parameters screen. Press **F4** to toggle between the two parameter-entry screens.

When you have entered the verify parameters, press **ENTER** to begin the verify operation. When UniSite-*xpi* is finished verifying, it displays an 8-digit sumcheck of the data in the device. (UniSite-*xpi* displays a 4-digit sumcheck when you verify a logic device.)

4. Commands

This chapter describes the commands you can access from UniSite-*xpi*'s menus.

Menu Organization

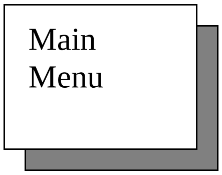
The interrelation of UniSite-*xpi* menus and commands is shown in the command tree in Figure 4-1.

Menu Maps

Each command description includes a map, which displays part of the command tree representing your location and shows you the path to the command.

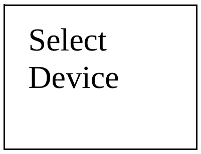
The maps, read from left to right, are composed of three or four boxes, each representing a screen type, as described below.

Shaded Box



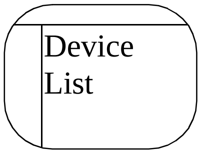
Main Menu Screen
A shaded box represents a top-level command, either a Main Menu command or the More Commands menu. These are the most frequently used commands. The shaded box is the start of your path to a particular command.

Box



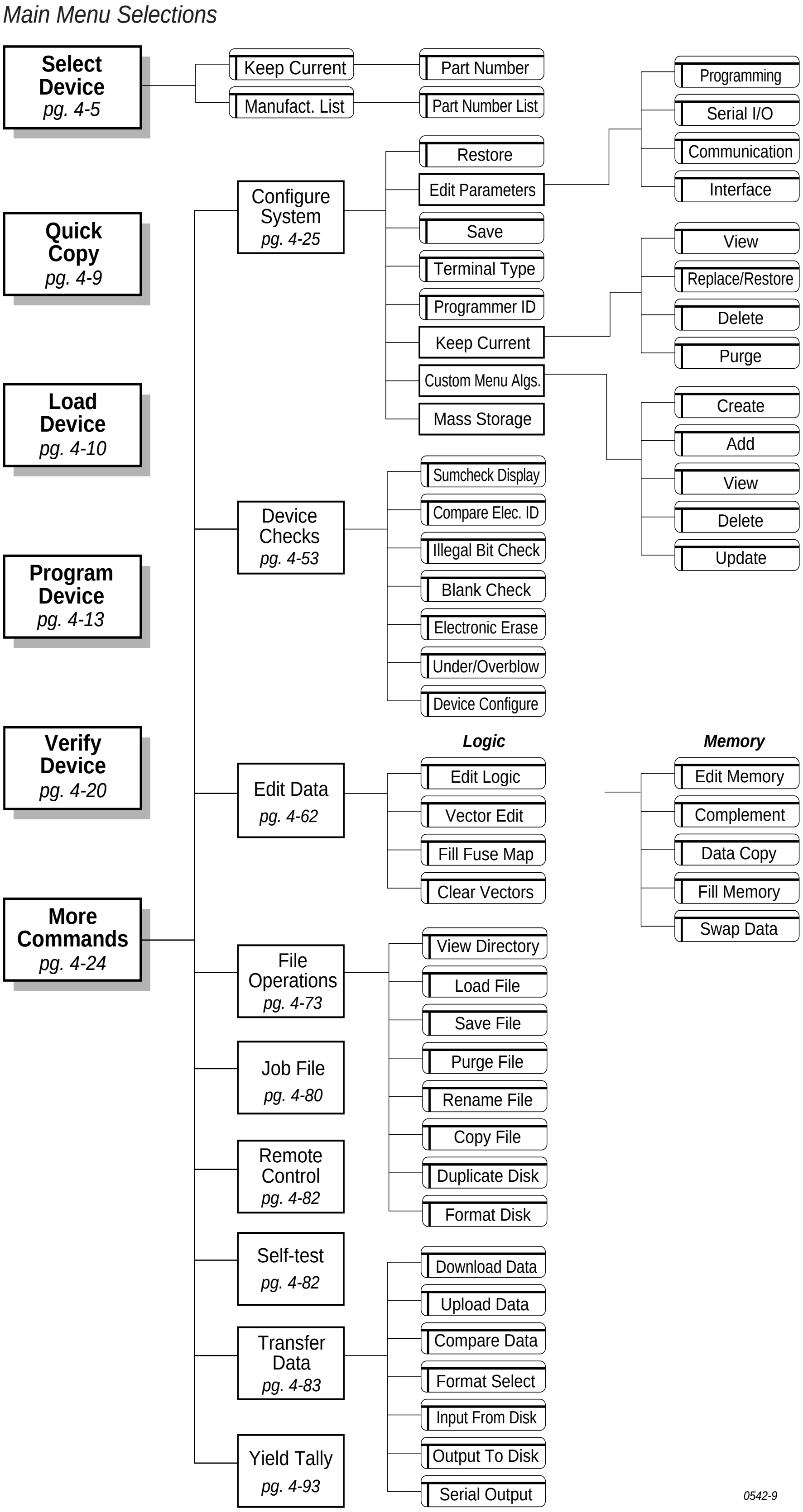
Other Menu Screens
The next step on the path to a command is a box, representing all other menu screens, most of which are accessible from the More Commands screen. When you select an item in a box, a T-box item or another screen is displayed.

T-Box



Dialog Window Screen
Selecting the item shown in a T-box brings you to your final destination, a dialog window.

Figure 4-1
The Command Tree



Overwriting User RAM

The following operations use User RAM as a temporary storage buffer, overwriting any data that may have been there previously.

- Duplicate disk
- Copy file
- Serial output from disk file
- Download data to disk file
- Self-test User RAM
- Upload data from disk file
- Output data to disk from disk file
- Input data from disk to disk file
- Compare data to disk file
- Create and Add Custom Menu Algorithms

Factory Default Settings

UniSite-*xpi*'s system parameters are initialized to certain settings at the factory. You can restore these factory defaults at any time by selecting configuration file number **0** from the More Commands / Configure System / Restore menu. From the Restore System Parameters menu, press **0**, **ENTER** to re-elect the factory settings. The power-up defaults (configuration file number **1**) are the same as the factory defaults when the unit is shipped from Data I / O.

Parameter	**Factory Default Setting**
Algorithm Type	D
Blank Check	Yes
Checksum Calculation	8
Continuity Check	Yes
Compare Electronic ID	Yes
Data source / destination	RAM
Data Word Width	8
Device Begin Address	0
Device Block Size	1000
Display device footnote	Yes
Enable download echo	No
Enable security fuse	No
Enable special data	No
Enable terminal beep	Yes
Enable yield tally option	No
EE bulk erase option	No
EOF delimiter flag (download)	No
EOF delimiter flag (upload)	No
File delimiter character (download)	1A (CTRL-Z)
File delimiter character (upload)	1A (CTRL-Z)
Filename	Blank
Fill RAM before downloading	No
Fill RAM with data (00 to FF)	00
High speed download	Yes

Parameter	Factory Default Setting
Host command (download)	Blank
Host command (upload)	Blank
Illegal Bit Check	Yes
Instrument control code (0, 1, 2)	0
I/O Address Offset	FFFFFFFF
I/O translation format	0 (no default)
I/O timeout	30 seconds
JEDEC I/O translate DIP/LCC option	Yes
Logic verification (all, fuse, vector)	All
Main menu job files	No
Manufacturer	Blank (no default)
Memory Begin Address	0
Number of lines between form feeds	0
Number of nulls	0
Odd/Even Byte Swap	No
Part number	(no default)
Power on CRC mode	No
Power up user RAM test	Yes
Program security fuse	No
RAM Device Selection	No
Reject option (commercial or single)	Commercial
Remote Off code	0
Remote On code	0
Remote serial port configuration	9600 baud, 8 data bits, no parity, 1 stop bit, active CTS/DTR
Security Fuse Data (0 or 1)	0
Serial set auto increment flag	No
Simple/complex parameter screen	Simple
Terminal serial port configuration	9600 baud, 8 data bits, no parity, 1 stop bit, active CTS/DTR
Terminal type	VT-100
Transmit pacing	0
User Menu Port*	T
User Data Size	0
Upload wait	0 seconds
Upload destination/download source	Remote
Upload record size	16
Verify Data Format	Hex
Verify Passes	2

* This parameter is not restored when the Restore Configuration operation is performed. However, it is used at powerup if it is saved as a powerup parameter.

Select Device

Before you can perform any device-related operations with the programmer, you must select the device you are using.

Before You Select a Device

Before selecting a device, set the **Algorithm Type** parameter to the type of algorithm that matches the device you wish to select.

To set the Algorithm Type, from the Main Menu select **More commands/ Configure system/Edit/Programming** (refer to “Algorithm Type” on page 4-27 for more information).

Note: You can press **F1** *to return to the Main Menu from any submenu.*

The Programming Parameters screen appears. On this screen, set the Algorithm Type to one of the following:

- **D** (default)—Set the algorithm type to **D** if you wish to select a device from the **Manufacturer List**. The Manufacturer List contains almost all of the devices supported by your programmer (the programming algorithms for these devices reside on your Algorithm disks). Selecting a device from this list is a two-step process: first, select the device manufacturer and then the device part number.
- **E** (extended algorithm)—Set the algorithm type to **E** if you wish to select a device from an **Extended Algorithm** list. An Extended Algorithm list contains devices that use extended algorithms for programming (the programming algorithms for these devices reside on a disk that contains the extended algorithms). Selecting a device from this list is a two-step process: first, select the device manufacturer and then the device part number.
- **K** (Keep Current)—Set the algorithm type to **K** if you wish to select a device from a **Keep Current** list. A Keep Current list contains devices that use Keep Current algorithms for programming (see “Keep Current” on page 4-43 for more information).

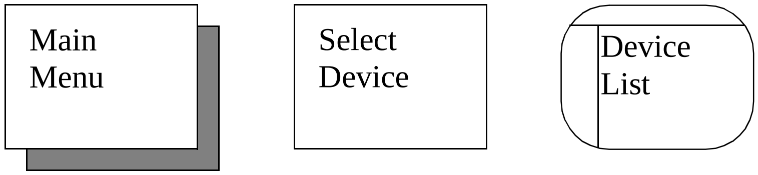
Note: You can also access Keep Current devices by setting the algorithm type to ***D*** *(default) and selecting* `KEEP CURRENT` *from the manufacturers list.*

- **C** (Custom Menu)—Set the algorithm type to **C** if you wish to select a device from a **Custom Menu** list. A Custom Menu list contains devices that you included in a Custom Menu (see “Custom Menu Algs” on page 4-46 for more information).

After setting the Algorithm Type parameter, press **F1** to return to the Main Menu.

Note: See “Save System Parameters” on page 4-38 for instructions on how to save your Algorithm Type and other parameter changes as power-up defaults.

Selecting a Device



To select a device, do the following:

1. From the Main Menu, select **Select Device**. (If you are not in the Main Menu, press **F1** to return to the Main Menu.)
2. You might be prompted to insert a disk. If so, insert the disk you are prompted for.
3. Depending on which Algorithm Type is selected, one of the following screens appears:
   - The **Device Manufacturer List** screen appears if the algorithm type is set to **D** (default). Selecting a device from this screen is a two-step process: first, select the device manufacturer and then the device part number. Continue with steps 4 and 5 of this procedure.
   - The **Extended Algorithm** screen appears if the algorithm type is set to **E** (extended). Selecting a device from this screen is a two-step process: first, select the device manufacturer and then the device part number. Continue with steps 4 and 5 of this procedure.
   - The **Keep Current** screen appears if the algorithm type is set to **K** (Keep Current). Skip to step 5 of this procedure.
   - The **Custom Menu** screen appears if the algorithm type is set to **C** (Custom Menu). Skip to step 5 of this procedure.

Note: If a different screen than the one you want appears, refer to "Before You Select a Device" on page 4-5 for instructions on how to select an Algorithm Type.

4. If your Algorithm Type is set to **D** (default) or **E** (Extended Algorithm), a list of device manufacturers supported by the programmer appears. The list also contains the following non-manufacturer choice:

   KEEP CURRENT Selecting KEEP CURRENT allows you to select a device whose algorithm was downloaded using Keep Current. This choice is the same as selecting **K** (Keep Current) as the Algorithm Type.

Note: Depending on what version of software you have, additional non-manufacturer choices may be available.

Select a manufacturer (or one of the non-manufacturer choices), by locating the manufacturer choice and entering the number that appears next to that choice in the manufacturer field. For example, if 2)KEEP CURRENT is a choice, type the number 2 for the manufacturer and press **ENTER**.

When you select a device manufacturer, you can set the device type filter to the appropriate setting for the device you are using. As the names indicate, Logic Only displays only logic devices, Memory & Emicros displays memory and emicros, and All displays all device types. Press **SPACE** to cycle through the filter types. The list of manufacturers displayed will change based on your choice.

The upper-right corner of the manufacturer screen displays how many screens (pages) of manufacturers there are and what page you are on. Press **CTRL** + **N** to go to the next page of manufacturers. Press **CTRL** + **P** to go to the previous page of manufacturers. When the screen repaints, you see the next (or previous) list of manufacturers.

5. A screen appears with a list of devices. To select the device, locate the device you are going to use and enter the number that appears next to that device.

   If you do not see the device you are looking for, press **CTRL** + **N** to advance to the next page of the device list. Press **CTRL** + **P** if you want to return to the previous page. After you have entered the number for the device you want, press **ENTER**. The programmer returns to the Main Menu and the status window shows the device you selected.

After You Select a Device

After you have selected a device, the following occurs:

Device Information

If there is online device-specific information for the selected device, the programmer automatically displays this information. If there is more than one screen of device specific information, press **CTRL** + **N** to view the next screen of information. Press **CTRL** + **P** to view the previous screen of information. Exit the screen by pressing **F1** or **F2**.

If the **Display Device Footnotes** parameter (in the More commands / Configure system / Edit / Programming screen) is set to **N**, device-specific information is not displayed. Instead the following message appears in the message bar if there is online device-specific information for the selected device.

```
Hit F3 or ? to view device specific message.
```

To view the online information, press **F3** or **?**. The screen clears, and the programmer displays the device-specific information.

Note: If only the footnote number appears in the message bar, the programmer could not access the ***devfnote.sys*** *file. Check to see if the correct disk is inserted into the programmer's drive.*

Footnote (device-specific) information can also be found in the ***Device List****.*

Device-related Selections Available

The manufacturer and part number of the device you selected appears in the status window, and the programmer displays only screens related to the type of device you selected. For example, if you choose a logic device, you will see only screens that are required to load, program, edit, and verify a logic device.

Also, some devices support special functions, such as the Program Signature and XNOR table. If the device you select supports a particular function, parameter entry fields associated with that function are displayed.

Cross Programming

Cross programming allows a single generic programmable logic device (PLD) to be configured as any one of many PLD architectures. Consequently, the generic device can take on the function of many subset devices. The term generic PLD is used to identify the superset device, such as a 16V8 generic PLD, which can be configured as a 16R4, 16R8, or a 16L8.

The generic PLD and the subset devices it can support are not restricted to the same manufacturer. For example, a 16V8 generic PLD from manufacturer A can be programmed using a fuse pattern originally designed for a 16L8 from manufacturer B. The cross-programming feature allows you to avoid recompiling source code for the generic PLD if the appropriate fuse pattern is available for a subset part.

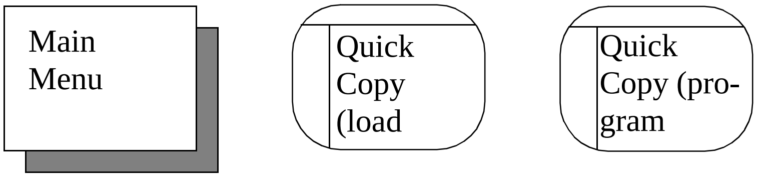
To view the subset devices, select a manufacturer with the **XPGM** extension. This screen lists all devices that can be replaced by the generic PLD of the selected manufacturer.

After selecting a device with the **XPGM** extension, load the fuse map of the subset PLD into User Memory by using either a Load From Device operation or a Download of a JEDEC file.

Select a manufacturer with an **XPGM** extension and press **ENTER**. Then, from the Part Menu for Manufacturer screen, select the appropriate device, such as **16V8 as 16L8** if a 16L8 fuse map has been loaded and a 16V8 generic PLD is to be programmed. After making your selection, press **ENTER** and the Main Menu appears. Select the Program Devices screen and `16V8 as 16L8` will be displayed in the PART # field at the top of the screen. Insert the 16V8 and press **ENTER**. The 16V8 is programmed as a 16L8.

Quick Copy

The Quick Copy command allows you to load data from a master device and program that device data into a target device quickly and easily.



Note: Before you use the Quick Copy command, you must first select a device.

To use the Quick Copy command, follow these steps:

1. Select the device that contains the data you are going to load. For more information, see the explanation of the Select Device command earlier in this chapter.
2. Select the Quick Copy command by pressing **Q** from the Main Menu. The dialog window displays the Quick Copy screen. The message bar displays the following message:

   ```
   Insert master device. Hit return.
   ```

3. Insert the master device and lock it into place.
4. Press **ENTER** to begin loading the master data into RAM. When the data in the master device has been loaded, UniSite-*xpi* displays

   ```
   OPERATION COMPLETE. Sumcheck = xxxxxxxx (8-bit) Hit
   return
   ```

Note: (Legacy) If you are using the Quick Copy function with SetSite, you must press Return again at this point. This allows you time to review the screen and sumcheck information.

5. Remove the master device and press **ENTER**. UniSite-*xpi* displays

   ```
   Insert blank device. Hit return.
   ```

6. Insert the target device (the device you want to program) and lock it into place.
7. Press **ENTER** to program the device.
8. When the programming operation is complete, UniSite-*xpi* displays

   ```
   OPERATION COMPLETE. Sumcheck = xxxxxxxx (8-bit) Hit
   return
   ```

9. The Quick Copy operation is complete. Remove the device.
10. To program another device, return to step 6.

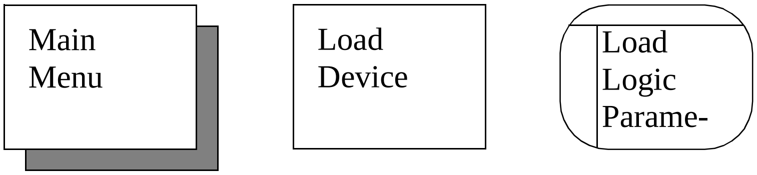
Note: (Legacy) If you are using the SetSite module, you may also use the Quick Copy command to load data for gang programming. Data Word Width will automatically be set to Device Word Width.

Load Device

The Load Device command allows you to copy data from a master device into RAM. Depending on whether you select a logic or memory device, either the Load Logic Device screen or the Load Memory Device screen will appear when you select this command. The Load Logic Device and Load Memory Device commands are explained below.

Load Logic Device

If you select a logic device and then select the Load Device command, the Load Logic Device menu will appear.



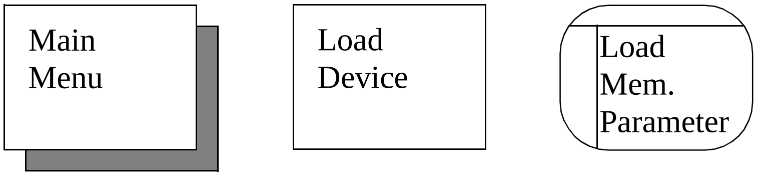
To load user memory with data from a logic device, follow these steps:

1. Select and socket a logic device.
2. Select Load Device from the Main Menu. The Load Logic Device screen appears.
3. Press **ENTER** to begin the loading.
4. When the load operation is complete, UniSite-*xpi* displays the following message in the message bar:

   ```
   OPERATION COMPLETE. Sumcheck = xxxx (8-bit)
   ```

Load Memory Device

If you select a memory device and then select the Load Device command, the Load Memory Device menu will appear.



To load user memory with data from a memory device, follow these steps:

1. Select and socket a memory device.
2. Select Load Device from the Main Menu. The Load Memory Device screen appears.
3. Specify the parameters you want, then press **ENTER** to begin the loading.
4. When the load operation is complete, UniSite-*xpi* displays the following message in the message bar:

```
OPERATION COMPLETE. Sumcheck = xxxxxxxx (8-bit)
```

The following parameters can be specified on the Load Memory Device screen:

- **Data Word Width**—Sets the word width of data to be loaded. For 8-bit (or larger) devices, the minimum word width is equal to the device width, and the maximum word width is 64. For 4-bit devices, your word width choices are 4, 8, 16, and 32. This value should match the data bus word width in the target system for the device being programmed. The default value is the device word width.
- **Next Device**—Designates the next device (next set member) in the set. For example, if you are using 8-bit devices and have specified a word width of 16 bits on the Load Memory Device screen, it requires two devices to store each 16-bit word. Typing **1** for the next set member directs UniSite-*xpi* to load the first device in the set at even-address bytes of the memory block. Typing **2** directs UniSite-*xpi* to load the second device at odd address bytes of the memory block.
- **Total Set Size**—Specifies how many virtual devices are in the set for device operations. For example, if you are loading 16-bit wide data from two 8-bit wide devices, your virtual device still equals one (one 16-bit virtual device). Any number between 1 and 99 can be entered. Automatic Set Size calculation will be attempted when any of the following parameters are changed: Device width, Data Word Width, Device Block Size or User Data Size. Total Set Size can be defined by the following equation:

$$\text{TOTAL SET SIZE} = \frac{\text{USER DATA SIZE}}{\text{DEVICE BLOCK SIZE}\left(\dfrac{\text{DATA WORD WIDTH}}{\text{DEVICE WIDTH}}\right)}$$

- **User Data Size**—Specifies the hexadecimal size, in bytes, of the data block used to load from the device to the destination. This value is normally equal to the device size or to a multiple of the device size for loading a set. If 0 or a value less than the device size is entered, the User Data Size is reset to the device size for the load. User Data Size works with Total Set Size to determine the total amount of bytes to load from a set of devices.
- **Next Operation Begins At**—This read-only parameter shows where in user memory the next data byte will be loaded. This value is calculated from the Data Word Width, Device Block Size, Memory Begin Address, device width, and next set member parameters.

Optional Parameter Screens

There are two types of parameter-entry screens: simple and complex. UniSite-*xpi* defaults to displaying the simple, Non-default Parameters screen. If you want to view the complex, All Parameters screen, press **F4** from the Non-default Parameters screen. If you have changed any of the default parameters, those changed parameters will also show up on the Non-default Parameters screen.

The following parameters are found on the complex, All Parameters Program Memory Device screen:

- **Memory Begin Address**—Specifies the first address, in hex, where the first byte of data is loaded from a device. If the destination is RAM, it is a beginning RAM address. If the destination is Disk, it is a beginning disk file address. The Memory Begin Address must be an even address if you have selected a 16-bit device. The default address is 0.
- **Device Begin Address**—Specifies the first hex master device address that will be loaded. The default value is 0.
- **Device Block Size**—Specifies the size, in hex, of device data used in device operations. After you have selected a device, the Device Block Size is set automatically to the device size and normally does not need to be changed. Device Block Size is set to a smaller value if the Device Begin Address is nonzero. This parameter can be changed if desired. Entering a zero sets the Device Block Size equal to the device size.
- **Set Auto-increment**—This option, used in serial set mode, automatically directs UniSite-*xpi* to the next block in the set that is to be loaded. For example, if you have four 1K x 8 devices to load into a 4K x 8 block of memory, using the auto-increment option directs UniSite-*xpi* to point to the first memory address of the next 1K block after each device had been loaded. For single device operations, this feature should be disabled and the Next Device parameter should be set to **1**.

*Note: Items with an * (asterisk) are visible on the screen only if the selected device supports the feature.*

* **Compare Electronic ID** (Y,N)—Compares the electronic ID of the socketed device against the electronic ID of the selected algorithm.

* **Odd/Even Byte Swap** (Y,N)—When enabled, allows the Most Significant Bytes (MSB) and the least significant bytes (LSB) of 16-bit words to be swapped when data is loaded from a 16-bit device. When disabled, the data from a 16-bit device is loaded into User RAM with the MSB stored at an odd memory address. When enabled, the MSB is stored at an even memory address.

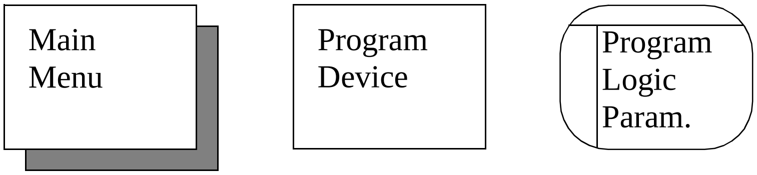
Program Device

The Program Device command allows you to copy data from RAM. Depending on whether you select a logic or memory device, either the Program Logic Device screen or the Program Memory Device screen appears when you select this command. The Program Logic Device command is explained first, and the Program Memory Device command is explained next.

Before you can program a device, you need to load the programming data into RAM.

Program Logic Device

If you select a logic device and then select the Program command, the Program Logic Device menu appears.



To program a logic device, follow these steps:

1. Select and socket a logic device.
2. Select Program Device from the Main Menu. The Program Logic Device screen appears.
3. Specify the parameters you want, then press **ENTER** to begin the programming.
4. When the programming is complete, UniSite-*xpi* displays the following message in the message bar:

```
OPERATION COMPLETE. Sumcheck = xxxx (8-bit)
```

The following parameters can be specified on the Program Logic Device screen:

- **Security Fuse Data** (0,1)—To program the security fuse, set this parameter to **1** and set the Program Security Fuse parameter to **Y**. This parameter defaults to **0**, which disables programming of the security fuse.

- **Program Security Fuse** (Y,N)—Enables or disables the programming of the security fuse. To program the security fuse, set this parameter to **Y** and set the Security Fuse Data parameter to **1**. This parameter defaults to **N**, which disables programming of the security fuse.

Optional Parameter Screens

There are two types of parameter-entry screens: simple and complex. UniSite-*xpi* defaults to displaying the simple, Non-default Parameters screen. If you want to view the complex, All Parameters screen, press **F4** from the Non-default Parameters screen. If you have changed any of the default parameters, those changed parameters will also appear on the Non-default Parameters screen.

The following parameters are found on the complex, all parameters Program Logic Device screen.

- **Illegal Bit Check** (Y,N)—Enables or disables the illegal-bit test. When enabled, this test compares data in a device against data in UniSite-*xpi*'s RAM to determine if the device has already-programmed locations of incorrect polarity. For example, UniSite-*xpi* returns an illegal-bit error in the following situation: data in RAM indicates a specific bit should be in an unprogrammed state while the corresponding bit in the device is in a programmed state. The device cannot be programmed if UniSite-*xpi* detects an illegal bit. This parameter is enabled by default.
- **Blank Check** (Y,N)—Enables or disables the blank check test. A Blank Check test checks a device for programmed bits. This parameter defaults to **Y**, which enables the test.
- **Enable Yield Tally** (Y,N)—When enabled, directs UniSite-*xpi* to keep a running tally of the programming yields for the last sixteen types of devices programmed. These totals show how many devices passed and failed, and what specific errors, if any, have occurred. This parameter defaults to **N**, which disables the test. A more complete description of this feature can be found in the "Yield Tally" section of this chapter.

- **Logic Verification** (A,F,V)—Specifies the type of logic verification to perform after programming. Select from Fuse Verification **F**, Vector Verification **V**, or All **A**. Fuse Verification checks the fuse pattern programmed into the device with the pattern in UniSite-*xpi*'s memory. Vector Verification functionally tests the device, using structured test vectors stored in memory. **A** directs UniSite-*xpi* to perform both fuse verification and vector verification. Press **SPACE** to step through the three choices. This parameter defaults to **A**.

  UniSite-*xpi* does not support vector testing for logic devices with more than 84 pins. Attempting to perform a vector test on a device with more than 84 pins will yield the following message:

  ```
  OPERATION COMPLETE. Sumcheck = hhhhhhhh (Vector test not
  supported)
  ```

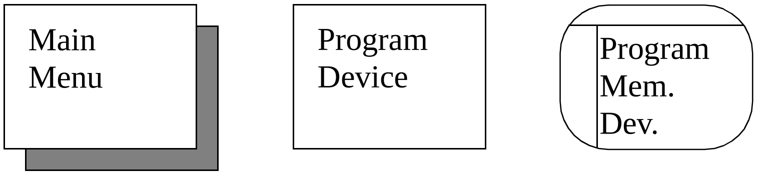
- **Verify Passes** (0,1,2)—Selects the number of times to test the device. **0** directs UniSite-*xpi* not to test the device. **1** directs UniSite-*xpi* to test the device once at the device manufacturer's nominal Vcc. **2** directs UniSite-*xpi* to verify the device at the device manufacturer's recommended high and low Vcc levels. This parameter defaults to **2**.

- **Reject Option** (C,S)—Selects the number of times the device is pulsed with programming voltage before it is rejected as unprogrammable. **C** selects the number of pulses specified by the manufacturer. **S** selects either a one-pulse or the military-specification number of programming pulses. Unless you are programming devices to a strict military specification, you should leave this option set at **C**. This option defaults to **C**.

Program Memory Device

If you select a memory device and then select the Program command, the Program Memory Device screen appears.

Note: If device configuration is supported for the selected device, you can set the program, erase, and protect separately for each sector using the More Commands/Device Checks/Device Configure command.



To program a memory device, follow these steps:

1. Select and socket a memory device.
2. Select Program Device from the Main Menu. The Program Memory Device screen appears.
3. Specify the parameters you want. Press **ENTER** to begin the programming.
4. When the programming is complete, UniSite-*xpi* displays the following message in the message bar:

```
OPERATION COMPLETE. Sumcheck = xxxxxxxx (8-bit)
```

The following parameters can be specified on this screen:

- **Data Word Width**—Sets the word width of the data to be programmed. For 8-bit (or larger) devices, the minimum word width is equal to the device width and the maximum is 64. For 4-bit devices, your word width choices are 4, 8, 16, and 32. This value should match the data bus word width in the target system for the device being programmed.
- **Next Device**—Designates the next device in the set. For example, if you are using 8-bit devices and have specified a word width of 16 bits on the Program memory device screen, two devices are required to store each 16-bit word. **1** directs UniSite-*xpi* to program the first device in the set with even-numbered addresses of the memory block. **2** directs UniSite-*xpi* to use odd-numbered addresses.
- **Total Set Size**—Specifies how many virtual devices are in the set for device operations. For example, if you are programming 16-bit wide data into two 8-bit wide devices, your virtual device still equals one (one 16-bit virtual device). You can enter any number between 1 and 99. Automatic Set Size calculation is attempted when you change any of the following parameters: Device width, Device Block Size or User Data Size. Total Set Size is defined by the following equation:

$$\text{TOTAL SET SIZE} = \frac{\text{USER DATA SIZE}}{\text{DEVICE BLOCK SIZE}\left(\dfrac{\text{DATA WORD WIDTH}}{\text{DEVICE WIDTH}}\right)}$$

0987-1

- **User Data Size**—Specifies the hexadecimal size, in bytes, of the data block to program into a device. Normally, this value is equal to the device size or to a multiple of the device size for set programming. Entering 0 sets the User Data Size to the device size. User Data Size works with Total Set Size to determine the total amount of bytes to program into a set of devices.
- **Next Operation Begins At**—This read-only parameter shows what address in user memory contains the next data byte to be programmed. This value is calculated from the Data Word Width, Device Block Size, Memory Begin Address, device width, and next set member parameters.

*Note: Items with an * (asterisk) are visible on the screen only if the selected device supports the feature.*

* **Program Security Fuse** (Y,N)—Enables or disables the programming of the security fuse. To program the security fuse, set this parameter to **Y**. This parameter defaults to **N**, which disables programming of the security fuse.

* **Program Signature**—Available on only a few devices, the program signature is a user-definable field that allows the user to program data into the program signature array.

* **Software Data Protection** (Y,N)—When enabled, prevents writing to a device.

Optional Parameter Screens

There are two types of parameter-entry screens: simple and complex. UniSite-*xpi* defaults to displaying the simple, Non-default Parameters screen. If you want to view the complex, All Parameters screen, press **F4** from the Non-default Parameters screen. If you have changed any of the default parameters, those changed parameters will also appear on the Non-default Parameters screen.

The following parameters are found on the complex, all parameters Program Memory Device screen.

- **Memory Begin Address**—Specifies the first address, in hex, of the first byte of data to be programmed. If the Source is RAM, it is a beginning RAM address. If the Source is Disk, it is a beginning disk file address. The Memory Begin Address must be an even address if you have selected a 16-bit device. The default address is 0.

Note: (Legacy) For a SetSite operation, Memory Begin Address specifies the first address of the device in the first socket.

- **Device Begin Address**—Specifies the first hex device address that will be programmed. The default is 0.
- **Device Block Size**—Specifies the size, in hex, of device data used in device operations. At device selection, Device Block Size is set to the device size and normally does not need to be changed. It is also automatically set to a smaller value if the Device Begin Address is nonzero. This parameter can be changed if desired. Entering zero sets the Device Block Size equal to the device size.
- **Set Auto-increment**—When enabled, directs UniSite-*xpi* (in serial set programming mode) to the starting memory address of the next block in the set to be programmed. For example, if you have four 1K x 8 devices to program from a 4K x 8 block of data, using the auto-increment option directs UniSite-*xpi* to point to the first address of the next 1K block after each device has been programmed. For single device operations, this feature should be disabled and the Next Device parameter should be set to **1**.

*Note: Items with an * (asterisk) are visible on the screen only if the selected device supports the feature.*

* **Illegal Bit Check** (Y,N)—Enables or disables the illegal-bit test. This test compares data in a device against data in UniSite-*xpi*'s RAM to determine if the device has already-programmed locations of incorrect polarity. For example, UniSite-*xpi* returns an illegal-bit error in the following situation: data in RAM indicates a specific bit should be in an unprogrammed state while the corresponding bit in the device is in a programmed state. The device cannot be programmed if UniSite-*xpi* detects an illegal bit. This parameter is enabled by default.

- **Blank Check** (Y,N)—Enables or disables the blank check test. A Blank Check test checks a device for programmed bits. This parameter defaults to **Y**, which enables the test.

- * **Compare Electronic ID** (Y,N)—When enabled, compares the electronic signature of the device against the electronic signature of the selected algorithm.
- • **Enable Yield Tally** (Y,N)—When enabled, directs UniSite-*xpi* to keep a running tally of the programming yields for the last sixteen types of devices programmed. These totals show how many devices passed and failed, and what specific errors, if any, occurred. This parameter defaults to **N**, which disables the test. A more complete description of this feature can be found in the “Yield Tally” section of this chapter.
- * **Odd/Even Byte Swap** (Y,N)—When enabled, allows the Most Significant Bytes (MSB) and the Least Significant Bytes (LSB) of 16-bit words to be swapped when data is programmed into a 16-bit device. The data is programmed into a device retrieving the Most Significant Byte from an odd memory address when the flag is **N** and an even memory address when it is **Y**.
- • **Reject Option** (C,S)—Selects the number of times the device is pulsed with programming voltage before it is rejected as unprogrammable. **C** selects the number of pulses specified by the manufacturer. **S** selects either a one-pulse or the military-specification number of programming pulses. Unless you are programming devices to a strict military specification, you should leave this option set at **C**. This option defaults to **C**.
- • **Verify Passes** (0,1,2)—Selects the number of times to test the device. **0** directs UniSite-*xpi* not to test the device. **1** directs UniSite-*xpi* to test the device once at the device manufacturer’s nominal Vcc. **2** directs UniSite-*xpi* to verify the device at the device manufacturer’s recommended high and low Vcc levels. This parameter defaults to **2**.
- * **Erase EE Device** (Y,N)—Allows you to erase electronically-erasable PROMs. Before the programming cycle, UniSite-*xpi* checks the device and displays a warning if the device if non-blank. If you enable the erasing of the device, UniSite-*xpi* erases the device before programming the device.

Enhanced Security Fuse Capability

The enhanced security fuse capability for EMICRO parts allows Security Fuse Data to be stored in a data file. Currently, some devices, including the Intel 8742AH, support this capability. For more information, or to see if a device supports this capability, see the device manufacturer’s data book.

Security Fuse Data Field

The Security Fuse Data field cannot be restored using the More Commands / Configure System / Restore command. Instead, Security Fuse Data will be restored from a data file.

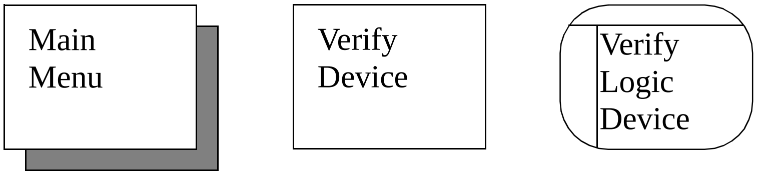
Verify Device

The Verify Device command compares data in a programmed device with data in RAM or in a disk file. Depending on which type of device is selected, either the Verify Logic Device screen or the Verify Memory Device screen appears when you select the Verify Device command.

Before you can verify a device, you need to load the data into RAM or select a disk file as your data source. RAM data can be loaded from a master device, from a disk file, or through the serial port.

Verify Logic Device

If you select a logic device and then select the Verify command, UniSite-*xpi*'s Verify Logic Device screen appears.



To verify a logic device, follow these steps:

1. Select and socket a logic device.
2. Press **ENTER** to begin the verifying process.
3. When the verify operation is complete, UniSite-*xpi* displays the following message in the message bar:

   ```
   OPERATION COMPLETE. Sumcheck = xxxx (8-bit)
   ```

Optional Parameter Screens

There are two types of parameter-entry screens: simple and complex. UniSite-*xpi* defaults to displaying the simple, Non-default parameters screen. If you want to view the complex, All Parameters screen, press **F4** from the Non-default Parameters screen. If you have changed any of the default parameters, those changed parameters will also show up on the Non-default Parameters screen.

The following parameters are found on the complex, All Parameters Verify Logic Device screen.

- **Logic Verification** (A,F,V)—Specifies the type of logic verification to perform. Select from Fuse Verification **F**, Vector Verification **V**, or All **A**. Fuse Verification checks the fuse pattern programmed into the device with the pattern in UniSite-*xpi*'s memory. Vector Verification functionally tests the device, using structured test vectors stored in memory. **A** directs UniSite-*xpi* to perform both fuse verification and vector verification. Press **SPACE** to step through the three choices. This parameter defaults to **A**.

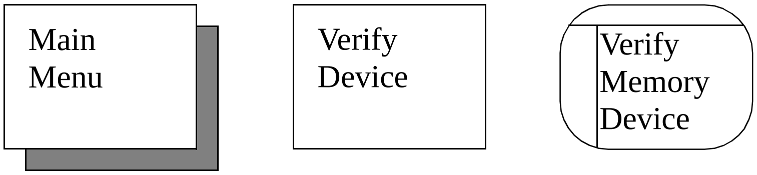
UniSite-*xpi* does not support vector testing for logic devices with more than 84 pins. Attempting to perform a vector test on a device with more than 84 pins will yield the following message:

```
OPERATION COMPLETE. Sumcheck = hhhhhhhh (Vector test not
supported)
```

- **Verify Passes** (0,1,2)—Selects the number of times to test the device. **0** directs UniSite-*xpi* not to test the device. **1** directs UniSite-*xpi* to test the device once at the device manufacturer's nominal Vcc. **2** directs UniSite-*xpi* to verify the device at the device manufacturer's recommended high and low Vcc levels. This parameter defaults to **2**.

Verify Memory Device

If you select a memory device and then select the Verify command, UniSite-*xpi*'s Verify Memory Device screen appears.



To verify a memory device, follow these steps:

1. Select and socket a memory device.
2. Specify the parameters you want. Then press **ENTER** to begin the verifying.
3. When the verify operation is complete, UniSite-*xpi* displays the following message in the message bar:

```
OPERATION COMPLETE. Sumcheck = xxxxxxxx (8-bit)
```

The following parameters can be specified in the Verify Memory Device screen.

- **Data Word Width**—Sets the word width of the data to be verified. For 8-bit (or larger) devices, the minimum word width is equal to the device word width and the maximum is 64. For 4-bit devices, your word width choices are 4, 8, 16, and 32. This value should match the data bus word width in the target system for the device being programmed.
- **Next Device**—Designates the next device in the set. For example, if you are using 8-bit devices and have specified a word width of 16 bits on the Verify Memory Device screen, then two devices are required to verify each 16-bit word. Typing **1** for the next set member directs UniSite-*xpi* to verify the first device in the set with even-numbered addresses of the memory block. Typing **2** directs UniSite-*xpi* to use odd-numbered addresses.
- **Total Set Size**—Specifies how many virtual devices are in the set for device operations. For example, if you are verifying 16-bit wide data into two 8-bit wide devices, your virtual device still equals one (one 16-bit virtual device). Any number between 1 and 99 can be entered. Automatic Set Size calculation is attempted when any of the following parameters are changed: Device width, Device Block Size or User Data Size. The following equation defines Total Set Size:

$$\text{TOTAL SET SIZE} = \frac{\text{USER DATA SIZE}}{\text{DEVICE BLOCK SIZE}\left(\dfrac{\text{DATA WORD WIDTH}}{\text{DEVICE WIDTH}}\right)}$$

0987-1

- **User Data Size**—Specifies the hexadecimal size, in bytes, of the data block used to verify the device with the source. This value is normally equal to the device size or to a multiple of the device size for verifying a set. If 0 is entered, it is reset to the device size. User Data Size works with Total Set Size to determine the total amount of bytes to verify with a set of devices.
- **Next Operation Begins At**—Specifies the address in user memory where the next data byte will be verified. This value is calculated from the Data Word Width, Device Block Size, Memory Begin Address, device width, and next set member parameters.

Optional Parameter Screens

There are two types of parameter-entry screens: simple and complex. UniSite-*xpi* defaults to displaying the simple, Non-default Parameters screen. If you want to view the complex, All Parameters screen, press **F4** from the Non-default Parameters screen. If you have changed any of the default parameters, those changed parameters will also show up on the Non-default Parameters screen.

The following parameters are found on the complex, All Parameters Verify Memory Device screen.

- **Memory Begin Address**—Specifies the first address, in hex, of the first byte of data to be verified. If the Source is RAM, it is a beginning RAM address. If the Source is Disk, it is a beginning disk file address. The Memory Begin Address must be an even address if you have selected a 16-bit device. The default address is 0.

Note: (Legacy) For a SetSite operation, Memory Begin Address specifies the first address of the device in the first socket.

- **Device Begin Address**—Specifies the first hex device address that will be verified.
- **Device Block Size**—Specifies the size, in hex, of device data used in device operations. When you select a device, Device Block Size is automatically set to the device size and normally does not need to be changed. It is also automatically set to a smaller value if the Device Begin Address is nonzero. This parameter can be changed if desired. If a zero is entered, the Device Block Size is set automatically to equal the device size.
- **Set Auto-increment**—When enabled, directs UniSite-*xpi* (when in a set verify mode) to the starting memory address of the next data block that is to be verified. For example, if you have four 1K x 8 devices to verify against a 4K x 8 block of data, using the auto-increment option directs UniSite-*xpi* to point to the first address of the next 1K block after each device has been verified. For single device operations, this feature should be disabled and the Next Device parameter should be set to 1.

*Note: Items with an * (asterisk) are visible on the screen only if the selected device supports the feature.*

* **Compare Electronic ID**—When enabled, compares the electronic signature of the device against the electronic signature of the selected algorithm.

* **Odd/Even Byte Swap** (Y,N)—When enabled, allows the Most Significant Bytes (MSB) and the Least Significant Bytes (LSB) of 16-bit words to be swapped when data is verified between a 16-bit device and memory. When disabled, data is verified with the MSB at an odd address. When enabled, the MSB is at an even address.

- **Verify Passes** (0,1,2)—Selects the number of times to test the device. **0** directs UniSite-*xpi* not to test the device. **1** directs UniSite-*xpi* to test the device once at the device manufacturer’s nominal Vcc. **2** directs UniSite-*xpi* to verify the device at the device manufacturer’s recommended high and low Vcc levels. This parameter defaults to **2**.

More Commands

In general, the commands found under the More Commands menu do things other than loading, programming, and verifying devices.

The More Commands menu is a multi-level menu with some commands nested three levels deep. The items on the top-level of the More Commands menu are described below:

- **Configure System**—Contains commands to perform the update operation, and to edit, save, and restore UniSite-*xpi*'s communications, interface, serial I/O, and programming parameters. (These are the items shown on the default parameters list at the beginning of this section.) From this menu, you can also select a new terminal type and access Keep Current Configuration functions. You could use these commands to set up unique parameter files for each device type you want to program and then save those values from the More Commands/Configure System/Save screen. These parameter settings can then be loaded at a later time using the More Commands/Configure System/Restore screen.
- **Device Checks**—Performs device tests on socketed devices.
- **Edit Data**—Allows you to edit RAM or disk data. Separate editing features exist for logic and memory devices.
- **File Operations**—Performs various operations on UniSite-*xpi*'s disk files, such as loading, saving, deleting, or renaming a file.
- **Job File**—Allows you to play back a series of keystrokes. This is useful if you are consistently programming the same devices. Up to ten job files may be stored on any Algorithm or System disk.
- **Remote Control**—Switches UniSite-*xpi* into remote mode, where it will accept commands sent from a remote computer. Chapter 5, "Computer Remote Control," lists the commands recognized by UniSite-*xpi* in remote mode.
- **Self-test**—Performs diagnostic checks on UniSite-*xpi*'s circuitry.
- **Transfer Data**—Allows you to upload or download data to or from UniSite-*xpi*. Also allows you to select or change the data translation format.
- **Yield Tally**—Allows you to view or clear programming statistics.

Configure System

The commands on the Configure System menu allow you to accomplish these basic tasks:

- Change communications protocols between UniSite-*xpi* and the other equipment connected to UniSite-*xpi*, such as a terminal or a host computer.
- Configure the Remote and Terminal ports so they will be compatible with your terminal or host computer.
- Edit, save, or restore a set of programming features unique to the device type you want to program.
- Access Keep Current algorithm files.
- Access Custom Menu algorithm files.
- Update and change settings for a Mass Storage Module (MSM).

The commands available on the Configure Systems menu are described on the following pages.

Edit

Use the commands on the Edit menu to change system parameters. These parameters include the settings of various options and features for the Programming, Serial I/O, Communication, and Interface screens.

Selecting Edit from the Configure System menu will display the Edit Parameter menu. From this menu you can choose Programming, Serial I/O, Communication, or Interface parameter screens which are described in the following subsections. Default parameter settings are shown in the table at the beginning of this chapter.

The system parameters that can be saved and restored with the More Commands/Configure System/Save and More Commands/Configure System/Restore commands are listed below.

Programming Parameters

Source / Destination
Reject Option
Logic Verification
Verify Passes
Verify Data Format
Data Word Width
Algorithm Source
User Data Size
Memory Begin Address
Device Begin Address
Device Block Size
High Speed Logic Drivers
Illegal Bit Check
Blank Check
Compare Elec ID
Enable Yield Tally
Erase EE Device
Odd / Even Byte Swap
Continuity Check
Compensated Vector Test
RAM Device Selection
Checksum Calculation

Communication Parameters

Source / destination
I / O Translation Format
I / O Address Offset
I / O Timeout
Upload Wait
Transmit Pacing
Download Echoing
Output Record Size
Number of Nulls
Instrument Control Code
Fill Memory Option
Fill Data
High Speed Download
User Menu Port*
JEDEC I / O Translate DIP / LCC Vectors
Upload: Use End-of-file Delimiter
Upload End-of-file Delimiter
Download: Use End-of-file Delimiter
Download End-of-file Delimiter
Upload Host Command
Download Host Command

Serial Port Parameters

Baud Rate
Parity
Data Bits
Stop Bits
Enable CTS / DTR

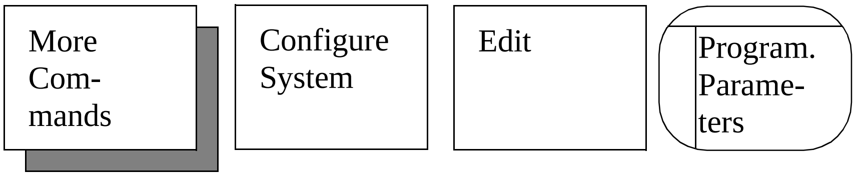
Interface Parameters

Power On CRC Mode
Enable Terminal Beeps
Remote On Code
Remote Off Code
Main Menu Job Files

* This parameter can be saved with the Save System Configuration command and is read on powerup, but this parameter is not restored when a Restore System Configuration command is performed.

Programming Parameters

Use the Edit Programming Parameters screen to specify programming options, to enter memory block parameters, and to enable/disable different tests.



If you want to save the settings for use in a future programming session, then you should save them in a configuration file. Use the Save command (part of the Configure System menu) to save the settings for later use.

The different programming parameters and settings are explained below.

- **Reject Option** (C,S)—Selects the number of times the device is pulsed with programming voltage before it is rejected as unprogrammable. **C** selects the number of pulses specified by the manufacturer. **S** selects either a one-pulse or the military-specification number of programming pulses. Unless you are programming devices to a strict military specification, you should leave this option set at **C**. This option defaults to **C**.
- **Logic Verification** (F,V,A)—Specifies the type of logic verification to perform. Select from Fuse Verification **F**, Vector Verification **V**, or All **A**. Fuse Verification checks the fuse pattern programmed into the device with the pattern in UniSite-*xpi*'s memory. Vector Verification functionally tests the device, using structured test vectors stored in memory. **A** directs UniSite-*xpi* to perform both fuse verification and vector verification. Press **SPACE** to step through the three choices. This parameter defaults to **A**.

  UniSite-*xpi* does not support vector testing for logic devices with more than 84 pins.
- **Verify Passes** (0,1,2)—Selects the number of times to test the device. **0** directs UniSite-*xpi* not to test the device. **1** directs UniSite-*xpi* to test the device once at the device manufacturer's nominal Vcc. **2** directs UniSite-*xpi* to verify the device at the device manufacturer's recommended high and low Vcc levels. This parameter defaults to **2**.
- **Verify Data Format** (B,H)—Specifies either **B** (binary) or **H** (hex) for the mis-verify data display format on the Verify Memory Device screen.
- **Algorithm Type** (D,E,K,C)—Selects the type of device algorithms. **D** is the default and directs UniSite-*xpi* to select algorithms from the algorithms included on the Algorithm disk. If you set Algorithm Source to **D** and then choose the Select Device command, UniSite-*xpi* displays the standard Manufacturer List screen, as shown in Figure 3-42.

  E directs UniSite-*xpi* to select algorithms from the alg.ext file. Extended algorithms (algorithms contained in the alg.ext file) are used by Data I/O to handle device approvals and special device algorithm updates. If you set Algorithm Source to **E** and then choose the Select Device command, UniSite-*xpi* displays a Manufacturer List screen containing only the manufacturers found in the alg.ext file. If this parameter is set to **E** and UniSite-*xpi* cannot find the alg.ext file, UniSite-*xpi* displays the following message in the message bar:

```
Cannot access system file. Insert System disk.
```

K directs UniSite-*xpi* to select algorithms from the Keep Current algorithms it finds on the disks in the disk drives. If you set Algorithm Source to **K** and then choose the Select Device command, UniSite-*xpi* displays the Keep Current Part List screen, as shown in Figure 3-45. Keep Current algorithms are downloaded from Data I/O's Web site and provide immediate support for new device algorithms and updated device algorithms. For more information, see the Keep Current documentation behind the Keep Current tab at the back of this binder.

C directs the programmer to select algorithms from the Custom Menu algorithms it finds on the disks in the disk drives. If you set Algorithm Source to **C** and then choose the Select Device command, the programmer displays the Custom Menu List screen.

Note: See "Save System Parameters" on page 4-38 for instructions on how to save your Algorithm Type and other parameter changes as power-up defaults.

- **Algorithm Media (File)** (F,M)—There are 2 algorithm media options: Floppy Disk (F) and Mass Storage Module (M). The device selection operation will select the device algorithm from the designated algorithm media.

Note: (Legacy) This parameter appears only if a Mass Storage Module (MSM) is installed in UniSite.

- **Checksum Calculation** (4,8,D,B)—Determines the word size and display of the checksum for all device operations. The Odd/Even Byte Swap option affects this calculation. **4** directs UniSite-*xpi* to calculate and display a 4-digit, 4-bit checksum. This is only for use on 4-bit devices. **8** directs UniSite-*xpi* to calculate and display an 8-digit, 8-bit checksum. This is the default and is the only option for 8-bit devices. **D** directs UniSite-*xpi* to calculate and display the checksum based on the number of bits per word in the device width. This option is for use with 16- and 32-bit devices. **B** directs UniSite-*xpi* to calculate and display the checksum based on the number of bits per word in the device width, and also to display the 8-bit checksum.
- **Data Word Width**—Should match the data bus word width in the target microprocessor system for the device being programmed. For 8-bit (or larger) devices, any word width between 4 and 64 may be typed in. For 4-bit devices, your word width choices are 4, 8, 16, and 32. When performing a Quick Copy, the Data Word Width is set to the device word width and restored to the original value after the Quick Copy is complete.

  UniSite-*xpi* always changes this parameter to match the selected device's width, with one exception: if the current Data Word Width is 16, your selected device's word width is 8, AND the previously-selected device's word width is also 8, this parameter does not change.

- **User Data Size**—Defines the hexadecimal size, in bytes, of the data block used in device operations. This value is normally equal to the device size, or to a multiple of the device size for set programming. User Data Size works with Total Set Size to determine the total amount of bytes for a set operation. This parameter can also indicate the number of bytes in a data transfer operation.
- **Memory Begin Address**—Specifies the first hex address in RAM to load data from a device. Also specifies the first address where a device is programmed / verified. If the user memory is RAM, it is a beginning RAM address. If the user memory is disk, it is a beginning disk file address. The Memory Begin Address must be an even address if you have selected a 16-bit device. The default address is 0.
- **Device Begin Address**—Specifies the first address used in device operations. This option is used for memory devices only.
- **Device Block Size**—Defines the size, in hex, of device data used in device operations. When you select a device, Device Block Size is automatically set to the device size and normally does not need to be changed. It is also automatically set to a smaller value if the Device Begin Address is nonzero. This parameter can be changed if desired. If a zero is entered, the Device Block Size is set to the device size.

- **Illegal Bit Check** (Y,N)—Enables or disables the illegal-bit test. This test compares data in a device against data in UniSite-*xpi*'s RAM to determine if the device has already-programmed locations of incorrect polarity. For example, UniSite-*xpi* returns an illegal-bit error in the following situation: data in RAM indicates a specific bit should be in an unprogrammed state while the corresponding bit in the device is in a programmed state. This parameter is enabled by default.
- **Blank Check** (Y,N)—Enables or disables the blank check test. A Blank Check test checks a device for programmed bits. This parameter defaults to **Y**, which enables the test.
- **Compare Elec ID** (Y,N)—Compares the electronic ID of the device against the electronic ID of the selected algorithm.
- **Enable Yield Tally** (Y,N)—When enabled, directs UniSite-*xpi* to keep a running tally of the programming yields for the last sixteen types of devices programmed. These totals show how many devices passed and failed, and what specific errors, if any, have occurred. This parameter defaults to **N**, which disables the test. A more complete description of this feature can be found in the "Yield Tally" section of this chapter.
- **Erase EE Device** (Y,N)—Bulk erases the electronically erasable devices before UniSite-*xpi* attempts to program them.
- **Odd/Even Byte Swap** (Y,N)—When enabled, swaps data at memory address locations during a load, program, or verify operation. The contents of user RAM are not altered.

  Swapping bytes is useful when manipulating 16- and 32-bit data for a target system that has a different architecture than the original file convention. For example, Motorola 16-bit data files store the Most Significant Bytes (MSB) at even-byte locations, and Motorola 32-bit data files store the significant bytes in descending order with the MSB in every first byte (byte 0) and the least significant in every fourth byte (byte 3). Intel 16-bit data files store the MSB at odd-byte locations and Intel 32-bit data files store the MSB in ascending order with the most significant in the fourth byte.

  The default for this parameter is **N**, which means the programmer maintains its RAM data and file data with the convention that the MSB of a 16-bit device resides in the odd byte of memory, and the MSB of a 32-bit word resides in the first of each four bytes of memory.

  When set to **Y**, for a 16-bit device, data is loaded / programmed / verified into user memory with the MSB at even addresses. For 32-bit devices, the significant bytes are placed in ascending order with the MSB placed in user memory in the fourth of each four bytes (the fourth byte is swapped with the first byte and the third byte is swapped with the second byte).
- **Continuity Check** (Y,N)—Checks for open device pins before programming a device. This parameter is enabled by default and when a new device type is selected.

- **Serial Vector Test** (Y,N)—When enabled, this test applies each vector’s input states serially, starting with pin one and stepping through the remaining pins. This test is a diagnostic tool designed to help debug and classify test vector failures. Specifically, this test is designed to isolate test vectors that are sequence dependent. If a sequence-dependent vector is found, it should be broken into two or more vectors to make them sequence independent.

  The JEDEC specification for test vectors requires that test vectors be sequence independent. If sequencing between pins is important, then the test vector should be separated into two or more vectors to make them sequence independent. This test helps isolate vectors that are sequence-dependent and that should be expanded.

  This switch is available only for logic devices and is disabled on power up. The switch is also returned to **N** when parameters are restored or when another device is selected.

- **High Speed Logic Drivers** (Y,N)—When enabled, this feature increases the speed of the logic transitions between **0** to **1** and **1** to **0** of the test vector input states.

  The speed of the logic transitions is increased by driving the **0** and **1** levels using the high speed logic drivers instead of a current limited driver.

  The JEDEC specification for test vector 0 and 1 input states defines that these inputs be current limited, so that the outputs of the device under test can overdrive the 0 or 1 level. However, current-limited drivers have inherently slow transition speeds. Enabling this feature reduces the possibility of doubling clocking due to slow transition times.

  This switch is available only for logic devices. It defaults to **Y** on power up and stays on until turned off.

CAUTION: If used with invalid test vectors that drive outputs, the High Speed Logic Drivers test may cause over current errors.

- **Compensated Vector Test** (Y,N)—**Y** enables load compensation on PLD output pins under test during vector testing. This may eliminate structured test errors when testing PLDs sensitive to output loading, where many of the device’s registers transition simultaneously. This test is available only for logic devices and defaults to **N** at power up. This parameter defaults to **Y** if you select a non Open Collector device, and defaults to **N** if you select an Open Collector device. This parameter can be saved / restored with the Save / Restore Configuration command.

- **Display Device Footnote** (Y,N)—When enabled, this feature enables the automatic display of device specific information (if any is available for the selected device).

- **Abort on Empty Socket** (Y,N)—When this option is enabled, all device operations will automatically be aborted on empty sockets.

- **RAM Device Selection**—When selected, allows the device programming algorithms to reside in user RAM. This feature requires UniSite-*xpi* to contain the 4 MB or 8 MB option and allows much faster device selection than the normal method of loading the algorithm from the disk each time a device is selected.

  After powerup, when you enable RAM Device Selection and select a device, UniSite-*xpi* loads the algorithms from the algorithm disk(s) you have inserted. The following message appears:

  ```
  Loading device algorithm file into user RAM
  ```

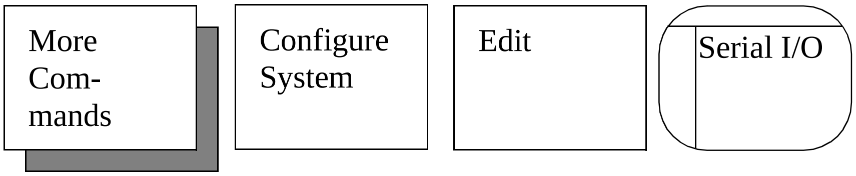
  If only one algorithm disk was present when you enabled RAM Device Selection, the first time you select a device whose algorithm is not already loaded into RAM, you will be prompted to insert the other algorithm disk, as follows:

  ```
  Cannot access system file. Insert Algorithm Disk X
  ```

  Similarly, if RAM selection is enabled when the programmer is powered up, the algorithms from the disk in drive B are loaded into RAM. The first time you select a device that requires algorithms from the other algorithm disk, UniSite-*xpi* prompts you to insert the other disk so that its data can be loaded into RAM.

Serial I/O Port Configuration

Use the Edit Serial I/O Port Configuration command to specify the communications parameters for UniSite-*xpi*'s two serial ports.



Use this command when connecting equipment to UniSite-*xpi*'s Terminal and Remote ports so they can be compatible with your terminal or host computer. If you want to save the settings of the two ports for use in a future session, you should save them in a configuration file.

The different parameters and their settings are explained below.

A change in the serial port parameters does not become effective until you press ENTER. If terminal settings are changed, a message appears, prompting you for another ENTER after you have altered your terminal to match the new settings. Output to the terminal is suspended until you press **ENTER a second time**.

- **Baud Rate**—Specifies the baud rate for both the Terminal and Remote ports. Press SPACE to cycle through the baud rates supported by UniSite-*xpi*.

  UniSite-*xpi* supports the following baud rates: 50, 75, 110, 134.5, 150, 200, 300, 600, 1050, 1200, 1800, 2000, 2400, 4800, 7200, 9600, and 19.2 kbaud (and 115.2 kbaud with HiTerm). However, not every baud rate works on one port when certain baud rates are selected for the other port. If you select incompatible baud rates, UniSite-*xpi* beeps and displays the following error message:

```
WARNING: Selection not compatible with other channel!
```

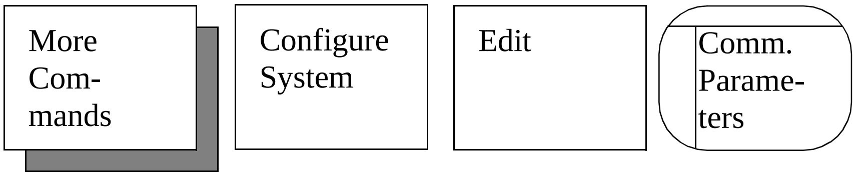
If one of the rates on the left is used for either port, then the corresponding rates on the right do not work on the other port.

50	75, 150
150	200, 1050
7200	75, 150, 1800, 2000, 19.2K
19.2K	50, 200, 1050, 7200

- **Parity** (N,O,E)—Three options are available for the parity setting: **N** (No parity), **O** (Odd parity), and **E** (Even parity). Press **SPACE** to cycle through the three values.
- **Data Bits** (7,8)—Specifies the number of data bits UniSite-*xpi* recognizes during serial communication. Press **SPACE** to toggle between the two values.
- **Stop Bits** (1,2)—Specifies the number of stop bits between data bytes. Two stop bits are generally used for baud rates of 110 or lower. Press **SPACE** to toggle between the two values.
- **Enable CTS/DTR** (Y,N)—Enables or disables CTS/DTR hardware handshaking. Press **SPACE** to toggle between the two values.

Communication Parameters

Use the Edit Communication Parameters screen to change UniSite-*xpi*'s I/O characteristics when downloading or uploading data.



If you want to save the settings of the communication parameters for use in a future session, you should save them in a configuration file. Use the Save command (part of the Configure System menu) to save the settings for later use.

- **Source/destination** (R,T)—Specifies the source/destination port of the data to be transferred. Press **SPACE** to toggle between **R** (Remote port) and **T** (Terminal port).
- **I/O Translation Format**—Specifies the two-digit decimal code that corresponds to the translation format in which the data will be transferred. For a detailed list and sample of each format, see the Translation Formats chapter of this manual.
- **I/O Addr Offset**—Specifies address offset to use during a data transfer. The I/O Address Offset is subtracted from the I/O Addresses during Input from Disk and Download operations so data is properly located in User RAM or on disk. During Output to Disk and Upload operations, the I/O Address Offset is added to the User Memory Address.

  Specifying FFFFFFFF instructs UniSite-*xpi* to set the I/O Address Offset to the first incoming address of data received from Input from

Disk and Download operations. During Output to Disk and Upload operations, setting the I/O Address Offset to FFFFFFFF is the same as setting the I/O Address Offset to 0, since no offset is added to the I/O addresses sent out.

To load your file absolutely, set the I/O Address Offset to 0. The data is stored at the addresses specified in the data file.

This parameter defaults to FFFFFFFF.

- **I/O Timeout**—Limits the time, from 0 to 99 seconds, UniSite-*xpi* waits for a data transfer to begin. A value of 0 disables the timeout. A value of 0 should not be used for data formats which don't use a start code and an end code because these formats need the timeout enabled in order to terminate a normal data transfer. Formats which fall into this category include codes 5, 6, 7, 9, 11, 13, 16, and 96.
- **Upload Wait**—Sets the period that UniSite-*xpi* waits before it begins sending data to the host computer after the host upload command is sent. The range of this parameter is 0 to 99 seconds.
- **Transmit Pacing**—Specifies the time-delay to insert between characters transmitted to a host during an upload. See the note about transmit pacing at the end of this section for more information.
- **Download Echoing** (Y,N)—Displays the data being downloaded. This may slow down UniSite-*xpi* in receiving data and is not recommended for high baud rates, such as 9600 and above. Download echoing may not be used with the binary formats.
- **Output Record Size**—Specifies the number of data bytes contained in each data record during upload. The range of this parameter is 0 to 256 bytes. Some formats have fixed record lengths for which this parameter does not apply.
- **Number Of Nulls**—Sets the number of null characters sent between each record in a data file after a carriage return and line feed. The range of this parameter is 0 to 254 nulls. Entering 255 specifies no nulls and suppresses the line feed.
- **Instrument Control Code**—Specifies how the data transfer will be controlled. Selecting 0 specifies regular XON/XOFF handshaking, selecting 1 or 2 specifies a special handshaking sequence (see the Translation Formats chapter for more information).
- **Fill Memory Option** (N,D,U)—Specifies what data user memory will be filled with before downloading begins. This option is used for Download Data and Input from Disk operations. If you select **N** (none), user memory will not be changed and whatever is in user memory will be overwritten by the downloaded data. If you select **D** (default), user memory will be filled with the data appropriate to initialize unused locations to the unprogrammed state for the device type selected. If you select **U** (user defined), then user memory will be filled with whatever you specify in the Fill Data option.

- **Fill Data**—Allows you to type in the hexadecimal data to be placed at unused locations of user memory during download. To use this option, you must also specify **U** for the Fill Memory option. User memory can also be filled with the specified data during the Edit Data operation.
- **High Speed Download** (Y,N)—When enabled, this parameter allows UniSite-*xpi* to download data from a PC at 115.2 kbaud. For high speed download to work, this parameter must be set to **Y** and the following conditions must be met:
  - You must have an IBM-compatible PC connected to UniSite-*xpi*'s Remote port.
  - You must have Data I/O's HiTerm software installed and running on the PC.
  - The data you are downloading must be stored in a data format supported by HiTerm for high speed downloads. The HiTerm manual contains a list of supported formats.
  - You must use HiTerm's **TR** command when you download data to UniSite-*xpi*. See the HiTerm manual for more information.

  Setting this parameter to **Y** causes the User Menu Port parameter to be set to **R** (Remote port). See the "Set up High Speed Download" section of Chapter 2 for more information.
- **User Menu Port** (R,T)—Specifies which of UniSite-*xpi*'s two ports should be used to send user menu information and to receive commands. (User menu information and commands are what you see when you operate UniSite-*xpi* from a workstation or a terminal.) Normally, this parameter is set to **T** (Terminal port).

  If you want to use the 115.2 kbaud high-speed download option, use this parameter to redirect the user menu information from the Terminal port to the Remote port. To redirect the user menu information, set this parameter to the port you want connected to the controlling PC and press **ENTER**. Then move the cable to the port specified on UniSite-*xpi*.

Note: When switching ports, make sure the communication settings of the terminal/PC and the port you switch to are the same. Also, make sure you switch between compatible terminals; for example, you cannot switch from an ANSI 3.64 compatible terminal to a VT-100 compatible terminal.

Setting the High Speed Download parameter to **Y** causes this parameter to be set to **R** (Remote port).

- **JEDEC I/O Translate DIP/LCC Vectors** (Y,N)—When enabled, translates test vectors for a device from its DIP package to its PLCC/LCC package. If this feature is selected, UniSite-*xpi* alters the test vectors during I/O translation, allowing for the different pinouts of the two package types. During downloading, vectors are converted from DIP to PLCC/LCC; during uploading, vectors are converted from PLCC/LCC to DIP. Use this feature if you have created test vectors for a DIP device but actually want to program the PLCC/LCC version of the same device.

- **Upload: Use End-of-file Delimiter** (Y,N)—When enabled, inserts an end-of-file character following an uploaded file. The delimiter character signals the host system that the upload is complete. During an upload, an end-of-file character is transmitted to the host. To invoke this feature, you must select this option and provide the two-digit hexadecimal number of the ASCII character you want to use as the end-of-file character. Select any value between 01 and 1F.

Note: Use this feature only if you are using a format with an end-of-text character. It cannot, for example, be used for files stored in a binary data translation format.

- **Upload End-of-file Delimiter** (1-1F)—Selects the two-digit hexadecimal number of the ASCII character you want to use as the end-of-file character in uploading data to a host computer. Select any value between 01 and 1F.
- **Download: Use End-of-file Delimiter** (Y,N)—When enabled, signals UniSite-*xpi* that the transmission of a particular file is complete. During a download operation, characters after the last record and before the end of the file would be ignored. If you wish to use this feature, you must select this option and you must provide the two-digit hexadecimal number of the ASCII character you want to use as the end-of-file character. Any value between 01 and 1F may be selected.
- **Download End-of-file Delimiter** (1-1F)—Selects the two-digit hexadecimal number of the ASCII character you want to use as the end-of-file character in downloading from a host computer. Any value between 01 and 1F may be selected.
- **Upload Host Command**—Type the command you want to use to tell the host system what to do with the data to be uploaded. The command may be up to 58 characters long. You can use host operating system commands on this line. For example, with UNIX, you would enter an upload command such as **cat > 27128.hex**. UniSite-*xpi* appends a return to the command.
- **Download Host Command**—Type the command you want to send to the host system to initiate a file transfer download to UniSite-*xpi*. The command may be up to 58 characters long. You can use host operating system commands on this line. For example, with UNIX, you would have a download command such as **cat 27128.hex**.

Transmit Pacing

Transmit pacing provides a time delay between characters sent to the host by UniSite-*xpi*, affecting data uploaded from UniSite-*xpi* as well as host commands sent by UniSite-*xpi*. If you encounter errors during upload operations to a host, you must use the transmit pacing delay feature to eliminate the errors. This will provide a character-by-character delay to prevent data overrun on the host. This type of condition may exist in spite of the use of hardware and / or software handshaking on the host and UniSite-*xpi*. This condition is most likely to occur on hosts that cannot accept incoming data fast enough at high baud rates.

Since the host may not explicitly report any error, to determine if errors are occurring during an upload process, transfer data to the host by using the upload function with an I/O format that utilizes checksums (such as Format 87) selected. Note the checksum reported by UniSite-*xpi* when the transfer is complete. Then transfer the same data back to UniSite-*xpi* while UniSite-*xpi* is performing the compare data function, again noting the checksums. If the checksums don't match or if an error is encountered, a transmit pacing delay should be used.

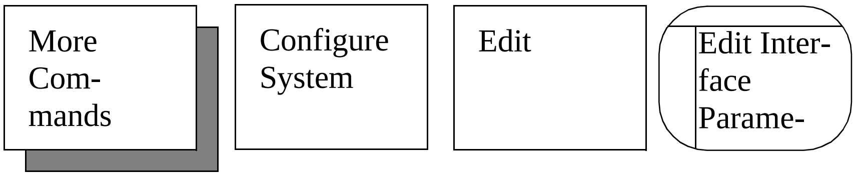
Typical delay values are presented in the following table. The transmit pacing value is specified in tenths of milliseconds delay. For example, a value of 12 represents 1.2 milliseconds delay. The minimum delay possible (other than zero) is 4 (0.4 milliseconds) and the maximum is 99 (9.9 milliseconds). The factory default is 0. The transmit pacing value required for reliable data transfers may vary somewhat from those presented in the table due to the particular characteristics of the host involved. This is primarily determined by the processor speed of the host and whether or not software other than the communication software is running at the same time on the host (in the latter case, larger delay values may be required).

Transmit Pacing Delay Values

	4800 and less	**9600**	**19.2K**
HiTerm			
PC	0	0	6
AT	0	0	6
VTERM			
PC	0	4	15
AT	0	0	8
PROCOMM			
PC	0	4	9
AT	0	0	6

Interface Parameters

Interface Parameters are parameters that are not related directly to uploading/downloading of files.

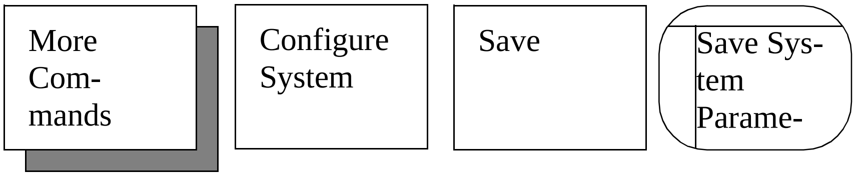


The parameters on the interface parameters screen are described below.

- **Power On CRC Mode** (Y,N)—When enabled, this feature allows you to power up UniSite-*xpi* and have it automatically enter computer remote control mode. You can exit CRC by pressing **SHIFT** + **Z** + **ENTER**.
- **Enable Terminal Beeps** (Y,N)—When this feature is enabled, UniSite-*xpi* beeps each time an error message is generated.
- **Remote On Code**—When this parameter is enabled, you can use an ASCII character to enable the Remote port. Type the two-digit hexadecimal number that represents the ASCII character you want to use to enable remote control.
- **Remote Off Code**—When this parameter is enabled, you can use an ASCII character to disable the Remote port. Type the two-digit hexadecimal number that represents the ASCII character you want to use to disable remote control.
- **Main Menu Job Files** (Y,N)—When this feature is enabled, you can start a Job File from the Main Menu rather than having to go to the Job Files menu.
- **Power Up User RAM Test**—This option is only used during powerup. The setting of this option is saved with the system parameters. When Power Up User RAM Test is enabled, a User RAM test is performed as part of the boot process. When this option is disabled, the User RAM test is skipped during the boot process.

Save System Parameters

System Parameters are all the parameters on the Programming, Serial I/O, Communication, and Interface screens. With the Save command, you can save a set of system parameters for future use.



This feature is useful if you want UniSite-*xpi* to power up with preset parameters. This feature is also useful if multiple users prefer to have their own set of parameters easily available.

To save a system configuration, follow these steps:

1. Before using the Save command, go through the above-mentioned Configure/Edit menus, setting the system parameters to fit your particular application.

2. When you have UniSite-*xpi* configured the way you want, go to the Save screen and select a file number in which to store the configuration file. The file number must be between two and nine. File numbers zero through two are reserved for factory defaults, power-up defaults, and CRC defaults.

3. Next, enter a description of the configuration file you are saving, for example, **Config file for Intel 27C256**. The name can be up to 30 characters long. After you enter the file description, press **ENTER**. UniSite-*xpi* displays the following message in the message bar:

   ```
   Parameter Entered
   ```

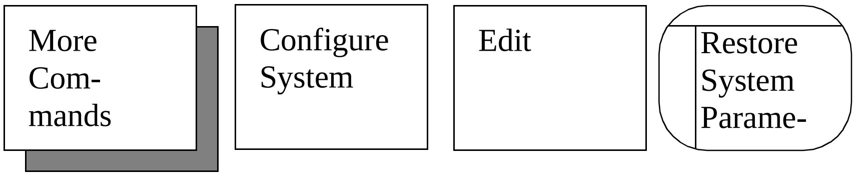
4. Finally, to save the current system configuration to a configuration file, press **ENTER**. While UniSite-*xpi* is saving the configuration, the action symbol rotates. When done, UniSite-*xpi* displays the following message:

   ```
   System parameters saved
   ```

Note: If a device has been selected, it is saved as part of the configuration file.

Restore System Parameters

System Parameters are all the parameters on the Programming, Serial I/O, Communication, and Interface screens.



With the Restore command, you can restore a set of previously saved system parameters.

To restore a system configuration, follow these steps:

1. Select Restore from the Configure System Parameters menu.

2. UniSite-*xpi* displays a list of all the configuration files that have been saved on the System disk. Look at the list of configuration files and find the file number of the file you want to restore.

3. Enter the file number of the configuration file you want to restore and press **ENTER**. UniSite-*xpi* loads the system parameters and displays the following message in the message bar when done:

   ```
   System parameters restored
   ```

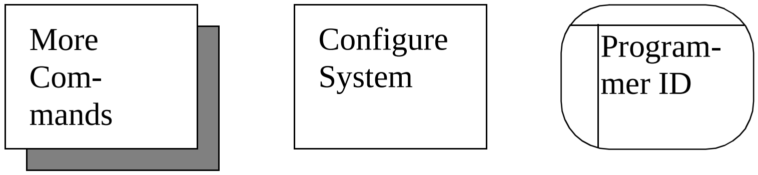
Note: When you restore a configuration, the device algorithm selected when the configuration file was saved (if any) will also be restored.

If you are restoring a configuration that had a device selected as part of the configuration and UniSite-xpi cannot find the algorithm for the selected device, UniSite-xpi will prompt you to insert another Algorithm disk.

When you return to a screen, the cursor will be where you left it. Restoring parameters to factory defaults or power-up defaults returns the cursor to its original position. Restoring parameters to user-defined defaults has no effect on cursor positions.

Programmer ID

Use this command to view your programmer ID.



Carrying a Configuration File Forward

Beginning with version 4.10 you can carry system configurations forward after you update to a new version of system software. Previously, you had to recreate system configurations when you updated to a new version of system software.

Note: If you carry the system configurations forward, the power up defaults will be carried forward in addition to any other user-defined system configurations. This is because the power-up defaults, in addition to the factory defaults and any user-defined configurations, are stored in a file named sysparm.sys on the UniSite-xpi System disk.

Now, the following scenario is possible:

- Under version X.3, create and save two different system configurations, such as one for memory devices and one for logic devices.
- Update to version X.4 system software.
- Carry the system configurations created under version X.3 forward to X.4.
- Use the two system configurations with version X.4.

Note: This procedure works only with system software version 4.10 and later.

For more information on saving and restoring system configurations, and on system configurations in general, see the "Restore System Parameters" and the "Save System Parameters" sections in this chapter.

Floppy or MSM

The procedure described in this section can be used to carry configuration files forward from an old System disk to the new System disk. If you wish to carry configuration files that are located on a Mass Storage Module (MSM) forward, refer to the update instructions included in the Update Kit.

What You Need

To carry system parameter settings forward from the old System disk to the new System disk, you will need a DOS-based IBM-compatible PC with a 3.5-inch floppy drive capable of reading and writing 720KB floppy disks.

Carrying the Configuration File Forward

Follow the steps below to carry your system configurations forward to a new version of system software.

Note: In the following steps "old" refers to the System disk containing the system configurations you want to carry forward to the "new" System disk.

1. Power up your programmer using the new System disk.
2. Make a backup copy of the new System and Algorithm disks. See the description of the Duplicate Disk command for more information on making a backup copy of your System and Algorithm disks.

Note: Store the master copies in a safe place. Use your backup copies for day-to-day operation.

3. Power up the PC. When the PC has finished booting, insert your working copy of the old System disk in the 3.5-inch drive in the PC.

Note: Make sure you insert the System disk with the configurations you want to carry forward.

4. Copy the sysparm.sys file to your hard drive. Enter the command

   copy *drive1*:\sysparm.sys *drive2*:

   where *drive1*: is the drive designator for the 3.5-inch disk drive and *drive2*: is the drive on which you will be temporarily storing the copy of the sysparm.sys file.

Note: We suggest you work with the backup copies of your System and Algorithm disks and not with the originals.

5. Remove the old System disk and insert the new **System** disk in the 3.5-inch drive. Make sure the new System disk is write enabled.

   Copy the sysparm.sys file to the new System disk. Enter the command

   copy *drive2*:sysparm.sys *drive1*:\sysparm.sys

   where *drive2*: is the drive you copied the sysparm.sys file to in step 4.

6. If you have more UniSite-*xpis* that you want to copy a configuration file to, repeat step 5 for each of the remaining UniSite-*xpi*s. For example, if you have three UniSite-*xpi*s in a lab, and if you used the same configuration files on all three, the following scenario would be possible:

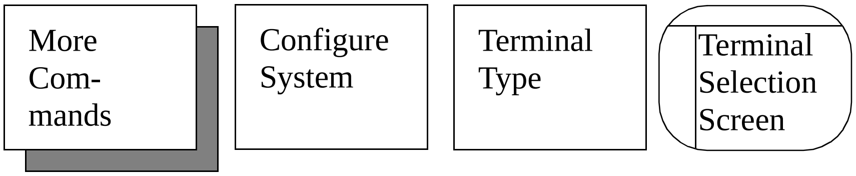
   - Update all three UniSite-*xpi*s to a new version of system software.
   - Make backups of the new System and Algorithm disks.
   - Perform steps 1 to 5 as described above on one UniSite-*xpi*.
   - Repeat step 5 for the remaining two UniSite-*xpi*s.

7. Remove the disk from the 3.5-inch drive and insert a new System disk into UniSite-*xpi*. In the More / Configure system / Restore screen, press **F4** to reconstruct the parameter list.

Note: If you do not reconstruct the parameter list, the next time your programmer is powered up with the new System disk, it prompts `System parameter field is out of date. Do you want to update it (Y/N)`. *Type* **Y** *and press* **ENTER** *to reconstruct the parameter list.*

Terminal Type

This command changes the current and default terminal types.



A list of compatible terminal types follows the instructions for changing the terminal type.

To change the current terminal type or default terminal type, follow these steps:

1. If you have not already done so, configure your terminal to match one of the compatible terminal types.
2. Select the Serial I / O Parameters screen from the More Commands / Configure System / Edit / Serial I / O menu and observe the settings of the port you want to connect the new terminal to. If the terminal's communications protocol does not match the port's, change the settings of the new terminal to match the port's settings.
3. Select the More Commands / Configure System / Terminal Type command. At this point UniSite-*xpi* displays the default and current terminal types, and the available terminal types. Select a terminal type, enter the number corresponding to that terminal type, and press **ENTER**. You have changed the terminal type for this current session.
4. UniSite-*xpi* then prompts you with the following:

   ```
   Save terminal type as power on default? (Y/N) [N]
   ```

   If you do not want to change the default terminal type, then continue with step 4a. If you want to change the default terminal type, then go to step 4b.

   a) If you do not want to change the default terminal, press **N**, then **ENTER**.

   b) To change the default terminal type, press **Y**, then **ENTER**. UniSite-*xpi* saves the new terminal type to disk. The new terminal type is now part of the power-on parameters.

The screen clears and UniSite-*xpi* returns to the Configure System Parameters menu. Resume normal operation.

Approved Terminals

UniSite-*xpi* is compatible with the terminal types listed below. If your terminal is not included in the list, refer to the manual supplied with the terminal to determine if it can emulate one of those listed below.

- ANSI 3.64 compatible terminals
- DEC VT-100 compatible terminals
- Qume QVT-101 compatible terminals
- TELEVIDEO TVI-910 compatible terminals
- Wyse WY-30 compatible terminals

Keep Current

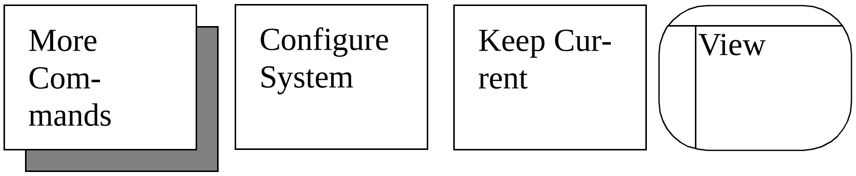
The commands in the Keep Current menu allow you to access Keep Current algorithm files (.KC*x*). Commands available on this menu include the following:

- View
- Replace / Restore
- Delete
- Purge

Note: For information on how to select a device supported by a Keep Current algorithm, refer to "Select Device" on page 4-5.

View

This command allows you to view a list of .KC*x* files.



View displays information on all .KC*x* files found on drives A and B. Compatibility between system software and Keep Current algorithms is not checked.

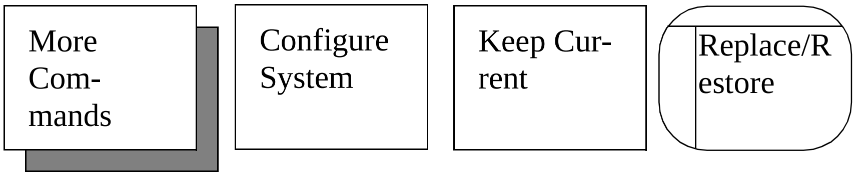
To view Keep Current algorithm files, follow these steps:

1. Insert the disk with the .KC*x* files you want to view into either disk drive.
2. When you select the View command, the dialog window fills with a directory listing. UniSite-*xpi* displays up to 10 files at one time. If there are more than 10 files, press **CTRL** + **N** to display the next page of files. Press **CTRL** + **P** to display the previous page of files.

   If you want to view files on another disk, press **F2**, insert another disk, and return to the beginning of this step.

Replace/Restore

This command displays the Replace / Restore screen. In the Replace / Restore screen, Keep Current algorithms can be toggled between “replaced” and “restored” status.



If a part is marked as “replaced,” the Keep Current algorithm is used during normal device selection instead of its corresponding algorithm in the default algorithm sets. Parts previously marked as “replaced” can be restored. In this case, the default algorithm is used during normal device selection for that part.

To toggle algorithm(s) between “replaced” and “restored” status, follow these steps:

1. Insert both the System disk (this has the **kcmarker.sys** file on it) and the disk with the .KC*x* files you want to replace or restore into the disk drives.

2. When you select the Replace / Restore command, UniSite-*xpi* loads the kcmarker.sys file and checks to see if alg_1.sys, alg_2.sys, alg_3.sys, or alg_4.sys has been loaded into RAM. If the file it needs has not been loaded, UniSite-*xpi* searches for the algorithm disk. If the algorithm disk is not found, the following message appears:

   ```
   Cannot access system file. Insert algorithm disk.
   ```

   If this message is displayed, insert the appropriate algorithm disk and try again.

   If you are performing these operations using an MSM, the Keep Current algorithm and system (.sys) files must be copied to the **I:** drive. Use the More Commands / File Operations / Copy File command.

   If no Keep Current algorithms are found, or the algorithms are not compatible with the current version of your system software, the following message is displayed:

   ```
   Insert Keep Current algorithm disk
   ```

   If this message is displayed, insert a disk with compatible Keep Current algorithms into one of the disk drives and try again.

3. On the Replace / Restore screen, the dialog window fills with a directory listing with parts marked as “replaced” displayed in reverse video. UniSite-*xpi* displays up to 10 files at one time. If there are more than 10 files, press **Ctrl** + **N** to display the next page of files. Press **Ctrl** + **P** to display the previous page of files.

Note that not all .KC*x* files are displayed. The only files that are displayed are those that

- Have a corresponding algorithm on one of the Algorithm disks (the algorithm can already be selected during the normal device selection)
- Are compatible with your version of the system software

If you want to view files on another disk, press **F2**, insert another disk, and return to the beginning of this step.

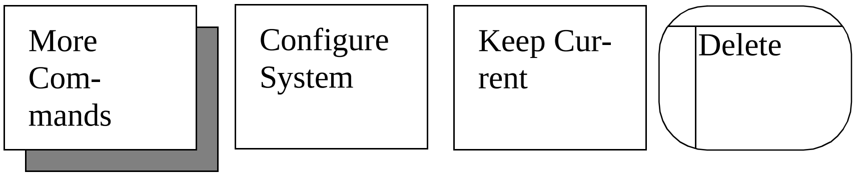
4. Move the cursor to the Replace/Restore field and enter the number corresponding to the file you want to replace or restore.
5. To toggle the file, press **ENTER**. If you do not want to toggle the file, press **F2** to return to the Keep Current Configuration menu.

   If you toggle the part to "replaced" status, it is displayed in reverse video. If the part was already marked as "replaced," it is toggled to "restored" status.

Note: The maximum number of replaced algorithms is 10.

Delete

This command deletes a .KC*x* file from a disk.



To delete a file from a disk, follow these steps:

1. Insert the disk with the .KC*x* file you want to delete into either disk drive.
2. When you select the Delete command, the dialog window fills with a directory listing. UniSite-*xpi* displays up to 10 files at one time. If there are more than 10 files, press **CTRL** + **N** to display the next page of files. Press **CTRL** + **P** to display the first page of files.

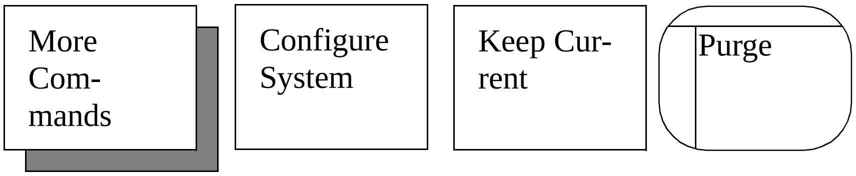
   If you do not see the file you want to delete, press **F2**, insert another disk, and return to the beginning of this step.
3. Move the cursor to the Delete field and enter the number corresponding to the file you want to delete.
4. Move the cursor to the **Are you sure?** field and press **Y**.

***CAUTION: If you do not want to delete the file, do not press* ENTER.**

5. To delete the file, press **ENTER**. If you do not want to delete the file, press **F2** to return to the Keep Current Configuration menu.

Purge

This command deletes all outdated .KC*x* files from a disk, leaving only the most up-to-date algorithms.



To purge files from a disk, follow these steps:

1. Insert the disk with the .KC*x* files you want to purge into either disk drive.
2. When you select the Purge command, the dialog window fills with a directory listing of outdated Keep Current files found on both drives A and B (outdated Keep Current algorithm files have version numbers older than the current system software version number). UniSite-*xpi* displays up to 10 files at one time. If there are more than 10 files, press **CTRL** + **N** to display the next page of files. Press **CTRL** + **P** to display the previous page of files.

   If you do not see the files you want to purge, press **F2**, insert another disk, and return to the beginning of this step.
3. In the **Are you sure?** field, press **Y**.

CAUTION: If you do not want to purge files, do not press* ENTER*.

4. To purge files displayed on the screen, press **ENTER**. If you do not want to purge files, press **F2** to return to the Keep Current Configuration Operations menu.

   If no more .KCx files are left on the disk(s), UniSite-*xpi* will return to the Keep Current Configuration menu.

Custom Menu Algs

The commands in the Custom Menu Algs menu allow you to build Custom Menus.

The most common use of a Custom Menu is to create a list of the devices you use most. Thus, when you wish to select a device, instead of scrolling through screens of devices that you rarely use, you can choose from a smaller, custom list of those devices you most often use.

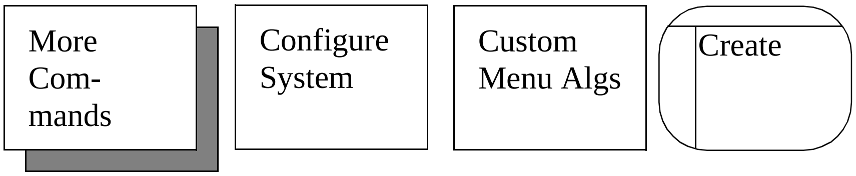
The commands available on this menu include the following:

- Create
- Add
- View
- Delete
- Update

Note: For information on how to select a device from a Custom Menu, refer to "Select Device" on page 4-5.

Create

This command displays the Create screen. In the Create screen, you can create a new Custom Menu by selecting the devices you wish to include in the Custom Menu. The Create command also copies device-related files to the new Custom Menu disk (the disk that will contain the Custom Menu).



To create a new Custom Menu, follow these steps:

1. Select **Create**. The programmer prompts you for

   ```
   Source Disk
   Algorithm Type (D,E,K)
   Algorithm Media (F,M)—only appears if MSM is installed
   Destination Disk
   Custom Menu Algorithm Disk
   ```

   where (D, E, K) is default algorithms, extended algorithms, or Keep Current algorithms, and (F, M) is floppy disk or Mass Storage Module (MSM).

2. Select the **algorithm media** and **algorithm source** that matches the first algorithm you wish to add to the Custom Menu. (You are not limited by the algorithm source you choose. Your Custom Menu can contain any combination of default, extended, and Keep Current algorithms.) In the **Custom Menu algorithm disk** field, select the disk drive where the Custom Menu algorithm files will be created.

3. If prompted, insert the Algorithm disk that contains the algorithms from which you want to choose into the disk drive. (If you do not have the correct disk inserted, the programmer will prompt you for the disk to insert when necessary.)

4. Press **ENTER** to continue. Follow the directions on the screen (inserting disks if prompted).

CAUTION: This operation uses RAM as a temporary storage buffer and alters the contents of RAM.

If you get the message

```
Need to clear user RAM file(s) prior to operation...
```

this means that the programmer does not have enough available RAM to perform the operation. If you press **ENTER**, the RAM file(s) are cleared and the operation continues. If you don’t want RAM files cleared, cancel the operation by pressing **CTRL** + **Z**.

5. When the programmer is ready to write the Custom Menu (CM) algorithm list to a disk, the following message is displayed:

   ```
   Insert Custom Menu algorithm disk. Hit Return to
   continue.
   ```

   Insert the disk that you want the Custom Menu files to be written to and press **ENTER**.

   If you get the message `File ERROR: Cannot allocate file space`, you need to delete some files on your target disk to make more room for the CM files, or insert a different disk that has enough space for the files.

6. After the programmer is done copying files, the programmer displays a list of devices (if algorithm type **K** was selected in step 2) or a list of device manufacturers (if algorithm type **D** or **E** was selected in step 2).

   Examine the list and select a device or device manufacturer by typing the number corresponding to the selection and pressing **ENTER**. If the device or device manufacturer you want is not listed, press **CTRL** + **N** to see the next page of devices or manufacturers.

7. If you selected a manufacturer in step 6, a list of devices produced by the selected manufacturer appears. Type the number corresponding to the device you wish to add to your Custom Menu, and press **ENTER** to add the device. If the device or device manufacturer you want is not listed, **CTRL** + **N** to see the next page of devices.

8. If prompted, insert the Algorithm disk that contains the algorithm you wish to add. (If you do not have the correct disk inserted, the programmer will prompt you for the disk to insert when it is necessary.)

   If you get the message `Cannot access file_name.sys`, the programmer could not find a file on the disk inserted into the programmer's drive. Insert the Algorithm disk that contains the file.

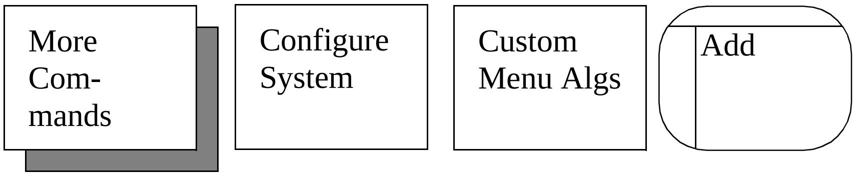
9. Select additional devices from the current screen (or press **F2** to return to the manufacturer list if you were selecting devices from a manufacturers list). Repeat steps 6 through 8 as necessary.

   When you are done adding devices, press **F2** twice until you are prompted to insert the Custom Menu disk.

10. Insert the Custom Menu disk and press **ENTER** to save your changes to disk. To add additional devices to your Custom Menu, use the Add command (described below).

Add

This command displays the Add screen. In the Add screen, you can add new devices to a Custom Menu.



Note: Use the Add command only to add devices to an already created Custom Menu (a Custom Menu created using the Create or Update command).

To add devices to a Custom Menu, follow these steps:

1. Select **Add**. The programmer prompts you for the

   `Source Disk`
   `Algorithm Type (D,E,K)`
   `Algorithm Media (F,M)`—only appears if MSM is installed
   `Destination Disk`
   `Custom Menu Algorithm Disk`

   where (D, E, K) is default algorithms, extended algorithms, or Keep Current algorithms, and (F, M) is floppy disk or Mass Storage Module (MSM).

2. Select the **algorithm media** and **algorithm source** of the device you wish to add to your Custom Menu. In the **Custom Menu algorithm disk** field, select the disk drive where the Custom Menu is located.

3. If prompted, insert the Algorithm disk that contains the algorithms from which you want to choose into the disk drive, and press **ENTER** to continue. Follow the directions on the screen (inserting disks if prompted).

CAUTION: This operation uses RAM as a temporary storage buffer and alters the contents of RAM.

4. The programmer displays a list of devices (if algorithm type **K** was selected in step 2) or a list of device manufacturers (if algorithm type **D** or **E** was selected in step 2).

   Examine the list and select a device or device manufacturer by typing the number corresponding to the selection and pressing **ENTER**. If the device or device manufacturer you want is not listed, press **CTRL** + **N** to see the next page of devices or manufacturers.

5. If you selected a manufacturer in step 4, a list of devices produced by the selected manufacturer appears. Type the number corresponding to the device you wish to add to your Custom Menu, and press **ENTER** to add the device. (**CTRL** + **N** to see the next page of devices.)

6. If prompted, insert the Algorithm disk that contains the algorithm you wish to add. (If you do not have the correct disk inserted, the programmer will prompt you for the disk to insert when it is necessary.)

   If you get the message `Cannot access file_name.sys` the programmer could not find a file on the disk in the programmer's drive. Insert the Algorithm disk that contains the file.

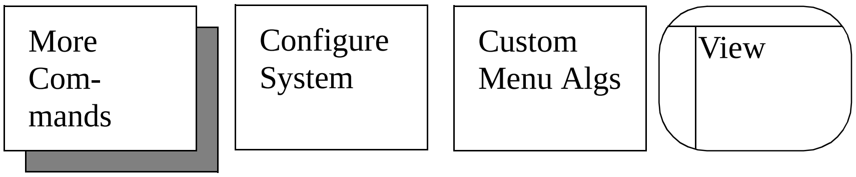
7. Select additional devices from the current screen (or press **F2** to return to the manufacturer list if you were selecting devices from a manufacturers list) or press **F2** (once or more if necessary) to return to the Add screen in order to select a different algorithm media or algorithm source. Repeat steps 2 through 6 as necessary.

   When you are done adding devices, press **F2** twice until you are prompted to insert the Custom Menu disk.

8. Insert the Custom Menu disk and press **ENTER** to save your changes to disk.

View

This command displays the contents of a Custom Menu.



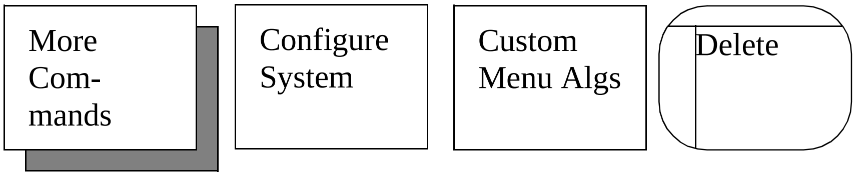
To view a Custom Menu, follow these steps:

1. Select the **View** command.
2. Insert the disk that contains your Custom Menu and press **ENTER** to continue. Follow the directions on the screen (inserting disks if prompted).

   When you are done viewing the Custom Menu, press **F2** to return to the Custom Menu Algs menu.

Delete

This command displays the Delete screen, which allows you to delete devices from a Custom Menu.



To delete devices from a Custom Menu, follow these steps:

1. Select the **Delete** command.
2. Insert the disk that contains your Custom Menu and press **ENTER** to continue. Follow the directions on the screen (inserting disks if prompted).

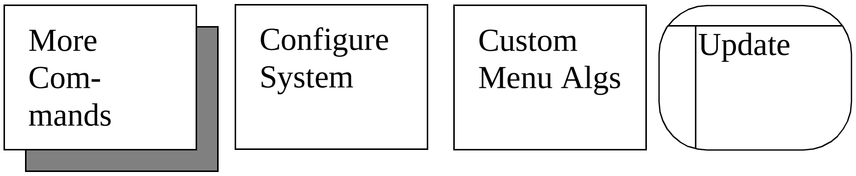
3. When the Custom Menu is displayed, in the **Delete** field enter the number corresponding to the file you want to delete, then move the cursor to the **Are you sure?** field and press **Y**.

***CAUTION: If you do not want to delete the file, do not press* ENTER.**

4. To delete the file, press **ENTER**. If you do not want to delete the file, or when you are done deleting devices, press **F2** to return to the Custom Menu Algs menu.

Update

This command displays the Update screen, which allows you to update the algorithms in a Custom Menu to the current version of the algorithms.



To update algorithms from a Custom Menu, follow these steps:

1. Select the **Update** command.

2. Set the following fields using the arrow keys to move from field to field and press **SPACE** to toggle a selected field:

Algorithm media (F, M) Select **F** if your current algorithm files are located on floppy disks. Select **M** if your current algorithm files are located on an MSM (Mass Storage Module). (Note for Legacy: This field appears only if an MSM is installed.)

Add/Create Select **C** (Create) to create a new Custom Menu disk and place the updated version of your old Custom Menu on the new disk.

New Custom Menu algorithm disk Select the drive letter that will contain your updated Custom Menu.

Old Custom Menu algorithm disk Select the drive letter that will contain you old Custom Menu.

Note: The algorithm source is always D (default) for an Update operation.

3. Insert the disk that contains your old Custom Menu and press **ENTER** to continue. Follow the directions on the screen (inserting disks if prompted).

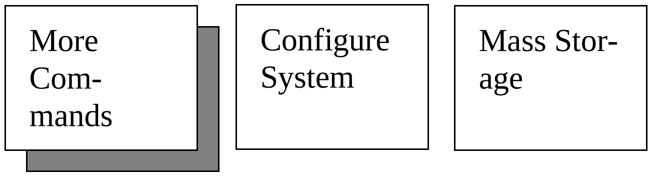
4. To update the Custom Menu, press **ENTER** (if you do not want to update the Custom Menu, press **F2**).

   When you are done updating Custom Menus, press **F2** to return to the Custom Menu Algs menu.

Mass Storage

Use this command to install the system software and algorithm files on the Mass Storage Module and configure your programmer to boot from the MSM.

Note: (Legacy) This parameter appears only if an MSM is installed in UniSite.



Note: For information on how to set up an MSM, refer to Chapter 2, "Setup and Installation."

To install system software and algorithm files, perform the following steps.

Note: To update system software and algorithm files stored on the MSM, refer to the directions in the User Notes and Update Instructions included in the Software Update Kit.

1. Select the **Mass Storage** command.
2. Set the following fields, using the arrow keys to move from field and pressing **SPACE** to toggle the selected field:

`Install the new version of software?`

Type **Y** if you want to install the new version of software on the MSM (drives H and I). Having files on the Mass Storage Module will allow for faster boot time (without a disk in drive A), and faster device selection operations. In order to install the new software, the programmer must be booted from the floppies of the version to be installed. If this option is enabled, the field Maintain Previous Configuration will appear.

`Maintain the Previous Configuration?`

Type **Y** if you want the previous configuration parameters to be maintained after you install the new version of software. If Y is selected, the programmer will automatically update the boot disk floppy with the configuration parameters, then load them onto the MSM. The factory default settings are always accessible through the Menu / Configure / Restore screen.

If this option is disabled, then the factory default configuration will be installed.

3. Press **ENTER** to start the installation.
4. Follow any instructions displayed on the screen. For instance, if the message `Insert Algorithm Set 3 Disk` appears, insert the Algorithm disk containing Algorithm Set 3.

CAUTION: Do not remove disks from UniSite-*xpi *during this operation unless UniSite-*xpi *prompts you to do so.

When the Mass Storage software installation is done, `OPERATION COMPLETE` is displayed in the message bar.

Device Checks

The commands on the Device Checks menu allow you to check devices you want to program and to check data in user memory. Commands available on this menu include the following:

- Sumcheck Display
- Compare Electronic ID
- Illegal Bit Check
- Blank Check
- Electronic Erase
- Underblow / Overblow (Logic Devices Only)
- Device Configure

Before you can execute a command described in this subsection, you must do the following:

1. Select a device. For more information, see the section titled “Select a Device” earlier in this chapter.
2. Insert and lock a device into the socket. (This applies only if you are checking a device, not if you are checking User RAM.)

Sumcheck Display

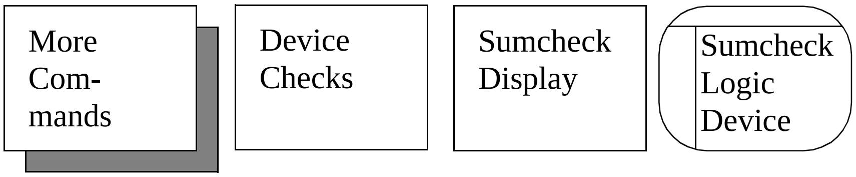
The sumcheck is a hexadecimal number that, when compared to the original data, allows you to verify that a copy of the data matches the original data. Remember, you must select a device before you calculate the sumcheck. The sumcheck is computed by adding each 8-bit byte in the specified data range into a 32-bit result with the carry dropped.

The sumcheck can be displayed as 8-bit, 16-bit, or both. To switch between displayed values, set the Checksum Calculation flag (located under Programming Parameters or in Sumcheck Display) to the desired value

Below, the options for sumchecking logic and memory are described, with logic devices explained first, followed by memory devices.

Sumcheck Logic Device

If you have selected a logic device, then the Sumcheck Logic Device screen appears.

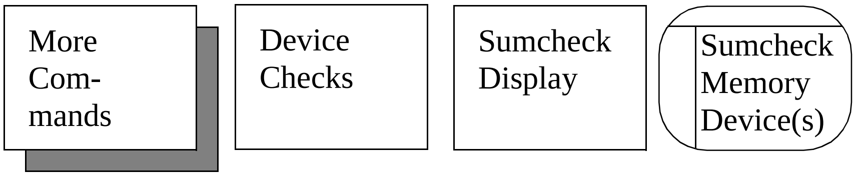


To sumcheck a logic device, follow these steps:

1. Select and socket a logic device.
2. Press **ENTER**, and UniSite-*xpi* calculates the 4-digit sumcheck of the fuse pattern. The sumcheck is displayed in the message bar.

Sumcheck Memory Device

For memory devices, UniSite-*xpi* calculates and displays the sumcheck for a single device, for each device in a set, or for an entire set.



Follow the steps below to calculate the sumcheck for a memory device:

1. Select and socket a memory device.
2. Enter the parameters described below.
3. Press **ENTER**, and UniSite-*xpi* calculates the 8-digit sumcheck. The sumcheck is displayed in the message bar.

The following parameters appear on this screen:

- **Sumcheck Entire RAM**—In addition to calculating the sumchecks for the device and set, this option specifies that the sumcheck operation calculate the sumcheck of the entire RAM. The sumcheck value is then displayed in the message bar.
- **Memory Begin Address**—Specifies the first address, in hex, of the first byte of data to be sumchecked (beginning RAM address). The default address is 0.
- **User Data Size**—Specifies the hexadecimal size of the data block to sumcheck. This value is normally equal to the device size or a multiple of the device size for sumchecking a set. Entering 0 resets User Data Size to the device size for sumchecking RAM or the file size for a sumchecking a data file.
- **Data Word Width**—Sets the word width, in bits, of the data to be sumchecked. For 8-bit (or larger) devices, the minimum word width is equal to the device word width and the maximum is 64. For 4-bit devices, valid choices are 4, 8, 16, and 32. Data Word Width should match the word width of the data bus in the target system for the device being programmed.
- **Total Set Size**—Specifies how many virtual devices are in the set to be sumchecked. Either enter a number between 1 and 99, or change one of the following parameters and UniSite-*xpi* will calculate the Total Set Size: Memory Begin Address, User Data Size, or Data Word Width. UniSite-*xpi* uses the following equation to calculate the Total Set Size:

$$\text{TOTAL SET SIZE} = \frac{\text{USER DATA SIZE}}{\text{DEVICE BLOCK SIZE}\left(\dfrac{\text{DATA WORD WIDTH}}{\text{DEVICE WIDTH}}\right)}$$

0987-1

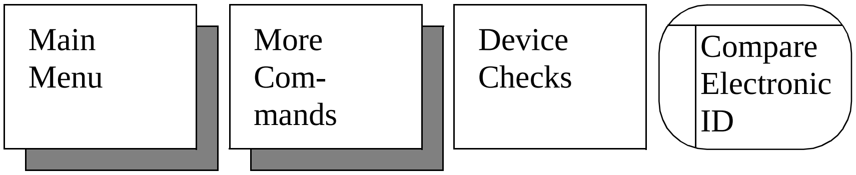
- **Next Operation Begins At**—A read-only field that specifies the address where the next sumcheck will start. This value is calculated from the Data Word Width, Device Block Size, Memory Begin Address, and device width.

- **For Member X of Y**—**X** specifies which device in the set is being sumchecked. **Y**, which is a read-only field, indicates how many devices are in the set(s). Values for **X** range from 1 to **Y**.
- **Individual Sumcheck**—A read-only field that displays the individual sumcheck for device number **X**.
- **Set Sumcheck**—A read-only field that displays the sumcheck for the entire set of **Y** devices.
- **Checksum Calculation** (4,8,D,B)—Determines the word size and display of the checksum for all device operations. The Odd / Even Byte Swap option affects this calculation. **4** directs UniSite-*xpi* to calculate and display a 4-digit, 4-bit checksum. This is only for use on 4-bit devices. **8** directs UniSite-*xpi* to calculate and display an 8-digit, 8-bit checksum. This is the default and is the only option for 8-bit devices. **D** directs UniSite-*xpi* to calculate and display the checksum based on the number of bits per word in the device width. This option is for use with 16- and 32-bit devices. **B** directs UniSite-*xpi* to calculate and display the checksum based on the number of bits per word in the device width, and also to display the 8-bit checksum.
- **Odd/Even Byte Swap**—When this option is enabled, you can swap high and low order bytes for 16-bit or 32-bit devices during a load, program, or verify operation. The organization of bytes in RAM is not altered. Swapping bytes is useful for manipulating 16- or 32-bit data when the target system has a different architecture than the original file convention. For example, Motorola 16-bit data files store the most significant bytes at even-byte locations; Intel 16-bit files store the most significant bytes at odd-byte locations.

Compare Electronic ID

This feature compares the electronic ID of a device with the electronic ID specified in the selected algorithm; this helps prevent accidental damage to a device.

Note: This command is not supported for devices selected by family/pinout code.



CAUTION: Turning this feature off could enable you to program a device with the wrong algorithm and could result in damage to that device. We recommend that you keep this feature turned on at all times.

To compare the electronic ID of a device with the ID stored in the selected algorithm, follow these steps:

1. Select and socket a device that supports electronic ID testing.

Note: You cannot use an electronic ID to automatically select the proper algorithm to program a device. You also cannot use this feature on devices that do not support electronic ID testing.

2. Press **ENTER**, and UniSite-*xpi* compares the electronic ID of the socketed device against the electronic ID of the selected device. If the electronic ID of the socketed device matches the electronic ID of the selected device, you see the following message in the message bar:

   ```
   OPERATION COMPLETE. Device = ssssssss
   ```

   where *ssssssss* is the electronic ID of the socketed device.

3. If UniSite-*xpi* detects an electronic ID that does not match the selected device type, you see the following message in the message bar:

   ```
   OPERATION FAILED: Electronic ID verify error. Device =
   ssssssss
   ```

   where *ssssssss* is the electronic ID of the socketed device.

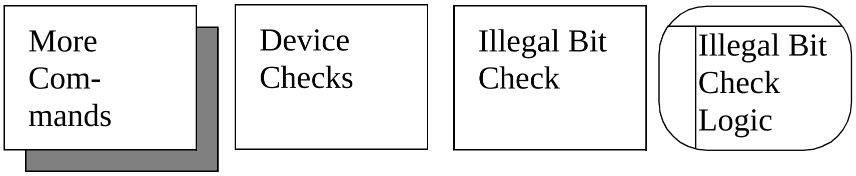
Illegal Bit Check

The Illegal bit test compares data in a device against data in UniSite-*xpi*'s RAM to determine if the device has already-programmed locations of incorrect polarity. For example, UniSite-*xpi* returns an illegal-bit error in the following situation: data in RAM indicates a specific bit should be in an unprogrammed state while the corresponding bit in the device is in a programmed state. The device cannot be programmed if UniSite-*xpi* detects an illegal bit.

If UniSite-*xpi* detects an illegal bit, it displays an error message. If the device is erasable, the illegal bit can be corrected by erasing the device. The Illegal Bit Check is supported for both logic and memory devices, but is not supported for electronically erasable devices.

Logic Device Illegal Bit Check

If you have selected a logic device, the Illegal Bit Check screen for logic devices appears.

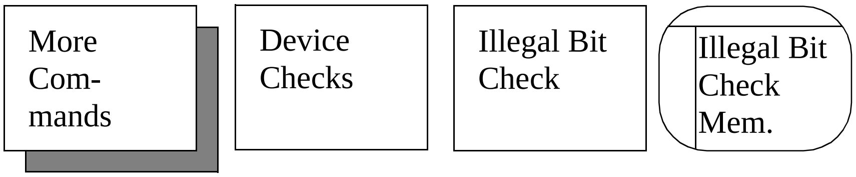


To check a logic device for illegal bits, follow these steps:

1. Select and socket a logic device.
2. Press **ENTER**, and UniSite-*xpi* begins the Illegal Bit Check. The results are displayed in the message bar.

Memory Device Illegal Bit Check

If you have selected a memory device, then the Illegal Bit Check screen for memory devices appear.



To check a memory device for illegal bits, follow these steps:

1. Select and socket a memory device.
2. Enter the parameters described below.
3. Press **ENTER**, and UniSite-*xpi* begins the Illegal Bit Check. The results are displayed in the message bar.

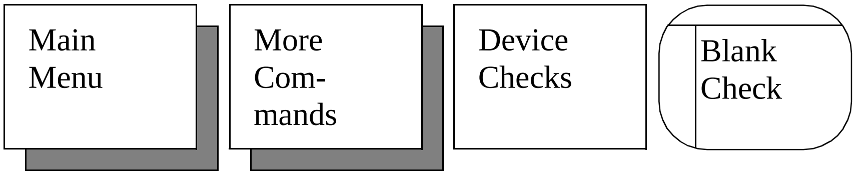
The following parameters appear on this screen:

- **User Data Size**—Specifies the size of the data block to check for illegal bits. This value is normally equal to the device size or a multiple of the device size for checking illegal bits of a set. If you enter 0, it is reset to device size for RAM or file size for a file operation. User Data Size works with Total Set Size to determine the total amount of bytes to check in a set.
- **Total Set Size**—The total number of parts in the set to check for illegal bits.

- **Data Word Width**—Sets the number of bits in the Data Word Width. For 8-bit (or larger) devices, the minimum word width is equal to the device width, and the maximum is 64. For 4-bit devices, your word width choices are 4, 8, 16, and 32. This value should match the data bus word width in the target system for the device being programmed.
- **Next Device**—Type the number corresponding to the next device in the set to check for illegal bits.
- **Next Operation Begins At**—This field is read-only and cannot be altered. It appears only to inform you where (at what hex address) the next operation begins.

Blank Check

The Blank Check command checks a device, ensuring that it is blank.



To blank check a device, follow these steps:

1. Select and socket a device.
2. Press **ENTER**. UniSite-*xpi* checks the device and respond with

   ```
   OPERATION FAILED: Non-blank device.
   ```

   if the device is non-blank, or

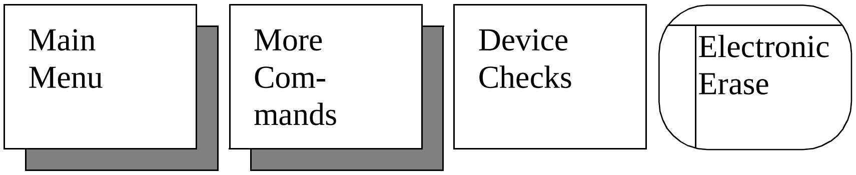
   ```
   OPERATION COMPLETE
   ```

   if the device is blank.

Electronic Erase

This command bulk erases an electronically erasable device

.



To erase a device, follow these steps:

1. Select and socket an electronically erasable device.
2. Press **ENTER** to erase the device. When finished, UniSite-*xpi* displays `Done` in the message bar.
3. If you try to erase a device that cannot be electronically erased, UniSite-*xpi* displays the following error.

```
Electronic bulk erase not supported by device.
```

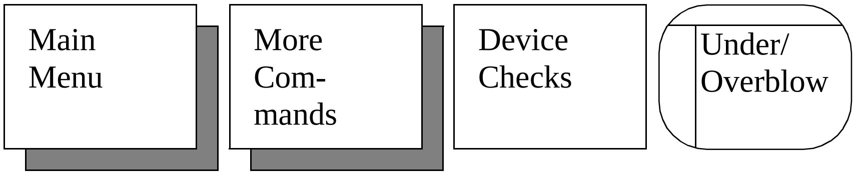
Note: You cannot access this screen if you have selected a device that cannot be electronically erased (a bipolar PROM, for example).

A blank check is run after a bulk erase operation if the blank check switch is enabled and if the selected device supports blank check.

This device operation is not necessary for most electronically erasable devices. The Electronic Erase is part of the normal programming cycle. Before programming an electronically erasable device, UniSite-*xpi* checks the device and displays a warning if the device is non-blank. If you enable the erasing of the device, UniSite-*xpi* erases the device before programming the device.

Underblow/Overblow (Logic Devices Only)

The under/overblow feature compares the fuse map of a logic device with the fuse map in RAM.



An underblow condition means that the device's fuse is intact, but the data in memory indicates that it should have been blown. An overblow means that the device's fuse is blown, but should have remained intact. (The under/overblow feature is not supported for POF devices.)

To use the under/overblow feature, follow these steps:

1. Select and socket a logic device.
2. Press **ENTER** to begin the test. The under/overblow screen is displayed. If the data source does not have proper fuse data (for the specified device type), a message appears saying that the file is not initialized. Type **C** to initialize the file.

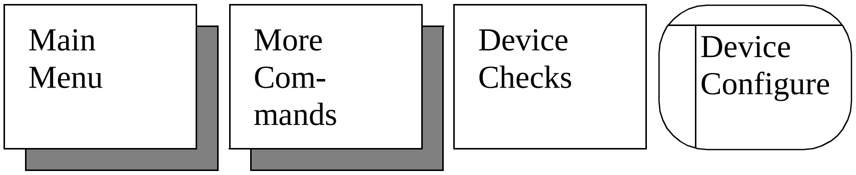
With two exceptions, the data shown on the screen is displayed in a format similar to that of the fuse editor. The first exception is that additional character symbols are used to display overblown (B) and underblown (U) data. The second is that, unlike in the fuse editor, no data can be edited. The fuse number corresponding to the cursor's location appears at the top of the screen. To move the cursor, use the arrow keys. The editor commands are described later in this chapter.

Underblow/Overblow Commands

Command	Keystrokes	Description
Next Block	**CTRL** + **N**	Displays the next page of under/overblow data
Prev Block	**CTRL** + **P**	Displays the previous page of under/overblow data
Jump to Fuse	**CTRL** + **B**	Moves the cursor to a specific fuse. A highlighted area appears just after the `^B: Jump to Fuse` prompt at the bottom of the screen. Type in the fuse number that you want to jump to and press **ENTER**.
Search Pattern	**CTRL** + **F**	Searches for one of four character symbols within the under/overblow data. The four characters you can search for are: **X** (intact fuse) **-** (blown fuse) **B** (overblown fuse) **U** (underblown fuse) After you select the search character, the search begins at the current cursor position and continues until either a match is found or the end of the fuse map is reached.
Exit	**F2**	Exits the Under/Overblow screen and returns UniSite-*xpi* to the Device Checks menu

Device Configure (Devices Supporting Sector Configuration Only)

The device configure feature allows you to set the switches for erasing, programming, and protecting sectors on devices that support sector configuration. These settings are used during the Program and Electronic Erase operations.



To use the device configure feature, follow these steps:

1. Select a device that supports sector configuration.
2. If you wish to enable electronic erase, set the `Erase EE device` field to **Yes** in the Program screen.
3. On the Device Configure screen, use the arrow keys to move from field to field and press **SPACE** to toggle between Y (Yes) and N (No).

   The following fields exist for each sector:

   - **Erase—**Set to Y if you wish the sector to be erased when a device erase operation is performed. Set to N to disable erase on the sector.
   - **Program—**Set to Y if you wish the sector to be programmed when a device program operation is performed. Set to N to disable programming on the sector.
   - **Protect—**Set to Y if you wish the sector protect to be enabled. Note that not all devices that support sector erase and program also have support for sector protect. Set to N to disable protect on the sector.
4. When you are finished setting the erase, protect, and program fields, press **F2** to return to the Device Checks menu, or press **F1** to return to the Main Menu.

Edit Data

Use the commands on the Edit Data menu to make changes to data stored in RAM or to data stored in a disk file. When you select the Edit Data command, UniSite-*xpi* displays a menu corresponding to the type of device that is currently selected. There are separate editors for memory and logic devices. For logic devices, there is a fuse map and test vector editor. For memory devices, there is a memory editor.

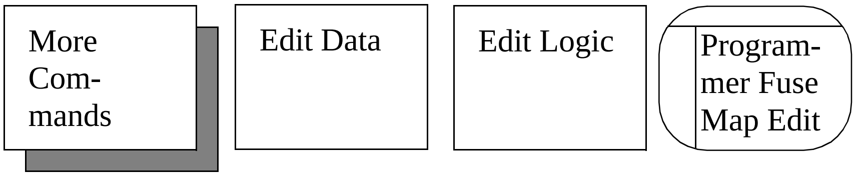
The commands on the Edit Logic menu are explained first, followed by those on the Edit Memory menu.

Edit Logic Menu

The Edit Logic menu appears if you have selected a logic device. This menu contains the Edit Logic, Vector Edit, Fill Fuse Map, and Clear Vectors commands.

Edit Fuse Map

This is the data editor for logic devices.



To edit fuse data, follow these steps:

1. Enter the parameters described below. If necessary, load the data into UniSite-*xpi*.
2. After you have selected the parameters, press **ENTER**. The screen clears and show the fuse map data.
3. If the data source does not have proper fuse data, a message appears indicating the file is not initialized. Press **C** to initialize the fuse map to an unprogrammed (blank) state.
4. While editing, you can enter either data or commands. To edit the fuse map data, move the cursor to the fuse you want to change. Press **SPACE** to toggle the fuse to the desired state.

   The fuse editor commands are described after the parameter list.

Note: In general, any paging command or an exit command causes all currently displayed data to be written to the data source.

The different options and commands for the editor are explained below.

- **Source** (R,D)—Specifies the source of the data to be edited. Press **SPACE** to toggle between **R** (RAM) and **D** (disk).
- **Filename**—Specifies the name of the disk file containing the fuse data to edit. This option appears only if you select disk as the Source. The filename must follow standard DOS conventions.
- **Data Representation** (X/–,0/1)—Specifies how the data in RAM or in a data file appears on the terminal's screen. The two choices for this parameter are **X** and **–**, or **0** and **1**. Press **SPACE** to toggle between the two options. **X** and **0** represent an unprogrammed state; **–** and **1** represent a programmed state.

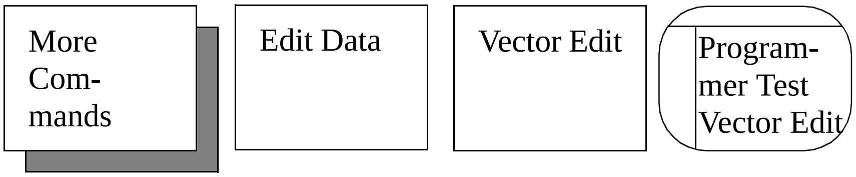
The commands described below are available when using the fuse editor.

Command	Keystrokes	Description
Prev Block	**CTRL + P**	Displays the previous block of fuse data.
Next Block	**CTRL + N**	Displays the next block of fuse data.
Jump to Fuse	**CTRL + B**	Moves the cursor to a specific fuse. A highlighted area appears just after the `^B: Jump to Fuse` prompt at the bottom of the screen. Type in the fuse number that you want to jump to and press **ENTER**.

Command	Keystrokes	Description
Restore Block	**Ctrl** + **U**	Returns the current page of fuse data to its original state (before editing that page). Only the data visible on the screen is affected by this command. This command works only if you have not moved off the currently displayed page of edit data since any changes were made.
Exit Editor	**F2**	Exits the fuse editor.

Vector Edit

The vector editor allows you to edit test vectors you have created for a particular logic device.



To edit test vectors, follow these steps:

1. Set the parameters for test vector editing. The parameters are described below.
2. Press **Enter**. The screen displays the test vectors (if any) for the selected device.
3. If the source data does not match the device type selected, a message appears indicating the file is not initialized. Type **C** to initialize it.
4. While editing, you can enter either data or commands. You may type only certain test conditions and use only certain keyboard commands in the vector editor. The available editor commands are described after the parameter list.

The available parameters are described below.

- **Source** (R,D)—Specifies the source of the test vectors to be edited. Press **Space** to toggle between **R** (RAM) and **D** (Disk).
- **Filename**—Specifies the name of the disk file containing the test vector data to edit. This option appears only if you select disk as the Source. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is **a:16r8.dat**.
- **Edit Begin Vector**—Specifies the first test vector you want to edit. Move the cursor to the Edit Begin Vector field and enter the desired vector number. The vector number you type must be less than or equal to the last vector in RAM or the disk file. This field defaults to 1.

Test Conditions

The test conditions and the allowed commands are listed in the following tables.

Vector Symbol	Description
0	Drives the specified input pin low.
1	Drives the specified input pin high.
2-9	Super-voltages, defined by the device's manufacturer.
B	Buried register preload.
C	Drives the specified input with a sequence of logic states: in this case low, high, and low (high clock).
D	A single transition that drives the specified input low using a fast slew rate.
F	Specifies that a particular input or output pin is to be floated (tri-stated).
H	Verifies that the specified output pin is high.
K	Drives the specified input with a sequence of logic states: in this case high, low, and high (low clock).
L	Verifies that the specified output pin is low.
N	Specifies that a particular input or output pin is floating (tri-stated). UniSite-*xpi*'s **F** and **N** conditions perform the same function.
P	Identifies a preload vector and invokes a preload algorithm. This character is allowed on the clock pin ONLY; otherwise it is treated as an X.
U	A single transition that drives the specified input high using fast slew rate; equivalent to **C** without returning to the low state. If more than 16 Ds or Us are used in any one test vector, the extra Ds or Us are ignored during test.
X	Ignores the state of an output pin. UniSite-*xpi* applies a logic level specified by a JEDEC file. X field value (1 or 0) or a low is used as the default value.
Z	Verifies the specified input or output pin has high impedance. UniSite-*xpi* will toggle the pin using a small current during this test.

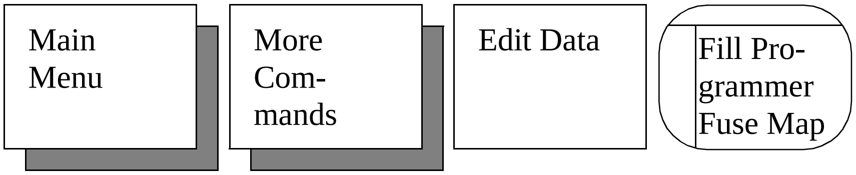
Note: C, K, U, and D are clock functions that allow setup time.

Vector Editor Commands

Command	Keystrokes	Description
Jump to Vector	**CTRL** + **B**	Moves the cursor to a specific vector. A highlighted field appears just after the `^B: Jump to Vector` prompt at the bottom of the screen. Enter the vector number to jump to and press **ENTER**.
Delete Vector	**CTRL** + **D**	Deletes the current vector where the cursor is located.
Insert Default	**CTRL** + **I**	Inserts a default vector, which consists of a vector of all Xs (the character that expresses the ignore input and output test condition). Use the default vector for creating new test vectors. To create a new test vector, insert a default vector and change that vector to contain the test conditions that you specify; the legal test conditions are listed in the previous table. When you enter this command, the default vector is placed in front of the vector highlighted by the cursor.
Next Block	**CTRL** + **N**	Displays the next block of vectors.
Prev Block	**CTRL** + **P**	Displays the previous block of vectors.
Restore Block	**CTRL** + **U**	Restores the current page of vector data to its original state (before editing this page began). Only the data visible on the screen is affected by this command. This command is effective only if there have been no paging commands since changes were made.
Save Vector	**CTRL** + **W**	Saves the current vector (at the cursor) to a temporary buffer.
Repeat Saved	**CTRL** + **V**	Inserts the vector that was last saved using the **CTRL** + **W** command. When you execute this command, the saved vector is placed in front of the vector highlighted by the cursor.
Exit Editor	**F2**	Exits the vector editor and returns to the previous screen.

Fill Fuse Map

This command enables you to fill the fuse map with a variable.



This command is useful if you have loaded a fuse map into memory and you want to clear the fuse map from memory.

You can also perform this operation automatically as part of a download operation in two different ways. First, you can use the **F** field in a JEDEC file, or second, you can enable the Fill Memory option on the Communication Parameters screen.

To fill the fuse map with a variable, follow these steps:

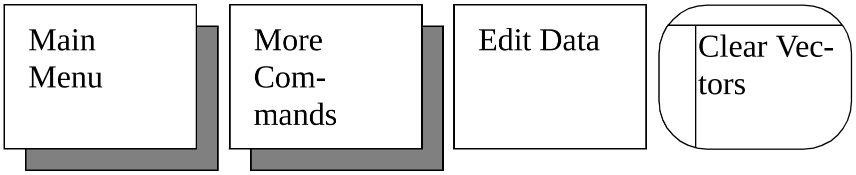
1. Enter the desired one-digit value (0 or 1) in the Fill Variable field. Press **SPACE** to toggle the variable between 1 and 0. **0** represents an unprogrammed state, while **1** represents a programmed state.
2. When the desired fill variable is displayed, press **ENTER**.
3. UniSite-*xpi* fills the fuse map with the specified variable. When done, UniSite-*xpi* displays

   ```
   Done
   ```

   in the message bar.

Clear Vectors

This command enables you to clear the current test vectors from memory.



To clear vectors, press **ENTER**.

Note: Only vectors in UniSite-xpi's RAM are cleared. This command cannot be used to clear vectors stored on a disk.

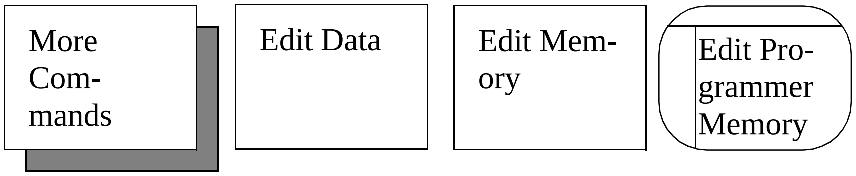
Edit Memory Menu

The Edit Memory menu appears if you have selected a memory device.

This menu contains the Edit Memory, Complement, Data Copy, Fill Memory, and Swap Data commands.

Edit Memory

Use the Edit Memory command to edit the data for a memory device.



To edit data stored in memory, follow these steps:

1. Specify the memory editing parameters.
2. Press **ENTER** to enter the editor. Depending on the selected word width, the 4-, 8-, or 16-bit word width memory editor screen appears.
3. To change data on the screen, move the cursor to the memory location to change and type the new characters over the old ones. Enter the data either in hex or ASCII mode; select the mode using the **TAB** command (see below). When the cursor is moved around, its location is represented at the top of the screen as a hex address.

The editor commands are described following the parameter list.

The available parameters are described below.

- **Source** (R,D)—Specifies the source of the data to be edited. Press **SPACE** to toggle between **R** (RAM) and **D** (Disk).
- **Filename**—Specifies the name of the disk file containing the data to edit. This option appears only if you select disk as the Source. The filename must follow standard DOS conventions and can contain a drive designator. An example of a valid filename is **a:27c256.dat**.
- **Edit Data Word Width** (4,8,16,32)—Selects a 4-, 8-, 16-, or 32-bit Data Word Width. Press **SPACE** to toggle between the 4-, 8-, 16-, and 32-bit options. If you select **8**, the editor treats all addresses as byte addresses. If you select **16** or **32**, the editor treats all addresses as 16- or 32-bit word addresses and the Edit Odd/Even Byte Swap feature is enabled.
- **Edit Address Offset**—Specifies the address you want assigned to the first byte of data in user memory. Using the address offset can save much calculation time on files written on a host system and then downloaded to UniSite-*xpi*. For example, if your host system data file was written using a begin address of 1000H, you could specify an offset of 1000H. Edit data would then be displayed on UniSite-*xpi*'s screen beginning with address 1000H.
- **Edit Begin Address**—Specifies the first address you wish to edit. Enter the 1- to 6-digit hex address. This address must be equal to or greater than the edit address offset. The edit address offset value subtracted from the edit begin address value cannot be greater than the user RAM size.

Memory Editor Commands

Only certain keyboard commands may be used in the memory editor. The allowed commands are listed in the following table.

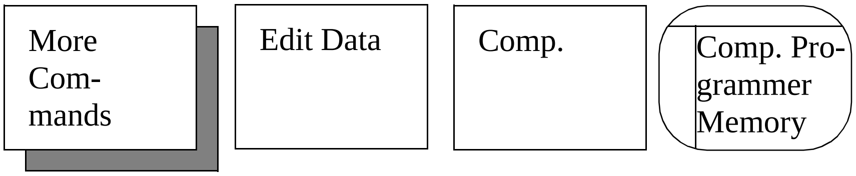
Command	Keystrokes	Description
Jump to address	**CTRL** + **B**	Moves the cursor to a specific memory address. When this command is selected, the cursor moves to the Jump to Address field. Enter the address you want to jump to and press **ENTER**.
Delete byte	**CTRL** + **D**	Deletes the entire byte with 8-bit data and deletes the entire word with 16-bit data. All the data after the current character position is moved one address position down. The end of RAM has an FF inserted. If a disk file is used, the file gets smaller.
Exchange	**CTRL** + **E**	Allows you to search for a certain pattern and replace it with another pattern: 1. Press **CTRL** + **E**. The cursor moves to the Exchange field at the bottom of the screen. 2. Type the pattern to search for (any hex value up to 8 digits), followed by **ENTER**. The cursor then moves to the With field at the bottom of the screen. 3. Type the pattern you want inserted in place of the existing pattern, followed by **ENTER**. If the pattern cannot be found, a message is displayed and the cursor remains in its original position. When exchange data is entered in 4-bit mode, the upper nibble of data is blank so only up to four characters can be entered in the field.
Next Block	**CTRL** + **N**	Displays the next block of memory data.
Prev Block	**CTRL** + **P**	Displays the previous block of memory data.
Restore Block	**CTRL** + **U**	Returns the current page of data to its original state (before editing began). The page is restored only if there have been no paging commands.

Command	Keystrokes	Description
Search Pattern	**CTRL** + **F**	Allows you to search for a particular hex pattern of up to 8 digits: 1. Press **CTRL** + **F**. The cursor moves to the Search field at the bottom of the screen. 2. Type in the pattern to search for (any hex value up to 8 digits), followed by **ENTER**. If the pattern cannot be found, a message is displayed and the cursor remains in its original position. When searching for data in 4-bit mode, the upper nibble of data is blank so only up to four characters can be entered in the search field.
Start/Stop Insert	**CTRL** + **T**	Toggles the state of the Insert mode between Insert and Overtype. If the Insert field is displayed in reverse video, the editor is in Insert mode. If the Insert field is displayed in normal video, the editor is in Overtype mode. In Insert mode, data is inserted at the current cursor position, and all data after that is moved up into higher memory or file addresses. If editing a RAM file, data at the end of RAM is lost. If editing a disk file, the file gets larger. The insert is not complete until the last hexadecimal character (8- or 16-bit) is entered. The cursor moves by byte for 8-bit data, or by word for 16-bit data. In Overtype mode, data entered replaces the current data. When not in Insert mode, the arrow keys move the cursor by character. For 8-bit data, the data is entered in bytes, and for 16-bit data, the data is entered in words.
Exit Editor	**F2**	Exits the memory editor and returns UniSite-*xpi* to the Edit menu.

Command	Keystrokes	Description
Toggle Hex/ASCII Modes	**TAB**	Toggles the mode for data entry. When in hex mode, data is entered on the left side of the screen and the only valid entries are hex characters. When in ASCII mode, data is entered on the right side of the screen and any printable ASCII character can be entered. ASCII mode is not allowed when in 4-bit mode.

Complement Data

The Complement command converts each bit of data within the specified data block to its opposite value (one's complement).



To complement data stored in memory, follow these steps:

1. Specify the parameters described below.
2. Press **ENTER** to begin the complement function. UniSite-*xpi* displays the following message when the operation is completed.

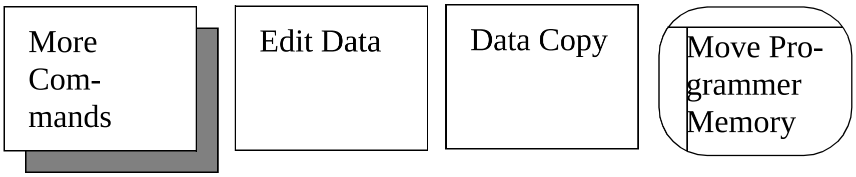
   ```
   Done
   ```

The available parameters are described below.

- **Memory Address**—The memory address at which the complement operation begins. The value entered may be any 1- to 6-digit hex address. The address cannot be greater than the User RAM size.
- **Block Size**—The number of bytes (in hex) that is complemented. Move the cursor to the block size window and enter the block size (from 1 to 6 hex digits). The block size, added to the memory address, cannot exceed the user memory size.

Data Copy

The Data Copy command copies a block of data from one location to another.



To copy data stored in memory, follow these steps:

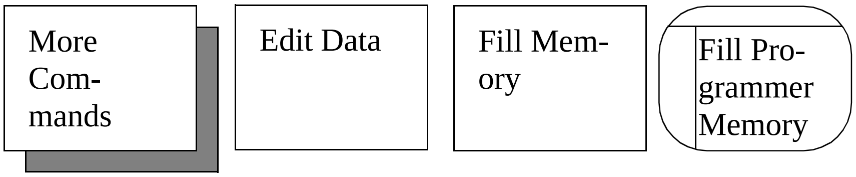
1. Specify the parameters described below.
2. Press **ENTER** to begin the Data Copy operation. UniSite-*xpi* displays `Done` when the process is complete.

The available parameters are described below.

- **From Memory Address**—The first memory address of the data block you want to copy data from. Enter any 1- to 6-digit hex address. The address cannot be greater than the user RAM size.
- **To Memory Address**—The first address of the data block you want to copy data to. Enter any 1- to 6-digit hex address. The address cannot be greater than the user RAM size.
- **Block Size**—The size (in hex) of the data block to be copied. If the sum of the block size and either the From Memory or the To Memory address values exceed user memory size, a warning message is displayed.

Fill Memory

The Fill Memory command fills a specified block of data with a 2-digit hex value.



To fill a block of memory, follow these steps:

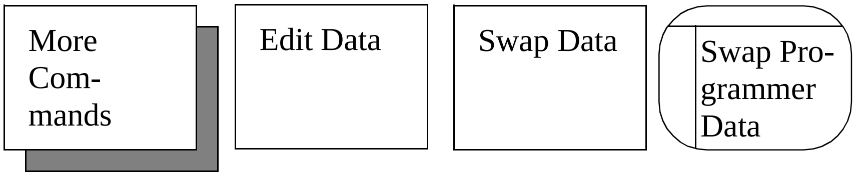
1. Specify the parameters described below.
2. Press **ENTER** to begin the Fill function. UniSite-*xpi* displays `Done` when the operation is completed.

The available parameters are described below.

- **Memory Begin Address**—The memory address at which the fill operation begins. Enter any 1- to 6-digit hex address. The address cannot be greater than the user RAM size.
- **Block Size**—The number of bytes (in hex) that are filled. Move the cursor to the block size window and enter the block size (from 1 to 6 hex digits). The block size, added to the memory address, cannot exceed the user memory size.
- **Fill Variable**—The 2-digit hex data variable that is used to fill the specified block. Enter any value between 00 and FF.

Swap Data

The Swap Data command performs either a byte swap or a nibble swap on the data in a specified block of User RAM.



To swap a block of memory, follow these steps:

1. Specify the parameters described below.
2. Press **ENTER** to begin the Swap function. UniSite-*xpi* displays `Done` when the swap is complete.

The available parameters are described below.

- **Swap Mode**—The type of swap to perform. Choose between Byte mode and Nibble mode. In Byte mode, the high byte and the low byte will be swapped. In Nibble mode, the high order nibble will be swapped with the low order nibble. Press **SPACE** to toggle between the two modes.
- **Memory Begin Address**—The memory address at which to begin the swap. Enter any 1- to 6-digit hex address.
- **Block Size**—The number of bytes (in hex) to be swapped. Enter any 1- to 6-digit hex block size. The block size, added to the memory address, cannot exceed the user memory size. Also, the block size must be an even number if using Byte mode.

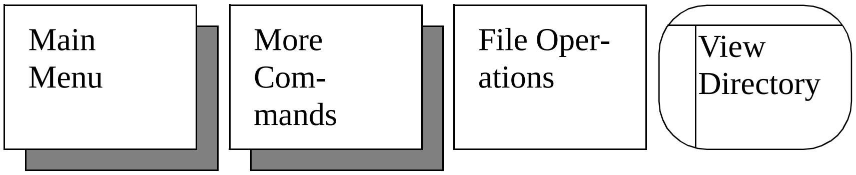
File Operations

From the File Operations menu, you can access several file manipulation and directory commands. These functions help you move and copy files, view file directories, and organize and maintain your disks. The following sections describe each of the File Operations commands in the order that they appear on the file menu screen.

Note: A file disk has a capacity of 720KB or 112 files, whichever comes first.

View Directory

This command displays the file directory of the disk in the disk drive.

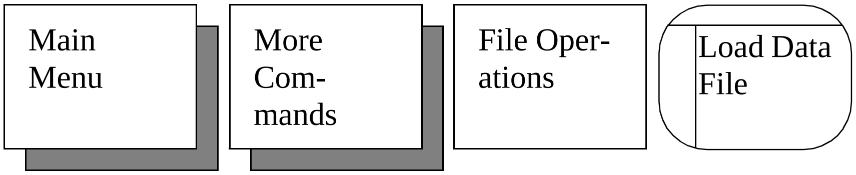


To view a directory, follow these steps:

1. Insert the disk you want to view into the disk drive.
2. Press **ENTER** to view the directory. The View Directory command can view the directory of any DOS-compatible 3.5” disk. The files on drive A are displayed first.
3. UniSite-*xpi* displays 28 files at one time. If your disk(s) contains more than 28 files, they are displayed on the next page(s). Press **CTRL** + **N** to advance to the next page of files.

Load File

This command loads a disk file into RAM. **Do not use this command to load files from a PC or from a file server.** See the Download Data command for information on transferring files to UniSite-*xpi* from a PC or from a file server.



To load a file into UniSite-*xpi*'s RAM, follow these steps:

1. Insert the disk containing the file you want to load into the disk drive.
2. When you select the Load File command, the dialog window displays a directory of the disk in the disk drive. If you do not see your file, press **F2** to return to the File Operations menu. Insert the disk containing your file into the disk drive and return to the beginning of this step.
3. UniSite-*xpi* displays 28 files at one time. If your disk has more than 28 files, they are displayed on the next page(s). Press **CTRL** + **N** to advance to the next page of files.
4. Specify the parameters described below. Be sure to include a filename.
5. Press **ENTER** to begin the loading. Once the disk file is in RAM, you may perform several operations, such as edit or program device, on the file; refer to these subsections of the Command chapters for more information on editing and programming.

Note: The User Data Size field does not appear on the Load File screen. However, this parameter is still updated to reflect the size of the file loaded for use in other screens.

The available parameters are described below:

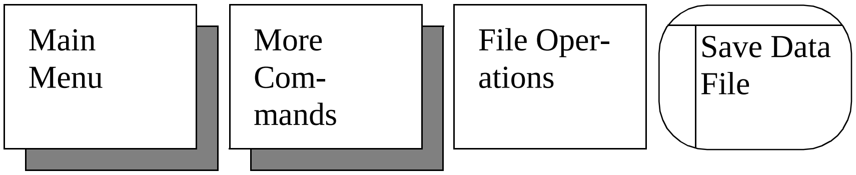
- **Filename**—Specifies the name of the disk file to load. The filename must follow standard DOS conventions and can contain a drive designator. An example of a valid filename would be **b:27c256.dat** or **a:16l8.dat**.
- **Memory Begin Address**—Specifies the first address in memory to load data into. This option appears only if you have selected a memory device. The default address is 0.

If you selected a logic device, the fuse map and vectors from the disk are loaded.

If your files contain data formatted in other than RAM Image Binary (Intel Hex, format 83, for example), use the Transfer Data / Input From Disk command.

Save File

This command allows you to save the data in RAM to a disk file. **Do not use this command to save a file on a PC or a file server.** See the Upload Data command for information on transferring files to a PC or a file server from UniSite-*xpi*.



To save data in RAM to disk, follow these steps:

1. Insert the disk you want to save the data to in the disk drive.
2. When you select Save File, the dialog window fills with a directory of the disk in the disk drive. If you do not want to save your file to this disk, press **F2** to return to the File Operations menu. Insert the disk you want to save the data to into the disk drive and return to the beginning of this step.
3. Specify the parameters described below. Be sure to include a filename.
4. Press **ENTER** to begin the save.

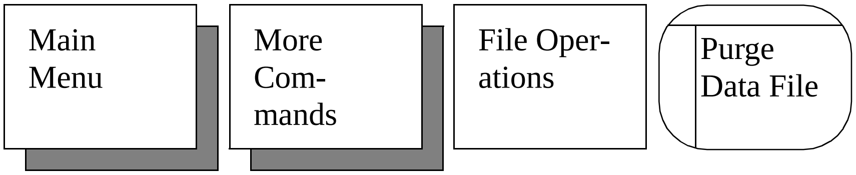
Note: If you are saving information for a logic device, the fuse map, security fuse state, and vectors are saved.

- **Filename**—Specifies the name of the disk file to save RAM data into. This may be a new filename, or an existing filename that you want to overwrite. If you are writing to an existing file, the data previously in the file is replaced by the new data. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename would be **a:27c256.dat** or **b:16l8.dat**.
- **Memory Begin Address**—Specifies the first address in RAM to take data to be saved. This option appears only if you have selected a memory device. The default address is 0.
- **User Data Size**—Specifies the size, in hex bytes, of the data block to save. This value is normally equal to the device size. This option appears only if you have selected a memory device.

A saved file is stored in RAM Image Binary format. If you want to store a file in some other format, use the Transfer Data / Output to Disk command.

Purge File

This command deletes a file, or group of files, from a disk.



To purge a file from a disk, follow these steps:

1. Insert the disk with the file you want to delete into one of the disk drives.
2. When you select the Purge File command, the dialog window fills with a directory listing. UniSite-*xpi* displays up to 28 files at one time. If there are more than 28 files, press **Ctrl** + **N** to display the next page of files. Press **Ctrl** + **P** to display the previous page of files.

   If you do not see the file you want to delete, press **F2**, insert another disk, and return to the beginning of this step.
3. Move the cursor to the Filename field and enter the name of the file you want to delete. Remember to specify which drive holds the file you want to delete.

Note: You can use an asterisk (*) as a wildcard. For example, to purge both 27512.dat and 27128.dat, you could type 27.dat.*

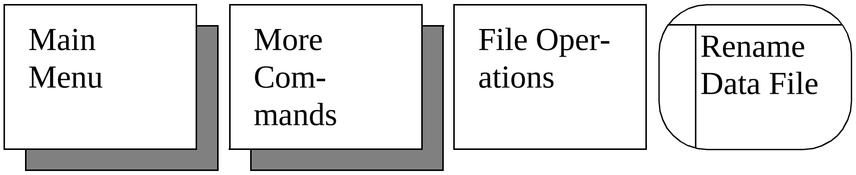
4. Move the cursor to the **Are you sure?** field and press **Y**.

CAUTION: If you do not want to delete the file, do not press Enter.

5. To delete the file, press **Enter**. If you do not want to delete the file, press **F2** to return to the File Operations menu.

Rename File

This command changes the name of a file.



To rename a file, follow these steps:

1. Insert the disk with the file you want to rename into one of the disk drives.

Note: You cannot rename a file to a different disk drive than the current one displayed on the screen.

2. When you select the Rename File command, the dialog window fills with the directory listing. UniSite-*xpi* displays up to 28 files at one time. If there are more than 28 files, press **Ctrl** + **N** to display the next page of files. Press **Ctrl** + **P** to display the previous page of files.

   If you do not see the file you want to rename, press **F2**, insert another disk, and return to the beginning of this step.

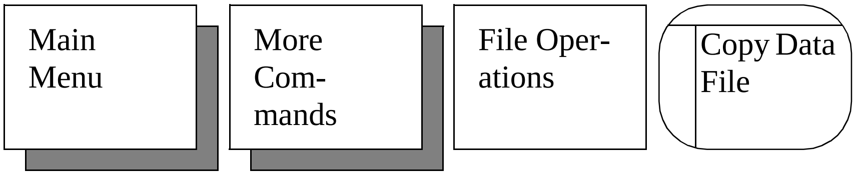
3. Move the cursor to the **From** field and enter the current name of the file you want to rename. Remember to specify which drive holds the file you want to rename.
4. Move the cursor to the **To** field and enter the new name for the file you want to rename. Remember to specify which drive holds the file you want to rename.

CAUTION: If you do not want to rename the file, do not press Enter.

5. To rename the file, press **ENTER**. If you do not want to rename the file, press **F2** to return to the File Operations menu.

Copy File

Use the Copy command to copy a file or a group of files.



To copy a file (or group of files), follow these steps:

1. Insert the disk with the file you want to copy into one of the disk drives.
2. When you select the Copy File command, the dialog window fills with the directory listing. UniSite-*xpi* displays up to 28 files at one time. If there are more than 28 files, press **CTRL** + **N** to display the next page of files. Press **CTRL** + **P** to display the previous page of files.

   If you do not see the file you want to copy, press **F2**, insert another disk, and go back to step 1.
3. Move the cursor to the **From** field. Enter the name of the source file. Remember to specify which drive holds the file(s) you want to copy.
4. Move the cursor to the **To** field and enter the name of the destination file. Remember to specify which drive holds the file(s) you want to copy.
5. Move the cursor to the **Single Drive File Copy to Different Disk** parameter. If you want to copy the file to a different disk and use the same drive, set this parameter to **Y**. UniSite-*xpi* prompts you to insert the source disk or destination disk at the appropriate times. This operation uses RAM as a temporary storage buffer, which alters the contents of RAM.

*Note: When copying with the Single Drive File Copy to Different Disk parameter set to **N**, you are prompted to swap disks when necessary. Only one file at a time can be copied using this method.*

Otherwise, if you want to copy the file from one drive to the other or to the same disk, set this parameter to **N**. Make sure the destination file has a different filename than the source file.

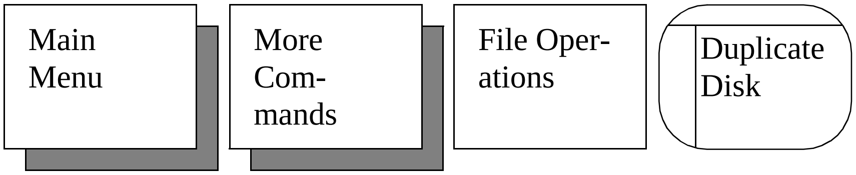
If you do not want to copy the file, do not press ENTER.

Note: When copying with the Single Drive File Copy to Different Disk parameter set to ***N****, you can copy a group of files by using an asterisk (*) as a wildcard in the name of the source and destination file.*

6. To begin the copying, press **ENTER**. If you do not want to copy the file, press **F2** to return to the File Operations menu.

Duplicate Disk

Use this command to duplicate an entire disk.



To duplicate an entire disk, you can use either the Duplicate Disk command, or you can use the DOS DISKCOPY command.

Using UniSite-*xpi*

Follow the steps below to duplicate a disk on UniSite-*xpi*.

1. Insert the disk you want to duplicate (the source disk) into drive A.

CAUTION: Do NOT use an Algorithm disk or the System disk as the destination disk because the original contents of the destination disk will be lost.

2. Insert the blank disk (the target disk) into drive B.
3. If you want to duplicate the disk, move the cursor to the **Are you sure?** field and press **Y**.

CAUTION: If you do not want to duplicate the disk, do not press Enter; this operation erases the contents of the destination disk. Also, this operation uses RAM as a temporary storage buffer and alters the contents of RAM.

4. Move the cursor to the **Do You Want To Verify Disk** field and enable the disk verification procedure.

   To disable the verification and speed up the process, press **N**. To enable the verification, press **Y**.

5. Finally, press **ENTER** to begin the disk duplication.

Note: After every disk duplication operation, the verification parameter returns to **Y** *and the* **Are you sure?** *parameter returns to* **N**.

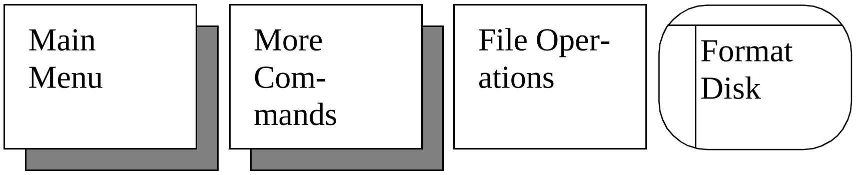
Using DOS

If you have access to a DOS-based PC with a 3.5” disk drive, we suggest that you use the DOS DISKCOPY command to make a copy of your Algorithm disk and System disk. If you use DOS, make sure you use DISKCOPY and not COPY. The backup must be an exact, bit-for-bit, sector-for-sector copy of the original. For more information, see your DOS manual.

Note: The Boot disks ***must first be formatted*** *using the UniSite-xpi Format Disk Operation.*

Format Disk

Use this command to prepare a data disk for use. A disk must be formatted before it can be used. Use this command, for example, if you need to create a new Boot Files disk or Algorithm disk for extended algorithms, Keep Current algorithms, or custom algorithms.



To format a disk, follow these steps:

1. Insert the disk to be formatted into either disk drive.
2. Press **SPACE** to select the drive that the new disk will be in when formatting.
3. Move the cursor to the **Disk Integrity Check** field and select whether or not you want UniSite-*xpi* to perform a disk integrity check. If enabled, the check is performed while the disk is being formatted, and looks for bad sectors on the disk. Although this test does take more time to complete, it helps ensure the integrity of the newly formatted disk.

   To enable the disk integrity check, press **Y**. To disable the disk integrity check, press **N**.
4. When you are ready to format the disk, type **Y** at the **Are You Sure?** prompt and press **ENTER**.

   If you are not ready to format a disk, press **F2** to return to the File Operations menu.
5. UniSite-*xpi* checks the disk in the disk drive, making sure it is not a System disk. If it is an Algorithm disk or a System disk, UniSite-*xpi* displays the following message:

   ```
   WARNING: system disk in drive. Hit return to continue, ^Z
   to abort.
   ```

CAUTION: Do NOT format a UniSite-*xpi* Algorithm disk or System disk sent to you by Data I/O, because the original contents of the disk will be lost.

To format a disk for use as an Algorithm or System disk, press **ENTER**. If you do not want to format a new Algorithm disk or System disk, press **CTRL** + **Z**.

Job File

Using the Job File feature, you can record a series of keystrokes which can then be replayed later.

Job files allow you to perform setup operations without re-keying all the parameters each time a new device is selected. For example, if you regularly program five different devices, you could create and save five different job files, each specifying particular options for a device.

You may store up to 10 job files on each System disk. Each job file may contain up to 499 keystrokes, although a typical job file contains 10 to 20 keystrokes. Job files can be viewed with the View Directory command, where they appear as JF*N*.JOB, where *N* is a number between 0 and 9. For example, if you have two job files on a disk, they may show up as JF0.JOB and JF2.JOB. If you have job files on another disk, insert the disk and press **F4** to display the job file directory for the current disk. Whenever you change disks with job files on them, pressing **F4** displays the current job file directory.

Suggestions on Construction of a Job File

You should not include any Quick Copy commands in your job files. Also, you should not include any operations requiring the insertion or removal of a disk or module. This is because a job file does not stop until the entire file has been played back.

The first command in a job file should be **F1**. This way, your job file always starts from the Main Menu, preventing “runaway” job files.

Note: Because screens and key functions may change when the software is updated, a particular job file may only be used with the version of software it was created with. For example, if you created a job file with one version of software, you should re-create the job file so it is compatible with the new software.

Recording a Job File

To record a job file, follow these steps:

1. Press **ESC**, **CTRL** + **J** to start recording the job file. Each keystroke entered from now on is stored in the job file.
2. Press **F1** as the first command in your job file. (Although this is not necessary, it helps prevent “runaway” job files.)
3. Enter all the parameters you want recorded. For example, you might want to select a device and then choose its programming parameters by using the Edit Programming Parameters command.
4. When you have entered all the keystrokes you want to store in the job file, press **ESC**, **CTRL** + **J** to stop recording the job file. The Job File screen appears.
5. Select a file number to store the newly created job file in. For example, if you want this job file to appear as job file number five on the Job Files screen, press **5**, **ENTER**. If you select a file number already in use, UniSite-*xpi* prompts you to press **ENTER** to overwrite the existing file. If you want to preserve the existing job file, press **CTRL** + **Z**.

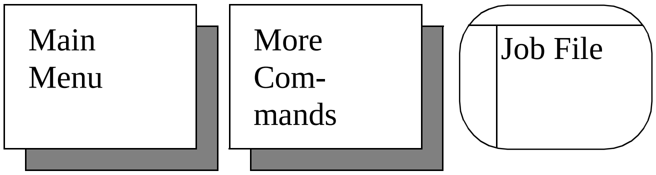
6. Move the cursor down to the **Enter Description** field and type in a name for the job file that has just been recorded. The description can be up to 31 characters long and should be followed by **ENTER**.

   To save the job file, press **ENTER**. While saving, the action symbol rotates. When done, you are returned to the last screen displayed during job file recording.

   To not save the job file, press **F2** or **F1**.

Playing Back a Job File

Use the Job File command to play back a pre-recorded job file.



To play back a job file, follow these steps:

1. Go to the Job Files screen.
2. You see a listing of the job files stored on the disk in the disk drive. To select a job file from this disk, type the number of the job file you want to play back and press **ENTER**. For example, to play back the fifth file on the list, press **5**, **ENTER**.

   To view a list of job files stored on another disk, insert the disk and press **F4**. Pressing **F4** reconstructs the job file directory.
3. UniSite-*xpi* now plays back the keystrokes that were recorded. Each screen displayed while you were recording keystrokes is shown (briefly).
4. After the job file is played back, UniSite-*xpi* displays the following message in the message bar:

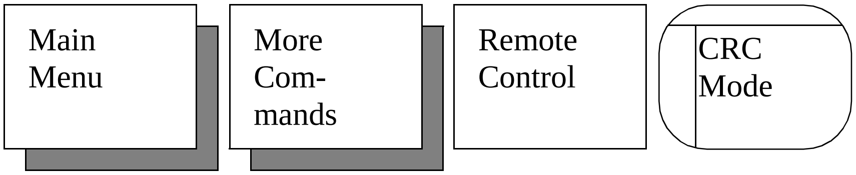
   ```
   Job file playback ended
   ```

   The last screen recorded while you were creating the job file is now displayed. If an error occurred during playback, an error message is displayed and the job file must be re-recorded.

Note: Because screens and key functions may change when the software is updated, a particular job file may only be used with the version of software it was created with. For example, if you created a job file with one version of software, you should re-create the job file so it is compatible with the new software.

Remote Control

This command puts UniSite-*xpi* in Computer Remote Control mode.

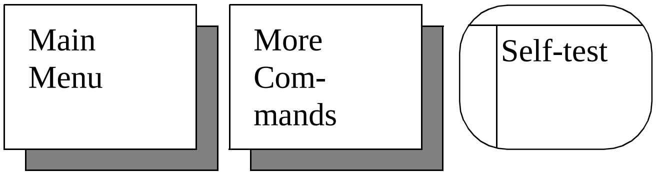


Chapter 2, "Setup and Installation," provides instructions on how to set up your system. Chapter 5, "Computer Remote Control," provides information on how to use CRC and describes the CRC command set.

To exit remote control, press **SHIFT** + **Z** + **ENTER** on the terminal's keyboard, or send a **Z**, **ENTER** command from the host computer.

Self-test

The Self-test command allows you to test circuits and subsystems in UniSite-*xpi*, verifying proper operation or isolating possible problem areas.



An automatic self-test is also performed each time UniSite-*xpi* is powered up. If errors occur during the power-up test, the Self-test screen is displayed, showing the areas that failed.

Some self-tests (such as User RAM test) can be disabled so they are not checked during powerup. Disabling and enabling self-tests is done in the **More Commands...Interface** menu.

Note: For details (help) on the function of each self-test, move the cursor to the test you want information about and press F3.

Halting a Self-test

You can stop a self-test anytime during its operation. To halt a self-test, press **CTRL** + **Z**.

Running the Self-test

To perform a Self-test, follow these steps:

1. Make sure all device sockets are empty.

CAUTION: Executing the System RAM test or the User RAM test erases any data in RAM.

2. Select the test mode. You can select either one-pass or continuous testing. To toggle modes, move the cursor to the Test Mode prompt and press **SPACE**. One Pass testing runs the specified test once. Continuous testing runs the specified test until there is a failure or until you halt the procedure by pressing **CTRL** + **Z**.

Note: There may be a delay before UniSite-xpi responds to the Ctrl-Z if the programmer is running the system RAM test.

3. To test all hardware, move the cursor to the **Perform All Tests** prompt and press **ENTER**. To test a particular item, move the cursor to the desired test and press **ENTER**.

Interpreting Self-test Results

Four conditions are used as status indicators on the self-test screen:

```
? UNTESTED

P PASS

F FAIL

- NOT INSTALLED
```

When testing begins, a **?** appears next to the untested areas. As each test completes, either **P** (pass) or **F** (fail) appears next to the test name, showing the results of the test performed.

If a hardware item is not installed, a **–** appears.

For example, if you are testing the Pin Driver boards on a UniSite-*xpi* that has ten Pin Driver boards installed (UniSite-*xpi* can hold 17), ten **?** symbols and seven **–** symbols appear before the testing begins. After testing, UniSite-*xpi* displays ten **P** symbols and seven **–** symbols if all the Pin Driver boards pass.

During testing, the message area of the self-test screen indicates that testing is in progress. During the System RAM test, the Remote and the Terminal indicators blinks to indicate that testing is in progress.

If **?** symbols still appear next to some test names when the testing has completed, it may be because some other test(s) need to pass before the indicated test may be run. For example, the Waveform board test must pass before the Pin Control Unit test executes. All the installed UniSite-*xpi* hardware must pass self-test.

Note: All of the installed UniSite-xpi hardware must pass self-test before any other operations can take place.

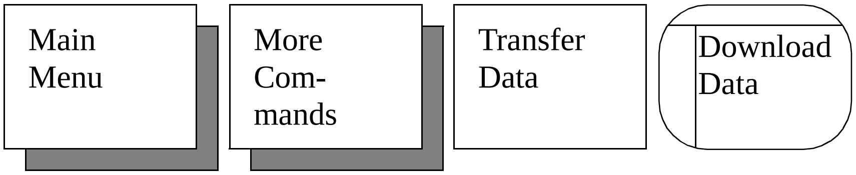
Transfer Data

Use the commands on the Transfer Data menu to move data files back and forth between UniSite-*xpi* and the host computer.

The Transfer Data menu contains the following seven commands: Download Data, Upload Data, Compare Data, Format Select, Input From Disk, Output to Disk, and Serial Output. Each of these commands is described in this subsection.

Download Data

Use the Download Data command to specify downloading parameters and to execute the download operation.



Downloading moves a data file from a host computer to UniSite-*xpi*'s RAM or disk.

Before you download data, specify the variables for the parameters listed below, then enter a command in the **Download Host Command** field. The information in the command line is a command that your host computer (the computer containing the file to download) recognizes as an instruction to begin the download operation. Press **ENTER** to execute the download. UniSite-*xpi* displays

```
Data transfer complete
```

in the message area when the download is complete.

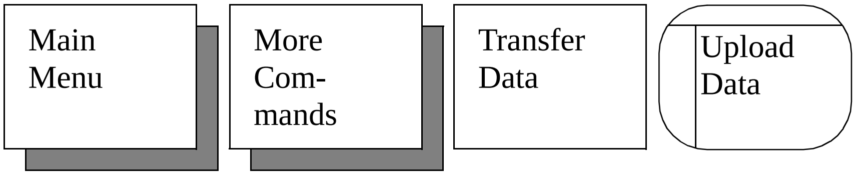
- **Source** (R,T)—Specifies which of UniSite-*xpi*'s ports is connected to your host computer. Press **SPACE** to toggle between **R** (Remote port) and **T** (Terminal port).
- **Destination** (R,D)—Specifies the destination of the data that is being downloaded from the host computer. Press **SPACE** to toggle between **R** (RAM) and **D** (disk).
- **Filename**—Specifies the name of the disk file in which to save the downloaded file. This option appears only if you specify disk as the Destination. The filename must follow standard DOS conventions, and can include a drive designator. An example of a valid filename is **a:27256.dat**.
- **I/O Translation Format**—Selects the translation format of the data in the file. A list of formats UniSite-*xpi* supports is available on the Format Select screen in the Transfer Data menu, and also in Chapter 6 of this manual. If you know the number for your format, you can enter it from this screen. If you do not know the correct code number, go to the Format Select screen, find the format you want, and enter the correct number from that screen. Entering the format number from the Format Select screen changes the Translation Format parameter on this screen. If you are using the Altera POF format, you must select the desired POF device before you perform a data transfer operation.
- **I/O Addr Offset**—Enter either the beginning hex address of the host computer's data file or the first address you want to capture within a file. This field appears only when a non-JEDEC format has been selected. UniSite-*xpi* subtracts this address from addresses received from the host computer to determine where, either in the user RAM or in the disk file, the data will be loaded. Entering **FFFFFFFF** sets the first address received as the I/O offset for the rest of the download.
- **Memory Begin Address**—Specifies the first address, in hex, where the first byte of data is stored from the source port. This field appears only when a non-JEDEC format has been selected. If the destination is RAM, it is a beginning RAM address; if the destination is disk, it is a beginning disk file address. The default address is 0.
- **User Data Size**—Specifies the hexadecimal size, in bytes, of the data block to be downloaded. This field appears only when a non-JEDEC format has been selected. The default is 0, which directs UniSite-*xpi* to receive all the data in the file. After the download is complete, a value equal to the number of bytes received is set here. If a value less than the size of the data received is entered, only the number of bytes equal to that value are actually stored.

- **Download Host Command**—Enter the appropriate host command line here to download the data. This line may be up to 58 characters long. UniSite-*xpi* generates a return character to terminate the line when transmitted to the host. To clear a previously entered command, enter a blank command by pressing **SPACE**, **ENTER**.

  If you are using HiTerm, use the **TR *filename*** command, where ***filename*** is the name of the file to download. For more information, see the *HiTerm User Manual* at the back of this binder. For an example of using HiTerm to download data from a PC, see Session 17 in Chapter 3.

Upload Data

Use the Upload Data command to specify uploading parameters and to execute the upload operation.



Uploading moves a data file from UniSite-*xpi*'s RAM or disk to the host computer. To upload a data file, follow these steps:

1. Before you upload data, specify the variables for the parameters listed below.
2. Enter a command in the **Upload Host Command** field. The information in the command line is a command that your host computer (the computer receiving the data file) recognizes as an instruction to begin the upload operation.
3. Press **ENTER** to start the upload. During the upload, the action symbol rotates. When done with the upload, UniSite-*xpi* displays the following message in the message bar:

```
Data Transfer complete. Data sum = xxxxxxxx (8-bit)
```

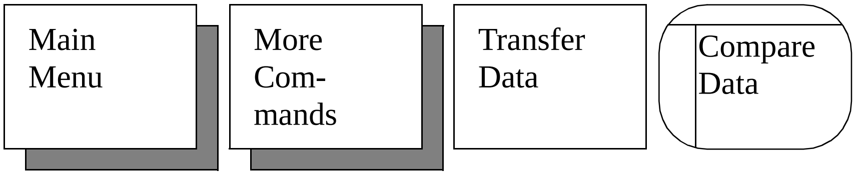
The available parameters are described below.

- **Source** (R,D)—Specifies where the data to be uploaded is located. Press **SPACE** to toggle between **R** (RAM) and **D** (disk).
- **Filename**—Specifies the name of the disk file to upload to the host. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions, and can include a drive designator. An example of a valid filename is **b:27256.dat**.
- **Destination** (R,T)—Specifies which port the data file is sent through. Press **SPACE** to toggle between **R** (Remote Port) and **T** (Terminal Port).

- **I/O Translation Format**—Specifies the translation format in which the file is to be generated. The format specified here must be the same as that expected by the host computer. A list of formats supported by UniSite-*xpi* appears on the Format Select screen in the Transfer Data menu, and also in the front of Chapter 6 of this manual. If you know the number for your format, you can enter it from this screen. If you do not know the format number, go to the Format Select screen, find the format you want and enter the number from that screen. Entering the format number from the Format Select screen changes the Translation Format parameter on this screen.
- **I/O Addr Offset**—Enter the beginning address of the upload file. This field appears only when a non-JEDEC format has been selected. This value is added to the address of the data in memory (relative to the Memory Begin Address of 0) and output as the I/O address. A value of FFFFFFFF sets the I/O Offset to 0.
- **Memory Begin Address**—Specifies the first address, in hex, from which the first byte of data is retrieved. This field appears only when a non-JEDEC format has been selected. If the source is RAM, it is a beginning RAM address. If the source is Disk, it is a beginning disk file address. The default address is 0.
- **User Data Size**—Specifies the hexadecimal size, in bytes, of the data block to be uploaded. This field appears only when a non-JEDEC format has been selected. Enter the value of the number of bytes to upload. Entering **0** directs UniSite-*xpi* to upload the entire contents of UniSite-*xpi*'s RAM. Or, if Disk is specified as the Source, entering **0** directs UniSite-*xpi* to upload the entire disk file.
- **Upload Host Command**—Enter the appropriate host command line here to direct the host to accept the uploaded data. This line may be up to 58 characters long. UniSite-*xpi* generates a return character to terminate the line when transmitted to the host. To clear a previously entered command, enter a blank command by pressing **SPACE**, **ENTER**.

Compare Data

The Compare Data command compares data in user memory with the data file downloaded from the host computer.



This command is useful if you want to verify that you transferred a complete and accurate copy of a data file.

The current I/O format is used to translate the incoming data from the serial port. (JEDEC format cannot be used with this command.) This operation is identical to a downloading operation, except data is compared with, rather than written to, memory.

1. Before you compare data, specify the variables for parameters listed below.
2. Enter a command in the Download host command field. The information in the command line is a command that your host computer recognizes as an instruction to begin the download operation.
3. Press **ENTER** to start this command. UniSite-*xpi* displays the following message if the two data files are identical.

   ```
   Data transfer complete
   ```

   If the data files are NOT identical, UniSite-*xpi* displays the following message:

   ```
   Data verify error. Data sum = XXXXXXXX
   ```

   If data in memory does not correspond with data sent from the host, and the terminal is not on the same port as the port receiving the data from the host, UniSite-*xpi* displays the following message:

   ```
   compare fail at AAAAAA:XX not YY
   ```

   where *AAAAAA* is the address, *XX* is the memory data, and *YY* is the host's data.

   If the terminal is on the same port, then the following message is displayed:

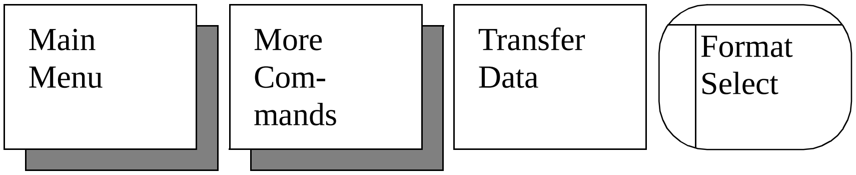
   ```
   Data verify error. Data sum = ssssssss
   ```

- **Source** (R,T)—Specifies which of UniSite-*xpi*'s ports is connected to the computer with the data file that is to be used to compare with the memory data. Press **SPACE** to toggle between **R** (Remote port) and **T** (Terminal port).
- **Data Location** (R,D)—Specifies where the data to be compared is located. Press **SPACE** to toggle between **R** (RAM) or **D** (disk).
- **Filename**—Specifies the name of the disk file you want compared. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions and can contain a drive designator. An example of a valid filename would be **a:27256.dat**.
- **I/O Translation Format**—Specifies the data translation format of the data in the file. A list of formats UniSite-*xpi* supports is available on the Format Select screen in the Transfer Data menu, and also in the front of Chapter 6 of this manual. If you know the number for your format, you can enter it from this screen. If you do not know the format number, go to the Format Select screen, find the format you want, and enter the format number from that screen. Entering the format number from the Format Select screen changes the Translation Format parameter on this screen.
- **I/O Addr Offset**—Specifies the beginning address of the downloaded data file to be compared. This field appears only when a non-JEDEC format has been selected. Entering FFFFFFFF causes UniSite-*xpi* to default to the first incoming address as the lowest address to be compared.

- **Memory Begin Address**—Specifies the first hexadecimal address of data to compare with data from the Source port. If the data location is RAM, it is a beginning RAM address. If the data location is Disk, it is a beginning disk file address. This field appears only when a non-JEDEC format has been selected. The default address is 0.
- **User Data Size**—Specifies the hexadecimal size, in bytes, of the data block to be downloaded and compared from the Source to the data location. This field appears only when a non-JEDEC format has been selected. Normally, you should enter a zero here so all of the data is compared. After the compare operation is complete, a value equal to the number of bytes compared is set here. If a value less than the size of the data received is entered, only the number of bytes equal to that value are actually compared.
- **Download Host Command**—Enter the appropriate host command line here to download the data. This line may be up to 58 characters long. UniSite-*xpi* generates a return character to terminate the line when transmitted to the host. To clear a previously entered command, press **SPACE**, then **ENTER**.

Format Select

The Format Select command selects the translation format to use.

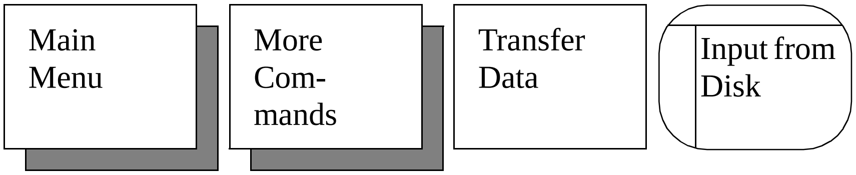


Translation formats (a form of transmission protocol) are used when uploading / downloading data between UniSite-*xpi* and a host computer. Only the formats listed in the Translation Formats chapter of this manual are recognized by UniSite-*xpi*. If your host computer does not generate code into one of the listed formats, edit it to match one of the supported formats.

The Format Select screen displays a list of all UniSite-*xpi* supported formats to choose from. At the bottom of this screen is a format entry field. Enter the number of the format that you want to use and press **ENTER**. When you select a translation format from this screen, the same format is entered in all of the other Transfer Data screens (i.e., Download Data, Compare Data).

Input From Disk

Use the Input From Disk command to load a data file from disk if the data is stored in a translation format. Depending on the settings of the Destination parameter, the data in the disk file is loaded either into RAM or into another disk file.



To input a data file from disk, follow these steps:

1. Insert the disk containing the file to input into the drive.
2. Specify the settings for the parameters listed below.
3. Press **ENTER** to start this command. UniSite-*xpi* displays the following message when the file has been loaded.

```
Data transfer complete
```

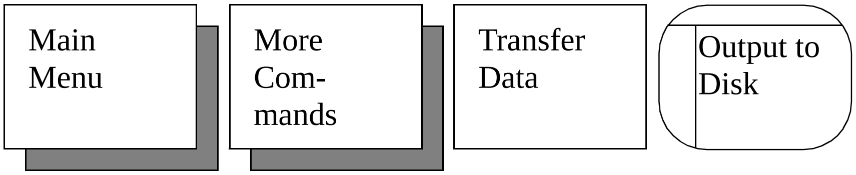
The following parameters appear on the Input from Disk screen:

- **Input Filename**—Specifies the name of the disk file from which formatted data is taken. The filename parameter must follow standard DOS conventions and can contain a drive designator.
- **Destination** (R,D)—Specifies the destination for the data. Press **SPACE** to toggle between **R** (RAM) and **D** (another disk file).
- **Filename**—Specifies the filename for the disk file into which data is sent. This option appears if you specify disk as the Destination. The filename must follow standard DOS conventions and can contain a drive designator.
- **I/O Translation Format**—Specifies the format for the data to be input. See Chapter 6 for a complete list of supported formats.
- **I/O Address Offset**—Enter the beginning hex address, or the first address you want to capture within a file, of the disk file's data. This field appears only if a non-JEDEC format has been selected. UniSite-*xpi* subtracts this address from addresses received from the host computer to determine where the data is loaded into memory. Entering FFFFFFFF sets the I/O Offset equal to the first address received.
- **Memory Begin Address**—Specifies the first address, in hex, to where the first byte of data is stored in memory. If the destination is RAM, it is a beginning RAM address. If the destination is disk, it is a beginning disk file address. The default address is 0. This field appears only if a non-JEDEC format has been selected.

- **User Data Size**—The User Data Size specifies how many bytes (in hex) is read during the Input from Disk operation. The default User Data Size is 0, which causes all of the data to be input. After the operation is complete, UniSite-*xpi* updates the User Data Size parameter to reflect the number of bytes stored to the destination. If a value less than the size of data file input is entered, the number of bytes equal to that value are actually stored. This field appears only if a non-JEDEC format has been selected.

Output To Disk

The Output To Disk command saves data from a disk file or from RAM to another disk file. The data in the newly created disk file can be stored in any of the supported translation formats.



This command is useful if you already have a data file and want to save it in another translation format. This command is just like the Upload Data command, except that the formatted data is sent to a disk file rather than to a port.

Follow the steps below to output data in a translation format and store it in a disk file.

1. Make sure there is enough space on the disk in the drive to hold the file you are writing.

   Use the View Directory command on the File Menu to check the amount of free space on the disk you are going to save the file to.

2. Specify the settings for the parameters listed below.

3. Press **ENTER** to initiate this command. UniSite-*xpi* displays

   ```
   Data transfer complete
   ```

   when the file has been output to the disk.

The following parameters appear on the Output To Disk File With Translation screen:

- **Source** (R,D)—Select the Source for the disk file data. Press **SPACE** to toggle between **R** (RAM) and **D** (disk).
- **Filename**—Specifies the name of the disk file from which the data is taken. This option appears only if disk is selected as the Source. The filename parameter must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is **b:27256.dat**.
- **Output Filename**—Specifies the name of the disk file you want the formatted data sent to. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is **a:27256.hex**.

- **I/O Translation Format**—Specifies the translation format for the data. A complete listing of the formats is given in the Translation Format chapter of this manual.
- **I/O Address Offset**—Enter the desired beginning address of the disk file. This field appears only if a non-JEDEC format has been selected. UniSite-*xpi* adds this value to the address of the data in memory (relative to the Memory Begin Address of 0) and output the sum as the I/O address. Entering FFFFFFFF sets the I/O Address Offset to 0.
- **Memory Begin Address**—Specifies the first address, in hex, from where the first byte of data is retrieved to write to the disk output file. If the source is RAM, it is a beginning RAM address. If the source is Disk, it is a beginning disk file address. The default address is 0. This field appears only if a non-JEDEC format has been selected.
- **User Data Size**—Specifies the hexadecimal size, in bytes, of the data block to be output, with translation, to the output file from the source. Enter the number of bytes to output. Entering zero sets User Data Size to the total number of hex bytes in UniSite-*xpi* User RAM or the size of the source disk file if disk is used as source. This field appears only if a non-JEDEC format has been selected.

Serial Output

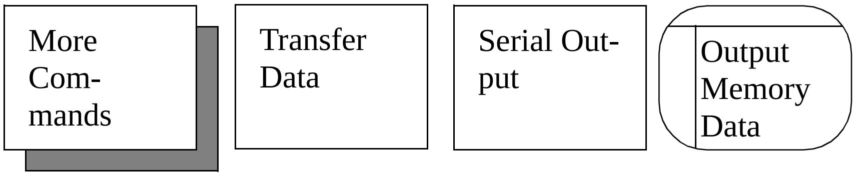
Use the Serial Output command to send data from UniSite-*xpi* to a serial device, such as a printer.

This command is a useful way to obtain a quick copy of programming or other device-related data. Serial Output does not do any data translating. If a logic device is selected, the fuse data is output by fuse number and the vector data is output by vector number. If a memory device is selected, the data is output by address in hex format.

Output Memory Data is explained first, followed by Output Logic Data.

Output Memory Data to Serial Port

When a memory device has been selected, the Serial Output command is output as a specified memory block to one of UniSite-*xpi*'s serial ports.



When all parameters described below have been entered, press **ENTER** to begin the transfer operation.

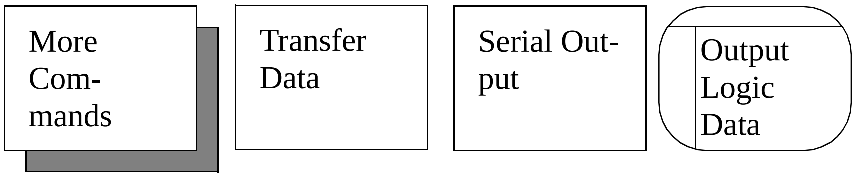
The parameters for this command are listed below.

- **Source** (R,D)—Specifies the source for the data. Press **SPACE** to toggle between **R** (RAM) and **D** (Disk).
- **Filename**—Specifies the disk file to use as the Source. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is **b:27128a.dat**.
- **Destination** (R,T)—Specifies the destination for the data. Press **SPACE** to toggle between **R** (Remote port) and **T** (Terminal port).

- **Number Of Lines Between Form Feeds**—Specifies the number of printed text lines you wish to have per page. The default is 0 (no form feed).
- **Memory Begin Address**—Specifies the first address, in hex, of the first byte of data to be retrieved and sent out the serial port. If the source is RAM, it is a beginning RAM address. If the source is disk, it is a beginning disk file address. The default address is 0.
- **User Data Size**—Specifies the hexadecimal size, in bytes, of the data block to be output. Enter the number of bytes to output. Entering zero sets the User Data Size to the total number of bytes in UniSite-*xpi* User RAM or the size of the source disk file if the disk is used as the source.

Output Logic Data to Serial Port

If you have selected a logic device, the Output Logic Data to Serial Port screen appears.



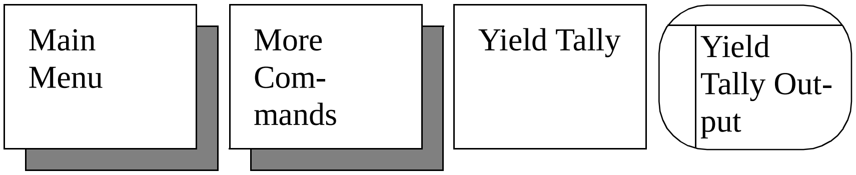
The parameters on that screen are described below:

When all parameters are entered, press **ENTER** to begin the transfer.

- **Source** (R,D)—Specifies the source for the data. Press **SPACE** to toggle between **R** (RAM) and **D** (Disk).
- **Filename**—Specifies the disk file to use as the data source. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions and can contain a drive designator. An example of a valid filename is **b:16l8.dat**.
- **Destination** (R,T)—Specifies the destination for the data. Press **SPACE** to toggle between **R** (Remote port) and **T** (Terminal port).
- **Number Of Lines Between Form Feeds**—Specifies the number of printed text lines you wish to have per page. The default is 0 (no form feed).
- **Starting Vector Number**—Specifies the first vector to be output. The default is 1, which causes the vector listing to start at the first vector.
- **Number Of Vectors**—Specifies the total number of vectors you wish to output. The default is 0, which causes no vectors to be output.

Yield Tally

The Yield Tally command allows you to maintain programming information on devices that have been programmed.



This information can be very useful in a manufacturing environment where device yield statistics must be kept. Yield statistics are maintained on the last 16 device types programmed. If you attempt a yield tally on a 17th device, UniSite-*xpi* drops the statistics for the oldest device. UniSite-*xpi* stores the manufacturer name and its part number or family / pinout codes as the device name in the yield tally record.

The yield data is stored on the System disk in a file called **ytally.sys**. If the file does not exist, enabling the Yield Tally option creates a blank copy of this file on the disk.

Note: You may upload the yield statistics while in CRC mode by using command ***43]****. The CRC command* ***46]*** *clears the yield tally statistics. CRC commands are described in Chapter 5, "Computer Remote Control."*

Space is allocated for the Yield Tally data files on the System disk. When the Yield Tally function is invoked without the System disk, the error message

```
System disk not found. Cannot access yield data.
```

is displayed on the terminal or, if you are in CRC mode, error code 9A is returned.

Yield Total

The Yield total does not include those devices with error conditions that are not recorded in one of the four categories: illegal bit, mis-verify, device not programmable, structured test failure. Examples of these conditions are continuity check, electronic ID error, or overcurrent.

The Yield Tally screen provides statistics for the following categories:

- **Device Name**—Manufacturer's name and part number, family / pinout codes. Statistics for the last sixteen device types are kept.
- **Total Count**—The number of individual devices of the same type that UniSite-*xpi* attempted to program.
- **Good Part**—The number of devices successfully programmed.
- **Illegal Bit**—The number of devices that failed because they did not pass non-blank test or Illegal Bit Check.
- **Verify Fail**—The number of devices that failed because they did not verify.
- **Struct Fail**—The number of logic devices of the same type that failed the logic structured vector test.

- **Device Not Programmable**—The number of devices that could not be programmed because they contained bits that required more programming pulses than were specified.

In the non-blank test, if you stop the operation without programming the device, the illegal bit count is incremented by one. If you proceed with the programming operation, the illegal bit count remains unchanged while the yield tally records the result of the programming operation.

To erase the entire set of statistics, press **CTRL** + **E**. Press **F2** to go to the previous menu or **F1** to return to the Main Menu. The total number of programming errors is not recorded. This value may be derived by adding the values in the individual error columns.

Transparent Mode

Transparent mode can be entered from all of UniSite-*xpi* screens with the following exceptions:

- Editor screens
- Under/overblow screen
- Yield Tally screen
- Help screens
- CRC mode

With Transparent mode, you can communicate with a host computer connected to one of UniSite-*xpi*'s ports. This mode causes the terminal connected to the other port on UniSite-*xpi* to act as if it were connected directly to the host computer. This mode is useful for establishing communication with the host (such as logging in and executing commands).

To enter and exit this mode, type **ESC**, **CTRL** + **T** from the terminal. Transparent mode does not support binary data transfers (this can be done via the upload and download commands using one of the binary data formats).

In transparent mode, all keystrokes entered on the terminal are passed directly to the host with one exception. The **ESC** character is stripped out since it represents a special meaning to UniSite-*xpi* (it is part of the Exit Transparent Mode command). To send an **ESC** character to the host, enter two consecutive **ESC** characters (the second one is passed to the host), or if **ESC** is followed by some character other than a **CTRL** + **T**, the escape and the character are sent to the host.

See Chapter 2 for information on configuring UniSite-*xpi* to operate in Transparent mode.

5. Computer Remote Control

The programmer can be controlled via a host computer using Computer Remote Control (CRC) protocol. CRC commands have been designed to be incorporated into a remote computer software program (driver) which will allow an operator to control the programmer. The driver generates commands and sends them to the programmer, which executes the commands. The programmer then returns a response character and, in some cases, data. The driver reacts to the response and uses it to generate messages and prompts for the user.

*Note: You do **not** need to use CRC if you are using PROMlink, TaskLink, or accessing the programmer's built-in menu system (using HiTerm or a similar product to communicate with the programmer). CRC commands offer you an alternative, allowing you to create your own custom interface with the programmer.*

This chapter is not intended to be a complete guide to using CRC commands. For a more detailed explanation of CRC commands, refer to the "UniSystem Computer Remote Control" Application Note available from Customer Support.

This chapter contains the following information:

- **System Setup**—Explains how to set up the programmer for remote control operation. Includes information on entering and exiting CRC mode.
- **CRC Summary**—Gives a listing of the available CRC commands.

Which Driver to Use?

If you are using CRC commands, you must use a driver program to send the CRC commands and receive the programmer's responses. You can either write your own software driver or use an already created driver (such as the **terminal.exe** program included with Windows).

System Setup

The programmer receives CRC commands and sends responses to the host computer through an RS-232C port using a 25-pin D connector in two possible configurations: either DTE or DCE. Only the Remote port supports CRC operation.

The pin designations for the Remote port are shown in Appendix B, “More About Cables.” Included in that appendix is a table of pin definitions, which explains the function of each pin for the two serial port configurations.

To ensure correct operation of the Remote port with the host computer, set the parameters for the Remote port according to the host computer requirements.

Entering CRC Mode

CRC mode can be entered in either of two ways: by the Remote Control menu, or automatically at powerup.

By Menu Commands

To enter CRC mode using the Remote Control menu, do the following:

1. Press **F1** to go to the Main Menu.
2. Type **M** to select the More Commands menu.
3. Press **R** to select Computer Remote Control from the More Commands menu.

The programmer is now in Remote Control mode.

On Power-up

The programmer enters either terminal or CRC mode during power-up based on the following combination of port connections and parameter settings.

Parameter Settings		**Port Connections**		
Power Up CRC	**User Menu Port**	**Terminal Connected**	**Remote Connected**	**Result**
Off	T (Terminal)	Yes	Yes	Terminal mode on Terminal port
Off	T (Terminal)	No	Yes	CRC mode on Remote port
Off	R (Remote)	X	Yes	Terminal mode on Remote port
On	X	X	Yes	CRC mode on Remote port
X	X	Yes	No	Terminal mode on Terminal port

Note: X = don’t care condition.

If you wish to have the programmer power up in CRC, perform the following steps:

1. Press **F1** to get to the Main Menu.
2. Type **M** to select the More Commands menu.
3. Press **C** to select the Configure System menu.
4. Press **E** to select Edit from the Configure Systems menu.
5. Press **I** to select Interface from the Edit menu. The programmer displays the interface parameters.
6. Move the cursor to the **Power on CRC** field and press **Y**. CRC is now selected. The following steps in this procedure save CRC mode as a powerup system parameter.
7. Press **F2** two times to return to the Configure System Parameters menu.
8. Press **S** to select Save from the Configure System Parameters menu. The screen displays the Save System Parameters menu.
9. Type **1** and press **ENTER** to select the Powerup Defaults file as the one where system parameters will be saved.
10. Press **ENTER** again so that the selection will be saved to the disk. The next time you power up the programmer, it will enter CRC mode automatically.

Which Interface Mode?

You can operate the programmer in one of two interface modes: Terminal and CRC. In Terminal mode you use screens and menus to interact with the programmer. In CRC mode you send single-line commands to the programmer, and the programmer responds with single line prompts, responses, and error codes.

Note: Terminal mode operations may be run from either the Terminal port or the Remote port. CRC mode operations must be run from the Remote port; CRC will not work on the Terminal port.

Depending on the equipment you have connected to the programmer, and on the settings of the User Menu Port and Power-on CRC parameters, you can select which mode is available on which port. (The User Menu Port parameter is found on the More Commands / Configure System / Edit / Communication Parameters screen.)

Factory defaults for the programmer are Terminal mode commands sent through the Terminal port and CRC mode commands sent through the Remote port. The factory default for power-up state is Terminal mode.

Exiting CRC Mode

Press **SHIFT** + **Z** + **ENTER** to exit CRC from an ASCII terminal on the Terminal port.

From a remote computer, send the **Z**, **ENTER** command. If you exit remote mode using the **Z, ENTER** command, the programmer's parameters are set to what they were **before** you entered remote mode.

Suspending CRC Mode

CRC Mode can be suspended temporarily to allow you to go into terminal mode to view or change parameter settings or the device data in memory. If the User Menu Port is set to T (Terminal port) press **SHIFT** +**Z** + **ENTER** to suspend CRC Mode. If you are controlling the programmer from the Remote port, send the **49]** command.

Halting CRC Operations

To halt any command or any ongoing CRC operation, use one of the following commands from the Remote port. Neither of the following two commands requires an **ENTER**. Both commands are immediate and both terminate any preceding command operation.

ASCII Command	**Hex Code**	**Description**
ESC	1B	Causes the programmer to unconditionally halt any operation except a binary transfer.
BREAK	n/a	Causes the programmer to unconditionally halt any operation in progress. This includes all data communications transfers. The data line must be held in the spacing condition for 110 ms to 700 ms.

CRC Default Settings

When CRC mode is entered, certain defaults are set prior to accepting any commands. The default settings are outlined below:

Description	Setting
Upload/download port	Remote port
Data source/destination	RAM
Security fuse data (0 or 1)	0
Program security fuse	No
Reject option (commercial or single)	Commercial
Algorithm Source	D (standard algorithms)
Logic verification option	All
Number of verify passes (0, 1, or 2)	2
Fill RAM before downloading	No
Illegal bit check option	No
Blank check option	No
Enable yield tally option	No
EE bulk erase option	No
Odd/even byte swap for 16 bit option	No
JEDEC I/O translate DIP/LCC option	Yes
Continuity check option	Yes
Compare electronic signature	Yes
Host command	Blank
I/O address offset	0
I/O format	MOS technology (format 81)
Instrument control code (0, 1, 2)	0
I/O timeout	30 seconds
Upload wait	0 seconds
Number of nulls	255
Serial set auto-increment mode	No
Programming mode	Single device
Total set size	1
Upload EOF delimiter flag	Disabled
Download EOF delimiter flag	Disabled

If you exit remote mode using the **Z**, **ENTER** command (or **SHIFT** +**Z** + **ENTER)**, UniSite-*xpi*'s parameters are set to what they were **before** you entered remote mode.

CRC Commands

CRC commands are a set of simplified commands for UniSite-*xpi*. The commands are designed to be received from a controlling computer. Because the commands are so simplified, they can be cryptic.

CRC Command Summary

You send CRC commands to UniSite-*xpi* by typing the command and then pressing the **ENTER** key. When UniSite-*xpi* receives a CRC command, the command is executed and a response is sent back, followed by a carriage return. If the response is an F, an error occurred. If the response is a `?`, UniSite-*xpi* did not understand the command. If the response is a `>`, the normal CRC prompt, the command executed properly. Some commands respond with both a value and the prompt. For example, UniSite-*xpi* might return `00284295>` when you send the Calculate Sumcheck command. In this case, the `00284295` is the sumcheck and the `>` indicates that the command executed properly. The **I**, **O**, and **C** commands perform any data transfer prior to sending the response.

Each command in the CRC command set is summarized in the following tables and then described in more detail on the pages that follow. The command tables are broken up into standard and extended CRC commands. Standard CRC commands are commonly used commands, such as load, program, and verify. Extended CRC commands are more specific device-related commands, such as Set Security Fuse, Fill Fuse Map, and Set Vector Test Options.

Note: While in CRC mode, UniSite-xpi recognizes only uppercase characters.

Except where noted, the commands use the following notation conventions:

- lowercase alphabetic characters indicate arguments that must be specified
- *h* represents a hexadecimal digit.
- *n* represents a decimal digit.
- *xxx...xxxx* represents a string of characters.

For example, ***nn*02]** indicates that you may precede the **02]** command with two decimal digits.

Summary of Standard CRC Commands

Command	Description	Response
&	Insert Parts Mode	None
–	Invert RAM	>
/	View device error status	XXYY>
hhhhhh:	Select device begin address	>
hhhhhh;	Select memory block size	> ; (CRC Command)
hhhhhh<	Select memory begin address	>
nn=	Select I/O timeout	>
*cff*A	Enter translation format	>
B	Blank check	>
C	Compare to port	>
D	Set odd parity	>
E	Set even parity	>
F	Error status inquiry	HHHHHHHH>
G	Configuration inquiry	RRDD>
H	No operation	>
I	Input from port	>
J	Set 1 stop bit	>
K	Set 2 stop bits	>
*nn*L	Load RAM from device	>
*hh*M	Enter record size	>
N	Set no parity	>
O	Output to port	>
*nn*P	Program device	>
Q	Swap nibbles	>
*nn*R	Return status of device	AAAAA/BB/C>
*nn*S	View sumcheck	HHHH>
*nn*T	Illegal-bit test	>
*hh*U	Set nulls	>
*nn*V	Verify device	>
*hhhhhhhh*W	Set I/O offset	>
X	Error code inquiry	HH....HH>
Y	Display parity errors	HHHH>
Z	Exit remote control	none
[	View device family/pinout code	FFFPPP>
\	Move memory block	>
hh^	Clear/fill RAM with data	>

Summary of Extended CRC Commands

Command	Description	Response
01]	Display system configuration	See Application Note
*nn*02]	Set upload wait time	>
*n*03]	Set device ID verify option	HHHHHHHH> or >
*nn*04]	Set Remote port baud rate	>
xxx...*xxxx*05]	Set host command	>
*n*06]	Select data bits	>
*n*07]	Set next set member	>
*n*08]	Select programming mode	>
*xx*09]	Set set size	>
*nn*22]	Set data word width	>
*n*23}	Select number of verify passes	>
*n*24]	Select security fuse option	>
*n*26]	Specify logic verify options	>
*n*27]	Set/clear enable/disable security fuse	>
*n*28]	Fill fuse map	>
*n*29]	Set reject count option	>
*hhh*2A] or *hh*2A]	Enable programming options	>
*hhh*2B] or *hh*2B]	Disable programming options	>
*nhh*2C]	Select memory fill option	>
*hh*2D]	Vector test options	
*nn*2F]	Return 8-character sumcheck	HHHHHHHH>
xxx...*xxxx*30]	Set data file name	>
*n*31]	Set data source/destination	>
xxx...*xxxx*33]	Select device manufacturer	>
xxx...*xxxx*34	Select device part number for device operations	>
xxx...*xxxx*38]	Load file from disk	>
39]	Delete all RAM files	>
xxx...*xxxx*3B]	Delete disk file	>
*n*3C]	Set data transfer port	>
xxx...*xxxx*3E]	Select Keep Current algorithm	>
40] or *n*40]	Upload device information	See Application Note
*n*41]	Run self-test and report results	AAA...AA>
43]	Upload yield tally	See Application Note
46]	Clear yield tally	>
49]	Suspend CRC mode	Displays terminal screen

Command	**Description**	**Response**
*n*4D]	Select Algorithm type	>
*n*52]	Select media for algorithms (floppy disk or MSM)	>
54]	Upload device footnote	See Application Note
55]	Upload device-specific message	See Application Note
56] or *nn*56]	Upload memory verify failure	ddPAAAAAAAAHHhh
57] or *nn*57]	Get checksum of operation	See Application Note
58]	Upload system ID	HHHH HHHH HHHH>
5A]	Display list of parameters	See Application Note
5B]	Clear vector data	>
5C]	Load system files for CM algorithm disk	>
5D]	Write system files to CM disk	>
5E]	Write algorithms to CM disk	>
*n*5F]	Select algorithm source drive for creating CM algorithms	>
60]	Get number of sectors	dd>
*n*61]	Get sector configuration settings	HHHH HHHH>
*nhhh...hhhh*62]	Set sector configuration settings	>
63]	Reboot the programmer	
*xxx...xxxx*64]	Select device part number for CM (use *xxx...xxxx*33] to select the manufacturer)	>
*n*67]	Set checksum calculation	>
A7]	Swap bytes	>
DC]	Device check	See Application Note
DF]	View status of sockets	HH HH...HH>
EB]	Input JEDEC data from host	>
EC]	Output JEDEC data to host	>
FC]	Restore CRC entry default parameters	>
FD]	Restore user-defined CRC parameters	>
FE]	Save user-defined CRC parameters	>

6. Translation Formats

Translation formats are different ways of encoding the data in a data file. A data file contains the information to be programmed into a device. The data file could contain the fuse pattern and test vectors for a logic device or the data for a memory device.

Generally, the data, such as the fuse pattern for a logic device, are created on a development platform and are then stored in a particular data translation format. When you want to transfer the data file to the programmer, you will need to set up the programmer to handle the correct translation format. During download, the programmer translates the formatted data and stores them in user memory as a binary image file.

Below you will find a list, in ascending numerical order, of all the translation formats supported by the programmer. Following the list is a description and, in most cases, an example of each translation format, presented in order by format number.

Format	**Code**	Format	**Code**
ASCII-BNPF	01 (05*)	RCA Cosmac	70
ASCII-BHLF	02 (06*)	Fairchild Fairbug	80
ASCII-B10F	03 (07*)	MOS Technology	81
Texas Instruments		Motorola EXORcisor	82
SDSMAC (320)	04	Intel Intellec 8 / MDS	83
5-level BNPF	08 (09*)	Signetic Absolute Object	85
Formatted Binary	10	Tektronix Hexadecimal	86
DEC Binary	11	Motorola EXORmacs	87
Spectrum	12 (13*)	Intel MCS-86 Hex Object	88
POF	14	Hewlett-Packard 64000	
Absolute Binary	16	Absolute	89
LOF	17	Texas Instruments	
ASCII-Octal Space	30 (35**)	SDSMAC	90
ASCII-Octal Percent	31 (36**)	JEDEC format (full)	91
ASCII-Octal		JEDEC format (Kernal	92
Apostrophe	32	Tektronix Hexadecimal	
ASCII-Octal SMS	37	Extended)	94
ASCII-Hex Space	50 (55**)	Motorola 32 bit (S3 record)	95
ASCII-Hex Percent	51 (56**)	Hewlett-Packard UNIX	
ASCII-Hex Apostrophe	52	Format	96
ASCII-Hex SMS	57	Intel OMF 386	97
ASCII-Hex Comma	53 (58**)	Intel OMF 286	98
		Intel Hex-32	99

* *This alternate code is used to transfer data without the STX start code and the ETX end code.*

** *This alternate code is used to transfer data using the SOH start code instead of the usual STX.*

Instrument Control Codes

The instrument control code is a 1-digit number that signals or controls data transfers. Specifically, the instrument control code can be used to implement a form of remote control that provides peripherals with flow control beyond that provided by software handshaking. When using computer remote control, the instrument control code is sent immediately preceding the 2-digit format code. The three values of the instrument control code and associated functions are described below.

0-Handshake Off

Input Function: Send X-OFF to stop the incoming transmission. Send X-ON to resume transmission.

Output Function: Data transmission will be halted upon receipt of an X-OFF character; transmission will resume upon receipt of an X-ON character.

1-Handshake On

Input Function: Transmit an X-ON character when ready to receive data; transmit X-OFF if the receiver buffer is full; transmit an X-ON if the receiver buffer is empty; transmit an X-OFF after all the data are received.

Output Function: Transmit a PUNCH-ON character prior to data transmission. Data transmission will be halted upon receipt of an X-OFF character and will resume upon receipt of an X-ON character. A PUNCH-OFF character is sent when the transmission is completed.

2-X-ON/X-OFF

Input Function: Send X-OFF to stop the incoming transmission. Send X-ON to resume transmission.

Output Function: Transmit data only after receiving an X-ON character. Data transmission will be halted upon receipt of an X-OFF character; transmission will resume upon receipt of an X-ON character.

Note: X-ON character is a CTRL-Q, or 11 hex.
X-OFF character is a CTRL-S, or 13 hex.
PUNCH-ON character is a CTRL-R, or 12 hex.
PUNCH-OFF character is a CTRL-T, or 14 hex.

General Notes

Some information about data translation is listed below:

Compatibility

When translating data, you may use any remote source that produces formats compatible with the descriptions listed in this section.

Formats with Limited Address Fields

Some formats are not defined for use with address fields greater than 64K. Thus, if you transfer a block greater than 64K, the address fields that would be greater than 64K may wrap around and overwrite data transferred in previous data records. Formats 70 through 86, and 90 may exhibit this characteristic.

Hardware Handshaking

Hardware handshaking may be used if compatible with the host interface by connecting the appropriate lines at the serial port interface.

Hardware handshake (CTS / DTR) is enabled as the default. However, if those signals aren't connected, the programming electronics sense this and communicate using software handshake (XON / XOFF). The programmer always uses software handshake regardless of whether hardware handshake is enabled.

Leader/Trailer

During output of all formats except 89 (HP 64000), a 50-character leader precedes the formatted data and a 50-character trailer follows. This leader / trailer consists of null characters. If the null count parameter is set to FF hex, then the leader / trailer is skipped. To set the null count, go to the More Commands / Configure / Edit / Communication Parameters screen and set the Number of Nulls parameter. If in CRC, use the CRC **U** command to set the null count.

Note: Formats 10, 11, and 89 do not function properly unless you select NO parity and 8-bit data.

ASCII Binary Format, Codes 01, 02, and 03(or 05, 06, and 07)

In these formats, bytes are recorded in ASCII codes with binary digits represented by Ns and Ps, Ls and Hs, or 1s and 0s, respectively. See Figure 6-1. The ASCII Binary formats do not have addresses.

Figure 6-1 shows sample data bytes coded in each of the three ASCII Binary formats. Incoming bytes are stored in RAM sequentially starting at the first RAM address. Bytes are sandwiched between B and F characters and are separated by spaces.

Figure 6-1
An Example of ASCII Binary Format

```
                                                 ②
                                                 ▼
FORMAT 01 (OR 05)  ①  BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF
                      BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF
                      BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF
                      BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF
                      BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF
                      BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF
                      BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF
                      BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF ③
                                                 ②
                                                 ▼
FORMAT 02 (OR 06)  ①  BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF
                      BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF
                      BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF
                      BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF
                      BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF
                      BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF
                      BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF
                      BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF ③
                                                 ②
                                                 ▼
FORMAT 03 (OR 07)  ①  B11111111F B11111111F B11111111F B11111111F
                      B11111111F B11111111F B11111111F B11111111F
                      B11111111F B11111111F B11111111F B11111111F
                      B11111111F B11111111F B11111111F B11111111F
                      B11111111F B11111111F B11111111F B11111111F
                      B11111111F B11111111F B11111111F B11111111F
                      B11111111F B11111111F B11111111F B11111111F
                      B11111111F B11111111F B11111111F B11111111F ③
```

LEGEND

① Start Code - nonprintable STX - CTRL B is the optional Start Code
② Characters such as spaces, carriage returns and line feeds may appear between bytes
③ End Code - nonprintable ETX - CTRL C

0074-2

Data can also be expressed in 4-bit words. The programmer generates the 4-bit format on upload if the data word width is 4 bits. Any other characters, such as carriage returns or line feeds, may be inserted between an F and the next B.

The start code is a nonprintable STX, which is a CTRL-B (the same as a hex 02). The end code is a nonprintable ETX, which is a CTRL-C (the same as a hex 03).

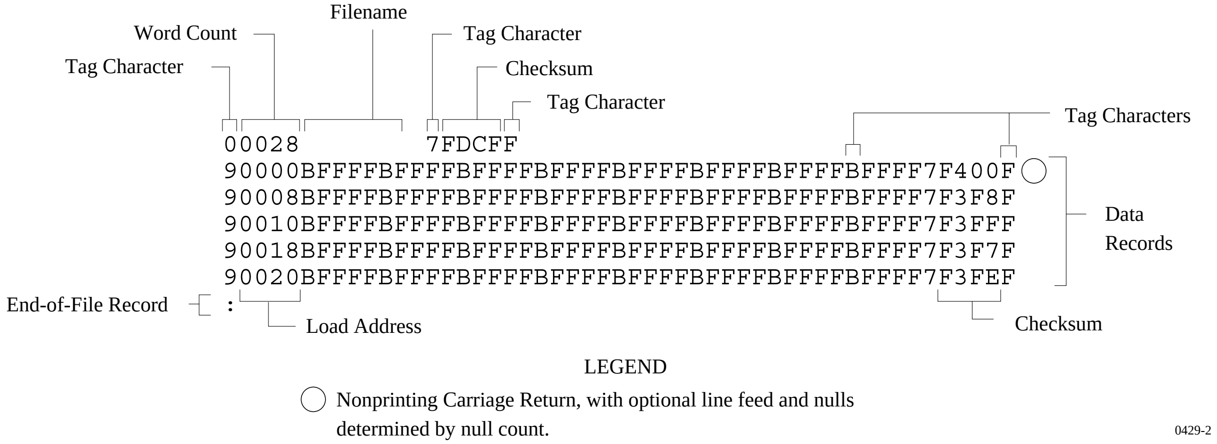
Note: Data without a start or end code may be input to or output from the programmer by use of alternate data translation format codes. These are ASCII-BNPF, 05; ASCII-BHLF, 06; ASCII-B10F, 07.

A single data byte can be aborted if the programmer receives an E character between B and F characters. Data will continue to be stored in sequential RAM addresses. Data are output in 4-byte lines with a space between bytes.

Texas Instruments SDSMAC Format (320), Code 04

Data files in the SDSMAC (320) format consist of a start-of-file record, data records, and an end-of-file record. See Figure 6-2. The format is used for Texas Instruments' 320 line of processors. It is very similar to format 90; the only difference is that the address fields represent 16-bit data words rather than bytes

Figure 6-2
An Example of TI SDSMAC Format



```
00028         7FDCFF
90000BFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFF7F400F
90008BFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFF7F3F8F
90010BFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFF7F3FFF
90018BFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFF7F3F7F
90020BFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFF7F3FEF
:
```

Each record is composed of a series of small fields, each initiated by a tag character. the programmer recognizes and acknowledges the following tag characters:

0 or K—followed by a file header.

7—followed by a checksum which the programmer acknowledges.

8—followed by a checksum which the programmer ignores.

9—followed by a load address which represents a word location.

B—followed by 4 data characters (16-bit word).

F—denotes the end of a data record.

*—followed by 2 data characters.

The start-of-file record begins with a tag character and a 12-character file header. The first four characters are the word count of the 16-bit data words; the remaining file header characters are the name of the file and may be any ASCII characters (in hex notation). Next come interspersed address fields and data fields (each with tag characters). The address fields represent 16-bit words. If any data fields appear before the first address field in the file, the first of those data fields is assigned to address 0000. Address fields may be expressed for any data word, but none are required.

The record ends with a checksum field initiated by the tag character 7 or 8, a 4-character checksum, and the tag character F. The checksum is the two's complement of the sum of the 8-bit ASCII values of the characters, beginning with the first tag character and ending with the checksum tag character (7 or 8).

Data records follow the same format as the start-of-file record but do not contain a file header. The end-of-file record consists of a colon (:) only. The output translator sends a CTRL-S after the colon.

During download or input from disk operations the destination address for the data is calculated in the following manner:

Memory address =
(load address x 2) – I/O address offset + begin address

During upload or output to disk operations the load address sent with each data record is calculated in the following manner:

Load address = I/O address offset / 2

The Memory begin address, I/O address offset, and User data size parameters represent bytes and must be even values for this format. The upload record size must also be even for this format (default is 16).

Note: If the data will be programmed into a 16-bit device to be used in a TMS320 processor-based system, the odd/even byte swap switch must be enabled.

The 5-Level BNPF Format, Codes 08 or 09

Except for the start and end codes, the same character set and specifications are used for the ASCII-BNPF and 5-level BNPF formats.

Data for input to the programmer are punched on 5-hole Telex paper tapes to be read by any paper tape reader that has an adjustable tape guide. The reader reads the tape as it would an 8-level tape, recording the 5 holes that are on the tape as 5 bits of data. The 3 most significant bits are recorded as if they were holes on an 8-level tape. Tape generated from a telex machine using this format can be input directly to a serial paper tape reader interfaced to the programmer. the programmer's software converts the resulting 8-bit codes into valid data for entry in RAM.

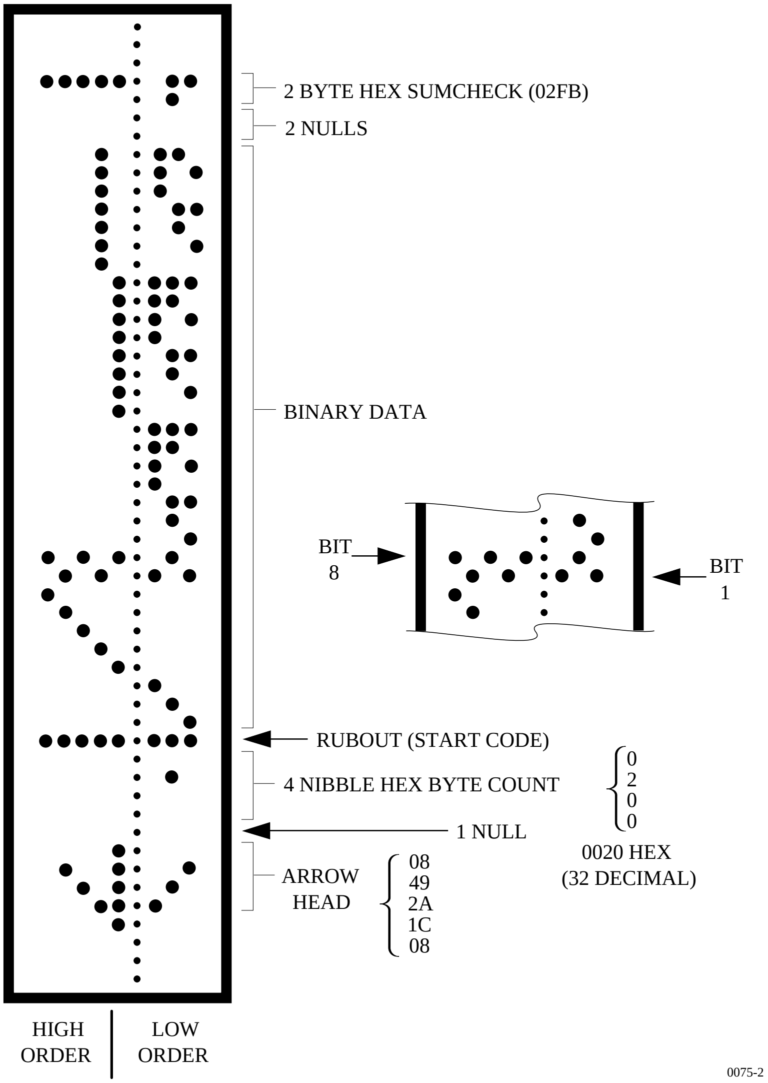
The start code for the format is a left parenthesis, (Figs K on a telex machine), and the end code is a right parenthesis, (Figs L on a telex machine). The 5-level BNPF format does not have addresses.

Note: Data without a start or end code may be input to or output from the programmer by use of the alternate data translation format code, 09. This format accepts an abort character (10 hex) to abort the transmission.

Formatted Binary Format, Code 10

Data transfer in the Formatted Binary format consists of a stream of 8-bit data bytes preceded by a byte count and followed by a sumcheck, as shown in Figure 6-3. The Formatted Binary format does not have addresses.

Figure 6-3
An Example of Formatted Binary Format

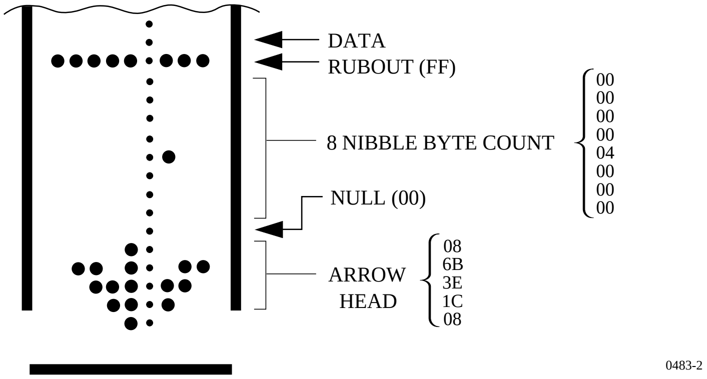


The programmer stores incoming binary data upon receipt of the start character. Data are stored in RAM starting at the first RAM address specified by the Memory Begin Address parameter and ending at the last incoming data byte.

A paper tape generated by a programmer contains a 5-byte, arrow-shaped header followed by a null and a 4-nibble byte count. The start code, an 8-bit rubout, follows the byte count. The end of data is signaled by two nulls and a 2-byte sumcheck of the data field. Refer to Figure 6-4.

If the data output has a byte count GREATER than or equal to 64K, an alternate arrow-shaped header is used. This alternate header (shown below) is followed by an 8-nibble byte count, sandwiched between a null and a rubout. The byte count shown here is 40000H (256K decimal). If the byte count is LESS than 64K, the regular arrowhead is used instead. Data that are input using Formatted Binary format will accept either version of this format.

Figure 6-4
An Example of Formatted Binary Format



In addition, a third variation of this binary format is accepted on download. This variation does not have an arrowhead and is accepted only on input. The rubout begins the format and is immediately followed by the data. There is no byte count or sumcheck.

DEC Binary Format, Code 11

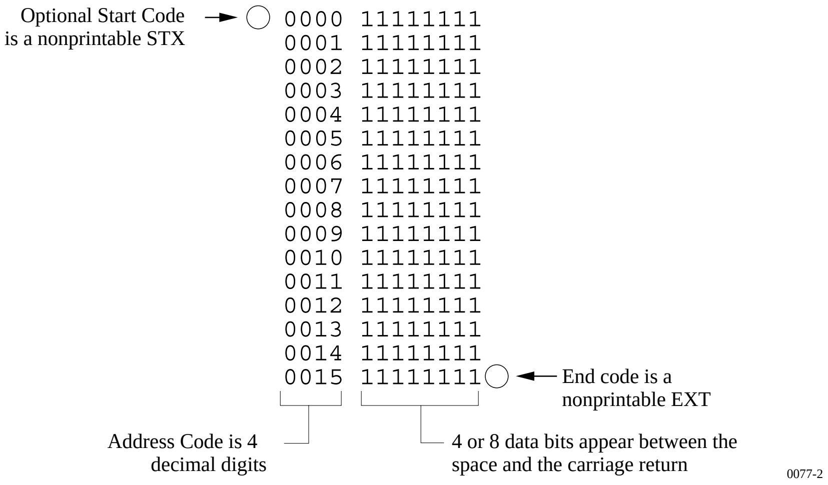
Data transmission in the DEC Binary format is a stream of 8-bit data words with no control characters except the start code. The start code is one null preceded by at least one rubout. The DEC Binary format does not have addresses.

Spectrum Format, Codes 12 or 13

In this format, bytes are recorded in ASCII codes with binary digits represented by 1s and 0s. During output, each byte is preceded by a decimal address.

Figure 6-5 shows sample data bytes coded in the Spectrum format. Bytes are sandwiched between the space and carriage return characters and are normally separated by line feeds. The start code is a nonprintable STX, CTRL-B (or hex 02), and the end code is a nonprintable ETX, CTRL-C (or hex 03).

Figure 6-5
An Example of Spectrum Format



```
0000 11111111
0001 11111111
0002 11111111
0003 11111111
0004 11111111
0005 11111111
0006 11111111
0007 11111111
0008 11111111
0009 11111111
0010 11111111
0011 11111111
0012 11111111
0013 11111111
0014 11111111
0015 11111111
```

Optional Start Code is a nonprintable STX
End code is a nonprintable EXT
Address Code is 4 decimal digits
4 or 8 data bits appear between the space and the carriage return
0077-2

Note: Data without a start or end code may be input to or output from the programmer by use of the alternate data translation format code, 13

POF (Programmer Object File) Format, Code 14

The POF (Programmer Object File) format provides a highly compact data format to enable translation of high bit count logic devices efficiently. This format currently applies to MAX™ devices, such as the Altera 5032.

The information contained in the file is grouped into "packets." Each packet contains a "tag," identifying what sort of data the package contains plus the data itself. This system of packeting information allows for future definitions as required.

The POF is composed of a header and a list of packets. The packets have variable lengths and structures, but the first six bytes of every packet always adhere to the following structure.

```
struct PACKET_HEAD
{
short tag;              /*tag number - type of packet */
long length;            /*number of bytes in rest of packet */
}
```

A POF is read by the program examining each packet and if the tag value is recognized, then the packet is used. If a tag value is not recognized, the packet is ignored.

Any packet except the terminator packet may appear multiple times within a POF. Packets do not need to occur in numerical tag sequence. The POF reader software is responsible for the interpretation and action taken as a result of any redundant data in the file, including the detection of error conditions.

The POF format currently uses the following packet types.

Note: In the following packet type descriptions, one of the terms—Used, Skipped, or Read—will appear after the tag and name.

Used: The information in this packet is used by the programmer.
Skipped: This information is not used by the programmer.
Read: This information is read by the programmer but has no direct application.

Creator_ID tag=1 Used

This packet contains a version ID string from the program which created the POF.

Device_Name tag=2 Used

This packet contains the ASCII name of the target device to be programmed, for example, PM9129.

Comment_Text tag=3 Read

This packet contains a text string which may consist of comments related to the POF. This text may be displayed to the operator when the file is read. The string may include multiple lines of text, separated by appropriate new line characters.

Tag_Reserved tag=4 Skipped

Security_Bit tag=5 Used

This packet declares whether security mode should be enabled on the target device.

Logical_Address_ and_Data_16 tag=6 Read

This packet defines a group of logical addresses in the target device and associates logical data with these addresses. The addresses comprise a linear region in the logical address space, bounded on the low end by the starting address and extending upward by the address count specified in the packet.

Electrical_Address _and_Data tag=7 Used

This packet defines a group of electrical addresses in the target device and associates data values with those addresses. The data field is ordered in column-row order, beginning with the data for the least column-row address, continuing with increasing row addresses until the first column is filled, then incrementing the column address, etc.

Terminator tag=8 Used

This packet signals the end of the packet list in the POF. This packet must be the Nth packet, where N is the packet count declared in the POF header. The CRC field is a 16-bit Cyclic Redundancy Check computed on all bytes in the file up to, but not including, the CRC value itself. If this CRC value is zero, the CRC check should be ignored.

Symbol table tag=9 Skipped

Test Vectors tag=10 Used

This packet allows the POF to contain test vectors for post programming testing purposes. Each vector is a character string and uses the 20 character codes for vector bits defined in JEDEC standard 3A, section 7.0.

Electrical_Address_and_ Constant_data tag=12 Skipped

Number of programmable elements tag=14 Read

This packet defines the number of programmable elements in the target device.

Logical_Address_and_ Data_32

tag=17 Read

This packet defines a group of logical addresses in the target device and associates logical data with these addresses. The addresses comprise a linear region in the logical address space, bounded on the low end by the starting address and extending upward by the address count specified in the packet.
The starting address and address count are each specified by 4-byte fields (32 bits).

Absolute Binary Format, Code 16

Absolute Binary format is a literal representation of the data to be transferred and no translation of the data takes place during the transfer. There are no overhead characters added to the data (i.e. no address record, start code, end code, nulls, or checksum). Every byte transferred represents the user’s data. This format can be used to download unformatted data such as an “.exe” file to the programmer.

Since this format does not have an end of file character, download transfers will terminate after no more data are received and an I/O timeout occurs. This is true for all data formats which don’t have an end of file indicator. For this reason do not use a value of 0 for the I/O timeout parameter on the communication parameters screen, since this will disable the timeout from occurring. A value between 1 and 99 (inclusive) should be used for the I/O timeout parameter when using formats which require the timeout to occur.

LOF Format, Code 17

The Link Object Format (LOF) is an extension of the standard JEDEC data translation format and is used to transfer fuse and test vector data between the programmer and a host computer. LOF is designed to support the Quicklogic QL8x12A family of FPGAs. An LOF data file is stored as an imploded ZIP file, which yields data compression approaching 95%.

Note: You must have PKZIP version 1.0 to use this format.

Note: The specification for the ZIP data compression algorithm allows for multiple data files to be compressed into one ZIP file. In addition, the ZIP data compression algorithm allows for multiple types of data compression.

The programmer's implementation of UNZIP supports only imploded data files and will extract only the first file in a ZIP file. All remaining files in the ZIP file will be ignored, as will all files not stored in the imploded format.

The LOF format contains both a subset and a superset of the JEDEC format described in this chapter. This section describes only the fields that are extensions of the JEDEC standard or that are unique to the LOF format. See the section explaining the JEDEC format for information on the standard JEDEC fields. See page 2-35 for information on obtaining a copy of the JEDEC Standard 3A.

LOF Field Syntax

The LOF character set consists of all the characters that are permitted with the JEDEC format: all printable ASCII characters and four control characters. The four allowable control characters are STX, ETX, CR (Return), and LF (line feed). Other control characters, such as Esc or Break, should not be used.

Note: This is Data I/O Corporation's implementation of Quicklogic's Link Object Format. Contact Quicklogic for a more in-depth explanation of the format and its syntax.

LOF Fields

The following fields are included in Data I/O's implementation of the LOF format:

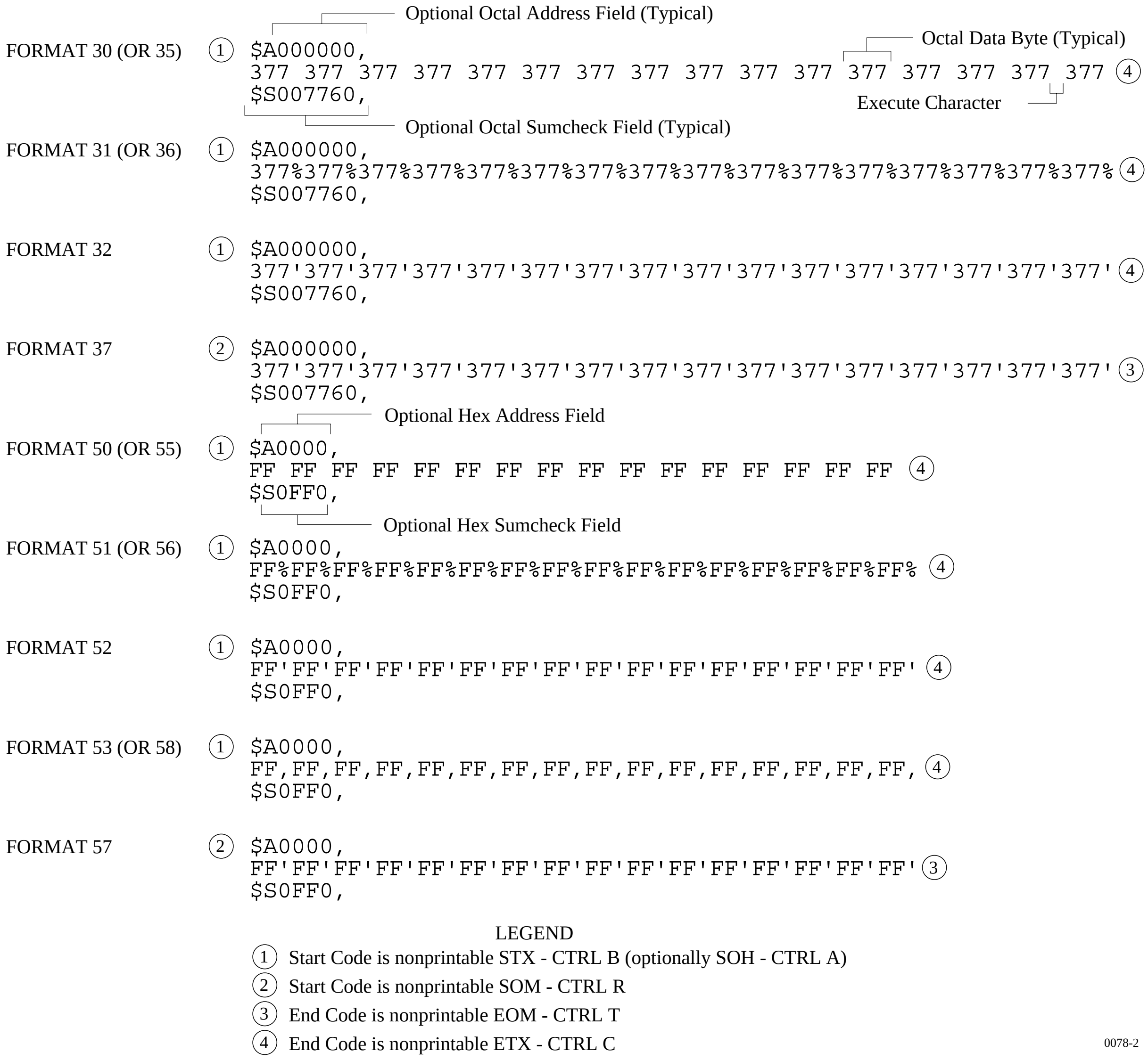
<STX>	*	Start of Data (ASCII Ctrl-B, 0x02 hex)
C	*	Fuse Checksum
K		Fuse data, followed by control words and pulse link cycles
N	*	Notes Field
QB		Number of bits per word
QC		Number of control words at the end of each K field
QF		Number of Fuses in Device (# of K fields)
QM		Number of macro cells in the data file
QP	*	Number of Device Package Pins
QS		Number of Hex-ASCII words in each K field and each control word
QV	*	Maximum Number of Test Vectors
R		Signature Analysis (reserved for future use)
S		SpDE Checksum
T		Signature Analysis (reserved for future use)
V	*	Test Vectors (reserved for future use)
X	*	Default Test Conditions (reserved for future use)
<ETX>	*	End of Data (ASCII Ctrl-C, 0x03 hex)

* *These fields are already defined as part of the JEDEC standard and will not be defined in this section.*

ASCII Octal and Hex Formats, Codes 30-37 and 50-58

Each of these formats has a start and end code, and similar address and checksum specifications. Figure 6-6 illustrates 4 data bytes coded in each of the 9 ASCII Octal and Hexadecimal formats. Data in these formats are organized into sequential bytes separated by the execute character (space, percent, apostrophe, or comma). Characters immediately preceding the execute character are interpreted as data. ASCII Octal and Hex formats can express 8-bit data, by 3 octal or 2 hexadecimal characters. Line feeds, carriage returns, and other characters may be included in the data stream as long as a data byte directly precedes each execute character.

Figure 6-6
An Example of ASCII Octal and Hex Formats


Optional Octal Address Field (Typical)
Octal Data Byte (Typical)
Execute Character
Optional Octal Sumcheck Field (Typical)
Optional Hex Address Field
Optional Hex Sumcheck Field

```
FORMAT 30 (OR 35)   ①  $A000000,
                       377 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377 ④
                       $S007760,

FORMAT 31 (OR 36)   ①  $A000000,
                       377%377%377%377%377%377%377%377%377%377%377%377%377%377%377%377% ④
                       $S007760,

FORMAT 32           ①  $A000000,
                       377'377'377'377'377'377'377'377'377'377'377'377'377'377'377'377' ④
                       $S007760,

FORMAT 37           ②  $A000000,
                       377'377'377'377'377'377'377'377'377'377'377'377'377'377'377'377' ③
                       $S007760,

FORMAT 50 (OR 55)   ①  $A0000,
                       FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ④
                       $S0FF0,

FORMAT 51 (OR 56)   ①  $A0000,
                       FF%FF%FF%FF%FF%FF%FF%FF%FF%FF%FF%FF%FF%FF%FF%FF% ④
                       $S0FF0,

FORMAT 52           ①  $A0000,
                       FF'FF'FF'FF'FF'FF'FF'FF'FF'FF'FF'FF'FF'FF'FF'FF' ④
                       $S0FF0,

FORMAT 53 (OR 58)   ①  $A0000,
                       FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, ④
                       $S0FF0,

FORMAT 57           ②  $A0000,
                       FF'FF'FF'FF'FF'FF'FF'FF'FF'FF'FF'FF'FF'FF'FF'FF' ③
                       $S0FF0,
```

LEGEND

① Start Code is nonprintable STX - CTRL B (optionally SOH - CTRL A)
② Start Code is nonprintable SOM - CTRL R
③ End Code is nonprintable EOM - CTRL T
④ End Code is nonprintable ETX - CTRL C

0078-2

Although each data byte has an address, most are implied. Data bytes are addressed sequentially unless an explicit address is included in the data stream. This address is preceded by a $ and an A, must contain 2 to 8 hex or 3 to 11 octal characters, and must be followed by a comma, except for the ASCII-Hex (Comma) format, which uses a period. The programmer skips to the new address to store the next data byte; succeeding bytes are again stored sequentially.

Each format has an end code, which terminates input operations. However, if a new start code follows within 16 characters of an end code, input will continue uninterrupted. If no characters come within 2 seconds, input operation is terminated.

After receiving the final end code following an input operation, the programmer calculates a sumcheck of all incoming data. Optionally, a sumcheck can also be entered in the input data stream. The programmer compares this sumcheck with its own calculated sumcheck. If they match, the programmer will display the sumcheck; if not, a sumcheck error will be displayed.

Note: The sumcheck field consists of either 2-4 hex or 3-6 octal characters, sandwiched between the $ and comma characters. The sumcheck immediately follows an end code. The sumcheck is optional in the input mode but is always included in the output mode. The most significant digit of the sumcheck may be 0 or 1 when expressing 16 bits as 6 octal characters.

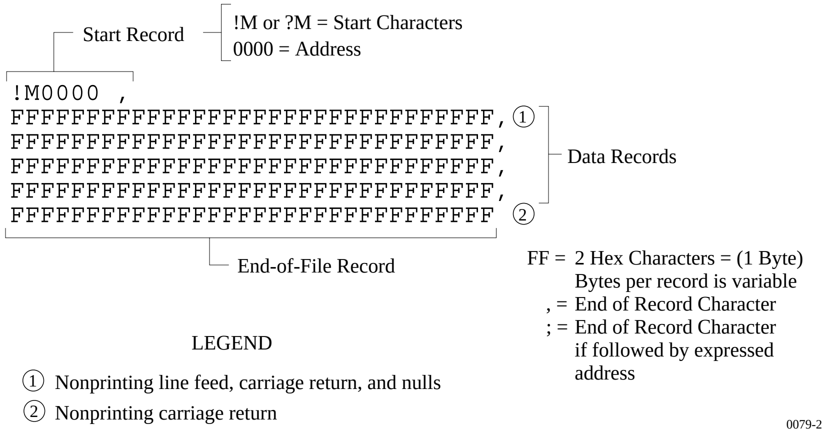
The programmer divides the output data into 8-line blocks. Data transmission is begun with the start code, a nonprintable STX character, or optionally, SOH.* Data blocks follow, each one prefaced by an address for the first data byte in the block. The end of transmission is signaled by the end code, a nonprintable ETX character. Directly following the end code is a sumcheck of the transferred data.

* *ASCII-Octal SMS and ASCII-Hex SMS use SOM (CTRL-R) as a start code and EOM (CTRL-T) as an end code.*

RCA Cosmac Format, Code 70

Data in this format begin with a start record consisting of the start character (!M or ?M), an address field, and a space. See Figure 6-7.

Figure 6-7
An Example of RCA Cosmac Format



```
!M0000 ,
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

Start Record
!M or ?M = Start Characters
0000 = Address
①
②
Data Records
End-of-File Record
FF = 2 Hex Characters = (1 Byte)
Bytes per record is variable
, = End of Record Character
; = End of Record Character
if followed by expressed
address
LEGEND
① Nonprinting line feed, carriage return, and nulls
② Nonprinting carriage return
0079-2

The start character ?M is sent to the programmer by a development system, followed by the starting address and a data stream which conforms to the data input format described in the ASCII-Hex and Octal figure. Transmission stops when the specified number of bytes has been transmitted.

Address specification is required for only the first data byte in the transfer. An address must have 1 to 4 hex characters and must be followed by a space. The programmer records the next hexadecimal character after the space as the start of the first data byte. (A carriage return must follow the space if the start code ?M is used.) Succeeding bytes are recorded sequentially.

Each data record is followed by a comma if the next record is not preceded by an address, or by a semicolon if it starts with an address. Records consist of data bytes expressed as 2 hexadecimal characters and followed by either a comma or semicolon, and a carriage return. The programmer ignores any characters received between a comma or semicolon and a carriage return.

The carriage return character is significant to this format because it can signal either the continuation or the end of data flow; if the carriage return is preceded by a comma or semicolon, more data must follow; the absence of a comma or semicolon before the carriage return indicates the end of transmission.

Output data records are followed by either a comma or a semicolon and a carriage return. The start-of-file records are expressed exactly as for input.

Fairchild Fairbug, Code 80

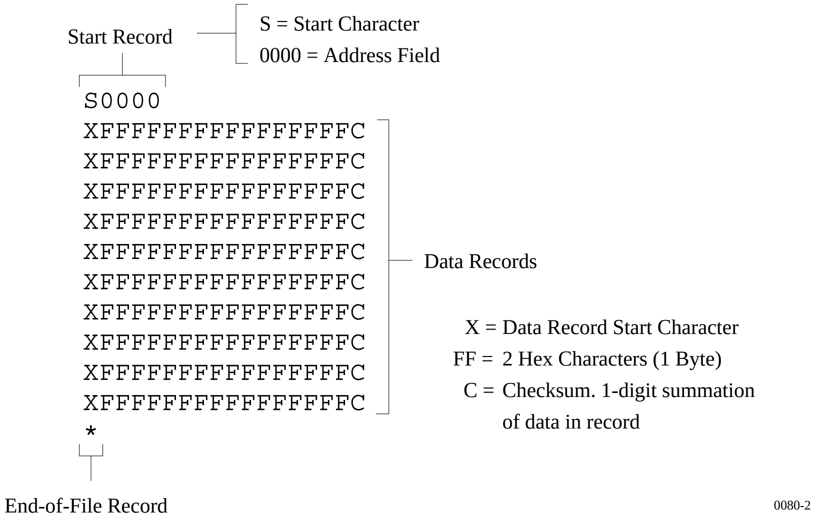
In the Fairbug format, input and output requirements are identical; both have 8-byte records and identical control characters. Figure 6-8 shows a Fairbug data file. A file begins with a 5-character prefix and ends with a 1-character suffix. The start-of-file character is an S, followed by the address of the first data byte. Each data byte is represented by 2 hexadecimal characters. The programmer will ignore all characters received prior to the first S.

Note: Address specification is optional in this format; a record with no address directly follows the previous record.

Each data record begins with an X and always contains 8 data bytes. A 1-digit hexadecimal checksum follows the data in each data record. The checksum represents, in hexadecimal notation, the sum of the binary equivalents of the 16 digits in the record; the half carry from the fourth bit is ignored.

The programmer ignores any character (except for address characters and the asterisk character, which terminates the data transfer) between a checksum and the start character of the next data record. This space can be used for comments.

Figure 6-8
An Example of Fairchild Fairbug



```
S0000
XFFFFFFFFFFFFFFFFC
XFFFFFFFFFFFFFFFFC
XFFFFFFFFFFFFFFFFC
XFFFFFFFFFFFFFFFFC
XFFFFFFFFFFFFFFFFC
XFFFFFFFFFFFFFFFFC
XFFFFFFFFFFFFFFFFC
XFFFFFFFFFFFFFFFFC
XFFFFFFFFFFFFFFFFC
XFFFFFFFFFFFFFFFFC
*
```

Start Record
S = Start Character
0000 = Address Field
Data Records
X = Data Record Start Character
FF = 2 Hex Characters (1 Byte)
C = Checksum. 1-digit summation of data in record
End-of-File Record
0080-2

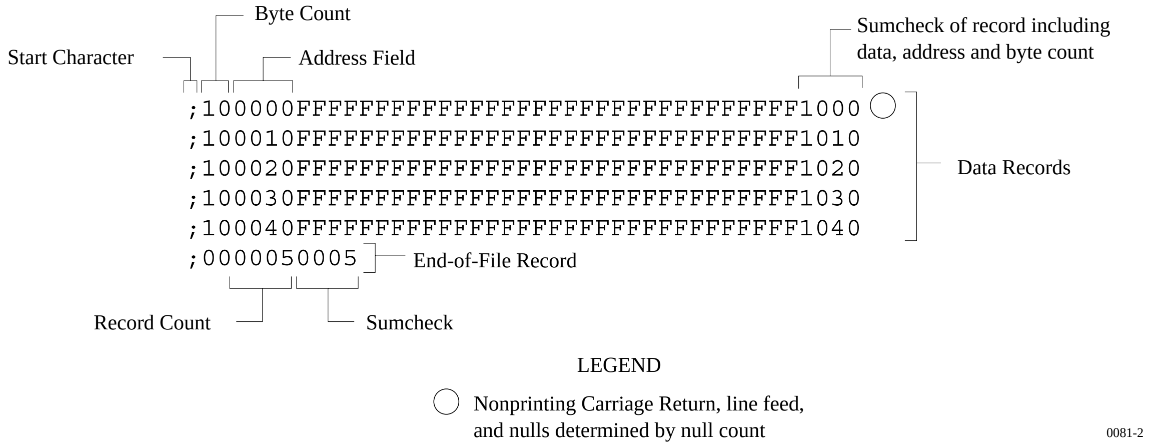
The last record consists of an asterisk only, which indicates the end of file.

MOS Technology Format, Code 81

The data in each record are sandwiched between a 7-character prefix and a 4-character suffix. The number of data bytes in each record must be indicated by the byte count in the prefix. The input file can be divided into records of various lengths.

Figure 6-9 shows a series of valid data records. Each data record begins with a semicolon. The programmer will ignore all characters received prior to the first semicolon. All other characters in a valid record must be valid hexadecimal digits (0-9 and A-F). A 2-digit byte count follows the start character. The byte count, expressed in hexadecimal digits, must equal the number of data bytes in the record. The byte count is greater than zero in the data records, and equals zero (00) in the end-of-file record. The next 4 digits make up the address of the first data byte in the record. Data bytes follow, each represented by 2 hexadecimal digits. The end-of-file record consists of the semicolon start character, followed by a 00 byte count, the record count, and a checksum.

Figure 6-9
An Example of MOS Technology Format



```
;100000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1000
;100010FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1010
;100020FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1020
;100030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1030
;100040FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1040
;0000050005
```

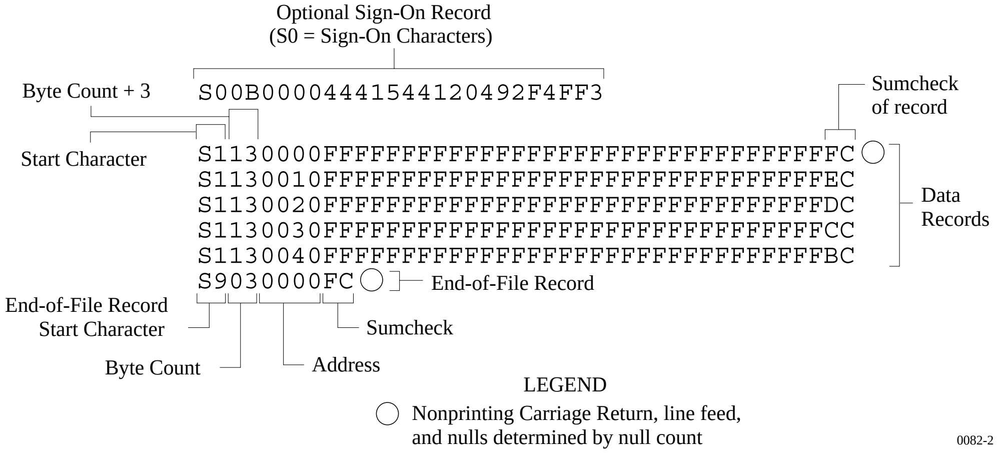
Byte Count
Start Character
Address Field
Sumcheck of record including data, address and byte count
Data Records
End-of-File Record
Record Count
Sumcheck
LEGEND
○ Nonprinting Carriage Return, line feed, and nulls determined by null count
0081-2

The checksum, which follows each data record, is a 2-byte binary summation of the preceding bytes in the record (including the address and byte count), in hexadecimal notation.

Motorola EXORciser Format, Code 82

Motorola EXORciser data files may begin with an optional sign-on record, which is initiated by the start characters S0. Valid data records start with an 8-character prefix and end with a 2-character suffix. Figure 6-10 shows a series of valid Motorola data records.

Figure 6-10
An Example of Motorola EXORciser Format



```
S00B00004441544120492F4FF3

S1130000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC
S1130010FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEC
S1130020FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDC
S1130030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFCC
S1130040FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBC
S9030000FC
```

Each data record begins with the start characters S1. The third and fourth characters represent the byte count, which expresses the number of data, address, and checksum bytes in the record. The address of the first data byte in the record is expressed by the last 4 characters of the prefix. Data bytes follow, each represented by 2 hexadecimal characters. The number of data bytes occurring must be three less than the byte count. The suffix is a 2-character checksum, which equals the one's complement of the binary summation of the byte count, address, and data bytes.

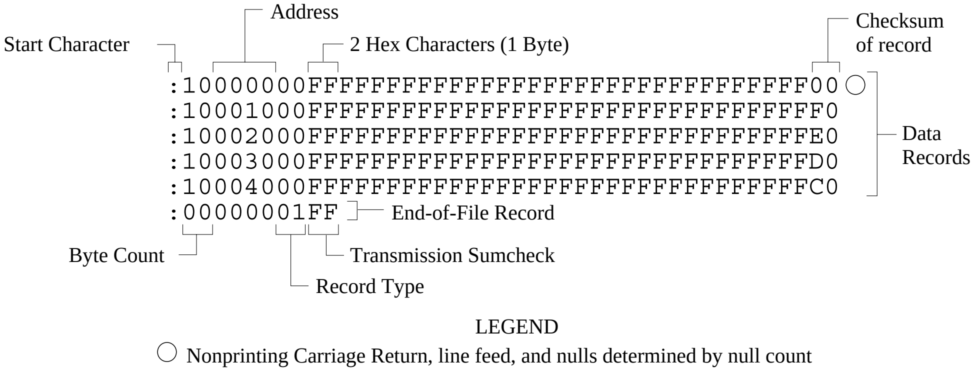
The end-of-file record consists of the start characters S9, the byte count, the address (in hex), and a checksum. The maximum record length is 250 data bytes.

Intel Intellec 8/MDS Format, Code 83

Intel data records begin with a 9-character prefix and end with a 2-character suffix. The byte count must equal the number of data bytes in the record.

Figure 6-11 simulates a series of valid data records. Each record begins with a colon, which is followed by a 2-character byte count. The 4 digits following the byte count give the address of the first data byte. Each data byte is represented by 2 hexadecimal digits; the number of data bytes in each record must equal the byte count. Following the data bytes of each record is the checksum, the two's complement (in binary) of the preceding bytes (including the byte count, address, record type, and data bytes), expressed in hex.

Figure 6-11
An Example of Intel Intellec 8/MDS Format



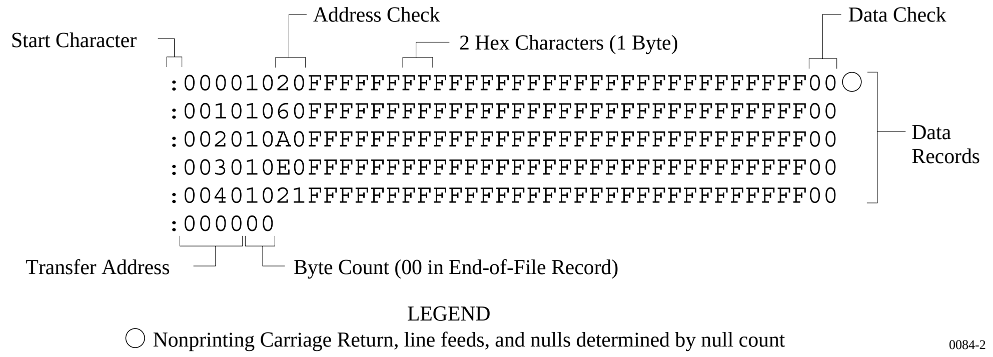
```
:10000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
:10001000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0
:10002000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
:10003000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD0
:10004000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0
:00000001FF
```

The end-of-file record consists of the colon start character, the byte count (equal to 00), the address, the record type (equal to 01), and the checksum of the record.

Signetics Absolute Object Format, Code 85

Figure 6-12 shows the specifications of Signetics format files. The data in each record are sandwiched between a 9-character prefix and a 2-character suffix.

Figure 6-12
An Example of Signetics Absolute Object Format



```
:00001020FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00○
:00101060FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
:002010A0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
:003010E0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
:00401021FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
:000000
```

Address Check
Data Check
Start Character
2 Hex Characters (1 Byte)
Data Records
Transfer Address
Byte Count (00 in End-of-File Record)
LEGEND
○ Nonprinting Carriage Return, line feeds, and nulls determined by null count
0084-2

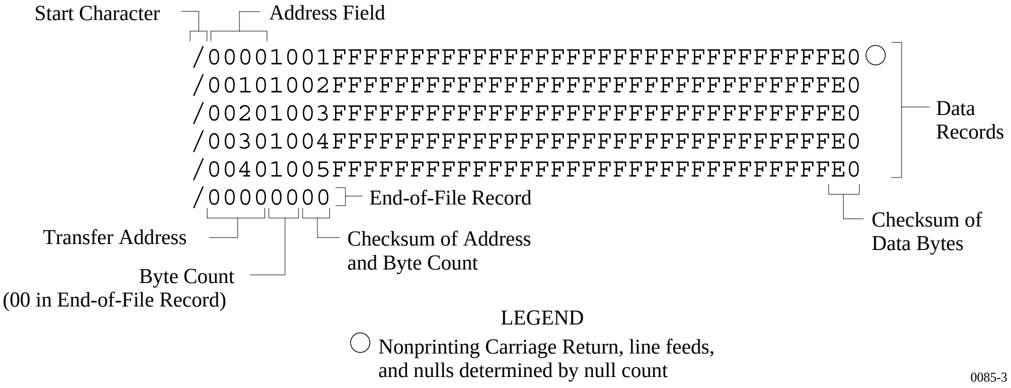
The start character is a colon. This is followed by the address, the byte count, and a 2-digit address check. The address check is calculated by exclusive ORing every byte with the previous one, then rotating left one bit. Data is represented by pairs of hexadecimal characters. The byte count must equal the number of data bytes in the record. The suffix is a 2-character data check, calculated using the same operations described for the address check.

The end-of-file record consists of the colon start character, the address, and the byte count (equal to 00).

Tektronix Hexadecimal Format, Code 86

Figure 6-13 illustrates a valid Tektronix data file. The data in each record are sandwiched between the start character (a slash) and a 2-character checksum. Following the start character, the next 4 characters of the prefix express the address of the first data byte. The address is followed by a byte count, which represents the number of data bytes in the record, and by a checksum of the address and byte count. Data bytes follow, represented by pairs of hexadecimal characters. Succeeding the data bytes is their checksum, an 8-bit sum, modulo 256, of the 4-bit hexadecimal values of the digits making up the data bytes. All records are followed by a carriage return.

Figure 6-13
An Example of Tektronix Hex Format



```
/00001001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
/00101002FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
/00201003FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
/00301004FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
/00401005FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
/00000000
```

Start Character
Address Field
Data Records
End-of-File Record
Checksum of Data Bytes
Transfer Address
Checksum of Address and Byte Count
Byte Count (00 in End-of-File Record)
LEGEND
○ Nonprinting Carriage Return, line feeds, and nulls determined by null count
0085-3

Data are output from the programmer starting at the first RAM address and continuing until the number of bytes in the specified block has been transmitted. The programmer divides output data into records prefaced by a start character and an address field for the first byte in the record.

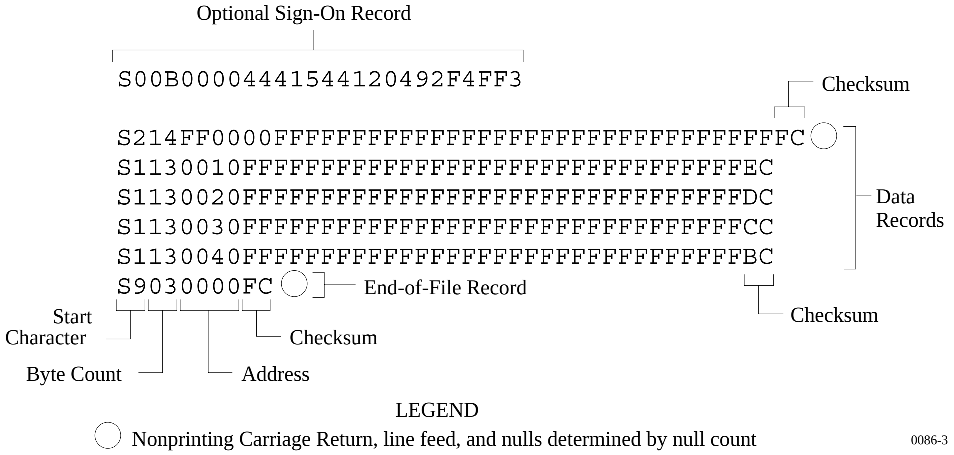
The end-of-file record consists of a start character (slash), followed by the transfer address, the byte count (equal to 00), and the checksum of the transfer address and byte count.

An optional abort record contains 2 start characters (slashes), followed by an arbitrary string of ASCII characters. Any characters between a carriage return and a / are ignored.

Motorola EXORmacs Format, Code 87

Motorola data files may begin with an optional sign-on record, initiated by the start characters S0. Data records start with an 8- or 10-character prefix and end with a 2-character suffix. Figure 6-14 shows a series of Motorola EXORmacs data records.

Figure 6-14
An Example of Motorola EXORmacs Format



```
S00B00004441544120492F4FF3

S214FF0000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC
S1130010FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEC
S1130020FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDC
S1130030FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFCC
S1130040FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBC
S9030000FC
```

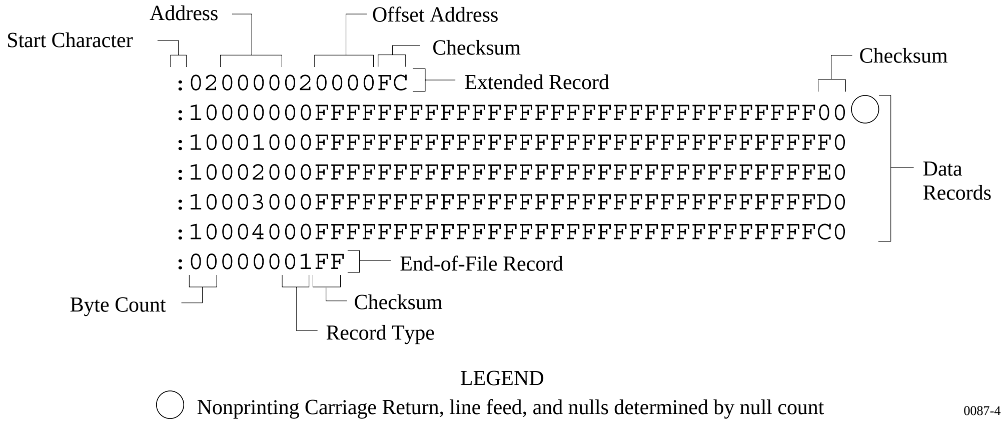
Each data record begins with the start characters S1 or S2: S1 if the following address field has 4 characters, S2 if it has 6 characters. The third and fourth characters represent the byte count, which expresses the number of data, address, and checksum bytes in the record. The address of the first data byte in the record is expressed by the last 4 characters of the prefix (6 characters for addresses above hexadecimal FFFF). Data bytes follow, each represented by 2 hexadecimal characters. The number of data bytes occurring must be 3 or 4 less than the byte count. The suffix is a 2-character checksum, the one's complement (in binary) of the preceding bytes in the record, including the byte count, address, and data bytes.

The end-of-file record begins with an S9 start character. Following the start characters are the byte count, the address, and a checksum. The maximum record length is 250 data bytes.

Intel MCS-86 Hexadecimal Object, Code 88

The Intel 16-bit Hexadecimal Object file record format has a 9-character (4-field) prefix that defines the start of record, byte count, load address, and record type and a 2-character checksum suffix. Figure 6-15 shows a sample record of this format.

Figure 6-15
An Example of Intel MCS-86 Hex Object



```
:020000020000FC
:10000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
:10001000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0
:10002000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
:10003000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD0
:10004000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0
:00000001FF
```

The four record types are described below.

00-Data Record

This begins with the colon start character, which is followed by the byte count (in hex notation), the address of the first data byte, and the record type (equal to 00). Following these are the data bytes. The checksum follows the data bytes and is the two's complement (in binary) of the preceding bytes in the record, including the byte count, address, record type, and data bytes.

01-End Record

This end-of-file record also begins with the colon start character. This is followed by the byte count (equal to 00), the address (equal to 0000), the record type (equal to 01), and the checksum, FF.

02-Extended Segment Address Record

This is added to the offset to determine the absolute destination address. The address field for this record must contain ASCII zeros (Hex 30s). This record type defines bits 4 to 19 of the segment base address. It can appear randomly anywhere within the object file and affects the absolute memory address of subsequent data records in the file. The following example illustrates how the extended segment address is used to determine a byte address.

Problem:

Find the address for the first data byte for the following file.

```
:02 0000 02 1230 BA
:10 0045 00 55AA FF.....BC
```

Solution:

Step 1. Find the record address for the byte. The first data byte is 55. Its record address is 0045 from above.

Step 2. Find the offset address. The offset address is 1230 from above.

Step 3. Shift the offset address one place left, then add it to the record address, like this:

```
   1230       Offset address (upper 16 bits)
+     0045    Record address (lower 16 bits)
   12345      20-bit address
```

The address for the first data byte is 12345.

Note: Always specify the address offset when using this format, even when the offset is zero.

During output translation, the firmware will force the record size to 16 (decimal) if the record size is specified greater than 16. There is no such limitation for record sizes specified less than 16.

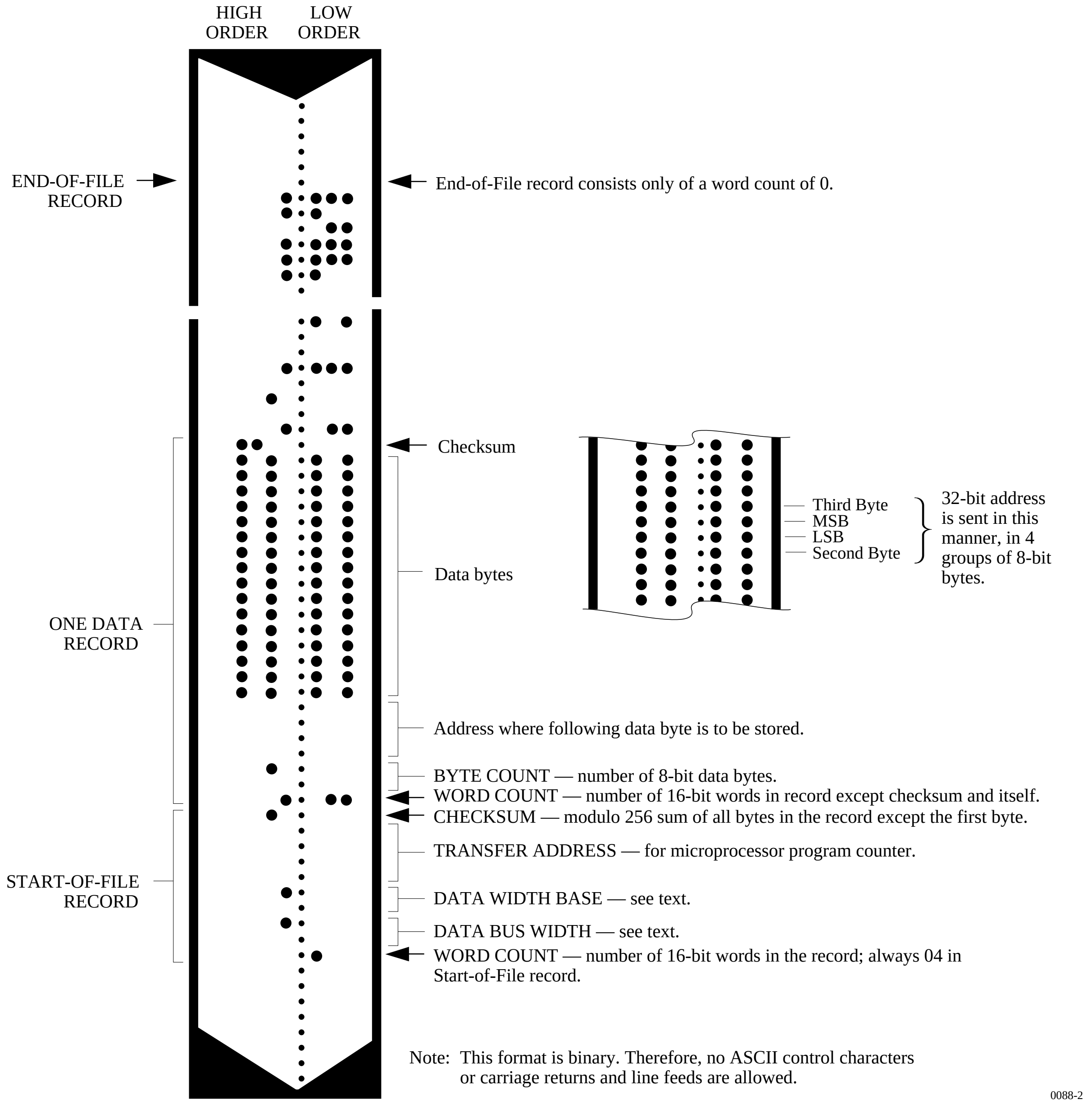
03-Start Record

This record type is not sent during output by Data I/O translator firmware.

Hewlett-Packard 64000 Absolute Format, Code 89

Hewlett-Packard Absolute is a binary format with control and data-checking characters. See Figure 6-16.

Figure 6-16
An Example of HP 64000 Absolute Format


HIGH ORDER
LOW ORDER
END-OF-FILE RECORD
End-of-File record consists only of a word count of 0.
Checksum
Data bytes
ONE DATA RECORD
Third Byte
MSB
LSB
Second Byte
32-bit address is sent in this manner, in 4 groups of 8-bit bytes.
Address where following data byte is to be stored.
BYTE COUNT — number of 8-bit data bytes.
WORD COUNT — number of 16-bit words in record except checksum and itself.
CHECKSUM — modulo 256 sum of all bytes in the record except the first byte.
TRANSFER ADDRESS — for microprocessor program counter.
START-OF-FILE RECORD
DATA WIDTH BASE — see text.
DATA BUS WIDTH — see text.
WORD COUNT — number of 16-bit words in the record; always 04 in Start-of-File record.
0088-2

Note: This format is binary. Therefore, no ASCII control characters or carriage returns and line feeds are allowed.

Data files begin with a Start-of-file record, which includes the Data Bus Width, Data Width Base, Transfer Address, and a checksum of the bytes in the record.

The Data Bus Width represents the width of the target system's bus (in bits). The Data Width Base represents the smallest addressable entity used by the target microprocessor.

The Data Bus Width and Data Width Base are not used by the programmer during download. During upload, the Data Bus Width will be set to the current Data Word Width, and the Data Width Base will be set to 8. The Transfer Address is not used by the programmer.

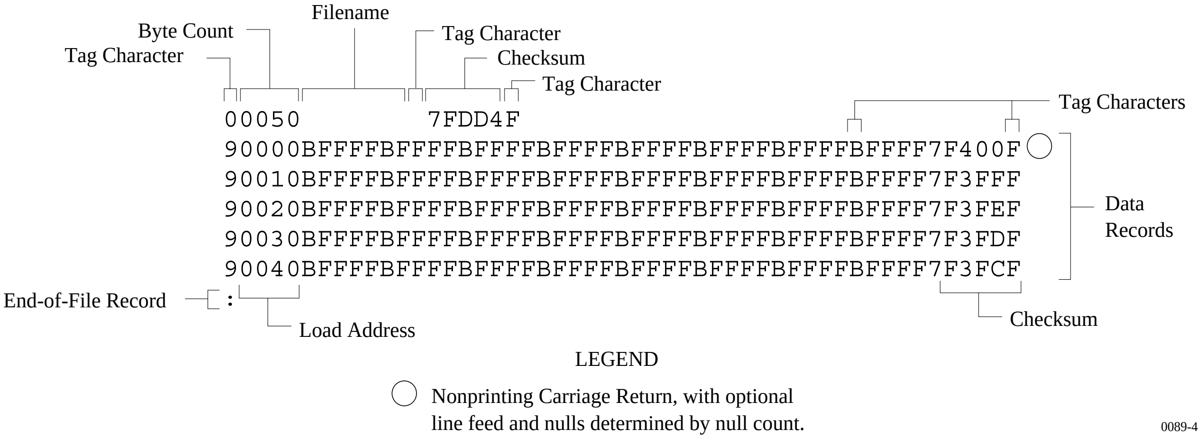
Data records follow the Start-of-file record. Each begins with 2 byte counts: the first expresses the number of 16-bit bytes in the record, not including the checksum and itself; the second expresses the number of 8-bit data bytes in the record. Next comes a 32-bit address, which specifies the storage location of the following data byte. Data bytes follow; after the last data byte is a checksum of every byte in the record except the first byte, which is the word count.

The End-of-file record consists of a one byte word count, which is always zero. Leader and trailer nulls, normally 50 each, are suppressed in this translation format.

Texas Instruments SDSMAC Format, Code 90

Data files in the SDSMAC format consist of a start-of-file record, data records, and an end-of-file record. See Figure 6-17.

Figure 6-17
An Example of TI SDSMAC Format



```
00050        7FDD4F
90000BFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFF7F400F
90010BFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFF7F3FFF
90020BFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFF7F3FEF
90030BFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFF7F3FDF
90040BFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFFBFFFF7F3FCF
:
```

Each record is composed of a series of small fields, each initiated by a tag character. The programmer recognizes and acknowledges the following tag characters:

0 or K—followed by a file header.

7—followed by a checksum which the programmer acknowledges.

8—followed by a checksum which the programmer ignores.

9—followed by a load address.

B—followed by 4 data characters.

F—denotes the end of a data record.

*—followed by 2 data characters.

The start-of-file record begins with a tag character and a 12-character file header. The first four characters are the byte count of the data bytes; the remaining file header characters are the name of the file and may be any ASCII characters (in hex notation). Next come interspersed address fields and data fields (each with tag characters). If any data fields appear before the first address field in the file, the first of those data fields is assigned to address 0000. Address fields may be expressed for any data byte, but none are required.

The record ends with a checksum field initiated by the tag character 7 or 8, a 4-character checksum, and the tag character F. The checksum is the two's complement of the sum of the 8-bit ASCII values of the characters, beginning with the first tag character and ending with the checksum tag character (7 or 8).

Data records follow the same format as the start-of-file record but do not contain a file header. The end-of-file record consists of a colon (:) only. The output translator sends a CTRL-S after the colon.

JEDEC Format, Codes 91 and 92

Introduction

The JEDEC (Joint Electron Device Engineering Council) format is used to transfer fuse and test vector data between the programmer and a host computer. Code 91 is full format and includes all the data fields (such as note and test fields) described on the following pages. Code 92 is the Kernel, or shorter, format. The JEDEC Kernel format includes only the minimum information needed for the programming; it does not, for example, include information fields or test vector fields. Prior to transferring a JEDEC file, the appropriate Logic device must be selected.

JEDEC's legal character set consists of all the printable ASCII characters and four control characters. The four allowable control characters are STX, ETX, CR (RETURN), and LF (line feed). Other control characters, such as ESC or BREAK, should not be used.

Note: This is Data I/O Corporation's implementation of JEDEC Standard 3A. For a copy of the strict standard, write to:

Electronic Industries Association
Engineering Department
2001 Eye Street NW
Washington, D.C. 20006

BNF Rules and Standard Definitions

The Backus-Naur Form (BNF) is used in the description here to define the syntax of the JEDEC format. BNF is a shorthand notation that follows these rules:

:: = denotes "is defined as."

Characters enclosed by single quotes are literals (required).

Angle brackets enclose identifiers.

Square brackets enclose optional items.

Braces {} enclose a repeated item. The item may appear zero or more times.

Vertical bars indicate a choice between items.

Repeat counts are given by a :n suffix. For example, a 6-digit number would be defined as:

<number> :: = <digit>:6

For example, in words the definition of a person’s name reads:

The full name consists of an optional title followed by a first name, a middle name, and a last name. The person may not have a middle name, or may have several middle names. The titles consist of: Mr., Mrs., Ms., Miss, and Dr.

The BNF definition for a person’s name is:

<full name> :: = [<title>] <f. name> {<m.name>} <l. name>

<title> :: = ‘Mr.’ | ‘Mrs.’ | ‘Ms.’ | ‘Miss’ | ‘Dr.’

The following standard definitions are used throughout the rest of this document:

<digit> :: = ‘0’ | ‘1’ | ‘2’ | ‘3’ | ‘4’ | ‘5’ | ‘6’ | ‘7’ | ‘8’ | ‘9’

<hex-digit> :: = <digit> | ‘A’ | ‘B’ | ‘C’ | ‘D’ | ‘E’ | ‘F’

<binary-digit> :: = ‘0’ | ‘1’

<number> :: = <digit> {<digit>}

<del> :: = <space> | <carriage return>

<delimiter> :: = <del> {<del>}

<printable character> :: = <ASCII 20 hex... 7E hex>

<control character> :: = <ASCII 00 hex... 1F hex> | <ASCII 7F hex>

<STX> :: = <ASCII 02 hex>

<ETX> :: = <ASCII 03 hex>

<carriage return> :: = <ASCII 0D hex>

<line feed> :: = <ASCII 0A hex>

<space> :: = <ASCII 20 hex> | “

<valid character> :: = <printable character> | <carriage return> | <line feed>

<field character> :: = <ASCII 20 hex... 29 hex> | <ASCII 2B hex... 7E hex> | <carriage return> | <line feed>

The Design Specification Field

<design spec> ::= {<field character>}'*'

The first field sent in a JEDEC transmission is the design specification. Both the full and kernel JEDEC formats accept the design specification field. This field is mandatory and does not have an identifier (such as an asterisk) signaling its beginning. The design specification field consists of general device information. It could, for example, consist of the following information: your name, your company's name, the date, the device name and manufacturer, design revision level, etc. This field is terminated by an asterisk character. Examine the sample transmission shown on the next page of this description—the first three lines of the file comprise the design specification field. The programmer ignores the contents of this field for downloads and places "Data I/O" in this field for upload operations.

Note: You do not need to send any information in this field if you do not wish to; a blank field, consisting of the terminating asterisk, is a valid design specification field.

The Transmission Checksum Field

<xmit checksum> ::= <hex digit>:4

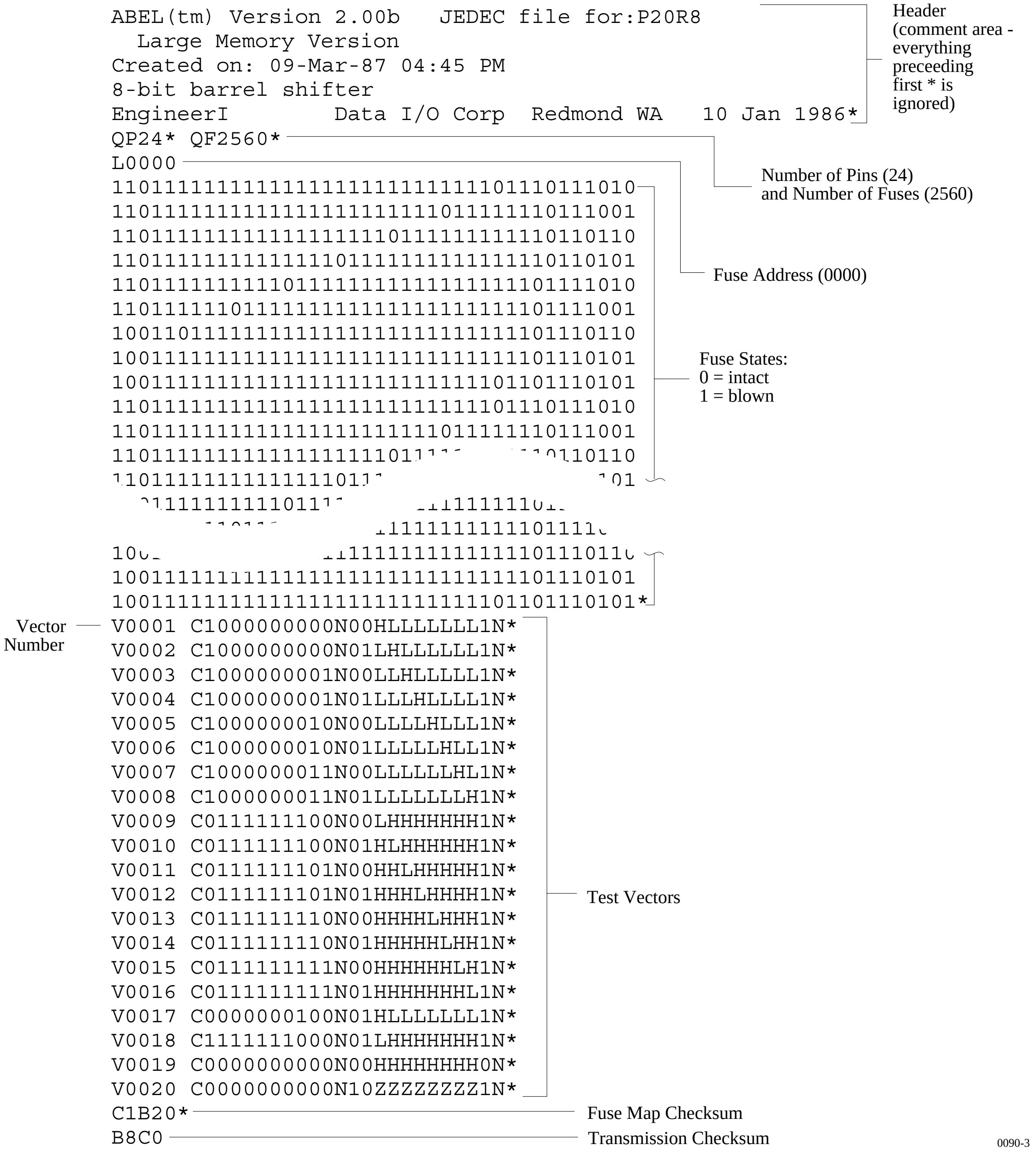
The transmission checksum is the last value sent in a JEDEC transmission. The full JEDEC format requires the transmission checksum. The checksum is a 16-bit value, sent as a 4-digit hex number, and is the sum of all the ASCII characters transmitted between (and including) the STX and ETX. The parity bit is excluded in the calculation of the transmission checksum.

Some computer systems do not allow you to control what characters are sent, especially at the end of a line. You should set up the equipment so that it will accept a dummy value of 0000 as a valid checksum. This zero checksum is a way of disabling the transmission checksum while still keeping within the JEDEC format rules.

JEDEC Full Format, Code 91

The full JEDEC format consists of a start-of-text character (STX), various fields, an end-of-text character (ETX), and a transmission checksum. A sample JEDEC transmission sent in the full format is shown in Figure 6-18. Each of the fields is described on the following pages.

Figure 6-18
An Example of JEDEC Full Format



```
ABEL(tm) Version 2.00b   JEDEC file for:P20R8
  Large Memory Version
Created on: 09-Mar-87 04:45 PM
8-bit barrel shifter
EngineerI        Data I/O Corp  Redmond WA   10 Jan 1986*
QP24* QF2560*
L0000
1101111111111111111111111111110111010111010
1101111111111111111111111011111110111001
1101111111111111111110111111111110110110
1101111111111111110111111111111110110101
1101111111110111111111111111111101111010
1101111101111111111111111111111101111001
1001101111111111111111111111111101110110
1001111111111111111111111111111101110101
1001111111111111111111111111101101110101
1101111111111111111111111111110111010111010
1101111111111111111111111011111110111001
110111111111111111111011111      1110110
110111111111111110111                 101
  011111111110111         11111111101
       110111         11111111111101111
100           1111111111111111101110110
1001111111111111111111111111111101110101
1001111111111111111111111111101101110101*
V0001 C1000000000N00HLLLLLLL1N*
V0002 C1000000000N01LHLLLLLL1N*
V0003 C1000000001N00LLHLLLLL1N*
V0004 C1000000001N01LLLHLLLL1N*
V0005 C1000000010N00LLLLHLLL1N*
V0006 C1000000010N01LLLLLHLL1N*
V0007 C1000000011N00LLLLLLHL1N*
V0008 C1000000011N01LLLLLLLH1N*
V0009 C0111111100N00LHHHHHHH1N*
V0010 C0111111100N01HLHHHHHH1N*
V0011 C0111111101N00HHLHHHHH1N*
V0012 C0111111101N01HHHLHHHH1N*
V0013 C0111111110N00HHHHLHHH1N*
V0014 C0111111110N01HHHHHLHH1N*
V0015 C0111111111N00HHHHHHLH1N*
V0016 C0111111111N01HHHHHHHL1N*
V0017 C0000000100N01HLLLLLLL1N*
V0018 C1111111000N01LHHHHHHH1N*
V0019 C0000000000N00HHHHHHHH0N*
V0020 C0000000000N10ZZZZZZZZ1N*
C1B20*
B8C0
```

JEDEC Field Syntax

<field> ::= [<delimiter>]<field identifier>{<field character>}’*’

<field identifier>::= ‘A’ | ‘C’ | ‘D’ | ‘F’ | ‘G’ | ‘K’ | ‘L’ | ‘N’ | ‘P’ | ‘Q’ | ‘R’ | ‘S’ | ‘T’ | ‘V’ | ‘X’

<reserved identifier>::= ‘B’ | ‘E’ | ‘H’ | ‘I’ | ‘J’ | ‘M’ | ‘O’ | ‘U’ | ‘W’ | ‘Y’ | ‘Z’

Following the design specification field in a JEDEC transmission can be any number of information fields. Each of the JEDEC fields begins with a character that identifies what type of field it is. Fields are terminated with an asterisk character. Multiple character identifiers can be used to create sub-fields (i.e., A1, A$, or AB3). Although they are not required, you may use carriage returns (CR) and line feeds (LF) to improve readability of the data.

Field Identifiers

Field identifiers which are currently used in JEDEC transmissions are shown above on the “field identifiers” line. The “reserved identifier” line indicates characters not currently used (reserved for future use as field identifiers). JEDEC field identifiers are defined as follows:

A	Access time	N	Note field
B	*	O	*
C	Checksum field	P	Pin sequence
D	Device type	Q	Value field
E	*	R	Resulting vector field
F	Default fuse state field	S	Starting vector
G	Security fuse field	T	Test cycles
H	*	U	*
I	*	V	Test vector field
J	*	W	*
K	Fuse list field (hex format)	X	Default test condition
L	Fuse list field	Y	*
M	*	Z	*

* *Reserved for future use*

Device Field (D)

Device selection by this field is not supported by the programmer. It has been replaced by the QF and QP fields and manual selection of devices.

Fuse Information Fields (L, K, F, C)

<fuse information> :: = [<default state>] <fuse list> {<fuse list>} [<fuse checksum>]

<fuse list> : = ‘L’ <number> <delimiter> {<binary-digit> [<delimiter>]} ' * '

<fuse list> :: = ‘K’ <number> <delimiter> {<hex-digit> [<delimiter>]} ‘*’

<default state> :: = ‘F’ <binary-digit> ‘* ‘

<fuse checksum> :: = ‘C’ <hex-digit>:4 ‘* ‘

Each fuse of a device is assigned a decimal number and has two possible states: zero, specifying a low-resistance link, or one, specifying a high resistance link. The state of each fuse in the device is given by three fields: the fuse list (L field or K field), the default state (F field), and the fuse checksum (C field).

Fuse states are explicitly defined by either the L field or the K field. The character L begins the L field and is followed by the decimal number of the first fuse for which this field defines a state. The first fuse number is followed by a list of binary values indicating the fuse states.

The information in the K field is the same as that of the L field except that the information is represented by hex characters instead of binary values. This allows more compact representation of the fusemap data. The character K begins the K field and is followed by the decimal number of the first fuse. The fuse data follow the fuse number and are represented by hex characters. Each bit of each hex character represents the state of one fuse, so each hex character represents four fuses. The most significant bit of the first hex character following the fuse number corresponds to the state of that fuse number. The next most significant bit corresponds to the state of the next fuse number, etc. The least significant bit of the first hex character corresponds to the state of the fuse at the location specified by the fuse number plus three.

The K field supports download operations only. The K field is not part of the JEDEC standard, but is supported by Data I/O for fast data transfer. The L and K fields can be any length desired, and any number of L or K fields can be specified. If the state of a fuse is specified more than once, the last state specified replaces all previous ones for that fuse. The F field defines the states of fuses that are not explicitly defined in the L or K fields. If no F field is specified, all fuse states must be defined by L or K fields.

The C field, the fuse information checksum field, is used to detect transmitting and receiving errors. The field contains a 16-bit sum (modulus 65535) computed by adding 8-bit words containing the fuse states for the entire device. The 8-bit words are formed as shown in the following figure. Unused bits in the final 8-bit word are set to zero before the checksum is calculated.

Word 00 Fuse No.	msb 7	6	5	4	3	2	1	lsb 0
Word 01 Fuse No.	msb 15	14	13	12	11	10	9	lsb 8
Word 62 Fuse No.	msb 503	-	-	-	499	498	497	lsb 496

Following is an example of full specification of the L, C, and F fields:

```
F0*L0 01010101* L0008 01010111* L1000 0101*C019E*
```

Following is an alternate way of defining the same fuse states using the K field:

```
F0*K0 55* K0008 57* K1000 5* C019E*
```

Another example, where F and C are not specified:

```
L0200     01101010101010101011
010111010110100010010010010*
```

The Security Fuse Field (G)

<security fuse>::='G'<binary-digit>'*'

The JEDEC G field is used to enable the security fuse of some logic devices. To enable the fuse, send a 1 in the G field:

```
G1*
```

The Note Field (N)

<note>::='N'<field characters>'*'

The note field is used in JEDEC transmission to insert notes or comments. The programmer will ignore this field; it will not be interpreted as data. An example of a note field would be:

```
N Test Preload*
```

The Value Fields (QF, QP, and QV)

JEDEC value fields define values or limits for the data file, such as number of fuses. The QF subfield defines the number of fuses in the device. All of the value fields must occur before any device programming or testing fields appear in the data file. Files with ONLY testing fields do not require the QF field, and fields with ONLY programming data do not require the QP and QV fields.

The QF subfield tells the programmer how much memory to reserve for fuse data, the number of fuses to set to the default condition, and the number of fuses to include in the fuse checksum. The QP subfield defines the number of pins or test conditions in the test vector, and the QV subfield defines the maximum number of test vectors.

The P Field

The P field remaps the device pinout and is used with the V (test vector) field. An asterisk terminates the field. The syntax of the field is as follows:

```
<pin list>::='P'<pin number>:N'*'

<pin number>::=<delimiter><number>
```

The following example shows a P field, V field, and the resulting application:

```
P 1 2 3 4 5 6 14 15 16 17 7 8 9 10 11 12 13 18 19 20 *

V0001 111000HLHHNNNNNNNNNN*

V0002 100000HHHLNNNNNNNNNN*
```

The result of applying the above P and V fields is that vector 1 will apply 111000 to pins 1 through 6, and HLHH to pins 14 through 17. Pins 7 through 13 and 18 through 20 will not be tested.

JEDEC U and E Fields

As of Version 2.5, the programmer supports the optional JEDEC U (user data) and E (electrical data) fields. The U and E fields are described below.

Note: Implementation of the JEDEC U and E fields is not part of the JEDEC-3C (JESD3-C) standard.

User Data (U Field)

The **U** field allows user data fuses that do not affect the logical or electrical functionality of the device to be specified in JEDEC files. For instance, the U field can be used to specify the User Data Signature fuse available in some types of PLD devices because this fuse contains information only (it has no logical or electrical functionality).

Note: To have the JEDEC U field processed correctly, you must select the device before downloading the JEDEC file.

The following guidelines apply to the U field:

- The U field must be included for devices with U fuses.
- Each U-field cell must be explicitly provided if the U field is present.
- The F (default fuse state) field does not affect U fuses.
- There can only be one U field in a JEDEC file.
- The U field fuses must be listed in the order they appear in the device.
- The U field must be listed after the L field and E field (if used), and before the V (test vector) field (if used).
- The U field is specified using binary numbers, since the full number of U-field cells is otherwise unknown.
- The number of cells specified in the U field is not included in the QF (number of fuses) field.

- The U-field cells are not included in the C (fuse checksum) field.
- The U field reads left to right to be consistent with the L (fuse list) and E fields.

The syntax for the U field is as follows:

```
<User Data Fuse List>::'U'<binary-digit(s)>'*'
```

The character U begins the U field and is followed by one binary digit for each U fuse. Each binary digit indicates one of two possible states (zero, specifying a low-resistance link, or one, specifying a high-resistance link) for each fuse.

For example,

```
QF24*
L0000
101011000000000000000000*
E10100111*
C011A*
U10110110*
```

Electrical Data (E field)

The **E** field allows special feature fuses that do not affect the logic function of the device to be specified in JEDEC files.

The following guidelines apply to the E field:

- The E-field cell must be explicitly provided if the E field is present.
- The F (default fuse state) field does not affect E fuses.
- There can only be one E field in a JEDEC file.
- The E field fuses must be listed in the order they appear in the device.
- The E field must be listed before the C (checksum) field. If the U field is used, the E field must come before the U (user data) field.
- The E field is specified using binary numbers, since the full number of E-field cells is otherwise unknown.
- The number of cells specified in the E field is not included in the QF (number of fuses) field.
- The E-field cells are included in the C (fuse checksum) field.
- The E field reads left to right for the purpose of checksum calculation.

The syntax for the E field is as follows:

```
<Electrical Data Fuse List>::'E'<binary digit(s)>'*'
```

The character E begins the E field and is followed by one binary digit for each E fuse. Each binary digit indicates one of two possible states (zero, specifying a low-resistance link, or one, specifying a high-resistance link) for each fuse. For example,

```
QF24*
L0000
101011000000000000000000*
E10100111*
C011A*
U10110110*
```

Test Field (V field)

<function test> :: = [<pin list>] <test vector> {<test vector>}

<pin number> :: = <delimiter> <number>

N :: = number of pins on device

<test vector> :: = 'V' <number> <delimiter> < test condition> :N '* '

<test condition> :: = <digit> 'B' | 'C' | 'D' | 'F' | 'H' | 'K' | 'L' | 'N' | 'P' | 'U' | 'X' | 'Z'

<reserved condition> :: = 'A' | 'E' | 'G' | 'I' | 'J' | 'M' | 'O' | 'Q' | 'R' | 'S' | 'T' | 'V' | 'W' | 'Y' | 'Z'

Functional test information is specified by test vectors containing test conditions for each device pin. Each test vector contains *n* test conditions, where n is the number of pins on the device. The following table lists the conditions that can be specified for device pins.

When using structured test vectors to check your logic design, do NOT use 101 or 010 transitions as tests for clock pins: use C, K, U, or D instead.

Test Conditions

0	Drive input low
1	Drive input high
2-9	Drive input to supervoltage #2-9
B	Buried register preload (not supported)
C	Drive input low, high, low
D	Drive input low, fast slew
F	Float input or output
H	Test output high
K	Drive input high, low, high
L	Verifies that the specified output pin is low
N	Power pins and outputs not tested
P	Preload registers
U	Drive input high, fast slew
X	Output not tested, input default level
Z	Test input or output for high impedance

Note: C, K, U, and D are clocking functions that allow for setup time.

The C, K, U, and D driving signals are presented after the other inputs are stable. The L, H, and Z tests are performed after all inputs have stabilized, including C, K, U, and D.

Test vectors are numbered by following the V character with a number. The vectors are applied in numerical order. If the same numbered vector is specified more than one time, the data in the last vector replace any data contained in previous vectors with that number.

The following example uses the V field to specify functional test information for a device:

```
V0001C01010101NHLLLHHLHLN*

V0002C01011111NHLLHLLLHLN*

V0003C10010111NZZZZZZZZZZN*

V0004C01010100NFLHHLFFLLN*
```

JEDEC Kernel Mode, Code 92

<kernel>::=<STX><design spec><min. fuse information><ETX><xmit checksum>

<design spec>::={<field character>}’*’

<min. fuse information>::=<fuse list>{<fuse list>}

You may use the JEDEC kernel format if you wish to send only the minimum data necessary to program the logic device; for example, if you do not want to send any test vectors. If you specify format code 92, the programmer will ignore everything except the design specification field and the fuse information field. The following fields will be ignored if format 92 is specified: C, F, G, Q, V, and X. Also, the security fuse will be set to zero and the transmission checksum will be ignored.

Figure 6-19 shows an example of a kernel JEDEC transmission.

Figure 6-19
An Example of JEDEC Kernel Mode Format

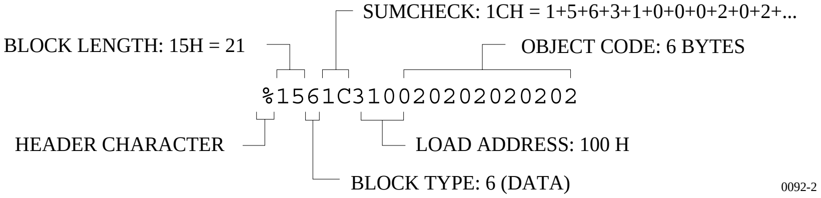
```
<STX>
Acme Logic Design  Jane Engineer   Feb. 29 1983
Widget Decode  756-AB-3456 Rev C Device Mullard 12AX7*
L0000 1111111011 1111111111  1111000000 0000000000
      0000000000 0000000000  0000000000 0000000000
      0000000000 0000000101  1111111111 1111111111
      0000000000 0000000000  0000111101 1111111111
      1111111111 1111110111  1111111111 1111111111*
L0200 1110101111 1111110000  0000000000 0000000000
      1111111111 1111011011  1111111111 1111111110
      0111111111 1111111111  1111111110 1111111111
      1111111111 1111101111  1111111111 1111101111
      0000000000 0000000000  0000*
<EXT>0000
```

0091-2

Extended Tektronix Hexadecimal Format, Code 94

The Extended Tektronix Hexadecimal format has three types of records: data, symbol, and termination records. The data record contains the object code. Information about a program section is contained in the symbol record (the programmer ignores symbol records), and the termination record signifies the end of a module. The data record (see sample below) contains a header field, a load address, and the object code. Figure 6-20 lists the information contained in the header field.

Figure 6-20
An Example of Tektronix Extended Format



Item	No. of ASCII Characters	Description
%	1	Signifies that the record is the Extended Tek Hex format.
Block length	2	Number of characters in the record, minus the %.
Block type	1	6 = data record 3 = symbol record (ignored by the programmer) 8 = termination record
Checksum	2	A 2-digit hex sum, modulo 256, of all the values in the record except the % and the checksum.

Character Values for Checksum Computation

The number of fields in the file will vary, depending on whether a data or a termination block is sent. Both data and termination blocks have a 6-character header and a 2-to-17 character address.

Character(s)	Value (decimal)	Character(s)	Value (decimal)
0 . . 9	0 . . 9	. (period)	38
A . . Z	10 . . 35	_(underline)	39
$	36	a . . z	40 . . 65
%	37		

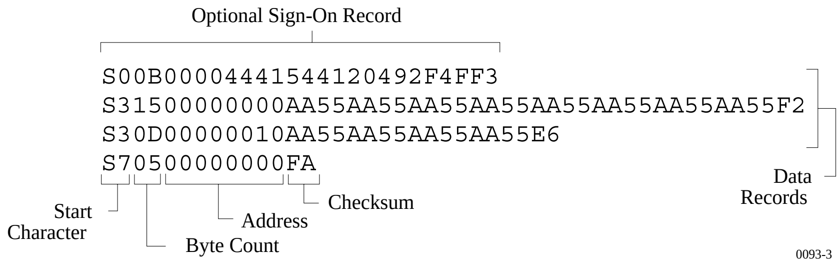
The load address determines where the object code will be located. This is a variable length number that may contain up to 17 characters. The first number determines the address length, with a zero signifying a length of 16. The remaining characters of the data record contain the object code, 2 characters per byte.

When you copy data to the port or to RAM, set the high-order address if the low-order is not at the default value.

Motorola 32-Bit Format, Code 95

The Motorola 32-bit format closely resembles the Motorola EXORmacs format, the main difference being the addition of the S3 and S7 start characters. The S3 character is used to begin a record containing a 4-byte address. The S7 character is a termination record for a block of S3 records. The address field for an S7 record may optionally contain the 4-byte instruction address that identifies where control is to be passed and is ignored by the programmer. Figure 6-21 shows a sample of the Motorola 32-bit format.

Figure 6-21
An Example of Motorola S3 Format



```
S00B000044415441204 92F4FF3
S31500000000AA55AA55AA55AA55AA55AA55AA55AA55F2
S30D00000010AA55AA55AA55AA55E6
S70500000000FA
```

Motorola data files may begin with an optional sign-on record, initiated by the start characters S0 or S5. Data records start with an 8- or 10-character prefix and end with a 2-character suffix.

Each data record begins with the start characters S1, S2, or S3: S1 if the following address field has 4 characters, S2 if it has 6 characters, S3 if it has 8 characters. The third and fourth characters represent the byte count, which expresses the number of data, address, and checksum bytes in the record. The address of the first data byte in the record is expressed by the last 4 characters of the prefix (6 characters for addresses above hexadecimal FFFF, and 8 characters for addresses above hexadecimal FFFFFF). Data bytes follow, each represented by 2 hexadecimal characters. The number of data bytes occurring must be 3, 4, or 5 less than the byte count. The suffix is a 2-character checksum, the one’s complement (in binary) of the preceding bytes in the record, including the byte count, address, and data bytes.

The end-of-file record begins with an S8 or S9 start character. Following the start characters are the byte count, the address, and a checksum. The maximum record length is 250 data bytes.

Hewlett-Packard UNIX Format, Code 96

This format divides the data file into data records, each with a maximum size of 250 bytes not including header information. An ID header is added to the beginning of the first record. Each subsequent record has its own header section. The section at the beginning of the file contains the following elements: the header 8004, filename, byte count for the processor information record, and the processor information record.

The header 8004 identifies the type of file being transferred. The first byte of this header (80) indicates that this file is binary, and the 04 indicates the type of file (absolute).

The ID header is followed by a 16-byte filename (not used by the programmer).

Next is the byte count, which indicates the size (minus one) of the Processor Information Record that follows. The Processor Information Record is divided into the following data words: Data Bus Width, Data Width Base, Transfer Address LS (least significant), and Transfer Address MS (most significant).

The Data Bus Width represents the width of the target system's bus (in bits). The Data Width Base represents the smallest addressable entity used by the target microprocessor.

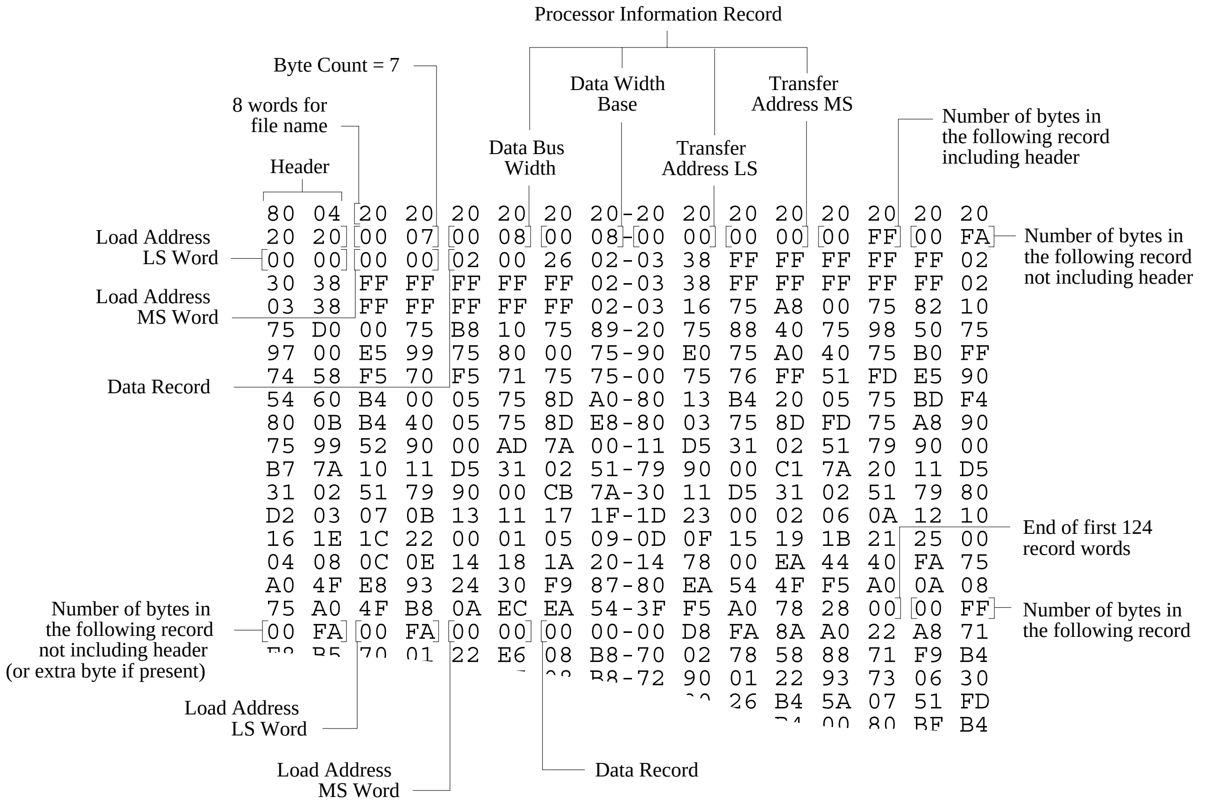
The Data Bus Width and Data Width Base are not used by the programmer during download. During upload, the Data Bus Width will be set to the current Data Word Width, and the Data Width Base will be set to 8. The Transfer Address LS and Transfer Address MS are not used by the programmer.

The data records consist of a header (8 bytes) and the data bytes. The first 2 bytes of the header indicate the size of the data record including the header (minus one). If the number of data bytes in the data record (not including the header) is odd, one extra byte will be added to the data record to ensure that an even number of data bytes exist in the data record. The maximum value for this field is 00FF hex. The next two bytes indicate the number of actual data bytes in the record, not including the header bytes and the extra byte (if present). The maximum value for this field is 00FA hex. The 4 bytes that follow represent the destination address for the data in this record. The rest of the bytes in the record are the data bytes.

This format has no end of file identifier.

The record length during upload is not affected by the upload record size parameter in the Configure / Edit / Communication screen. It is automatically set to transfer records using the maximum size (250 bytes) except for the last record. The size of the last record will be set according to the remaining number of data bytes.

Figure 6-22
Hewlett-Packard 64000 Unix Format



Processor Information Record
Byte Count = 7
Data Width Base
Transfer Address MS
8 words for file name
Number of bytes in the following record including header
Data Bus Width
Transfer Address LS
Header
Load Address LS Word
Number of bytes in the following record not including header
Load Address MS Word
Data Record
End of first 124 record words
Number of bytes in the following record not including header (or extra byte if present)
Number of bytes in the following record
Load Address LS Word
Load Address MS Word
Data Record
0474-2

```
80 04 20 20 20 20 20 20-20 20 20 20 20 20 20 20
20 20 00 07 00 08 00 08-00 00 00 00 00 FF 00 FA
00 00 00 00 02 00 26 02-03 38 FF FF FF FF FF 02
30 38 FF FF FF FF FF 02-03 38 FF FF FF FF FF 02
03 38 FF FF FF FF FF 02-03 16 75 A8 00 75 82 10
75 D0 00 75 B8 10 75 89-20 75 88 40 75 98 50 75
97 00 E5 99 75 80 00 75-90 E0 75 A0 40 75 B0 FF
74 58 F5 70 F5 71 75 75-00 75 76 FF 51 FD E5 90
54 60 B4 00 05 75 8D A0-80 13 B4 20 05 75 BD F4
80 0B B4 40 05 75 8D E8-80 03 75 8D FD 75 A8 90
75 99 52 90 00 AD 7A 00-11 D5 31 02 51 79 90 00
B7 7A 10 11 D5 31 02 51-79 90 00 C1 7A 20 11 D5
31 02 51 79 90 00 CB 7A-30 11 D5 31 02 51 79 80
D2 03 07 0B 13 11 17 1F-1D 23 00 02 06 0A 12 10
16 1E 1C 22 00 01 05 09-0D 0F 15 19 1B 21 25 00
04 08 0C 0E 14 18 1A 20-14 78 00 EA 44 40 FA 75
A0 4F E8 93 24 30 F9 87-80 EA 54 4F F5 A0 0A 08
75 A0 4F B8 0A EC EA 54-3F F5 A0 78 28 00 00 FF
00 FA 00 FA 00 00 00 00-00 D8 FA 8A A0 22 A8 71
E8 B5 70 01 22 E6 08 B8-70 02 78 58 88 71 F9 B4
                     08 B8-72 90 01 22 93 73 06 30
                              26 B4 5A 07 51 FD
                                 B4 00 80 BF B4
```

This data translation format was generated by a "dump utility" for illustrative purposes. Actual data files are in binary code and are typically generated by the appropriate development software.

Intel OMF386 Format, Code 97

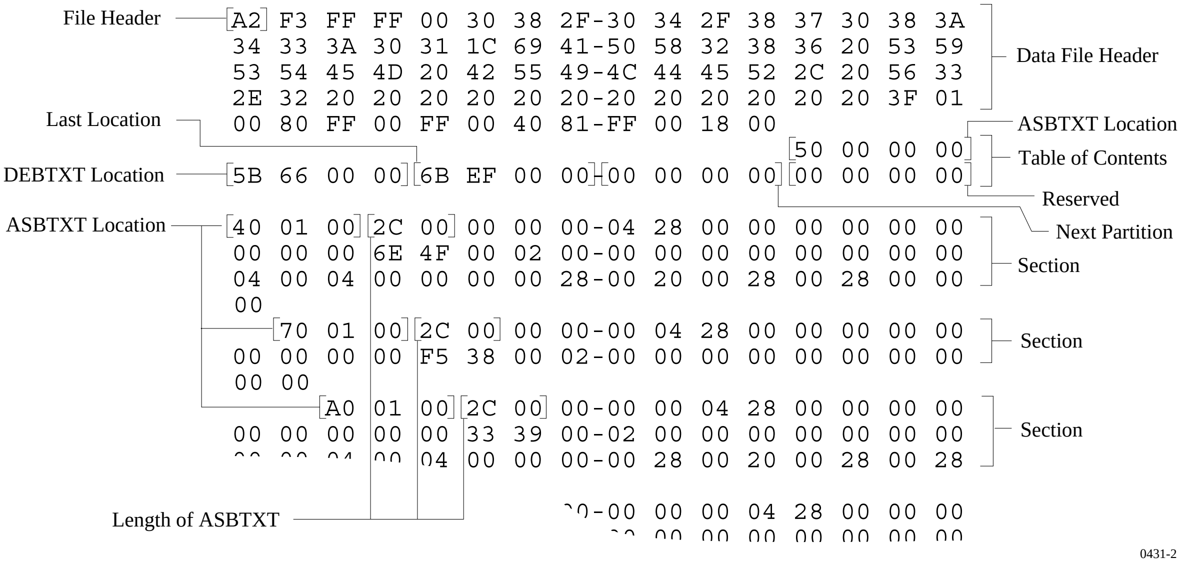
This data translation format is considered by Intel to be proprietary information. Contact your local Intel representative or call (408) 987-8080 for information about the structure of this format.

Intel OMF286 Format, Code 98

The Intel OMF286 format is a dynamically allocatable file format.

This format has three basic parts: the file header, data file module, and a 1-byte checksum. The file header is hexadecimal number (A2) that identifies this file as an Intel OMF 286 format file. See Figure 6-23.

Figure 6-23
A Sample of the Intel OMF286 Format

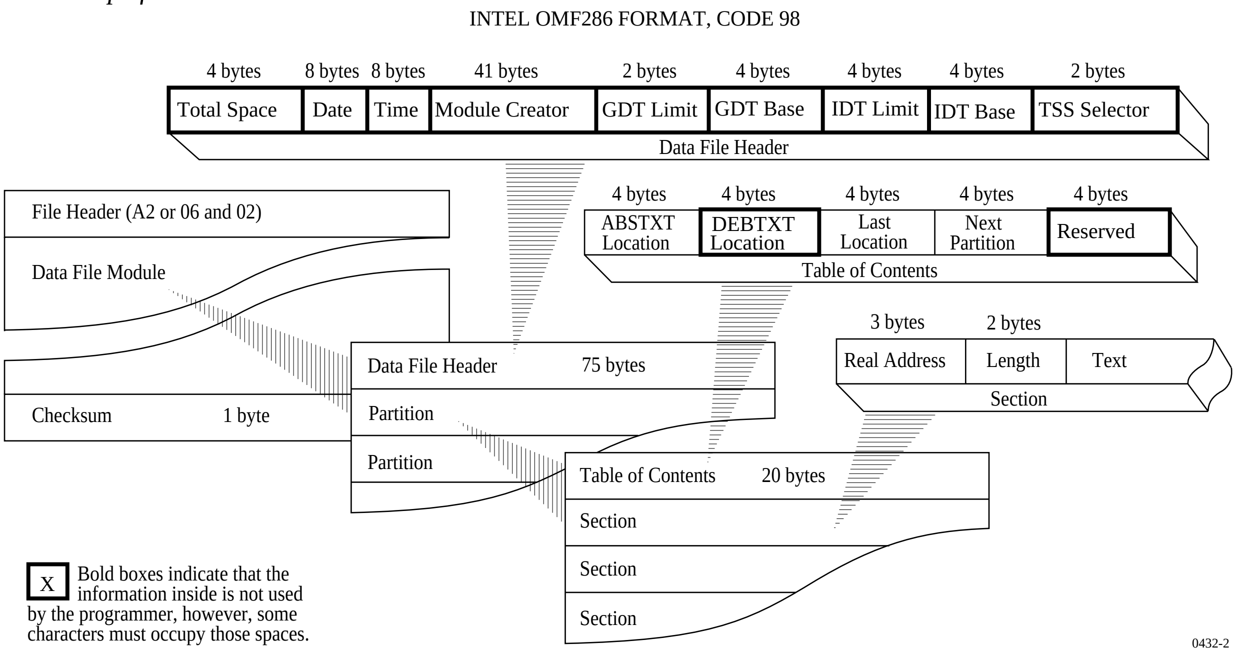


```
A2 F3 FF FF 00 30 38 2F-30 34 2F 38 37 30 38 3A
34 33 3A 30 31 1C 69 41-50 58 32 38 36 20 53 59
53 54 45 4D 20 42 55 49-4C 44 45 52 2C 20 56 33
2E 32 20 20 20 20 20 20-20 20 20 20 20 20 3F 01
00 80 FF 00 FF 00 40 81-FF 00 18 00
                                    50 00 00 00
5B 66 00 00 6B EF 00 00-00 00 00 00 00 00 00 00
40 01 00 2C 00 00 00 00-04 28 00 00 00 00 00 00
00 00 00 6E 4F 00 02 00-00 00 00 00 00 00 00 00
04 00 04 00 00 00 00 28-00 20 00 28 00 28 00 00
00
   70 01 00 2C 00 00 00-00 04 28 00 00 00 00 00
00 00 00 00 F5 38 00 02-00 00 00 00 00 00 00 00
00 00
      A0 01 00 2C 00 00-00 00 04 28 00 00 00 00
00 00 00 00 00 33 39 00-02 00 00 00 00 00 00 00
00 00 04 00 04 00 00 00-00 28 00 20 00 28 00 28

                        00-00 00 00 04 28 00 00 00
                           00 00 00 00 00 00 00
```

The first 75 bytes of the data file module is the data file header. The header information is generated and used by the development system and is not used by the programmer, although some characters must fill those bytes. The rest of the data file module consists of one partition.

The partition begins with a 20 byte table of contents. The table of contents specifies the locations of ABSTXT (absolute text), DEBTXT (debug text), the last location of this partition, and the location of the next partition. The OMF286 format consists of only one partition so this field will be zeros. The rest of the partition consists of sections. The actual data are located in the sections. The first 3 bytes in each section specify the real address of the text. The next 2 bytes state the length of the text, and the remainder of the section is the text (or data). Following the final section of the final partition is a 1-byte checksum representing the complement of the sum of all the bytes in the file, including the header. The sum of the checksum byte and the calculated checksum for the file should equal zero. The programmer ignores this checksum.

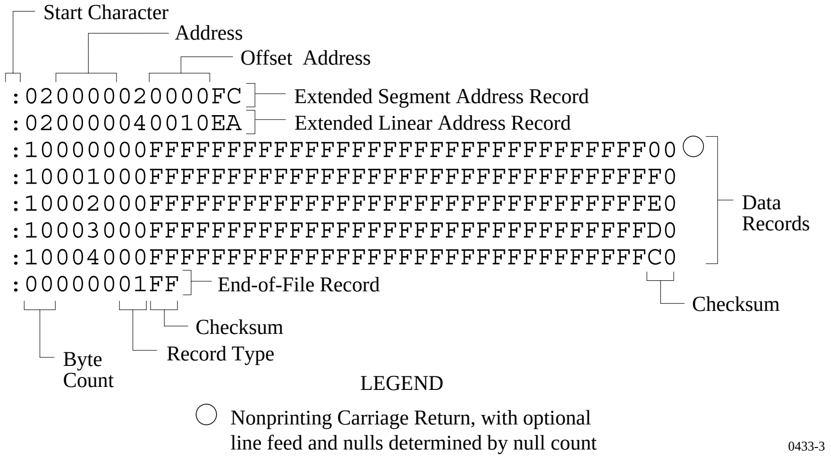
Figure 6-24
A Close-up of the Intel OMF286 Format



Intel Hex-32, Code 99

The Intel 32-bit Hexadecimal Object file record format has a 9-character (4-field) prefix that defines the start of record, byte count, load address, and record type, and a 2-character checksum suffix. Figure 6-25 illustrates the sample records of this format.

Figure 6-25
An Example of the Intel Hex-32 Format



```
:020000020000FC
:020000040010EA
:10000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
:10001000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0
:10002000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
:10003000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD0
:10004000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0
:00000001FF
```

Start Character
Address
Offset Address
Extended Segment Address Record
Extended Linear Address Record
Data Records
End-of-File Record
Checksum
Checksum
Record Type
Byte Count
LEGEND
○ Nonprinting Carriage Return, with optional line feed and nulls determined by null count
0433-3

The six record types are described below.

00-Data Record

This record begins with the colon start character, which is followed by the byte count (in hex notation), the address of the first data byte, and the record type (equal to 00). Following these are the data bytes. The checksum follows the data bytes and is the two's complement (in binary) of the preceding bytes in the record, including the byte count, address, record type, and data bytes.

01-End Record

This end-of-file record also begins with the colon start character and is followed by the byte count (equal to 00), the address (equal to 0000), the record type (equal to 01), and the checksum, FF.

02-Extended Segment Address Record

This is added to the offset to determine the absolute destination address. The address field for this record must contain ASCII zeros (Hex 30s). This record type defines bits 4 to 19 of the segment base address. It can appear randomly anywhere within the object file and affects the absolute memory address of subsequent data records in the file. The following example illustrates how the extended segment address is used to determine a byte address.

Problem

Find the address for the first data byte for the following file.

```
:02 0000 04 0010 EA
:02 0000 02 1230 BA
:10 0045 00 55AA FF ..... BC
```

Solution:

Step 1. Find the extended linear address offset for the data record (0010 in the example).

Step 2. Find the extended segment address offset for the data record (1230 in the example).

Step 3. Find the address offset for the data from the data record (0045 in the example).

Step 4. Calculate the absolute address for the first byte of the data record as follows:

`00100000`	Linear address offset, shifted left 16 bits
`+  12300`	Segment address offset, shifted left 4 bits
`+   0045`	Address offset from data record
`00112345`	32-bit address for first data byte

The address for the first data byte is 112345.

Note: Always specify the address offset when using this format, even when the offset is zero.

During output translation, the firmware will force the record size to 16 (decimal) if the record size is specified greater than 16. There is no such limitation for record sizes specified less than 16.

03-Start Segment Address Record

This record, which specifies bits 4-19 of the execution start address for the object file, is not used by the programmer.

04-Extended Linear Address Record

This record specifies bits 16-31 of the destination address for the data records that follow. It is added to the offset to determine the absolute destination address and can appear randomly anywhere within the object file. The address field for this record must contain ASCII zeros (Hex 30s).

05-Start Linear Address Record

This record, which specifies bits 16-31 of the execution start address for the object file, is not used by the programmer.

Highest I/O Addresses

The following table shows the highest I/O addresses accepted for each Data Translation Format.

Format Number	Format Name	Highest Address (hex bytes)
01-03	ASCII (BNPF, BHLF, and B10F)	N/A
04	Texas Instruments SDSMAC (320)	1FFFF (FFFF words)
05-07	ASCII (BNPF, BHLF, and B10F)	N/A
11	DEC Binary	N/A
12-13	Spectrum	270F
16	Absolute Binary	N/A
17	LOF	N/A
30-32	ASCII-Octal (Space, Percent, and Apostrophe)	3FFFF (777777 octal)
35-37	ASCII-Octal (Space, Percent, and SMS)	3FFFF (777777 octal)
50-52	ASCII-Hex (Space, Percent, and Apostrophe)	FFFF
55-58	ASCII-Hex (Space, Percent, SMS, and Comma)	FFFF
70	RCA Cosmac	FFFF
80	Fairchild Fairbug	FFFF
81	MOS Technology	FFFF
82	Motorola EXORciser	FFFF
83	Intel Intellec 8/MDS	FFFF
85	Signetics Absolute Object	FFFF
86	Tektronix Hexadecimal	FFFF
87	Motorola EXORmacs	FFFFFF
88	Intel MCS-86 Hex Object	FFFFF
89	Hewlett-Packard 64000 Absolute	FFFFFFFF
90	Texas Instruments SDSMAC	FFFF
91, 92	JEDEC (Full and Kernel)	N/A
94	Tektronix Hexadecimal Extended	FFFFFFFF
95	Motorola 32 bit (S3 record)	FFFFFFFF
96	Hewlett-Packard UNIX Format	FFFFFFFF
97	Intel OMF 386	FFFFFFFF
98	Intel OMF 286	FFFFFF
99	Intel Hex-32	FFFFFFFF

7. Messages

This chapter lists and describes UniSite-*xpi*'s system and error messages. If a message requires action, instructions for that action are included in the message's description. Messages are listed in alphabetical order.

To access the online help for a message, press **F3** or **?** when the message appears in the message bar. The online Help describes the problem and its solution in more detail.

Note: PSM (Package Specific Module) and FSM (Function Specific Module) refer to the modules that are installed on UniSite-xpi's front panel. The PSM is the small module located on the left side of the top panel; the FSM is the optional, large module on the right.

Troubleshooting Guides containing more detailed information about the following problems may be found at the end of this chapter.

Message List

`0 div err`

UniSite-*xpi* has experienced a divide-by-zero error that it cannot recover from. This is a fatal error; turn UniSite-*xpi* off and reboot the system. If the error recurs, contact Data I/O Customer Support.

`Adapter for selected device not supported.`

Adapter for selected device not supported. Update your system software.

`Addr err`

UniSite-*xpi* has experienced an error that it cannot recover from; turn UniSite-*xpi* off and reboot the system.

`Address out of range.`

The address you tried to select is beyond the selected device's range. Select an address that is within the limits of the device, or select a different device. This message appears while you are in memory editor, fuse editor, or using the under/overblow feature.

`Algorithm files recovered.`

The algorithm file transfer operation has failed and original algorithm files have been recovered. This is due to a non-fatal feature: for example, CTRL + Z have been pressed or there is no disk in drive A.

`Algorithm files not recovered.`

The algorithm file transfer operation has failed, but original algorithm files have not been recovered. You should format your Mass Storage Module's hard disk on the Format Disk screen, then perform the operation again.

`Algorithm revision is not compatible with system software.`

This message will appear if you are using the wrong version of the algorithm files.

If you are using an algorithm disk, remove the disk and check the version number on it. The programmer requires that the Boot and Algorithm disks be of the same software version (both V2.0, for example).

If you are using algorithm files stored in the Mass Storage Module, perform the Mass Storage operation on the Configure screen to update the algorithm files in the Mass Storage Module, then try again.

`Altera POF translator must be selected for POF devices.`

This message appears when you have selected an Altera POF device and attempt a data transfer operation, such as a download, without having selected the POF data translation format. Select the POF format as the data translation format and try the operation again.

`ASCII entry not allowed in 4-bit mode.`

This message appears in the memory editor when you attempt to go into ASCII entry mode when a 4-bit device is selected. Re-select device or edit in hex mode only.

`Bad algorithm data.`

An error was detected with the algorithm disk. Use another copy of the algorithm disk and repeat the operation.

`Bad Directory name.`

The directory name entered is not a valid directory name.

`Base for the device selected is not installed.`

The device you selected cannot be programmed in the Base that is presently installed. If you select a device that is not supported by the installed Base, the Load, Program, Verify, and Device Check screens cannot be entered. Install the correct Base and try the operation again.

`Begin address too large.`

The beginning address you selected in the memory editor was too large and is beyond the limits of the selected device. Change the begin address to one within the device's range.

`Beginning of file`

This message appears when you are viewing the first block of data and press **CTRL** + **P** (previous page) when you are using the memory editor, vector editor, fuse editor, or the under/overblow feature.

`Booting non-system disk. Insert system disk. Type ESC and CTRL W to reboot.`

This message will appear if UniSite-*xpi* detects a disk other than the Boot Files disk installed in drive A during power up. Insert the Boot Files disk.

`Buffer full. Insert Custom Menu algorithm disk.`

The Custom Menu algorithm building process uses user RAM as a buffer. This message appears when all available user RAM has been used and it is time to save the contents of the buffer onto the Custom menu algorithm disk.

`Bus err`

UniSite-*xpi* has experienced an error that it cannot recover from; turn UniSite-*xpi* off and reboot the system. If the error recurs, contact Data I/O Customer Support.

`Bytes copied = nnnnnn.`

This message appears while the Copy File operation is in progress; *nnnnnn* refers to the number of bytes copied.

`Calculating sumcheck.`

This message appears when you are using the Device Check's Sumcheck screen. It informs you that the RAM sumcheck is being calculated.

`Cannot access file.`

This message indicates that a file cannot be accessed. If you are using a floppy disk, insert the correct disk. If you are using the Mass Storage Module, you need to update your Mass Storage Module.

`Cannot access system disk.`

This message indicates a non-system disk is installed. Make sure the Algorithm disk and Boot Files disk are installed.

`Cannot allocate file space.`

The disk is full. Store the data on another disk or delete some current disk files that are not longer needed.

`Cannot begin job file recording from this screen.`

This message appears when you attempt to create a new job file from an illegal screen: one that cannot be part of a job file. Use a different screen to start the job file.

`Cannot copy multiple files into one file.`

This message will appear in the Copy Files screen if you attempt to use the wildcard file designation to copy multiple files to another single file (for example, if you enter “27*.DAT” for the filename, and the files 27128.DAT and 27512.DAT both exist on the disk).

`Cannot execute job file while recording job file.`

A job file cannot be executed while another is being recorded. You may “chain” job files together by ending the recording of a job file in the job file execution screen.

`Cannot purge file.`

This message means that, for some reason other than the disk’s being write-protected, the file cannot be removed from the disk. If the disk is write-protected, a different message will be generated.

`Cannot REPLACE more than 10 Keep Current algorithms.`

This message will appear when you try to REPLACE the 11th Keep Current algorithm in a REPLACE/RESTORE operation.

`Cannot update yield tally.`

This message will appear after a programming session or during device selection if the System Files disk is full, write-protected, or contains data errors. The yield tally information is stored on the System Files disk. Check your disk to see whether it is write-protected. If the disk is NOT write-protected, the cause of the problem is probably a data error on the System Files disk. User another copy of the System Files disk and try the same operation; if the problem persists, contact your nearest Data I/O Service Center for assistance.

`Cannot use Variable Parameter and Variable Vcc Verify on same device.`

This message may appear in two cases:

1) When you try to invoke Variable Parameter on a device algorithm after you have used Variable Vcc Verify on it.
2) When you try to invoke Variable Vcc Verify on a device algorithm after you have used Variable Parameter on it.

In either case, to remove the blocking of your desired operation simply select a different device, then select the device you want and try the operation again.

`Checksum error.`

The checksum of the data that you downloaded did not match the checksum calculated by the programmer. Try the operation again. If the problem continues, verify that the checksums generated from the host are indeed correct.

`Compare operation not supported for logic formats.`

The compare operation is only available for memory formats.

`[Computer Remote Control: enter Control-Z to exit.]`

This message informs you that UniSite-*xpi* is now in remote control mode and all programmer commands are now read from the remote port. Pressing **CTRL** + **Z** returns control to terminal mode.

`Constant over-current fault.`

This message indicates that an over-current condition exists and UniSite-*xpi* is unable to clear the condition. The over current could be caused by a hardware failure in UniSite-*xpi*. Reboot the system. If the condition persists, contact Data I/O Customer Support.

Constructing Job File Directory.

This job files are now being read in order to put together a job file directory. You can then select one of the files for playback.

Copying file1.ext to file2.ext. Bytes copied = xxxx.

This message appears during a Copy operation if you are using the wildcard designation. *xxxx* in the display is the number of bytes copied into the destination file.

Copying sectors ssss - ssss+120 Reading source disk.

This message appears while the Disk Copy command is proceeding. The information presented in this message displays the number of sectors copied in each pass. There are 1440 sectors on each disk. This message is accompanied by the message "Copying sectors ssss - ssss Writing destination disk" which appears while UniSite-*xpi* is writing data onto the destination disk.

Copying sectors ssss - ssss+120. Writing destination disk.

This message appears during the Disk Copy routine, indicating that the data is being copied.

Could not initialize default system parameters from disk.

When the programmer was booting up, either the terminal selection information couldn't be saved on the Boot Files disk, or the power-up system parameters could not be loaded. If the Boot Files disk is write-protected, reboot the programmer with the Boot Files disk not write-protected. If the disk wasn't write-protected, reboot the programmer with a different Boot Files disk, or call you nearest Data I/O Service Center for assistance.

Could not initialize default system parameters from disk.

When UniSite-*xpi* was booting up, the default and programming system parameters could not be loaded. Reboot UniSite-*xpi* with a different system disk, or contact Data I/O Customer Support for assistance.

Current system and disk system have different revisions.

This message appears when you are trying to update the Mass Storage system with a Boot Files disk that has a different revision number than the current system. Use a Boot Files disk that has the same revision number as the current system and try again.

Data conversion error.

This data transfer error indicates that the selected I/O translator could not handle the data properly (a mismatch between selected translator and data). Check your data format for illegal characters. Chapter 6 has sample listings of each translation format.

Data Load Err

An error occurred during a Gang/Set Load operation and data was not correctly loaded from this part.

Data record error.

The data that you attempted to transfer did not conform with the data translation format selected. Put the data into the correct format and attempt the operation again.

Data size exceeded user memory.

This message appears when you try to load a data file that is too large for user RAM. Specify the size of the data file before attempting the operation again.

`Data Source/Destination set to RAM file.`

This message appears after you exit CRC mode (using CTRL + Z) with the Source/Destination parameter set to RAM file. Device operations will use the RAM file(s) unless the parameter is changed to RAM or disk.

`Data transfer complete.`

This message appears after a data transfer with an external source was successfully completed.

`Data transfer complete. Data Sum = ssssssss.`

After a data transfer, this message appears. The data sum represents the calculated sumcheck for the data bytes transferred.

`Data transfer complete. Data Sum = ssss. Xmit = ssss.`

After a data transfer of a JEDEC file, this message appears. The data sum represents the calculated checksum for the data bytes in the fusemap section of the data transferred. The Xmit sum represents the calculated checksum for all the bytes transferred.

`Data transfer complete. Data Sum = ssss. POF CRC = ssss.`

After a data transfer of a POF file, this message appears. The data sum represents the calculated checksum for the data bytes transferred. The POF CRC represents the calculated Cyclic Redundancy Check for all the bytes in the POF file up to, but not including the CRC value.

`Data operation complete: data saved on disk.`

After a data file is downloaded to disk, this message appears.

`Data verify error.`

Data in RAM (or disk) was not the same as the data that was transferred from the remote system, or not the same as device data in a verify operation.

`Destination file already exists.`

The filename you selected for the file is the same as an existing one. Create a different filename.

`Destination file already exists. Hit return to continue, ^Z to abort.`

The filename that you have designated as the destination for the data already exists, so existing data will be written over if you execute the operation. This is a precautionary message that occurs on any file operation which could overwrite an existing file.

`Device Alg Err`

A device algorithm error has occurred.

`Device algorithm not found:`

This message will occur if you select a device using family/pinout codes, and the codes selected correspond to an algorithm which is not supported by the programmer.

Device Insertion Warning. Press F3.

The pins on the inserted device are not making good contact with the programming fixture. Check the following:

- Is the device properly justified in the socket or properly oriented in the MatchBook? (See the programmer User Manual for more information on inserting devices.)
- Check the device for bent or damaged pins/leads. Repeat the operation with a similar device from the same manufacturer. If the operation with the second device is successful, the first device is probably defective.
- Try to perform the operation with the Continuity Check disabled. If the device loads without errors, try to program the device. If the programming operation is successful, contact Data I/O Customer Support. Continuing to program with Continuity Check disabled will not compromise programming integrity.
- If you are using a base that contains an Elastomer PAD, try cleaning the PAD with distilled alcohol or compressed air. Also, using distilled alcohol, try cleaning the contacts between the programmer and the adapter.
- Check the socket for bent or damaged pins.

See page 7-40 for more information.

Device must be previously selected:

Before you may enter the Load, Program, Verify, or Device Checks screens, you must first select a device. This message appears if you have not yet selected a device but try to go into one of those screens.

DEVICE SPECIFIC FAILURE: Backward device.

Device specific backwards test failure. Remove the device and re-insert it correctly.

DEVICE SPECIFIC FAILURE: Cannot synchronize to device.

This error message may appear if you are trying to program Motorola microprocessors. Try the operation again. This message usually indicates a faulty device.

DEVICE SPECIFIC FAILURE: DAC setting out of range.

System algorithm error. Contact Data I/O.

DEVICE SPECIFIC FAILURE: Device programming error.

This message appears if the programmer detects a defective location in the device during the programming operation. If this message appears, try erasing the device and programming it again if it is erasable; otherwise, try another device.

DEVICE SPECIFIC FAILURE: Device verification error.

This error message may appear if you are trying to program devices that perform verification during programming. This message indicates that the part failed to verify.

DEVICE SPECIFIC FAILURE: Device verification error.

A device verification error occurred. This device performs the verify operation internally and issues a pass/fail signal at the end of the operation. The location of the failure is not available.

DEVICE SPECIFIC FAILURE: EEPROM at $B800 not allowed.

This error message may appear if you are trying to program Motorola 86HC811 devices. This message means that you have attempted to specify the EEPROM data block beginning address at location B800. Using this address is not allowed. Edit the configuration byte at location 103F, changing the upper nibble to indicate any OTHER starting address.

DEVICE SPECIFIC FAILURE: Encryption Array already Programmed.

This message appears if you try to reprogram a device on which the encryption array has already been programmed. Reprogramming of the main array is not allowed once the encryption array has been programmed.

DEVICE SPECIFIC FAILURE: Generic fatal device-specific error.

This message may occur when a unique error code or message has not been assigned.

DEVICE SPECIFIC FAILURE: Illegal bit error.

This message may appear if you are trying to program Cypress devices, family 0F0 and 0EE. This message will be generated if a non-blank error occurs before programming has begun. Try erasing the part (if possible), and then try to program the part again. If the message shows again, the device may be faulty.

DEVICE SPECIFIC FAILURE: Illegal output configuration.

This error message occurs when the MMI 10H20EV8 or 10020EV8 has an output configuration fuse not in the JEDEC fuse map that is associated with the register bypass fuse. The output configuration fuse is programmed when the bypass fuse is programmed. This error happens when loading a device that has one of these fuses programmed, but not the other.

DEVICE SPECIFIC FAILURE: Insufficient Pin Driver Boards Installed.

This device requires additional pin driver boards. This algorithm may require more pin drivers than the number of pins on the device. Contact Data I/O in order to upgrade your system.

DEVICE SPECIFIC FAILURE: Invalid algorithm instruction.

The algorithm contains a call to an invalid algorithm instruction. System algorithm error. Contact Data I/O.

DEVICE SPECIFIC FAILURE: Invalid Architecture Word.

Data entered into User RAM for the architecture word, or control word, is not valid for this device. Refer to the manufacturer's specification for information regarding which bits are not programmable, or which bits must be programmed.

DEVICE SPECIFIC FAILURE: Invalid argument.

The algorithm contains a call with an invalid argument value. System algorithm error. Contact Data I/O.

DEVICE SPECIFIC FAILURE: Invalid device ID.

Certain devices contain an ID byte which is used to select the proper programming voltages for the device in the socket. This error is generated when the ID read from the device is not recognized by the algorithm. An invalid device ID may indicate that an algorithm update is needed to support the device in the socket. This error may also be generated if the device in the socket has been damaged. See page 7-45 for more information.

DEVICE SPECIFIC FAILURE: Invalid RAL or GAL in socket.

This error message may appear if you are trying to program Lattice family 36V or VTI family 02F devices. This error indicates that the manufacturer's signature read from the device does not match the one stored on the programmer's disk.

DEVICE SPECIFIC FAILURE: Pullup/Pulldown group assignment error.

System algorithm error. Contact Data I/O.

`DEVICE SPECIFIC FAILURE: Security fuse programmed or bad device.`

The device is not accessible either because the device is faulty, or because the security fuse is programmed. This error message may appear if you are trying to program Motorola 68705xx devices.

`DEVICE SPECIFIC FAILURE: Software data protect failure.`

The “Software data protect” option has been enabled when programming, and the device(s) has not been successfully placed in the protected state. Alternatively, the device data may be incorrect.

`DEVICE SPECIFIC FAILURE: Test fuse programming error.`

For certain devices, there are “transparent” test fuses that are automatically programmed during a program operation. These fuses cannot be used, seen, or read by the user. If these fuses fail to program correctly, this error occurs and the device may be faulty. Use another device.

`DEVICE SPECIFIC FAILURE: Un-supported Operation.`

The selected operation is not supported for this device.

`DEVICE SPECIFIC FAILURE: Verify error due to non-blank device.`

This error message may appear if you are trying to program Signetics device, family 087. Due to the characteristics of these devices, it is impossible to perform blank and illegal-bit checks. Consequently, it is assumed that the device has been properly erased prior to programming. Try erasing the part and then try programming it again. If the message reappears, the device may be faulty.

`DEVICE SPECIFIC WARNING: Generic non-fatal device-specific error.`

This message may appear when a unique error code or message has not been assigned.

`DEVICE SPECIFIC WARNING: Part has been programmed over 128 times.`

This error message may appear if you are trying to program Lattice family 036 or VTI family 02F devices. This message indicates that the part has been reprogrammed more than 128 times, which is more than the manufacturer recommends. This is only a warning message; the program operation was successful.

`Disk access error.`

Disk access error.

`Disk boot err`

UniSite-*xpi* has experienced an error that it cannot recover from. Turn UniSite-*xpi* off and reboot the system. If the problem persists, use another copy of the Boot Files disk.

`Disk data error.`

The read or write operation that was attempted could not be completed because there is a problem with the disk. Try the operation again with a different disk.

`Disk duplication overwrites user RAM. Hit Return to continue, ^Z to abort.`

If you copy a disk, UniSite-*xpi* uses User RAM as a buffer. Anything already in user RAM will be overwritten. If you don’t want to change User RAM, press **CTRL** + **Z** to halt the disk duplication. Press **ENTER** to proceed with the operation.

`Disk Error - Defective or Wrong Size Disk. Press F3.`

The read or write operation that was attempted could not be completed because there is a problem with the disk; try the operation again with a different disk.

`Disk error, terminal type not saved!`

If you try to save the terminal type as one of the power-up parameters and there is a write problem with the disk (the disk is either full or defective), this message will appear.

`Disk open error. Type ESC and Control W to reboot.`

This message appears if you try to boot UniSite-*xpi* without the Boot Files disk in the disk drive. Insert the Boot Files disk in the disk drive and reboot UniSite-*xpi*.

`Disk write-protected, terminal type not saved!`

If you try to save the terminal type as one of the power-up parameters and the disk is write-protected, this message will appear. Move the write-protect slide so that the hole through the disk is blocked.

`Done.`

The operation is completed. Proceed to the next operation you want to perform.

`Done. Bytes copied = nnnnnn.`

This message appears after the Copy File operation is complete. It displays the size in hexadecimal bytes of the file that was copied. Proceed to the next operation you want to perform.

`Done: Certain tests not performed due to Socket not empty.`

This message appears if a device was present in the socket when the “perform all test function” was invoked. Any test requiring the device socket to be empty will not be performed. Remove the socketed device and try the operation again.

`Edit begin address must be greater than or equal to offset.`

This message indicates that the begin address is less than the lowest location of system user memory. The memory data editor’s offset is used as a reference for user memory addresses. The begin address represents the beginning address of user data. Reset the parameters.

`Elec Erase Err`

A device error occurred when UniSite-*xpi* attempted to electronically erase this part.

`Elec ID Err`

The electronic ID of the socketed device does not match the one UniSite-*xpi* has stored in its memory. Make sure the correct device type is installed or selected.

`Electronic bulk erase not supported by device.`

You tried to electronically erase a device that was not electronically erasable. Erase the device with ultraviolet light, or select a device that can be electrically erased.

`Electronic ID not supported by device.`

This message appears after the programmer tried to read the electronic ID of a device that does not have an ID. Check to make sure you selected the correct device.

`Empty Socket.`

The selected device operation requires that a device be present in the device socket. Place the appropriate device in the socket and repeat the operation.

`Empty Socket`

UniSite-*xpi* has detected that the socket is empty.

`End of file.`

This message appears when you are viewing the last block of data and press **CTRL** + **N** (next page) when you are using the memory editor, vector editor, fuse editor, or the over/underblow feature.

`Entry into CRC not allowed during job file recording or playback.`

This message appears if you try to invoke CRC during a job file recording or playback operation. Reprogram you job file without invoking CRC.

`ER_ALG_DYN_LINK`

Algorithm Dynamic Linker error. Please contact Data I/O.

`Extension '.SYS' is reserved for system use.`

This message appears when you attempt to use ".SYS" as an extension for one of your filenames. Give your file another extension, such as ".DAT" or "DOC."

`Factory setting save file is protected:`

This message appears if you try to save a parameter file to zero. Parameter file number zero is the file that contains the set of configuration or system parameters set by the factory when the programmer is shipped: it cannot be changed.

`Fatal system err`

UniSite-*xpi* has experienced an error that it cannot recover from; turn UniSite-*xpi* off and reboot the system. If the error recurs, contact Data I/O Customer Support.

`File already exists.`

The destination file already exists on the disk.

`File does not exist.`

The file you tried to use is not on this disk. Try another disk or another filename.

`FILE ERROR: Cannot access yield data.`

This message appears if yield data could not be updated for one of the following reasons: the System Files disk is not installed, the System Files disk is write-protected, or the yield file has been corrupted. Correct the problem and try the operation again.

`FILE ERROR: Can't reach track 0.`

If this message appears, a fatal disk error has occurred. The disk drive may be faulty. Contact Data I/O Customer Support.

`FILE ERROR: Data transfer error.`

This message appears if UniSite-*xpi* could not successfully read or write user or system data to the disk. If trying the operation again still causes the error messages to appear, a new disk (and a new copy of whatever software or data the unit needs) must be used.

`FILE ERROR: Disk change detected.`

This message appears if the programmer has detected a disk change during a disk file operation. Try the operation again and leave the disk in the drive until the operation is finished.

`FILE ERROR: Disk is write-protected.`

The disk cannot be written to because it is write-protected. Move the write-protect slide so that the hole through the disk is blocked, or use another disk.

`FILE ERROR: Error in sector preamble.`

This error appears when UniSite-*xpi* detects an error with the format of a disk. Use a different disk or reformat the existing disk and try the operation again.

`FILE ERROR: Mass Storage Module is unusable, (requires formatting).`

This message is displayed if the Mass Storage Module gave an error when it was accessed. This may indicate that the Mass Storage Module requires formatting. Perform the Mass Storage format operation. If the error persists, contact your nearest Data I/O Service Center.

`FILE ERROR: No disk in drive.`

This message appears when UniSite-*xpi* is trying to access a disk file but the disk drive is empty. Insert the disk with the file to be used.

`FILE ERROR: Sector not found.`

The programmer was unable to find the desired sector during disk operation. Try a different disk.

`FILE ERROR: Source file does not exist.`

This message appears during the Copy File operation if the file that you want to copy is NOT on the disk installed in the disk drive. Insert the disk that has the source file into the disk drive and attempt the operation again.

`FILE ERROR: Track not found.`

This message appears if UniSite-*xpi* cannot find the disk track associated with the system file, or cannot find the data needed to support whatever action you just requested. If you try the operation again and the error message reappears, a new disk (or a new copy of whatever software or data the unit needs) must be used.

`File must have ".EXE" extension.`

The file name must have a “.EXE” extension to execute as a user program. Rename the file and try the operation again.

`File name too long for HiTerm.`

The file name that was entered on the host command line has too many characters. A legal file name consists of up to eight characters in the name, and up to three characters in the extension. A path may precede the filename; however, the length of the path together with the filename must not be greater than 25 characters.

`File not initialized! Enter ‘C’ to initialize, any other key to quit.`

This message appears within the fuse editor, vector editor, or when the under/overblow feature is selected. The file that you have selected is not in a format that is compatible with the feature that you want to use. If you want to use the fuse editor, and the data file you have is not formatted for the device you have selected, typing **C** reformats the data file to be compatible with the device.

`File opened exclusively. (obsolete)`

File opened exclusively.

`Formatting and initializing user disk.`

This message appears while a disk is being formatted.

`Framing error during transfer. Data sum = sssssss.`

This error indicates that a received character did not contain a valid stop bit. If this error occurs, try using a slower baud rate. If this error occurs after performing a high speed download, the communication board in your PC may not be able to transfer data at 15.2 Kbd. In this case, either disable the high speed download feature, or try a different communication board in the PC.

`FSM/PSM for the device/mode selected is not installed.`

The device you selected cannot be programmed in the PSM or FSM that is presently installed. If you select a device that is not supported by an installed PSM or FSM, the Load, Program, Verify, and Device Check screens cannot be entered. Install the correct module and try the operation again.

`Fuse checksum error. Data sum = ssss. Xmit sum = ssss.`

After a download or input from disk operation of a JEDEC file, this error indicates the checksum calculated for the fusemap data received does not match the fusemap checksum bytes in the file. The data sum bytes represent the calculated value for the fusemap data bytes. The Xmit sum indicates the calculated checksum for all of the bytes received.

`HiTerm file I/O error.`

This message appears if there are any file I/O errors (such as input file not found, corrupted input file, or input file not readable) when downloading a file from the PC to the programmer using HiTerm in high speed mode.

`HiTerm protocol error.`

This message appears when an error occurs during parameter passing between the programmer and HiTerm. Ensure that the mode shown on the HiTerm help screen is ‘Programmer’ mode, and that the interface cable is securely connected between the programmer and the PC. The HiTerm mode is specified in the configuration file.

`Hit PF3 or ? to view device specific message.`

The selected device has specific information associated with it.

`Hit return to continue, ^Z to abort.`

This message appears after a Verify operation has failed. If you want to ignore the warning and proceed with the verify operation without examining the errors, press **ENTER**. If you want to investigate the verify errors, press **CTRL** + **Z**, and the Verify screen will reappear.

`Hit return to switch user menu port, ^Z to abort.`

This message is displayed whenever you toggle the User Menu Port parameter. To cancel the port switch operation, press **CTRL** + **Z**, otherwise press **ENTER** to switch the port. The cable between the programmer and the PC (or terminal) should then be moved to the port specified by the parameter.

`I/O address beyond range of data format selected.`

An I/O address in the data received exceeded the highest value allowed in the address field of the data format selected. Prior to performing an upload or output to disk operation, the programmer calculates the highest I/O address that will be output based on the parameters supplied by the user. The programmer aborts the operation if the I/O address is too large for the data format selected. The formula for calculating the highest I/O address is:

Highest I/O address = I/O addr offset + User data size - 1

To avoid this error condition, either select a different data format (one which will support the I/O addresses for the transfer operation), or decrease the value of the I/O offset address and/or the User data size to achieve I/O addresses within the range of the present data format selected. The I/O addr offset parameter is also considered an unsigned value if it is treated as a value of 0.

`I/O timeout error. Press F3. Data checksum = ssssssss Xmit sum = ssssssss.`

This error occurs when a file is being transferred from systems such as PC, Sun, or Vax over RS-232 or via programmer disk drive. If the programmer displays data sum = `00000000`, no data was transferred. Other hex values indicate the file transferred partially or completely. Check the following:

- Wrong Download Command Sent to Host: Your host machine will transfer a file receipt of the proper command. Under HiTerm, the Download Host command must begin with “tr” or “transfer,” followed by the appropriate drive letter, path, and filename.
- Wrong I/O Translation Format Selected: The format of the file being transferred must match the description in the User Manual. If it doesn’t, enter the proper I/O translation format number and transfer the file again.
- Unrecognized Characters in Beginning of File: The data file must begin with characters that match the appropriate format described in the User Manual. Make sure the file has a “start character.”
- No Recognizable End-of-File Character: The data file must end with the proper end-of-file character or record.

See page 7-50 for more information.

`I/O timeout error. Data sum = sssssss.`

This error occurs when a file is being transferred from systems such as PC, Sun, or Vax over RS-232 or via programmer disk drive. If the programmer displays data sum = `00000000`, no data was transferred. Other hex values indicate the file transferred partially or completely. Check the following:

- Wrong Download Command Sent to Host: Your host machine will transfer a file receipt of the proper command. Under HiTerm, the Download Host command must begin with “tr” or “transfer,” followed by the appropriate drive letter, path, and filename.
- Wrong I/O Translation Format Selected: The format of the file being transferred must match the description in the User Manual. If it doesn’t, enter the proper I/O translation format number and transfer the file again.
- Unrecognized Characters in Beginning of File: The data file must begin with characters that match the appropriate format described in the User Manual. Make sure the file has a “start character.”
- No Recognizable End-of-File Character: The data file must end with the proper end-of-file character or record.

See page 7-50 for more information.

I/O translation format error.

There is a compatibility problem with the data translation format you are using. Check the format of your data; if the format is JEDEC, ensure that the device to be programmed is selected prior to downloading the file. The Computer Remote Control section of this manual contains a description of all the data translation formats supported by the programmer.

For LOF, the format error may be caused by an incompatible XIP compression format. Ensure that the LOF file is compressed using the ZIP “imploded” format.

I/O translation format is not compatible with the device selected.

This message appears if you attempt a data transfer operation without using the appropriate type of I/O translation format. Memory devices require the use of one of the memory type of I/O translation formats. Likewise, logic devices require the use of a logic type of I/O translation format. Alter POF devices require the use of the POF I/O translation format.

I/O translator format not selected.

The data translation format must be selected before an external data transfer may be performed. Select a data translation format from the Format Select screen and attempt the operation again.

IOX init err

UniSite-*xpi* has experienced an error that it cannot recover from; turn UniSite-*xpi* off and reboot the system.

Illegal bit

UniSite-*xpi* has detected already-programmed locations of incorrect polarity in this socketed device. An illegal bit is a programmed device bit whose corresponding memory bit is unprogrammed.

Illegal instr err

UniSite-*xpi* has experienced an error that it cannot recover from; turn UniSite-*xpi* off and reboot the system.

ILLEGAL KEY ENTERED.

A character entered from the terminal was not recognized by the programmer.

Illegal Key Input: Type control-Z to abort parameter entry.

You pressed a key that is illegal for the field where the cursor is positioned. For example, if you enter a hex number for Data Word Width field (only decimal numbers are allowed), this message will appear.

Illegal operation for the device selected.

The device you selected using the Select device screen does not support the operation that you are attempting to enable.

Illegal options for this command.

This message appears when you attempt an operation with an illegal option. For example, the Verify function requires that the “Verify passes” be set to either “1” or “2.” When this message appears, go to the Programming Parameters screen or to the Options screen and enable the appropriate option. Try the operation again.

Illegal or odd Block Size not allowed.

This message will appear if you attempt to perform a Swap Data operation in Byte mode with the Block Size (bytes) set to an odd number. This message also appears if you attempt to perform a Swap Data operation in 32-bit mode with the Block Size (bytes) set to a number which is not a multiple of 4.

`Illegal parameter specified.`

This message indicates an illegal parameter was specified for the data format selected. If the TI SDSMAC (320) format was selected, this error indicates that the memory begin address, user data size, I/O address offset parameter, or upload record size parameters were odd values. This format requires that these parameters be even.

`Illegal parameter value.`

This message will appear if you use an inappropriate number to specify a parameter. Enter a legal value.

`Illegal terminal type!`

The terminal type number you entered was not one of the choices presented on the Terminal Type screen. Type in a valid terminal type number.

`Improper use of buried register preload.`

During programming of the device, an error occurred because either the buried register preload function is specified incorrectly in a test vector, or the buried register index is out of range. Check your structured vectors and try the operation again with a different device.

`Incompatible algorithm file set.`

This message appears when the revision numbers of the files on your algorithm disk are not identical. Check the integrity of your algorithm disk and try again.

`Incompatible baud rates between remote & terminal ports: no changes entered.`

If you selected incompatible baud rates for an operation and then pressed the carriage return key, this message will appear. Select a compatible baud rate. Refer to the Configure system command description in Chapter 4 for a listing of compatible baud rates.

`Incompatible software revision numbers. Overlay file not loaded.`

This message appears if the core version number (the middle version number on the screen) does not match the overlay file. The overlay file will not be loaded. Use another Boot Files disk with the same version number, or reboot the system with the Boot Files disk in drive A.

`Incompatible User Data Size and Data Word Width.`

This message appears if you try a 16-bit (or greater) device operation and the user Data Size is not compatible with the Data Word Width selected. Frequently, this happens when you are using a 16-bit device with an odd byte count User Data Size which is defined in bytes. Adjust your User Data Size.

`Incomplete boot/algorithm file set.`

1) If yu are selecting a device, this message appears when there is not a complete set of algorithm files stored in the Mass Storage Module. Perform the Mass Storage operation and try again.

2) If you are updating the Mass Storage algorithm, this message appears when there is not a complete set of algorithm files stored on your algorithm disk. Check the integrity of your algorithm disk and try again.

3) If you are updating the mass Storage system, this message appears when there is not a complete set of boot files stored on your Boot Files disk. Check the integrity of your Boot Files disk and try again.

`Incompatible disk formats.`

This message indicates that you are trying to duplicate a disk and are using disks of different density. For example you might be trying to copy a quad density disk to a double density disk. make sure your target disk is of the same density as the source disk is and try the operation again.

`Insert algorithm disk.`

This message will appear during the Select device operation if the algorithm file is not present on the selected drive. Check the “Algorithm type” and “Algorithm media” settings.

Insert a disk containing the appropriate type of algorithms.

This message will also appear in the following situations:

4) The programmer needs adapter information when a device operation is attempted using the PPI base.
5) When attempting to view device specific information if the algorithm disk is not present. In this case, insert the algorithm disk, select the device again, and then view the device-specific information.

`Insert Algorithm Set 1 disk.`

This message will appear if you attempt to select a device from Algorithm Set 1 and the disk is not present.

Insert the Algorithm Set 1 disk. Press return to perform the operation again.

`Insert Algorithm Set 2 disk.`

This message will appear if you attempt to select a device from Algorithm Set 2 and the disk is not present.

Insert the Algorithm Set 2 disk. Press return to perform the operation again.

`Insert Algorithm Set 3 disk.`

This message will appear if you attempt to select a device from Algorithm Set and the disk is not present.

Insert the Algorithm Set 3 disk. Press return to perform the operation again.

`Insert blank device. Hit return.`

This message appears during a Quick Copy operation. Remove the newly programmed device or the master device from the device socket, place a blank device in the socket and press **ENTER**. UniSite-*xpi* will then begin programming the blank device with RAM data loaded from the master device.

`Insert blank device. Hit return to continue or push START lever forward.`

(Legacy) This message appears during a Quick Copy operation if you are using the SetSite module. Place the blank devices into the SetSite sockets, lock them into place, and press **ENTER** or move the socket lever to the START position. UniSite will then start to program the parts with data from RAM.

`Insert Boot Files disk.`

The programmer needs the Boot Files disk present in order to access a file which is on that disk. Insert the Boot Files disk.

`Insert Custom Menu algorithm disk.`

This message will appear during the Select device or Custom Menu algorithm building operation if the Custom Menu algorithm disk is not present on the selected drive. Check “Algorithm type” and “Algorithm media” settings.

`Insert destination disk. Hit return to continue.`

This message may appear during the Duplicate Disk or Copy File operation. When this message appears, remove the source disk, insert the destination disk (the disk where you want the data to go) and press **ENTER**. Make sure to use a formatted disk. If you insert an unformatted disk, UniSite-*xpi* will abort the operation and display “Sector not found.” If this occurs, perform the Format Disk operation and restart the Duplicate Disk or Copy File operation.

`Insert Extended algorithm disk.`

This message will appear during the Select device operation if the algorithm file ALG.EXT is not present on the selected drive. Check the “Algorithm type” and “Algorithm media” settings.

`Insert Keep Current algorithm disk.`

This message will appear when you try to perform any operations related to Keep Current algorithms, but no Keep Current algorithms can be found on either drive of your machine.

In the case of a PURGE operation, it simply means that no outdated Keep Current algorithms can be found.

`Insert master device. Hit return to continue.`

This message appears during a Quick Copy operation. Place the master device into the device socket, lock it into place, and press **ENTER**. UniSite-*xpi* will then start to load RAM with data from the master device.

`Insert master device. Hit return to continue or push START lever forward.`

(Legacy) This message appears during a Quick Copy operation if you are using the SetSite module. Place the master device into device socket number one, lock it into place, and press **ENTER** or move the socket lever to the START position. UniSite will then start to load RAM with data from the master device.

`Insert New Custom Menu algorithm disk.`

The drive name of the New Custom Menu algorithm disk is displayed on the screen. Insert your new Custom Menu algorithm disk in the specified drive.

`Insert System Files disk.`

The programmer needs the System Files disk present in order to access a file which is on that disk. Insert the System Files disk.

`Insert source disk. Hit return to continue.`

This message appears during a Duplicate Disk operation. When this message appears, remove the destination disk from the disk drive, insert the source disk, and press **ENTER**.

`Insertion Err`

A device insertion error has occurred.

`Install`

This message appears when the right Module and / or Base adapter has not been installed. The device that you have selected requires that the listed Module and / or Base adapter be installed for a device operation to be performed.

`Install ChipSite or PinSite with PLCC and LCC Base`

A device operation was attempted on a device that could be programmed on a PinSite Module with a PLCC and LCC Base or on a ChipSite, but neither module was installed.

`Install ChipSite or PinSite with SOIC Base`

A device operation was attempted on a device that could be programmed on a PinSite Module with an SOIC Base or on a ChipSite, but neither module was installed.

`Install PinSite with PGA Base`

A device operation was attempted on a device that could be programmed on the PinSite Module with a PGA Base, but that module was not installed.

`Install PinSite with PLCC and LCC Base`

A device operation was attempted on a device that could be programmed on the PinSite Module with a PLCC and LCC Base, but the module was not installed.

`Install PinSite with SOIC Base`

A device operation was attempted on a device that could be programmed on a PinSite Module with an SOIC Base, but the module was not installed.

`Install PGA Base`

A device operation was attempted on a device that could be programmed on a PinSite Module with a PGA Base, but only the PinSite Module is installed. Either no Base or the wrong Base was installed.

`Install PLCC and LCC Base`

A device operation was attempted on a device that could be programmed on a PinSite Module with a PLCC and LCC Base, but only the PinSite Module was installed. Either no Base or the wrong Base was installed.

`Install PPI Base`

A device operation was attempted on a device that could be programmed on a PinSite Module with a PPI Base, but only the PinSite Module is installed. Either no Base or the wrong Base was installed.

`Install SOIC Base`

A device operation was attempted on a device that could be programmed on a PinSite Module with an SOIC Base, but only the PinSite Module was installed. Either no Base or the wrong Base was installed.

`Insufficient pin driver boards for the module/device selected.`

The device you are trying to load, program, verify, or check requires more Pin Driver boards than are installed in your programmer. Each Pin Driver board drives 4 pins. See the User Note for a table of all Device Pin / Pin Driver Board requirements.

`Insufficient pin driver boards for the module/device selected.`

The device you are trying to load, program, verify, or check requires more Pin Driver boards than are installed in your programmer. Each Pin Driver board drives 44 pins.

`Intermediate name not directory. (obsolete)`

Intermediate name not directory.

`Invalid file name.`

The filename you entered either contained characters that are not part of the programmer's character set or do not conform to the DOS file conventions. Enter a valid filename.

`Invalid medium format.`

If this message appears when you attempt to mount a disk, reformat your disk and try again.

If this message appears when you attempt to format a disk, your disk might require service. Contact your nearest Data I/O Service Center.

`Invalid operation on directory. (obsolete)`

Invalid operation on directory.

`Job file already exists.`

The filename you selected for the job file is the same as an existing one. Crete a different filename.

`Job file operation aborted due to an out of date system parameter file.`

This message indicates that a job file recording or playback operation was aborted when the user attempted to perform a parameter save or restore operation using a system parameter file that is not the same revision as the current system software. You should perform a save or restore operation while not recording or using job files, and direct the programmer to update the system parameter file when it reports that the file is out of date. After updating the parameter file, attempt the job file operation again.

`Job file playback ended.`

This message informs you that a job file's playback has ended and you may continue with the operation where the job file left off.

`Job file save aborted. Keystrokes not recorded.`

This message appears in the following situation: If you attempt to end job file recording, and either the system disk is not in the drive or UniSite-*xpi* has difficulty reading the disk, an error message will appear. If you press **Ctrl** + **Z** after seeing that message, the above message will appear.

`Key error: Hit space bar to toggle parameter or hit character desired.`

This message appears if you tried to type in a parameter using an inappropriate character, such as a letter instead of a number. Enter a valid parameter.

`Keystroke recording ended. Select job file for saving.`

This message appears after you have pressed **Esc**, **Ctrl** +**J** a second time to end recording keystrokes for a job file. Specify a job file number by typing a number between 0 and 9. Then type in a job file description.

`Keystroke recording for job file has begun.`

After you press **Esc**, **Ctrl** + **J** once, this message will appear. You are now in the job file record mode: every keystroke that you make will be recorded. Press **Esc**, **Ctrl** + **J** a second time to end the session.

`Loading data from file.`

This message appears while data is being loaded into User RAM from a disk's data file.

`Loading device algorithm.`

When you restore a set of system parameters that include a specific device, this message will appear while the programming algorithm is being loaded.

`Loading device algorithm file into user RAM.`

This message appears when the device programming algorithm is being loaded into User RAM at the first device selection operation after the RAM device Selection parameter is enabled.

`Loading device menu data.`

This message indicates that UniSite-*xpi* is loading the device and manufacturer selection files.

`Loading from disk.`

This message appears when UniSite-*xpi* is reading system information or routines from the disk.

`Loading programming parameters.`

When you restore a set of system parameters from the Configuration file directory, this message will appear while the programming parameters are being loaded.

instead of a number. Enter a valid parameter.

`Logic device must be previously selected.`

This message appears if you try to use the under/overblow feature but have not selected a logic device. Use the Verify command (if you are checking a memory device), or select a logic device and load the data to be verified.

`Logic device must be selected for logic translation.`

This error occurs when a logic type of data translator is selected while a memory type of device is selected. In order to use the logic translators, a log type of device must be selected.

`Logic translator must be selected for logic device.`

This message appears if you have a logic type of device selected and you attempt a data transfer operation without using a logic type of translation format.

`Loop count nnnn = Hit CTRL Z to abort this test.`

This message appears while a self-test is running in the continuous mode. The loop count *nnnn* is the number of times the selected test has been repeated.

`Memory parity error at:hhhhhh.`

UniSite-*xpi* has experienced an error that it cannot recover from; turn UniSite-*xpi* off and reboot the system. If the problem persists, record the location at which the error is occurring (represented by *hhhhhh*), and contact Data I/O Customer Support Office.

`Memory Part must be selected for Array Verify.`

The array verify operation only works with memory devices. Select a memory device and try the operation again.

`Missing description.`

You did not type in a description for the system parameters save area or the recorded job file. Type in a description or press the PF1 or PF2 keys to escape from the screen.

`MODULE ERROR: Base ID not recognized. Reinsert Base.`

This message appears if there is a problem with the Base. Remove the Base and inspect the interconnection pins to see if they are all intact. If the pins are OK, reinsert the Base and watch for this message to appear. If the message reappears, you are using an obsolete version of software.

`MODULE ERROR: Base mis-socketed. Reinsert Base.`

This message appears when the Base is not installed properly. Remove the Base and inspect the interconnection points to see if they are all intact. If the pins are OK, reinsert the Base and try the operation again.

`MODULE ERROR: FSM ID not recognized. Reinsert module.`

This message appears if there is a problem with the FSM module. Remove the module and inspect the interconnection pins to see if they are all intact. If the pins are OK, reinsert the module and watch for this message to appear. If the message reappears, you are using an obsolete version of software.

`MODULE ERROR: FSM mis-socketed. Reinsert module.`

This message appears if the FSM is inserted incorrectly. Remove the module and inspect the interconnection pins to see if they are all intact. If all the pins are straight and free of debris, reinstall the module, making sure the two retaining hooks are inserted into their slots. When the FSM is reinstalled, see if the message reappears. If the message reappears, the FSM may be defective. Call your nearest Data I/O Service Center for assistance.

MODULE ERROR: PSM ID not recognized. Reinsert module.

This message appears if there is a problem with the PSM module. Remove the module and inspect the interconnection pins to see if they are all intact. If the pins are OK, reinsert the module and watch for the message to reappear. If the message does not appear, you may resume normal operation. If the message does reappear, you are using an obsolete version of software.

MODULE ERROR: PSM mis-socketed. Reinsert module.

This message appears if the PSM is inserted incorrectly. Remove the module and inspect the interconnection pins to see if they are all intact. If all the pins are straight and free of debris, reinstall the module, making sure the two retaining hooks are inserted into their slots. When the PSM is reinstalled, see if the message reappears. If the message continues to reappear, the PSM may be defective. Call your nearest Data I/O Service Center for assistance.

Multiple device errors.

This message appears during a set Load, Program, or Verify operation if more than one error occurred: for example, if both a high and low Vcc misverify occurred.

Need to clear user RAM file(s) prior to this operation.

The Custom Menu algorithm building process uses the user RAM file system as a buffer. All user RAM files have to be deleted prior to a Custom Menu algorithm Create or Add operation.

Need to load device menu data. Insert Algorithm disk.

This message appears when you attempt to update the MSM algorithms before the device menu data (stored in DEV.SYS of the Algorithm disk) has been loaded into system RAM. Insert your Algorithm disk and try again.

Need to load device menu data. Insert Boot Files disk.

This message appears when you attempt to update the MSM algorithms before the device menu data (stored in DEV.SYS of the Boot Files disk) has been loaded into system RAM. Insert your Boot Files disk and try again.

No Base installed or Base ID not recognized.

This message appears when you have selected a device-dependent operation and a Base is not installed. The Load, Program, Verify, Device Select, and Device Check operations all require the presence of a Base. Install the correct Base and try the operation again.

No Custom Menu algorithm files found.

No Custom Menu algorithm files found for the selected Custom Menu algorithm operation.

No device configuration features are supported for this device.

The selected device does not contain any features which require this screen for configuration.

No disk in drive A.

There is no disk in the disk drive. Insert the Boot Files disk into drive A and try the operation again.

No FSM or PSM installed or module ID not recognized.

This message appears when you have selected a device-dependent operation and an FSM or PSM is not installed. The Load, Program, Verify, Device select, and Device check operations all require the presence of an FSM or PSM. Install the correct module and try the operation again.

No Keep Current algorithm files found.

No Keep Current Algorithms files were found for the selected Keep Current operation. If you are performing View or Delete, no Keep Current algorithm files are found.

If you are performing Replace/Restore, no restorable Keep Current algorithm files are found.

If you are performing Purge, no incompatible Keep Current files are found.

Non-blank device. Hit Return to continue, ^Z to abort.

This message appears after UniSite-*xpi* has performed a blank check on a device and has detected bits that are not in their erased or blank state, and are not illegal bits. If you press **ENTER**, UniSite-*xpi* proceeds with the Programming operation and programs over the existing data. If you press **CTRL** + **Z**, the Program screen will reappear and you can try the operation again with another device. The Blank Check parameter must be enabled before this test can be performed.

Non-blank Err

UniSite-*xpi* has detected programmed locations in this device. You may still program **over** the existing data if you wish (by pressing **ENTER** or moving the lever to START); this is merely a status message.

No old Custom menu algorithm files found.

Cannot find any Custom Menu algorithm files that are not compatible with the current system software. Check the files on your installed disk. The drive name for the old Custom menu algorithm files is displayed on the screen.

Odd Memory Begin Address is not allowed.

This message appears when the Memory Begin Address is set to an odd number and you try a device operation on a 16-bit (or larger) device. Set the Memory Begin Address to an even number and retry the device operation.

OPERATION ABORTED BY USER (review: ESC key for download?)

If you press **CTRL** + **Z** or **CTRL** + **C** to abort an operation, this message will appear. After you have aborted an operation, it must be reexecuted; the programmer will not allow you to resume the same operation.

OPERATION ABORTED: Cannot erase partial device.

The entire device size must be specified if you want to bulk-erase an EEPROM. Either set the Device Begin Address to 0000 and Block Size equal to the device's size, or disable the EEPROM erase option.

OPERATION ABORTED: Capacitor configuration error.

The capacitor configuration on your programming module is not compatible with the pinout of the device selected. A capacitor is located on a pin that is not a GND, VCC, or NC. This can result in unreliable operations. Install a programming module with capacitors located only on GND or VCC pins.

This error also will appear if there was a device in the socket when you most recently selected a device. That would prevent the programmer from accurately identifying the capacitor locations.

Additionally, if you change capacitor configurations without changing programming modules or reselecting a device, the programmer will not have identified the current capacitor locations.

If the correct test site configuration is installed, reselect the device with the socket empty and try the operation again.

Operation aborted due to HiTerm error.

When this message appears, refer to the HiTerm status line at the bottom of the screen for specific HiTerm errors.

`OPERATION ABORTED: Product security violation.`

This message appears when you have performed an operation that violates Data I/O's product security policy.

`Operation aborted. System parameter file not compatible with system software.`

This message appears when you attempt to use a system parameter file from a different version of software that is not compatible with the version you are currently running. System parameter files are transportable across most versions of software; however, the particular version you are trying to use is not transportable to your present version of software. You will need to copy the original system parameter file from the original master Boot Files disk to your working Boot Files disk. Next you should set the system parameters on the programmer to the desired values and then perform the "SAVE SYSTEM PARAMETERS" command while using your working Boot Files disk.

`OPERATION COMPLETE.`

The operation you selected has been completed; you may now proceed with other operations.

`OPERATION COMPLETE. Device = hhhhhhhh.`

This message appears after a successful Compare Electronic ID operation. *hhhhhhhh* represents the device's electronic ID.

`OPERATION COMPLETE. Sumcheck = hhhhhhhh.`

This message appears after the completion of a Program, Load, or Verify operation. *hhhhhhhh* represents the sumcheck of the data that was programmed into the device. "(Vector test not supported)" will be appended to the message when you attempt to perform a structured vector test on devices with more than 84 pins.

`OPERATION COMPLETE. Sumcheck = hhhhhhhh. Set Sumcheck = ssssssss.`

This message appears after the completion of a Set Program, Load, or Verify operation. *hhhhhhhh* is the sumcheck of data that was just programmed into the last set member. *ssssssss* is the sumcheck of all the set members that have been programmed.

`OPERATION FAILED: Cannot erase device data.`

This message appears after the programmer was not able to erase an EEPROM. The device may be defective; try another device.

`OPERATION FAILED: Data load error.`

Data could not be read from the device during a load operation. Try the operation again. If the message reappears, the device may be faulty.

`OPERATION FAILED: Device over-current fault.`

This message appears when you attempt to program a socketed device that requires more programming current than the device you selected on-screen. The device may be faulty. Insert another device into the socket and try the operation again. See page 7-42 for more information.

`OPERATION FAILED: Device programming error. Press F3.`

A device programming error is reported when a repeated attempt to program a particular cell or fuse has failed. This error may be caused by a faulty device or an improper programming algorithm. Please check the following:

- Improper Device Selected: Make sure the device selection matches the manufacturer and part number of your device as precisely as possible.
- Faulty Device(s): Program at least one more device labeled with the same date/lot code. If the operation is successful, the original device is probably faulty. If other devices with the same date/lot code fail, try to program devices labeled with different date/lot codes. If devices with different date/lot codes are programmed successfully, the devices from the original date/lot code are probably faulty.
- Specification Change: If this error occurs only on devices with recent date/lot codes, the devices may require a modified programming algorithm.

See page 7-44 for more information.

`OPERATION FAILED: Electronic ID verify. Press F3. (Elec. ID vfy error)`

When this error occurs, the electronic ID of the device is displayed. You can also determine the ID of the part by selecting Compare Elec ID in the Device Checks window. The Compare Elec ID parameter is also located in the Load, Program, Verify, and Programming Parameters windows. To display the full parameter list in the Load, Program, and Verify windows, press F4. Disabling the Electronic ID check is a viable workaround for most memory devices.
Check the following if this error occurs:

- Improper Device Selected: Make sure the device selected matches the manufacturer and part number of your device as precisely as possible.
- Faulty Device(s): If disabling the Compare Elec ID parameter causes an operation (Load, Program, Verify) to fail, try the operation on other devices with the same date/lot code. If the operation is successful on these devices, the original device may be defective.
- Manufacturer Changed Electronic ID: If the device is labeled with a recent date/lot code, the manufacturer may have placed a new electronic ID on the device. To minimize ID errors, use the latest version of your programmer software. If you are using the current version of programmer software, disable the Compare Elec ID parameter.

See page 7-46 for more information.

`OPERATION FAILED: Illegal bit error. Press F3.`

This error indicates that at least one location in the device contains data, while its corresponding location in RAM calls for not data to be programmed into the device. Check the following:

- Windowed Device: Perform a blank check on the device. If the blank check reports “non-blank device,” place the device under a UV lamp and allow sufficient time for it to be erased, then reprogram the device.
- Electronically Erasable Device: Make sure that the device’s electronic erase feature is enabled.
- One Time Programmable Device: Perform a blank check on the device. If the blank check reports “non-blank device,” your OTP device has previously been programmed and cannot be reprogrammed.
- Faulty Device: Program at least on other device with the same date/lot code. If it programs successfully, the original device was probably faulty. If other devices with the same date/lot code fail, program devices with a different date/lot code. If these program correctly, the devices with the original date/lot code are probably faulty.

See page 7-48 for more information.

OPERATION FAILED: Incompatible user data for the device selected. Press F3.

The data in RAM is the wrong size or format for the device selected. Try the following:

- Check to make sure the JEDEC file is intended for the device you have selected.
- Confirm that you have selected the correct device.
  - If both the device and JEDEC file are correct, reload the JEDEC file into RAM and attempt the programming operation again.
- If you are using a logic device, you may need to select the “cross programming” option for this manufacturer.

See page 7-52 for more information.

OPERATION FAILED: Incompatible vector data for the device selected.

This message will appear when you attempt an operation involving test vectors but the number of device pins are not equal to the number of pins represented in the vectors.

OPERATION FAILED: Partial device operations not allowed. Press F3.

This message appears if you specify part of a device for a device operation. To specify the entire device, the Device Begin Address must be 0, and the Device Block Size and User Data Size must be equal to the size of the selected device. To set the Device Block Size and User Data Size to their default values, enter 0 in their parameter fields.

OPERATION FAILED: Preload not supported or illegal preload vector.

Device does not support preload operation or an illegal test condition is in the preload vector. Either remove or modify preload test vector.

OPERATION FAILED: Security fuse programming error.

This message appears when the programmer cannot program the security fuse. If this message appears, the device you are trying to program may be defective; try programming another device.

OPERATION FAILED: Structured test vector error. Press F3.

In many cases, vector failures are due to faulty devices. To determine whether this is the cause, try programming three of the same devices. If only one device out of three has the problem, it usually means the one device is faulty. If all three fail, then this could be related to a combination of the factors stated below. It is also possible for a device to fail in the programmer and yet be a fully functional device. Failures can be caused by some of the following:

- The specific characteristics of the Logic device being tested.
- The manner in which the test vectors have been written.
- The specific design of the Logic device as created by the design engineer.
- The specific defaults used by the programmer’s hardware and firmware in applying the vectors.

To determine exactly which vector/pins are failing, select the Verify option and reverify. Sometimes these errors can be solved by using a combination of three vector “test” options: 1) High speed logic drivers, 2) Serial vector test, and 3) Compensated vector test. These options can be enabled one at a time or in any combination. A hardware option is also available. Refer to Application Note #983-0372-001.

`OPERATION FAILED: Structured test vector error (Both Passes). Press F3.`

In many cases, vector failures are due to faulty devices. To determine whether this is the cause, try programming three of the same devices. If only one device out of three has the problem, it usually means the one device is faulty. If all three fail, then this could be related to a combination of the factors stated below. It is also possible for a device to fail in the programmer and yet be a fully functional device. Failures can be caused by some of the following:

- The specific characteristics of the Logic device being tested.
- The manner in which the test vectors have been written.
- The specific design of the Logic device as created by the design engineer.
- The specific defaults used by the programmer's hardware and firmware in applying the vectors.

To determine exactly which vector/pins are failing, select the Verify option and reverify. Sometimes these errors can be solved by using a combination of three vector "test" options: 1) High speed logic drivers, 2) Serial vector test, and 3) Compensated vector test. These options can be enabled one at a time or in any combination. A hardware option is also available. Refer to Application Note #983-0372-001.

`OPERATION FAILED: Structured test vector error (First pass). Press F3.`

In many cases, vector failures are due to faulty devices. To determine whether this is the cause, try programming three of the same devices. If only one device out of three has the problem, it usually means the one device is faulty. If all three fail, then this could be related to a combination of the factors stated below. It is also possible for a device to fail in the programmer and yet be a fully functional device. Failures can be caused by some of the following:

- The specific characteristics of the Logic device being tested.
- The manner in which the test vectors have been written.
- The specific design of the Logic device as created by the design engineer.
- The specific defaults used by the programmer's hardware and firmware in applying the vectors.

To determine exactly which vector/pins are failing, select the Verify option and reverify. Sometimes these errors can be solved by using a combination of three vector "test" options: 1) High speed logic drivers, 2) Serial vector test, and 3) Compensated vector test. These options can be enabled one at a time or in any combination. A hardware option is also available. Refer to Application Note #983-0372-001.

`OPERATION FAILED: Structured test vector error (Second pass). Press F3.`

In many cases, vector failures are due to faulty devices. To determine whether this is the cause, try programming three of the same devices. If only one device out of three has the problem, it usually means the one device is faulty. If all three fail, then this could be related to a combination of the factors stated below. It is also possible for a device to fail in the programmer and yet be a fully functional device. Failures can be caused by some of the following:

- The specific characteristics of the Logic device being tested.
- The manner in which the test vectors have been written.
- The specific design of the Logic device as created by the design engineer.
- The specific defaults used by the programmer's hardware and firmware in applying the vectors.

To determine exactly which vector/pins are failing, select the Verify option and reverify. Sometimes these errors can be solved by using a combination of three vector "test" options: 1) High speed logic drivers, 2) Serial vector test, and 3) Compensated vector test. These options can be enabled one at a time or in any combination. A hardware option is also available. Refer to Application Note #983-0372-001.

`OPERATION FAILED: Verify data error.`

This message appears when the programmer has performed a Verify and has found a memory cell that did not program correctly.

`OPERATION FAILED: Verify Data Error (Both passes).`

This message appears when the programmer has performed a verify operation and has found a memory or fuse cell that was not programmed correctly. The device was verified with its lowest and highest operating voltages applied and an error was found in both. If this message appears, try another device.

`OPERATION FAILED: Verify data error (First pass).`

This message appears when the programmer has performed a Verify and has found a memory cell that was programmed incorrectly. If you selected “2” for the number of Verification passes, the device was verified with its lowest operating voltage applied. If you selected “1” for the number of Verification passes, the device was verified with its nominal operating voltage applied. If this message appears, try another device.

`OPERATION FAILED: Verify data error (Second pass).`

This message appears when the programmer has performed a Verify and has found a memory cell that was not programmed correctly. The device was verified with its highest operating voltage applied. If this message appears, try another device.

`Operation not allowed because the device was selected by fam/pincode.`

This message appears if you try to perform an operation that the programmer does not allow since the current device was selected by fam / pincode. Re-select the device by manufacturer and part name and then try again.

`Operation not supported for I/O format selected.`

The Upload and Output to Disk operations are not supported for the I / O data format selected. Select a different format in order to perform the operation desired.

`Options installed. Hit Return after changing your terminal settings.`

This message appears on the Serial Port Configuration screen after serial port parameters have been changed and **ENTER** has been pressed. When this message appears, UniSite-*xpi* suspends screen output until you press **ENTER** a second time. Make sure you configure the terminal to match the new settings for the serial port.

`Overcurrent`

An overcurrent condition was detected during the previous operation. One or more of the socketed devices may be faulty.

`Overrun error during transfer. Data sum = sssssss.`

This error indicates that the data received by the programmer caused an overrun condition on the programmer’s communication port. Make sure that the programmer’s communication parameters match those of the machine it is communicating with. If this error occurs after performing a high speed download, the communication board in your PC may not be able to transfer data at 115.2 Kbd. In this case, either disable the high speed download feature, or try a different communication board in the PC.

`Parameter Entered.`

This message acknowledges that the parameter you entered was accepted.

`Parameter field full. Hit Return or arrows to enter, ^A to abort.`

This message appears when you have tried to enter too many characters into a parameter field. Press the carriage return, the PF1 or PF2 keys, or use the arrow keys to enter the parameter.

`Parameter Field Full. Hit return or arrows to enter, CTRL Z to abort.`

This message appears when you try to enter too many characters into a parameter field. Press **ENTER**, **F1**, or **F2**, to enter the parameter.

`Parity error during transfer. Data sum = ssssssss.`

This error indicates that the parity of a received byte did not correspond with the parity selected on the programmer. Make sure that the parity settings of the programmer match those of the machine it is communicating with. If this error occurs after performing a high speed download, the communication board in you PC may not be able to transfer data at 115.2 Kbd. In this case, either disable the high speed download feature, or try a different communication board in the PC.

`PASS`

The operation just performed was successful for this device.

`Power Down.`

UniSite-*xpi* has experienced a power down condition.

`POWER UP SELFTEST FAILED: Base error.`

An error was detected with the Base module. Perform the self-test on the Base; if the error persists, contact your nearest Data I/O Service Center.

`POWER UP SELFTEST FAILED: Base socket not empty.`

If you get this error during the power-up self-test, check to make sure there are no devices in the programming socket. If this error appears some time after self-test, it could mean there is a bad Pin Driver board.

`POWER UP SELFTEST FAILED: FSM error.`

An error was detected with the FSM module. Perform the self-test on the FSM module; if the error persists, contact your nearest Data I/O Service Center.

`POWER UP SELFTEST FAILED: FSM/PSM socket not empty.`

If you get this error during the power-up self-test, check to make sure there are no devices in the programming sockets. If this error appears some time after self-test, it could mean there is a bad Pin Driver board.

`POWER UP SELFTEST FAILED: Mass Storage Module mount error. Reformat drive x.`

This message is displayed if the Mass Storage module gave an error when it was accessed. This may indicate that the Mass Storage Module requires formatting. Perform the Mass Storage format operation; if the error persists, contact your nearest Data I/O Service Center.

`POWER UP SELFTEST FAILED: No Base installed.`

If you get this error, install the Base and cycle the power (turn the programmer off, and then on again). If the Base is installed and you get this message, there may be a Controller board problem.

`POWER UP SELFTEST FAILED: No FSM or PSM installed.`

If you get this error, install the FSM or PSM and cycle the power (turn UniSite-*xpi* off and then on again). If the FSM and/or PSM are installed and you get this message, there may be a Controller board problem.

`POWER UP SELFTEST FAILED: No pin drivers installed.`

UniSite-*xpi* must have a Pin Driver board installed in connector J5, the left-most connector, in order to pass the self-test. Make sure that the Pin Driver boards are properly installed.

`POWER UP SELFTEST FAILED: PCU error.`

If this error appears, cycle the power. If the message returns, then the PCU (pin control unit) is bad, there is a bad Pin Driver board in the left-most position, or the Waveform board is faulty.

`POWER UP SELFTEST FAILED: Pin Driver error.`

If this message appears, go to the self-test screen and check to see which Pin Driver board is faulty. Call your nearest Data I/O Service Center and arrange replacement of the board.

`POWER UP SELFTEST FAILED: PSM error.`

An error was detected with the PSM module. Perform the self-test on the PSM module; if the error persists, contact your nearest Data I/O Service Center.

`POWER UP SELFTEST FAILED: System RAM failed.`

This power-up message is displayed if the power-on version of the System RAM test fails. Perform the self-test for the System RAM; if the error persists, contact your nearest Data I/O Service Center.

`POWER UP SELFTEST FAILED: User RAM failed.`

This power-up message is displayed if the power-on version of the User RAM test fails. Perform the self-test for the User RAM; if the error persists, contact your nearest Data I/O Service Center.

`POWER UP SELFTEST FAILED: Waveform board error.`

There is an error detected with the Waveform board. Call you nearest Data I/O Service Center and arrange replacement of the board.

`Pre-format check.`

This message appears when you have selected the Format Disk operation and means that UniSite-*xpi* is checking to see if the disk you want to format is a system disk.

If this error occurs, go to the self-test screen and re-execute the test(s) that show status of **F** (Fail). If the test(s) fails while the device socket is empty, UniSite-*xpi* may require service. Contact Data I/O Customer Support.

`Premature End-of-file detected.`

If file delimiters are used for data transfer, this message will indicate that the file delimiter was detected during an unexpected condition (for example, within a data field). End-of-file delimiters are non-printable ASCII characters used only with printable ASCII-type translator formats.

`Product security violation.`

This message indicates that the ID of the machine is not valid.

`Program Err`

This display indicates that a programming error has occurred during the last operation. Try another device. The socketed device may be faulty.

`Programmer set size limit (99) exceeded.`

This error message occurs during a device serial set load operation, when the set member counter has been exceeded, yet the available memory has not. The programmer limit is 99 devices.

Programming hardware hasn’t passed self-test. See Self-test screen.

This message appears if you attempt to perform a device operation (such as load, program, verify, etc.) when a self-test has previously failed. The tests must complete successfully before device operations are allowed. For these tests to be performed, the device sockets must be empty at power-up and when modules are changed. If this error occurs, go to the self-test screen and execute the test(s) that shows a “FAIL” status. If the test fails while the device sockets are empty, the module and/or the programmer require service. Contact your nearest Data I/O Service Center.

Purging filename.ext.

This message will appear if you are using the wildcard (*) designation to purge more than one file at once: for example, type **27*.dat** to delete both the files **27512.dat** and **27256.dat**.

RAM file area exhausted.

An attempt was made to create a RAM file, but it could not be done because either there was not enough RAM space remaining for RAM files, or there were not enough directory entries left for RAM files.

RAM file buffer exceeded.

A data write was attempted beyond the end of a previously allocated RAM file. The size of a RAM file cannot be changed once it is created.

RAM file create error.

An attempt was made to create a RAM file but for some reason it could not be created. Some possible reasons are: an illegal filename was used or the filename already exists.

RAM file not found.

A device operation was attempted using a RAM file, but the RAm file could not be found in memory. Ensure that the file name entered is correct and repeat the operation.

Reading user data file size.

This message appears while UniSite-*xpi* is reading the data file size from disk.

Recording system state parameters.

This message appears after you select a file number for the set of system parameters that you want to save. This message remains until UniSite-*xpi* is finished recording the parameters.

Remote port is not properly connected.

This message appears when the programmer is in transparent mode or is performing a data transfer operation, and the remote/handler port is not properly connected. The front panel LEDs are lit when the remote/handler port is properly connected.

Restore operation aborted: save are empty

This message appears when the programmer is in transparent mode or is performing a data transfer operation, and the remote port is not properly connected. The front panel LEDs are lit when the remote port is properly connected.

Restoring system state variables.

This message appears while UniSite-*xpi* is reading the recorded system variables from the selected file.

`Root directory full.`

This message appears when file operations are attempted on a medium that already has the maximum number of files that medium can hold. The maximum number of files for a double density diskette is 112, for a quad density diskette is 205, for drive C is 512, for drive D is 512, for drive H is 320, and for drive I is 320.

This message appears when file operations are attempted on a double density disk with more than 112 files, or on a quad density disk with more than 205 files.

`RTC err`

UniSite-*xpi* has experienced an error that it cannot recover from; turn UniSite-*xpi* off and reboot the system. If the error recurs, contact Data I/O Customer Support.

`RTE init err`

UniSite-*xpi* has experienced an error that it cannot recover from; turn UniSite-*xpi* off and reboot the system. If the error recurs, contact Data I/O Customer Support.

`Save operation aborted due to an out of date system parameter file.`

The “SAVE SYSTEM PARAMETERS” operation requires that the version number of the System parameter file match the system version number. You should direct the programmer to update the parameter file when it detects and reports this condition during the save or restore operations.

`Saving data to file.`

This message appears while data is being written to a file on disk.

`Saving parameters.`

This message appears when UniSite-*xpi* is saving the selected variables onto the disk.

`Saving job file.`

This message appears when UniSite-*xpi* is saving a job file.

`Search pattern not found.`

If you specified a data pattern for a file that does not contain that pattern, this message will appear. This message appears while UniSite-*xpi* is in the memory editor or in the under/overblow display.

`Security device not recognized.`

`Secur Prog Err`

An error occurred when UniSite-*xpi* attempted to program the security fuse of this device.

`Security fuse violation.`

This message appears when a device operation attempt is made on a device on which the security fuse has already been blown.

`Security fuse violation. Hit return to continue, ^Z to abort.`

This message appears when you try to program an EE device with the security fuse already blown. If you continue by pressing **ENTER**, the program operation will be performed and previous data in the device will be overwritten.

`Security Violation`

This device cannot be programmed because its security fuse bit is already set to the programmed state.

`Selection process aborted due to algorithm revision number change.`

The Boot and Algorithm disks are not the same version. Check your disks to make sure that the two match. Re-select the desired device, and the device algorithm will be loaded after the new device menu data is loaded.

`SELFTEST FAILED:`

An error was detected during the self-test.

`SELFTEST FAILED: Failure of Mass Storage Module - Drive x.`

This message appears if the Mass Storage Module fails the on-line self-test. This error may be caused by a hardware problem.

`Single drive copy overwrites user RAM. Hit Return to continue, ^Z to abort.`

If you try to perform a single drive disk copy, the programmer uses user RAM as a buffer. Therefore, anything already in user RAM will be overwritten when copying to the same drive. If you don't want the user RAM to change, press CTRL + Z.

`System error. Please contact Data I/O.`

Contact Data I/O Customer Support.

`System Files disk not found.`

When the yield tally option is enabled, the System Files disk will be searched. If the disk is not found, this message appears.

`System initialization failed.`

An error occurred while the programmer was booting. The problem may be caused by a defective disk, try another copy of the Boot Files disk.

`System parameters restored.`

This message appears when you have restored a configuration file from the Restore System Parameters screen.

`System parameters saved.`

This message appears when you save a set of system parameters.

`Task error.`

UniSite-*xpi* has experienced an error that it cannot recover from; turn UniSite-*xpi* off and reboot the system. If the error recurs, contact Data I/O Customer Support.

`Terminal port is not properly connected.`

This message will appear if the user attempts to perform an operation involving the terminal port when it is not connected properly. This may occur when the user has terminal functions assigned to the remote port and an attempt is made to perform a data transfer operation over the terminal port when nothing is connected to it. This could also be the result of the user's attempting to enter transparent mode with nothing connected to the terminal port.

`TEST ABORTED: Base not installed.`

If you selected the Adapter/Relay test from the Self-test screen, and a Base is not installed, this message will appear. Install a Base and try the self-test again.

`TEST ABORTED: Both calibration & PCU must pass before Pin driver test.`

If you selected self-tests for the Pin Driver boards, and either the calibration or the Pin Control Unit did not pass the self-test, this message will appear. Perform separate self-tests on both the calibration and the Pin Control Unit prior to testing the Pin Driver boards.

`TEST ABORTED: Both waveform board & PCU must pass before Pin driver test.`

If you selected self-tests for the Pin Driver boards, and either the Waveform board or the Pin Control Unit did not pass the self-test, this message will appear. Perform separate self-tests on both the Waveform board and the Pin Control Unit prior to testing the Pin Driver boards.

`TEST ABORTED: Calibration test must pass before PCU can be tested.`

This message appears when you have attempted to perform a self-test on the Pin Control Unit (PCU) before the Calibration test has passed. Perform the Calibration test prior to testing the PCU.

`TEST ABORTED: Diskette not installed.`

If you selected either drive A or drive B from the self-test screen, but a formatted disk is not in that drive, this message will appear. Insert a formatted disk into the drive and try the test again.

`TEST ABORTED: Drive B not installed.`

If you selected drive B from the self-test screen, but drive B is not installed, this message will appear. Install the disk drive or select drive A.

`TEST ABORTED: FSM and PSM not installed.`

If you selected a self-test for the Waveform board, Pin Control Unit, or Pin Driver board, and the PSM or FSM is not installed, this message will appear. Install on or both of the modules.

`TEST ABORTED: FSM not installed.`

If you selected the FSM test from the self-test screen, and an FSM is not installed, this message will appear. Install the module and try the self-test again.

`TEST ABORTED: PSM not installed.`

If you selected the PSM test from the self-test screen and a PSM is not installed, this message will appear. Install the module and try the self-test again.

`TEST ABORTED: Selected pin driver board is not installed.`

The Pin Driver board you attempted to test is not installed. Install the board or select a different board to test.

`TEST ABORTED: Socket must be empty to perform test.`

The device socket(s) must be empty in order to perform the self-test. Remove any device(s) which may be present in the device socket(s). If this message is being reported while the sockets are empty, the programmer hardware may be defective. Contact your local Data I/O Service Center if the problem persists.

`TEST ABORTED: Unknown adapter installed.`

This message appears if the adapter installed is not recognized by the programmer. Update your system/algorithm disk.

`TEST ABORTED: Waveform board test must pass before PCU can be tested.`

This message appears when you have attempted to perform a self-test on the Pin Control Unit (PCU) before the Waveform board has passed its self-test. Perform the Waveform board self-test prior to testing the PCU.

`TEST HALTED: Socket not empty, hit return to continue, ^Z to abort.`

The self-test that you are attempting requires that the device socket not contain any devices. Remove the socketed part and try the operation again, or type **Ctrl** + **Z** to abort the operation.

CAUTION: ***If you press the carriage return key, UniSite-*****xpi** ***will run the test, and the socketed device could be damaged.***

`Testing`

This message appears when a self-test is in progress.

`The operation was not enabled.`

This message appears when you have not entered ‘Y’ at the “Are you sure?” prompt on the screen for the operation you tried to perform. Enter ‘Y’ at the prompt. Retry your operation.

This message also appears when the operation is blocked on your programmer for data security reasons.

`Transferring data.`

This message appears while a data transfer operation is being performed.

`[transparent mode]`

This message appears on the screen when UniSite-*xpi* enters the transparent mode. To exit transparent mode, press **ESC**, **CTRL** + **T**.

`Trc init err`

UniSite-*xpi* has experienced an error that it cannot recover from; turn UniSite-*xpi* off and reboot the system. If the error recurs, contact Data I/O Customer Support.

`Too many revisions of the same Custom Menu algorithm.`

The last letter of the file extension of a Custom Menu algorithm file runs from ‘A’ to ‘Z.’ This message appears when all 26 letters have been used by the Custom Menu algorithm files of the same device.

`Unable to load algorithm. Perform the Restore System Parameters operation.`

The programmer was not able to load the device algorithm which was previously saved as part of the power up default parameters. In order for the programmer to load the algorithm, the disk containing the algorithm must be present during the boot process. In order to load the algorithm after this error, go to the Restore System Parameters screen and load the power up default parameters or reboot the programmer with the correct algorithm disk in the disk drive.

`Unable to load power up device algorithm.`

The programmer was unable to load the algorithm which was previously saved by the user as one of the power up default parameters. This can be caused by the user’s not inserting the correct algorithm disk in the drive. Ensure that the version number of the algorithm disk matches that of the Boot Files disk.

`Unknown command.`

This is a system error. Contact Data I/O.

`Unknown I/O format.`

The I/O format number selected is not recognized by the programmer. Enter a different format number.

`Updating device algorithms.`

This message appears when the algorithm disk is being updated in the Update Device Algorithms operation. Do not remove the algorithm disk or turn off system power for the duration of the operation.

`User disk space exhausted.`

There is insufficient space available on the disk to store data for the operation attempted. Delete some user files from the disk to increase the space available, or use a different disk.

`User RAM full.`

This message appears when user RAM becomes completely full during a device loading operation. Change the memory block parameters and try the operation again.

`User RAM sumcheck = ssssssss.`

This message contains the sumcheck for all of User RAM and is generated in the Sumcheck device check screen. This calculation is done regardless of whether user data is in RAM or on disk.

Using Keep Current algorithm in *filename*.KC*x*.

This message appears when the “replaced” Keep Current algorithm is used during a normal device selection operation where *filename*.KC*x* is the Keep Current algorithm file.

`Variable Vcc Verify not supported on device selected.`

This message appears when any of the Vcc Verify voltages specified in the selected algorithm are out of the range allowed by Variable Vcc Verify, which is from 3.00 to 7.00 Volts.

`Vector out of range.`

The vector you tried to select does not exist for the device you have selected. Select a vector that is within the limits of the device, or select a different device. This message may appear while you are using the vector editor.

`Verify (1st pass)`

The socketed device failed the verify test at the manufacturer’s low Vcc level or nominal Vcc level.

`Verify (2nd pass)`

The socketed device failed the verify test at the manufacturer’s high Vcc level.

`Verify (2 passes)`

The socketed device failed the verify test at manufacturer’s low and high Vcc levels.

`Waiting for self-test completion.`

This power-up message shows up only if you are changing the terminal selection before the power-up self-test has been completed.

`WARNING Algorithm disk in drive. Hit return to continue, ^Z to abort.`

This message will appear if you are attempting a file operation and have the Algorithm disk installed in the disk drive.

`WARNING: Boot files on disk. Hit Return to continue, ^Z to abort.`

WARNING: End of file exceeded.

This message may appear when the memory block size and memory begin address parameters you specified in the Programming screen are too large for the data file you intend to use for programming. When this message appears, change the memory block size and memory begin address file size parameters so they are small enough to accommodate the data file. You can perform the operation without changing anything, but only part of the device will be programmed. If this message appears during the input from disk operation, it indicates that the User data size parameter was set larger than the amount of data in the file. The User data size will automatically be set to the amount of data in the file when this warning message appears.

WARNING: End of device exceeded.

There is not enough room in the device to hold all the data you have specified. You may have the device beginning address set too high, the block size set too high, or you may need a larger device. Although the operation may still be performed, only part of the device will be programmed.

WARNING: End of file exceeded.

This message may appear when the memory block size and memory begin address parameters you specified in the Programming screen are too large for the data file you intend to use for programming. When this message appears, change the memory block size and memory begin address file size parameters so they are small enough to accommodate the data file. You can perform the operation without changing anything, but only part of the device will be programmed. If this message appears during the input from disk operation, it indicates that the User data size parameter was set larger than the amount of data in the file. The User data size will automatically be set to the amount of data in the file when this warning message appears.

WARNING: End of user RAM exceeded.

There is not enough user RAm for the amount of data you want to load into it or program from it. You may have the device block size set too large, or the beginning RAM address too high. The operation may still be performed, but only part of the device will be programmed.

WARNING: Integrity check fails. Do not use disk as Boot/System/Algorithm disk.

This message appears after a disk format operation. If the disk integrity test is not passed, this warning appears. Use a new disk.

WARNING: No device selected. Previous algorithm remains installed.

This message appears if you are attempting a programming operation and have not selected a device using the Select Device screen. This is a reminder: the device that was selected the last time the programmer was used is still in effect.

`WARNING: Partial/no transfer performed. Data sum = xxxxxxxx Press F3.`

This warning appears in the download screen and indicates that a portion of your file's data has not been transferred into the programmer RAM. Transfer problems of this nature occur more often with files that have been generated with addresses in non-sequential order, where the lowest address is embedded somewhere in the middle of the file. Check the following:

- I/O Address Offset = FFFFFFFF and first file is not lowest: To transfer data from file to corresponding RAm locations, enter an I/O address of 00000000.
- Offset Translation: If the file data must begin at the programmer RAM location 0000H, find the lowest file address and use that value as the I/O address offset.
- Improper Use of Begin RAM Address/User Data Size: Your file must be transferred to the proper locations in programmer RAM. Edit programmer RAM to determine whether data has been transferred properly. If not, make sure your Begin RAM Address and User Data Size parameters are appropriate. Refer to the Download Data section of your User Manual for definitions and usage.

See page 7-51 for more information.

`WARNING: Selection not compatible with other channel!`

This message appears if you are scrolling through the list of baud rates for one port and encounter a baud rate that is incompatible with the selected baud rate of the other port.

`WARNING: Set size limited by memory available.`

This message will appear if the set size you selected is larger than the programmer's user memory. Check the set size: take the device word width, divide that by the data word width, and multiply that result by the number of devices in the set.

`WARNING: Set size may be greater than programmer limit (99).`

This message appears when the device set's determining variables (memory block size, data word width, and device block size) indicated a set larger than 99. Adjust the variable sizes so the set number is smaller than 99.

`WARNING: System disk in drive. Hit return to continue, ^Z to abort.`

This message appears during any file operation that displaces disk data. Any information currently on the disk will be erased and is not retrievable. Press **ENTER** to go ahead with the operation. Press **CTRL** + **Z** to cancel the operation.

`Xmit checksum error. Data sum = ssss. Xmit sum = ssss.`

After a download or input from disk operation of a JEDEC file, this error indicates the calculated checksum for all the bytes received does not match the transmission checksum bytes in the file. The Xmit sum value displayed represents the calculated checksum for all the bytes in the file. The data sum represents the calculated checksum for the fusemap data bytes.

Troubleshooting Guides

On the following pages you will find a collection of Troubleshooting Guides. Each guide covers one message and explains what the message means and how to diagnose what caused the message.

Device Insertion Error When Using Elastomeric Pad

Probable Cause	Solution
Device inserted improperly	Ensure that the device is properly justified in the socket or properly oriented in the MatchBook. ***CAUTION: Do not press on the lid of the MatchBook to improve continuity. It compresses the pad, scratches and dents the base, and bends the pins on the device.***
Faulty device(s)	Check the device for bent or damaged pins / leads. Repeat the operation with similar devices from the same manufacturer, and from other manufacturers. If the operation proves successful with similar devices, then the suspect part is probably defective. If the device has been in circuit or in a test or burn in socket, it may have bent pins. If so, it is not likely that the device will be able to be programmed with an elastomeric base. A PPI adapter with a discrete socket may be required to successfully program the device.
Pad is dirty or worn	Examine the pad for debris and wear. Clean or replace the pad as necessary. Refer to documentation for cleaning and replacement instructions. A raised or discolored pad is not necessarily bad. Examine it carefully prior to changing it.
Base is dirty or worn	Separate the compression ring from the base and examine the base with a magnifying glass or microscope. If any gold has worn off or there are scratches or dents in the gold traces, replace the base.
Possible defect in programmer software associated with continuity check	If this error occurs during an attempt to load the device (via Load Device from the Main Menu), disable the Continuity Check parameter (change from “Y” to “N”) in the Programming Parameters screen (via More / Configure system / Edit / Programming options from the Main Menu). If the device is loaded successfully without insertion errors, try to program the device. If the device programs successfully, you’ve found a reasonable workaround. *Note: Contact Data I/O Customer Support and report your findings.*
Continuity problem with device / programmer interface	If following the steps described in the previous section causes the device to fail programming, a subtle continuity problem may exist. **Workaround:** Refer to your programmers’ Device List and note the earliest version of the device that is supported by your programmer. Boot your programmer with any previous software version that supports the device and attempt the operation again. If the operation is successful with the earlier software, then you’ve found a temporary workaround. *Note: Contact Data I/O Customer Support and report your findings.*

Additional Information

The “Device insertion error” message can only be caused by a failure of the continuity check. The continuity check is activated prior to device programming. During the continuity check, the programmer applies low level current to each pin on the device to determine whether it is making good contact with the programming fixture.

After you disable the continuity test, we suggest you load a device rather than program one. A load operation is less apt to harm the device because no programming voltages are applied. Attempts to program a device that is not making proper contact may result in overcurrent or programming errors which may damage the device.

Device Over-current Fault

Probable Cause	Solution
Improper device selected	Make sure the device selection matches the manufacturer and part number of your device as precisely as possible. If it doesn't, select the proper characteristics and perform the operation again. *Note: Choosing a wrong manufacturer and/or part number (via the Select device option from the Main Menu) causes the programmer to expect an electronic ID different from your device's ID.*
Faulty device(s)	Load the suspect device (using the Load device option from the Main Menu). If an overcurrent error occurs, load new devices. If other devices load with no errors, a single device may be faulty. If no errors occur during load operation, try to program another device (using the Program Device option from the Main Menu). If other devices program without error, a single device may be faulty. If other devices also produce overcurrent errors during load or program operations, check the date code. If failures occur only on devices with a particular date code, while other parts are programmed or loaded successfully, the problem is probably device related. *Note: If the problem appears to be device related, you may wish to notify the device manufacturer.*
Possible bug in software associated with device tests	If other devices produce overcurrent error during load a program operations, disable the Continuity check parameter displayed on the Programming Parameters screen (via More / Configure system / Edit / Programming options from the Main Menu). If this error occurs during a programming operation but not during a load operation, a device test may be causing the problem. Disable all device checks (such as Device check and Illegal bit check) listed on the Program Device screen by changing the appropriate fields from "y" to "N." Press the F4 key to display all parameters.
Improper algorithm applied by programmer due to possible bug in software	If this error occurs on devices with old and recent date codes, there may be a software bug in the device's programming algorithm. **Workaround:** On the programmer's Device List, find an earlier version of programmer software that supports your device, boot your programmer with this version, and repeat the operation. If the operation is successful, you've found a temporary workaround. *Note: Contact Data I/O Customer Support and report your findings.*
Programmer hardware problem	An overcurrent error occurring with different devices may indicate a hardware problem. To determine whether a hardware malfunction exists, perform a self-test (using the More / Self-test options from the Main Menu) with no device sin the socket. If the self-test shows a malfunction, make the necessary service arrangements.

Additional Information

This error is reported by the hardware overcurrent detection circuitry on the programmer. The trip level for the overcurrent error is set by the programming algorithm. From the error alone, it is not possible to determine which operation the programmer was performing (device tests, program, verify / read) when the overcurrent condition was detected. To determine the nature of the problem, you need to isolate the operation that is being performed.

Device Programming Error

Probable Cause	**Solution**
Improper device selected	Make sure the device selection matches the manufacturer and part number of your device as precisely as possible. If it doesn't, select the proper characteristics and perform the operation again. *Note: Choosing the wrong manufacturer and/or part number (via the Select device option from the Main Menu) causes the programmer to expect an electronic ID that differs from the ID in you device.*
Faulty device(s)	Program at least one more device labeled with the same date code. If the operation is successful, the original device is probably faulty. If other devices with the same date code fail, try to program devices labeled with different date codes. If devices with different date codes are programmed successfully, the devices with the original date code are probably faulty. *Note: You may wish to contact the device manufacturer and report your findings.*
Improper algorithm applied by programmer due to recent change in manufacturer specifications	If this error occurs only on devices with recent date codes, the devices may require a modified programming algorithm. *Note: You may wish to contact the device manufacturer to determine whether the programming specifications for the device have changed. If they have, please notify Data I/O Customer Support.*
Improper algorithm applied by programmer due to possible flaw in programmer software	If this error occurs on devices spanning old and recent date codes, it may indicate an algorithm-related problem in your programmer software. **Workaround:** On the programmer's Device List, find the earliest version of programmer software that supports your device, boot your programmer with this version, and repeat the operation. If the operation is successful, you've found a temporary workaround. *Note: Call Data I/O Customer Support and report your findings.*
Additional Information	A device programming error is reported when a repeated attempt to program a particular cell or fuse has failed. This error may be caused by a faulty device or an improper programming algorithm.

Invalid Device ID on Logic Device

Probable Cause	**Solution**
Improper device selected	Make sure the device selection matches the manufacturer and part number of your device as precisely as possible. I it doesn't, select the proper characteristics and perform the operation again. Note: *Choosing the wrong manufacturer and/or part number (via the Select device option from the Main Menu) causes the programmer to expect an electronic ID that differs from the ID in your device.*
Device manufacturer has changed the device ID of the part, and the programmer doesn't recognize it	If the device is labeled with a recent date code, the semiconductor manufacturer may have assigned the device a new ID that is not recognized by your programmer. To minimize ID errors, use the latest version of your programmer software. Note: *If your programmer is at the current version, you may wish to contact the device manufacturer to determine if they changed the device's ID. If they did, contact Data I/O Customer Support.*
Faulty device(s)	Attempt the operation with at least one more device labeled with the same date code. If the operation is successful, the original device is probably faulty. Note: *If a high percentage of parts produce ID errors, you may wish to contact the device manufacturer and report your findings.*
Possible bug in programmer software	A high percentage of parts failing with ID errors may also indicate that a bug exists in the programming algorithm software associated with the part. **Workaround:** On your programmer's Device List, note the earliest version of programmer software that supports the device, boot your programmer with this version, and repeat the operation. If the operation is successful, you've found a temporary workaround. Note: *Contact Data I/O Customer Support and report your finding.s*
Additional Information	Most logic devices are uniquely identified by their device IDs. Semiconductor manufacturers issue new IDs to reflect changes in manufacturing processes that may produce modifications to device programming algorithms. Data I/O engages in ongoing communication with semiconductor manufacturers and is usually informed when device IDs change.

Electronic ID Verify Error on Memory Device

Probable Cause	Solution
Improper device selected	Make sure the device selection matches the manufacturer and part number of your device as precisely as possible. If it doesn't select the proper characteristics and perform the operation again. Note: *Choosing the wrong manufacturer and/or part number (using the Select device option from the Main Menu) causes the programmer to expect an electronic ID that differs from the ID in your device.*
Device manufacturer has changed the electronic ID of the part, and the programmer does not recognize it	If the device is labeled with a recent date code, the manufacturer may have placed a new electronic ID on the device that is not recognized by your programmer. To minimize ID errors, use the latest version of your programmer software. **Workaround:** If your programmer is at the current version, disable the Compare elec ID parameter (change from "Y" to "N") under the Load Device, Program Device, Verify Device, or Programming Parameters screen. Note: *You may wish to contact the device manufacturer to find out if they have changed the ID on the device. If they have, please notify Data I/O Customer Support.*
Faulty device(s)	If disabling the Compare elec ID parameter causes an operation such as load, program, or verify to fail, try the operation on other devices with the same date code. If the operation is successful on these devices, the original device may be defective. Note: *If a high percentage or parts fails the operation, you may wish to contact the device manufacturer and report your findings.*
Possible bug in programmer software	If disabling the Compare elec ID parameter causes the operation to fail on a high percentage of parts across several date codes, there may be a software bug in the programming algorithm associated with the part. **Workaround:** On the programmer's Device List, find the earliest version of programmer software that supports the device, boot your programmer with this version, and repeat the operation. If the operation is successful, you've found a temporary workaround. Note: *Contact Data I/O Customer Support and report your findings.*

Additional Information

Most memory devices are uniquely identified by their silicon signatures (electronic IDs). At times, device manufacturers change the electronic IDs of devices that have undergone changes in the manufacturing process. Typically, the electronic ID is altered to promote automatic device selection and usually does not reflect a change in the device programming specifications. Consequently, disabling the electronic ID check is a viable workaround for most memory devices.

When this error occurs, the electronic ID of the device is displayed. You can also determine the electronic ID of your part by selecting the Compare Elec ID Device checks option (using the More / Device checks option from the Main Menu).

The Compare elec ID parameter is located in the Load Device, Program Device, Verify Device, and Programming Parameters screens. To display the full parameter list under the Load Device, Program Device, and Verify Device screens, press the F4 function key.

Illegal Bit Error

Probable Cause	Solution
Windowed device contains data	Perform a blank check on the device (using the More / Device checks options from the Main Menu) to determine whether it is truly blank. If blank check reports “non-blank device,” place the device under a UV lamp and allow sufficient time for it to be fully erased. After erasure, reprogram the device.
Electrically erasable (EE) device contains data, and device’s electronic erase feature is not enabled	Maker sure that the device’s electronic erase feature (Erase EE Device parameter under the Program Device screen) is enabled (set to “Y”). Reprogram the device after the electronic erase feature has been enabled. *Note: To display all parameters on the Program Device screen, press the F4 function key.*
One Time Programmable (OTP) device contains data	Perform a blank check on the device to determine whether the device contains data. If so, your OTP device had previously been programmed and most likely cannot be over-programmed with different data or fuse pattern. Program another device.
Faulty device(s)	Program at least one more device labeled with the same date code. If it programs successfully, your original device was probably faulty. If other devices with the same date code also fail, attempt to program devices labeled with different date codes. If these devices program successfully, the devices from the original date code are probably faulty. *Note: You may wish to contact the device manufacturer and report your findings.*
Improper algorithm applied by programmer due to recent change in manufacturer specifications	If this error occurs only on devices with recent date codes, the devices may require a modified programming algorithm. *Note: You may wish to contact the device manufacturer to determine whether the programming specifications for the device have changed.*
Improper algorithm applied by programmer due to possible bug in programmer software	This error, if it occurs on devices with old to recent date codes, may indicate an algorithm-related problem in your programmer software. **Workaround:** On your programmer’s Device List, find the earliest version of programmer software that supports the device, boot your programmer with this version, and attempt the operation again. If the operation is successful, you’ve found a temporary workaround. *Note: Contact Data I/O Customer Support and report your findings.*

Additional Information

An “illegal bit error” indicates that at least one location in the device contains data (programmed state) while its corresponding location in RAM has no data (unprogrammed state). For example, the unprogrammed state of a PROM is 0, while its programmed state is 1. If a particular 8-bit PROM’s memory locations contains 09 hex (00001001 binary), and the corresponding memory location in RAM contains F0 hex (11110000 binary), then an “illegal bit error” will occur because the programmer is not able to unprogram the first and fourth least significant bits.

I/O Timeout Error

Probable Cause	**Solution**
Wrong download command sent to host	Your host machine (PC-DOS, Sun, VAX, etc.) will transfer a file upon receipt of the proper command. Under HiTerm, for example, the Download Host Command must begin with "tr" or "transfer" followed by the appropriate drive letter, path, and filename. Refer to Chapter 4, "Commands," the *HiTerm User Manual*, or your host machine's documentation for more information.
Wrong I/O translation format code selected	The format of the file being transferred must match the description in this User Manual. If it doesn't, enter the proper I/O translation format number and transfer the file again.
Unrecognized characters in beginning of file	The data file must begin with characters that match the appropriate format described in this User Manual. Remove any characters in the data file that the programmer will not recognize. In general, ensure that the format of the data file conforms to the description in Chapter 6, "Translation Formats."
No recognizable end-of-file character or record in file	The data file must end with the proper end-of-file character or record, as described in Chapter 6, "Translation Formats," under the I/O translation format type selected. Add the end-of-file character or record, if it is missing. Of course, this does not apply to binary files, which have no end-of-file character or record.
Additional Information	This error occurs when a file is being transferred from systems such as a PC, Sun, or VAX over RS-232, or when transfer is occurring via the programmer disk drive (using More/Transfer/Input from disk). If the programmer displays `data sum = 00000000`, no data was transferred. Other hex values indicate the file transferred partially or completely. The I/O translation format parameter is selected in the download screen. All I/O translation formats and their corresponding codes are listed in Chapter 6, "Translation Formats." To tell which format your data file corresponds to, view the file with an ASCII editor (or hex editor, if the file is binary).

Partial or No Transfer Performed

Probable Cause	Solution
I/O Address Offset = FFFFFFFF and first file address is not the lowest	Absolute translation: To transfer data from file to corresponding RAM locations, enter an I/O Address Offset value of 00000000. Offset translation: If file data must begin at the programmer RAM location 0000H, find the lowest file address and use that value as the I/O Address Offset.
Improper use of Begin RAM Address and/or User Data Size	Your file must be transferred to proper locations in programmer RAM. Edit Programmer RAM to determine whether data has been transferred properly. If it hasn't, make sure your Begin RAM address and User Data Size parameters are appropriate. Refer to the Download Data section of your User Manual for definitions and usage.
File addressing places data outside RAM address range	View your file with an ASCII editor and look for addresses that exceed your programmer's RAM address range. To convert the decimal value of your programmer RAM size to its hex equivalent, refer to the "Data I/O Memory Chart" Application Note.
Additional Information	This warning message appears in the download screen (reached via More/Transfer/Download from the Main Menu) and indicates that a portion of your file's data has not been transferred into programmer RAM. Transfer problems of this nature occur more often with files that have been generated with addresses in non-sequential order, where the lowest address is embedded somewhere in the middle of the file. **More About I/O Address Offset:** In general, the following formula represents where in programmer RAM data will be transferred. **Physical RAM address = [(File Address) - (I/O Offset Address)] + (Begin RAM Address)** The default I/O Address Offset, FFFFFFFF, does not represent a numerical hex value. It is simply a flag to indicate that the first address in your file will be used as the I/O Address Offset. By default, the programmer interprets the first file address as the I/O Address Offset and subtracts that value from all of the remaining addresses in the file. Consequently, the data contained in address locations lower than the first address will be lost.

Incompatible User Data File for Device Selected

Probable Cause	Solution
The wrong device is selected in the programmer	Select the correct device in the programmer menu.
The wrong device is selected in the development tool	Recompile the source file with the correct device selected in the development tool.
The user has transferred the .JED file from the drive on the programmer using the More / File Operations / Load file command	Use the More / Transfer data / Input from Disk command and program the device from RAM. *Note: The More/File Operations/Load File command is a transfer for an absolute binary format. The programmer must receive a JEDEC format for logic devices.*
The user is attempting to program the device from disk and the disk data is still in the .JED format	Use the More / Transfer Data / Input from Disk command and program the device from RAM.

QF and QP fields

When programming a device, the programmer looks at the QF and QP fields in the JED file. If the fields do not contain one of the acceptable values, the programmer returns the error. QF is for the number of fuses in the device. QP is for the number of pins in the device.

Calculating the Number of Fuses in a Device

There are three ways of determining the number of fuses in a device:

1. Review the device manufacturer’s specification for the number of fuses.
2. Using the programmer in terminal mode, perform the following:
   a. Select the device.
   b. Choose More commands / Edit Data / Fill Fuse Map.
   c. Choose More commands / Edit Data / Edit Logic.
   d. Page down to the end of the fuse map. Note the number of the last fuse and add 1 to it (fuse numbers start at 0). Compare it to the QF field in the JED file.
3. Using the programmer in terminal mode, perform the following:
   a. Select the device.
   b. Choose More commands / Edit Data / Fill Fuse Map.
   c. Choose More commands / Transfer Data / Upload Data.
   d. Enter the file name of the .JED file to be created.
   e. View the new JED file with the text editor. Locate the QF and the QP fields and compare them with the QF and QP fields in the failing JED file.

Glossary

Action Symbol Found in the upper left corner of the screen, the action symbol rotates to indicate that the programmer is performing an operation.

Address A coded instruction designating the location of data or program segments in storage.

Address Offset A value subtracted from addresses during input translation and then added to addresses during output translation.

Algorithm The software file containing information to program a specific device, usually contained on a floppy disk.

All Parameters Also referred to as the Complex Parameters screen, the All Parameters screen displays all applicable parameters on a parameter entry screen.

Approval Indication that a device manufacturer has tested an algorithm to support a specific device on a programmer. The level of an approval varies by device manufacturer, but an approval usually indicates both yield and waveform analysis.

AutoBaud A special feature on the programmer that senses the baud rate of equipment connected to the programmer and sets the programmer's baud rate to match the equipment's baud rate. AutoBaud is designed for equipment that cannot support baud rates as high as 9600 baud, such as 1200/2400 baud modems.

Base The interface between the module (PinSite) and the device. The Base routes the signals between the device and the Universal Pin Drivers inside the programmer.

Baud Rate A measure of data flow. The number of signal elements per second based on the duration of the shortest element. When each element carries one bit, the Baud rate is numerically equal to bits per second.

BGA An acronym for Ball Grid Array, a type of device package. Usually a square device with one side populated with small solder balls as leads.

Blank Check A device check that checks a device for programmed bits. If no programmed bits are found, the device is considered blank.

Block Size The hexadecimal number of bytes to be transferred in a data transfer. The beginning of the block is defined by a begin address, and the end of the block is the sum of the block size and the begin address minus one.

Byte Swap See Odd/Even Byte Swap.

Cards Also called PC Cards, these are 68-pin, integrated circuit cards. PC Cards usually follow the PCMCIA/JEIDA (PC Memory Card International Association) standards, which govern 68-pin memory or I/O type cards.

Checksum The sum of all the bytes, expressed in hexadecimal format.

Command Window The left side of the screen. At the top of the window is the menu name, displayed in uppercase letters. Below the menu name, the available commands are displayed in upper- and lowercase characters.

Communications Parameters The various settings that determine the I/O characteristics of your equipment. The parameters include baud rate, stop bits, data bits, and handshaking.

Compare Electronic ID A command that compares the electronic signature of the socketed device with the electronic signature specified in the currently selected algorithm.

Compensated Vector Test A device test that enables load compensation on PLD output pins under test during vector testing. This may eliminate structured test errors when testing PLDs sensitive to output loading, where many of the devices register transitions simultaneously.

Complex Parameter Screen See All Parameters.

Computer Remote Control A command set that may be used to operate a programmer remotely. These commands are usually the basis for external programmer drivers, which may operate a programmer from a PC or other host. See also Remote Mode.

Context-sensitive Information that changes depending on the cursor position. With the programmer, every time you move the cursor to a different field, the information on the online help screen changes.

Continuity Check A device check that tests for open device pins before performing a device operation.

CRC An acronym for Computer Remote Control. See Remote Mode.

Cross Programming A programming operation that allows a single generic programmable logic device (PLD) to be configured as any one of many PLD architectures. Consequently, the generic device can take on the function of many subset devices. For example, a 16V8 generic PLD can be configured as a 16R4, 16R8, or 16L8.

Data Bits A communication parameter that specifies the number of bits per byte.

Data Representation The manner in which the data in user memory appears on the screen. You can select either **X** and **-**, or **0** and **1**, where **X** and **0** represent an unprogrammed state, and **-** and **1** represent a programmed state.

Data Word Width The word width of the data to be used during a device operation. For 8-bit (or above) devices, the maximum is 64, and the minimum word width is equal to the device width. For 4-bit devices, the word width can be 4, 8, 16, or 32. This value should match the word width of the data bus in the target system for the device being programmed.

Destination The place where you are sending something. The “something” you are sending to the destination is almost always data. The destination you are sending the data to can be RAM, a disk file, or one of the programmer’s serial ports.

Device Begin Address The first hexadecimal address of device data to use for a device operation. If programming, it represents the first address to program. If verifying, it represents the first address to verify.

Device Block Size The size of device data to be used in device operations.

Device Operation Usually a term that refers to loading, programming, or verifying. However it can also refer to other available commands, such as device checks and electronic erasing.

Device Word Width The number of bits in the data word of the device.

DIP An acronym for Dual In-line Package, a type of device package. **SDIP** (Shrink Dual Inline Package) is similar to DIP but with more leads at tighter pitch (a separate adapter is required to program SDIPs).

Dialog Window The largest window on the screen. The dialog window displays different information and system parameters, depending on the selected command.

Download Data A file operation that moves a data file from a host computer to the programmer’s RAM or disk.

Download Echoing Displays the data being downloaded.

Download Host Command A command that is sent from the programmer to the host during a download. The command tells the host to begin sending data to the programmer.

E-MICRO An acronym for Programmable Microcontroller, a type of device technology.

EPROM An acronym for Erasable Programmable Read-Only Memory. (Usually refers to UV erasable memories.)

EEPROM An acronym for Electronically Erasable Programmable Read-Only Memory. The device can be either completely or partially erased electrically in circuit or on the programmer.

Electronic ID The combination of bytes that identify the device number and manufacturer of a programmable device.

Enhanced Security Fuse Capability Found on EMICROs, the Enhanced Security Fuse Capability allows security fuse data to be stored in a data file. For more information, or to see if a device supports this capability, see the device manufacturer’s data book.

ESD An acronym for Electro-Static Discharge.

Extended Algorithm

Extended algorithms (algorithms contained in the alg.ext file) are used by Data I/O to handle device approvals and special device algorithm updates.

False Positive

In programming, a misprogrammed fuse that retains minimal operational characteristics so that it passes the fuse test. These may be inadequately programmed, or over-programmed so that they will fail later in circuit.

File Transfer Operations

An operation involving the transfer of data between the programmer and a host. Upload and download are file transfer operations.

Filename

The name of the disk file to use during file operations. The filename must follow standard DOS conventions: up to eight alphanumeric characters, followed by an optional three-character file extension, with the two fields separated by a period. An example of valid filenames would be **27256.dat** or **filename.c**.

Fuse Verification

A type of post-programming device check that checks the fuse pattern programmed into a logic device with the pattern in user memory.

Fusemap

The fuse-level description portion of a programmable integrated circuit. Fusemaps are typically files in JEDEC Standard #3A and are downloaded to PLD programmers for device implementation.

Handshaking

The required sequence of signals for communication between two units. The I/O bus protocol for a unit defines its handshaking requirements. This is especially true for asynchronous I/O systems in which each signal requires a response to complete an I/O operation.

High-speed Download

A special feature of the programmer that allows the programmer to download data from a PC at 115.2K baud. To take advantage of this feature, you must be using HiTerm on a PC, you must enable the High Speed Download parameter, and you must send User Menu Data to the Remote port.

High-speed Logic Drivers

A device test that increases the speed of the logic transitions between 0 to 1 and 1 to 0 of the test vector input states. This test is a diagnostic tool designed to help debug and classify test vector failures. Specifically, this test is designed to help identify vector transitions that are speed dependent.

Host

A micro, mini, or mainframe computer used to control the programmer in Remote mode. You must use a software driver, such as Data I/O's PROMlink to allow the computer to communicate with the programmer.

Host Command (download & upload)

The command that is sent from the programmer to the host system during uploading/downloading. See Download Host Command and Upload Host Command.

I/O Address Offset

This value influences the beginning address where data is stored during a file transfer operation. For uploads, the I/O Offset represents the address to start loading a formatted data file. For downloads, the I/O Offset is subtracted from the beginning address in the formatted data file. The result is then added to the memory begin address to determine where the block of data is loaded.

I/O Timeout The amount of time that the programmer will wait for a data transfer to begin.

I/O Translation Format See Translation Formats.

Illegal Bit An illegal bit occurs when a device contains a programmed location and the data file specifies that the location should be unprogrammed.

Illegal Bit Check A test that determines whether or not a socketed device contains any illegal bits.

Instrument Control Code A 1-digit number that signals or controls data transfers. It also implements a form of remote control that provides peripherals with flow control beyond that provided by software handshaking.

JEDEC Joint Electron Design Engineering Committee: a committee of programmer and semiconductor manufacturers that provide common standards for programmable issues. Examples include: acceptable test characters for PLDs and standard data transfer / programming formats for PLDs. JEDEC Standard #3 is the industry standard for PLD formats.

JEDEC Standard #3A The standard PLD data translation format, as defined by JEDEC for PLD design software to communicate with PLD programmers. It defines the states of all fuses in the device (the fusemap) and may include test vectors for DEVICE testing.

JEDEC I/O translate DIP/ LCC Vectors A feature on the programmer that translates test vectors for a device from its DIP package to its PLCC / LCC package, allowing for the different pinouts of the two package types.

JLCC An acronym for J-style Leadless Chip (or Ceramic) Carrier, a type of device package. A device with hooked leads that are “open” at one end (leads are usually on all four sides).

Job File A sequence of keystrokes that have been stored in a disk file and that can be played back at a later time.

Keep Current Subscription Service A one-year renewable subscription to keep your product and documentation up-to-date with the latest features and device support. A Keep Current algorithm is an algorithm that has been downloaded from the Keep Current area of the Data I / O Web site.

LCA An acronym for Logic Cell Array.

LCC An acronym for Leadless Chip Carrier, a type of device package. A 4-sided ceramic package with pads on the underside for surface mount applications.

LED An acronym for Light Emitting Diode. The programmer has five LEDs: four on the top cover and one on the disk drive.

Load Data A device operation that moves device data into the programmer. You can load the programmer with data from a device, from the programmer’s internal disk drive, or from a serial port (for example, from the Remote port).

Load Device A device operation that copies data from a master device into User memory.

Logic Verification After programming a device, you can select test vector verification, fuse verification, or both types of verification.

Master Device A device that contains data you wish to program into another device. For example, you would load data from a master device and then program that data into a blank device.

MatchBook A new type of socketing technology that makes handling surface-mount devices easier.

Memory Begin Address The first address, in hex, of the first byte of data to be used in device operations. If the data source / destination is RAM, the memory begin address is a RAM address. If the data source / destination is disk, the memory begin address is the offset for a disk file.

Message Bar The fourth line of the screen. The programmer displays system and error messages in the message bar. The action symbol is also located in the message bar.

Next Device Used during serial set programming, this value specifies the next device in the set. For example, if you are using 8-bit devices and have specified a word width of 16 bits, it will require two devices to store each 16-bit word. Depending on the value entered, the data programmed into the next device will come from either even addresses or odd addresses.

Non-default Parameters Also referred to as the Simple Parameter screen, the Non-default parameters screen displays a selected group of parameters on a parameter entry screen. To display all the available parameters, select the All Parameters screen, which is also referred to as the Complex Parameters screen.

Odd/even Byte Swap Used during device operations for 16-bit devices, this option swaps the Most Significant Bytes (MSB) and the Least Significant Bytes (LSB) of 16-bit words. The programmer stores RAM data and disk file data with the convention that the LSB of a 16-bit word resides in the even byte of memory.

Online Help Available throughout the programmer menus, the help screens provide you with both general help and context-sensitive help. The Help screen is divided into four sections: the key listing, the general help, the specific help, and the reminder bar.

Output Record Size The number of data bytes contained in each data record during upload.

Overblow A condition in which fuses are blown that should not have been.

Overblown Fuse A fuse that has been over-programmed such that the surrounding area may have been damaged or such that fuse material splatter was created. Splatter (or rattlers) can cause intermittent shorting.

PAL An acronym for Programmable Array Logic. PALs are devices with programmable AND and fixed OR arrays. This is a slightly different architecture from a PROM or an FPLA. Other examples of PAL-type architectures from other manufacturers include PEEL and GAL.

Parallel Test Vector Application

Use of internal registers to hold and release a full set of test vectors (e.g., 20 for a 10-input 10-output device) at once. In contrast to serial application, parallel application does not require accommodations for clocking contention, and parallel better matches in-circuit PLD operation and board test suites.

Part Number

The number on the device. For example if you are using an Intel 27C256, then the part number of the device is 27C256.

Pin-driver

The electric circuit reading or applying voltage and current pulses to the individual pin of a device, for programming or testing. See also Universal Pin Driver.

PGA

An acronym for Pin Grid Array, a type of device package. Usually a square device with one side populated with small pins as leads.

PLCC

An acronym for Plastic Leaded Chip Carrier, a device package with J-shaped leads extending from four sides downward, used for surface mount applications.

PLD

An acronym for Programmable Logic Device, a type of programmable integrated circuit. Architectures range from very simple to very complex. Most PLDs contain two levels of logic: an AND array followed by an OR array.

PROM

An acronym for Programmable Read-Only Memory. A device with fixed AND and programmable OR arrays. This is a slightly different architecture from an FPLA or a PAL.

Program

The controlled application of electrical pulses to program specific fuses or cells.

Program Device

A device operation that copies device data into a socketed device. The programming is done according to the programming algorithm selected in the select device stage. The programming operation can also include a verify operation.

Program Security Fuse

A programming parameter that enables / disables the programming of the device’s security fuse.

Program Signature

Available on only a few devices, the Program Signature is a user-definable field that allows the user to program data into the program signature array. For example, the Program Signature could contain the revision level or modification date of the data in the remainder of the device.

Programmable Integrated Circuit

One of the four basic categories of ASICs: the other three being gate arrays, standard cells, and full custom devices. PICs are ICs that are user configurable. PLDs and PGAs are examples of programmable integrated circuits.

QFP

An acronym for Quad Flat Pack, a type of device package. A square or rectangular device with leads on all four edges (leads can be either straight or gull-wing). **BQFP** (Bumper Quad Flat Pack) is a QFP with “bumpers” on the corners to prevent damage to leads. **CQFP** is a Ceramic QFP or a QFP with carrier. **Long Lead Carrier QFP** is an ACTEL product with long leads attached to a carrier grid for handling ease. **PQFP** (Plastic carrier QFP) is a QFP with a plastic protective carrier for handling ease. **QFPCAR** is a QFP in a carrier. **TQFP** is a thin QFP.

QUIP

An acronym for Quad Inline Package, a type of device package. Similar to DIP, but with staggered long & short leads (leads on long sides).

Reboot

The process of re-initializing the programmer. After rebooting, the programmer is in the same state as if it had just been turned on.

Registered Devices

Devices that contain registers, rather than being combinatorial only. Registered devices are typically used for sequencers and state machine designs. Typical examples are 16R8, 82S159, and 22V10.

Reject Option

A post-programming device check that pulses the programmed device with voltage to see if the device has programmed per specification. The number of times a device is pulsed varies by manufacturer and by the reject option you select.

Reminder Bar

The bottom line of the screen. The reminder bar tells you what function keys are available and what they will do if pressed.

Remote Mode

Operating mode in which the programmer is controlled from a host running a driver program. Device data files can be stored on the programmer’s disk and on the host.

Security Fuse

A location in a programmable device that, when programmed, secures the device from readback: the data in the device is unreadable.

Security Fuse Data

The actual data to program into the device’s security fuse.

Select Device

A procedure that tells the programmer what device you will be using. You can select a device by selecting the manufacturer and the device part number.

Self-test

A built-in self-diagnosis command that allows you to test various circuits and subsystems in the programmer, verifying proper operation or isolating possible problem areas.

Serial Set

A method of set programming in which the devices of the set are programmed one at a time instead of all at once.

Serial Test Vector Application

The process of applying test vectors in a serial fashion, one input at a time.

Serial Vector Test

A device test that applies test vector input states serially, starting with pin one and stepping through the remaining pins. This test is a diagnostic tool designed to help debug and classify test vector failures. Specifically, this test is designed to isolate test vectors that are sequence dependent.

Set Programming

A type of programming where a large data file is partitioned and programmed into multiple memory devices.

SIMM

An Acronym for Single Inline Memory Module, a type of device package. A rectangular device with leads on one long edge.

Simple Parameter Screen See Non-default parameters.

SmartPort A feature of the programmer that automatically detects and adjusts the programmer to the presence of DCE / DTE protocol.

SOIC An acronym for Small Outline Integrated Circuit, a type of device package. A rectangular device with gull-wing leads on the long sides. **SOP** is Small Outline Package. **SO** is Small Outline.

Source The place from which something comes. The "something" the source is sending is almost always data. The source can be RAM, a disk file, or one of the programmer's serial ports.

SSOP An acronym for Shrink Small Outline Package, a type of device package. A SOP with more leads at higher and finer pitch. Also called TSOP II.

Status Window The top three lines of the screen. The following information is displayed in the status window: the name of the data file, the amount of user RAM, the version numbers of the algorithm / system disk, the device manufacturer and part number, the family / pinout code, and the data translation format.

Structured Test Vectors A string of test conditions applied to a PLD in a programmer / tester to stimulate inputs and test outputs to ensure functionality. A test vector is one such string. For instance, 20 characters for a 20-pin PLD, with 10 input signals and 10 expected outputs.

Structured Test Vectors (design) Structured vectors created by the design engineer to confirm that the design is operating as intended. For instance, that a 10-bit counter is counting to 10. Design vectors are used both in preprogramming simulation and in manufacturing.

Structured Test Vectors (device) Structured vectors created by the design engineer, test engineer, or an automatic test vector generation program, which confirm that the device is operating properly after programming. For instance, structured vectors can ensure that nothing can happen in the device to prevent the 10-bit counter from operating correctly. An exhaustive set of device vectors will assure that no undetectable faults may occur.

Sumcheck A 4- or 8-digit hexadecimal number that, when compared to the original data, allows you to verify that a copy of the data matches the original data. Memory devices have 8-digit sumchecks and logic devices have 4-digit sumchecks. For devices in a set, you can calculate the individual sumcheck of the device and the sumcheck of the entire set.

Terminal Emulator A program to enable a PC or other computer to act as an ASCII terminal. Allows a PC to be used to communicate with a programmer in terminal mode or with a mainframe.

Terminal Mode One of the programmer's three operating modes. The programmer is controlled from either a dedicated terminal or from a workstation running a terminal emulation package. Device data files can be stored on the programmer's disk (and on the workstation).

Test Vector Test vectors functionally test the device, using structured test vectors stored in memory or in a disk file.

Test Vector Stretching Conversion of DIP test vectors to equivalent PLCC test vectors by adding don't care vector characters into the string to correspond with the PLCC's dead pins.

Total set size Used during serial set programming, this value specifies how many devices are in a set.

Translate DIP/LCC Vectors See JEDEC I/O translate DIP/LCC Vectors.

Translation Formats A form of transmission protocol, these formats are used when transferring data between the programmer and a host computer. The different formats represent different ways of encoding the device data in a data file. The data file could contain the fuse pattern for a logic device or the data for a memory device.

Transmit Pacing The number of milliseconds the programmer will insert as a time-delay between characters transmitted to the host computer during uploading. The time delay is specified in tenths of milliseconds.

Transparent Mode One of the programmer's three operating modes. The programmer is controlled from either a dedicated terminal or from a workstation running a terminal emulation package. In addition, the Remote port is connected to a host computer or file server. The programmer performs all of its programmer operations, and can act as a transparent link between the terminal/workstation and the host, eliminating the need for a switch box. Thus, you can control both the programmer and the host from the same terminal/workstation. With one exception, all keystrokes entered on the terminal will be passed directly to the host. The exception is the Escape command.

TSOP An acronym for Thin Small Outline Package, a type of device package. A rectangular device with gull-wing leads on the short sides. Also called TSOP I.

Underblow A condition in which fuses that should have been blown or programmed were not.

Underblown Fuse A fuse that did not disconnect as per manufacturer's specifications. These fuses may test properly but tend to be more prone to grow back when in circuit, rendering the PLD useless.

Universal Pin Driver A pin driver with the ability to supply power and ground to every pin. With Universal Pin Drivers, you can program and test devices without having to use pin out adapters and characterizers.

Universal PLD Programmer A programmer that can apply power, ground, and any programming pulse required to program any fuse technology device.

Upload Data A file transfer operation that involves sending data from the programmer to a host.

Upload Host Command A command that is sent from the programmer to the host during an upload. The command tells the host what to do with the incoming data.

Upload Wait The length of time the programmer will wait before it begins sending data to the host computer after the host upload command is sent.

User Data Size The hexadecimal number of bytes of a data block to use for a device operation. Normally, this value is equal to the device size. During serial set operations, this value works with Total Set Size to determine the total amount of bytes to program into a set of devices.

User memory The workspace used during device operations. It can be either internal RAM or a disk file. Normally, RAM is used for small, quick device operations, such as programming a single device, while disk is used for larger device operations, such as serial set programming.

User Menu Data The information you see when you look at the programmer screen. It includes such items as the dialog window, reminder bar, and message bar.

User Menu Port The port the user menu data is sent to. You can re-direct user menu data to either the Terminal port of the Remote port.

User RAM The RAM in the programmer. User RAM can be used as a source/destination for an operation. Several operations use User RAM as a a temporary storage buffer, overwriting any data that may have been there previously.

Verify Device A device operation that compares data in a programmed device with data in RAM or in a disk file. With logic devices, verifying can also include functional testing. Verify is an automatic part of the program operation, but additional verify operations can provide useful information about any errors.

Verify Pass A verify pass is a trip through a device at a specified Vcc to see if the device programmed properly. The pass is usually done once at 5V. The pass can also be done twice, with the first pass at 5.5V and the second pass at 4.5V.

Waveforms Images of the programming pulses that program a device. Usually created by programmer manufacturers and submitted to device manufacturers as part of the approval process and to record the correct programming spec for a specific device.

Wildcard Used when entering filenames, a wild card represents one or more characters in a filename. For example, **27*.dat** represents both **27512.dat** and **27128.dat**.

Workstation A PC or other micro computer used for local control of the programmer. You must use terminal emulation software, such Data I/O's HiTerm, to allow the programmer and the PC (or other micro) to communicate. The programmer is designed to be compatible with all popular design workstations, including both DOS and UNIX-based workstations.

Yield The percentage of successfully programmed devices.

Yield Tally

The yield tally function keeps track of the programming statistics for the last 16 types of devices programmed. The following statistics are kept for each device type: the manufacturer name and part number, the family / pinout code, the number of devices attempted, the number of devices that programmed successfully, the number of devices that failed non-blank test or illegal bit check, the number of devices that failed to verify, the number of devices that could not be programmed because they contained bits that required more programming pulses than were specified, and, for logic devices only, the number of devices that failed structured vector test.

ZIF Socket

An acronym for Zero Insertion Force. A socket in which the device can be dropped in and engaged via a lever.

A. Using Modules

This appendix describes the use of the PinSite and SetSite Programming Module. It contains the following sections:

Introduction to PinSite

The PinSite Programming Module provides single-site support for integrated circuits in surface mount and pin grid array (PGA) packages.

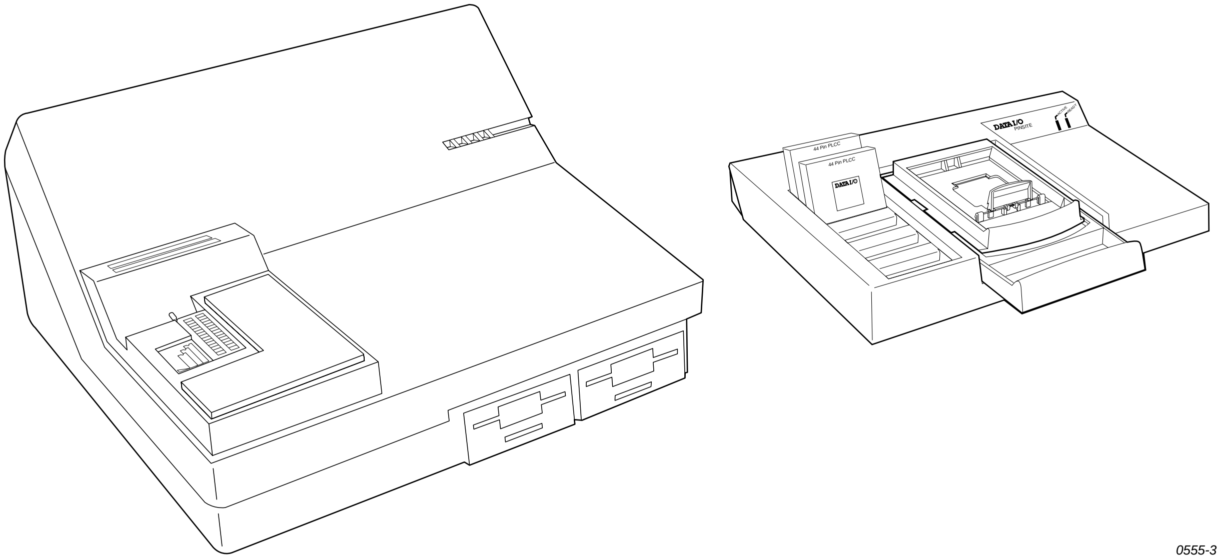
With PinSite, PGAs and surface mount devices (SMDs), including plastic leaded chip carriers (PLCCs), leadless chip carriers (LCCs), and JEDEC standard small outline ICs (SOICs), can all be programmed and tested in one programming site. With a PPI (Package or Pinout Interface) base installed, a variety of high pin count and unusual packages, such as QFPs and PCMCIA cards, can be programmed.

The standard PinSite Module contains 16 pin drivers, extending UniSite-*xpi* device support to 84 pins. The Device List shipped with each UniSite-*xpi* software update lists the devices supported by PinSite in the “Module” column.

Configuration

The PinSite module can program most types of surface mount devices. Figure A-1 shows UniSite-*xpi* and the PinSite Module. Several configuration options are available and are described in the following paragraphs.

Figure A-1
UniSite-xpi and PinSite Module



PinSite Module

The standard PinSite Module consists of one internal circuit board and can program parts with up to 84 pins.

Base Options

Optional bases available are described below.

- PLCC Base — comes with MatchBooks for PLCC devices from 20 to 84 pins.
- LCC Base— comes with MatchBooks for LCC devices from 20 to 84 pins.

- PGA Base — contains a 15 x 15 universal Pin Grid socket, enabling you to program PGA devices from 44 pins to 84 pins.
- SOIC Base — enables you to program SOICs with leads on 0.050-inch centers up to 44 pins.
- PPI Base— accepts a variety of PPI (Package or Pinout Interface) adapters. The PPI adapters support a variety of high pin count and unusual packages, including QFP, Shrink DIP (0.070-inch centers), TSOP, and PCMCIA card.

Compatibility

Whether or not PinSite is compatible with your UniSite-*xpi* depends on your memory requirements and on the version of UniSite-*xpi* software you are using. The PinSite module is fully compatible with any UniSite-*xpi* running Version 3.2 (or later) system software.

Note: The software version and RAM size appear in the upper right corner of most UniSite-xpi VT100 screens.

Pin Driver Requirements

PinSite requires a minimum number of pin driver boards for programming specific device packages. The following table lists the minimum number of pin driver boards necessary to program PLCC, LCC, SOIC, and PGA devices.

		Number of
Package Type	Number of Pins	Pin Driver Boards
PLCC and LCC	20	7
PLCC and LCC	28	7
PLCC and LCC	32	8
PLCC and LCC	44	11
PLCC	52	13
PLCC and LCC	68	17
PLCC and LCC	84	17
SOIC	8	7
SOIC	14	7
SOIC	16	7
SOIC	18	7
SOIC	20	7
SOIC	24	7
SOIC	28	7
SOIC	32	8
PGA	28	17*
PGA	44	17
PGA	68	17
PGA	84	17
PGA	88	17†

* *28-pin PGA devices require the PSBASE-0402 revision of the PGA Base.*

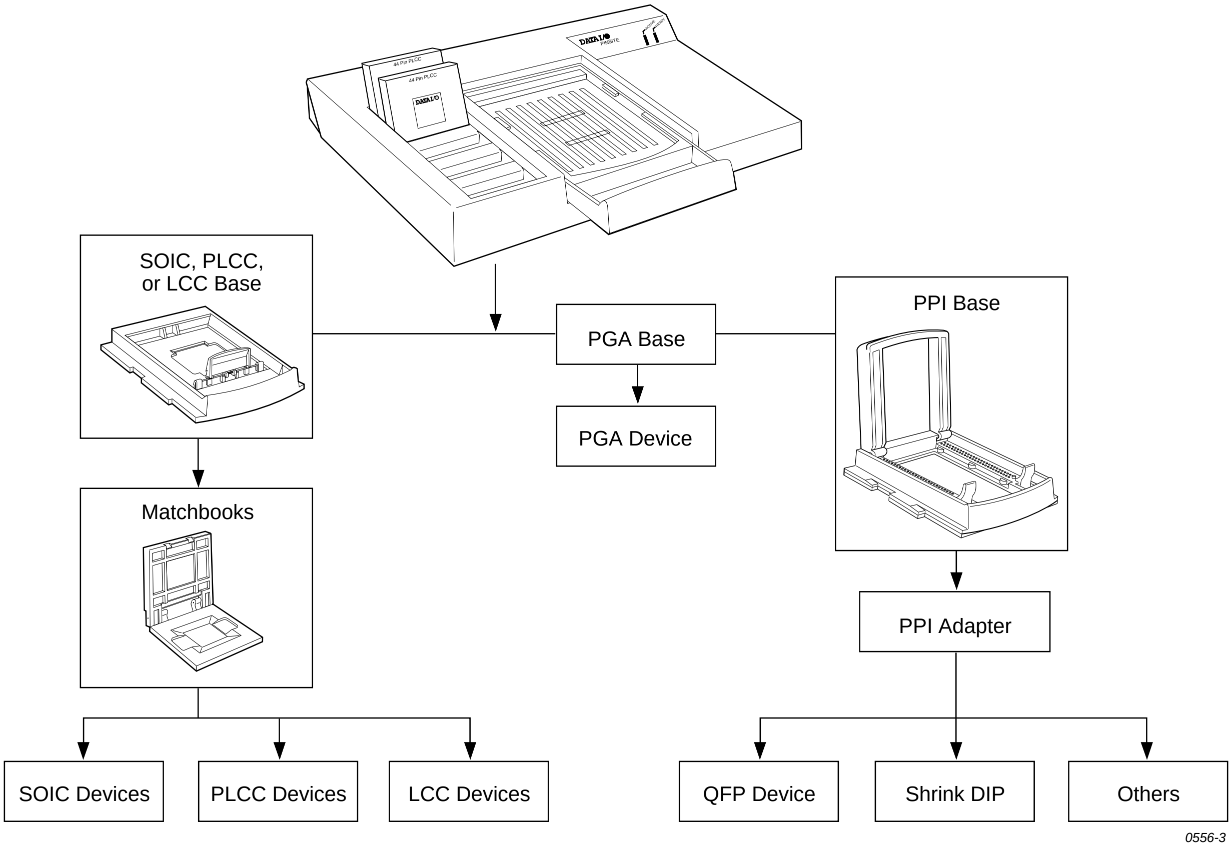
† *See page A-13.*

Installing the PinSite Module

This section describes the various PinSite and Base installation possibilities. Your specific configuration depends on the Base and device package type you will be using.

Before continuing with the PinSite installation procedures, make sure you have correctly set up your UniSite-*xpi* as described in the Chapter 2, “Setup and Installation.” Figure A-2 shows the relationship between the PinSite Module, the Bases, the MatchBooks, and the devices.

Figure A-2
Installation Flow Chart



Installing the PinSite Module in UniSite-*xpi*

Depending on when your UniSite-*xpi* was manufactured, your UniSite-*xpi* might have been shipped with a blank module cover installed in place of a large module. See “Removing the Blank Module Cover” on page 2-3 for more information.

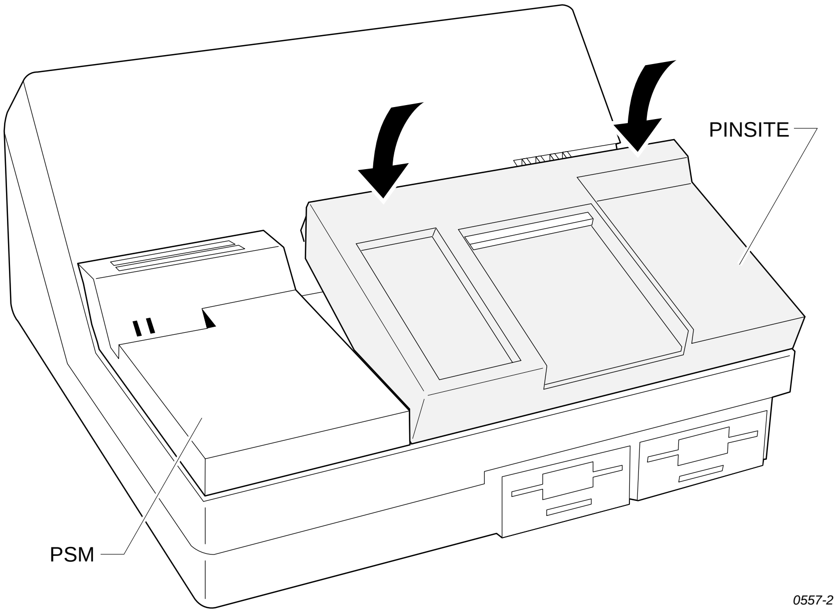
Note: You must first remove the top cover and the blank module cover, then replace the top cover before installing the PinSite module.

PinSite Installation

Follow the steps below to install the PinSite Module into your UniSite-*xpi*.

1. Insert the two retainer hooks on the bottom of the PinSite Module into the two right slots on the top of UniSite-*xpi*.
2. Slowly lower the back of the module until the module connector touches its mating connector on top of UniSite-*xpi*. (See Figure A-3.)
3. To ensure complete contact, firmly press down on the rear of the module.

Figure A-3
PinSite Installation



4. To remove the module, lift the rear of the module until the retaining hooks on the front of the module are disengaged from the slots on the top of UniSite-*xpi*.

Installing a Base in PinSite

The PinSite module is designed to accept one Base at a time. The Base serves as the interface between the device and the PinSite Module. Bases are available with MatchBook technology for PLCC, LCC and SOIC devices or with Zero Insertion Force (ZIF) sockets as used on the PGA Base. The PPI Base, also for use with PinSite, provides a package and pinout interface.

The following instructions show how to install a PLCC Base into the PinSite Module, but the installation procedure applies to all Bases.

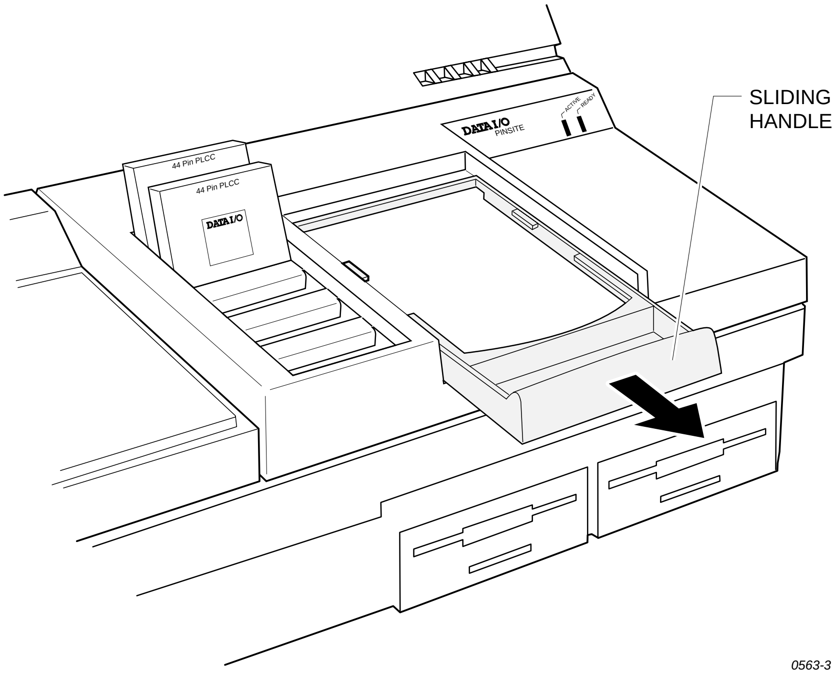
Note: You can install and remove a Base with the power on as long as you are NOT performing a device operation.

1. Position UniSite-*xpi* so that the disk drive slots are facing toward you.
2. Examine the Base receptacle opening. It has a sliding handle on the front half. As shown in Figure A-4, pull the sliding handle toward you until the handle stops. Do not remove the handle.

CAUTION: Do not insert a foreign object into the Base receptacle opening. Also, do not touch any of the gold pins in the Base receptacle opening. Pressing or touching the pins may cause the pins to stick and will cause operational problems.

If the pins do get stuck, call Customer Support for more information.

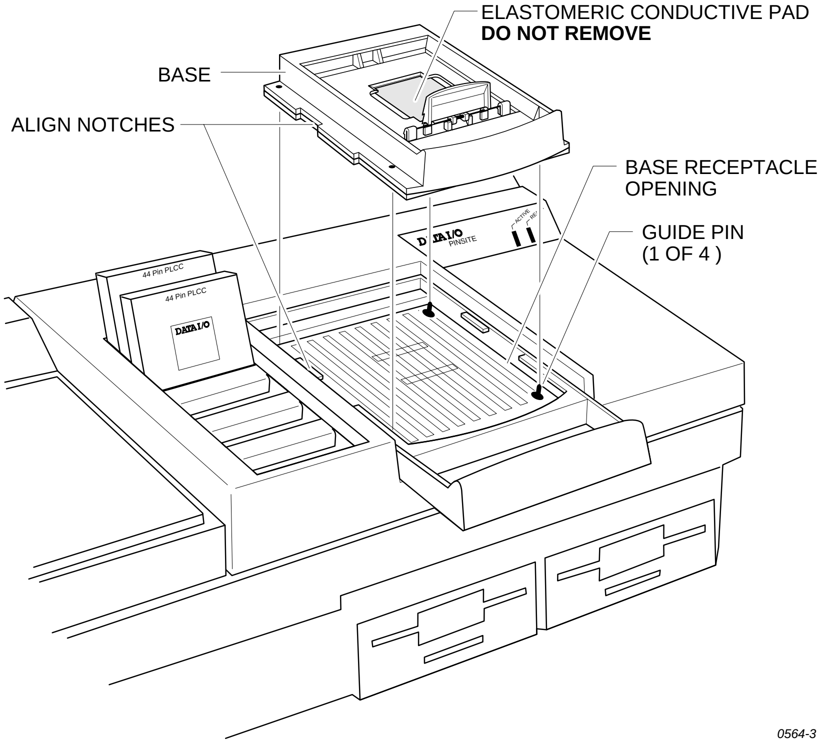
Figure A-4
Base Receptacle Opening



3. Remove the cardboard cover that protects the conductive pad.

4. Align the notches on the Base with the notches in the opening, as shown in Figure A-5.

Figure A-5
Aligning the Base



5. Line up the four guide pins in the Base opening with the holes in the Base, and insert the Base into the opening.

6. Finally, squeeze the Base and module handles together, locking the Base into place. You do not need to use excessive force when compressing the handles.

CAUTION: You can damage PinSite by squeezing too hard on the handles.

Conductive Pad

The conductive pad, shown in Figure A-5, is required to complete the electrical connection between the programmable device pins and the Base's circuit board.

CAUTION: Do not remove this conductive pad unless you are cleaning or replacing it.

See page A-20 for more information on conductive pad care and replacement procedures.

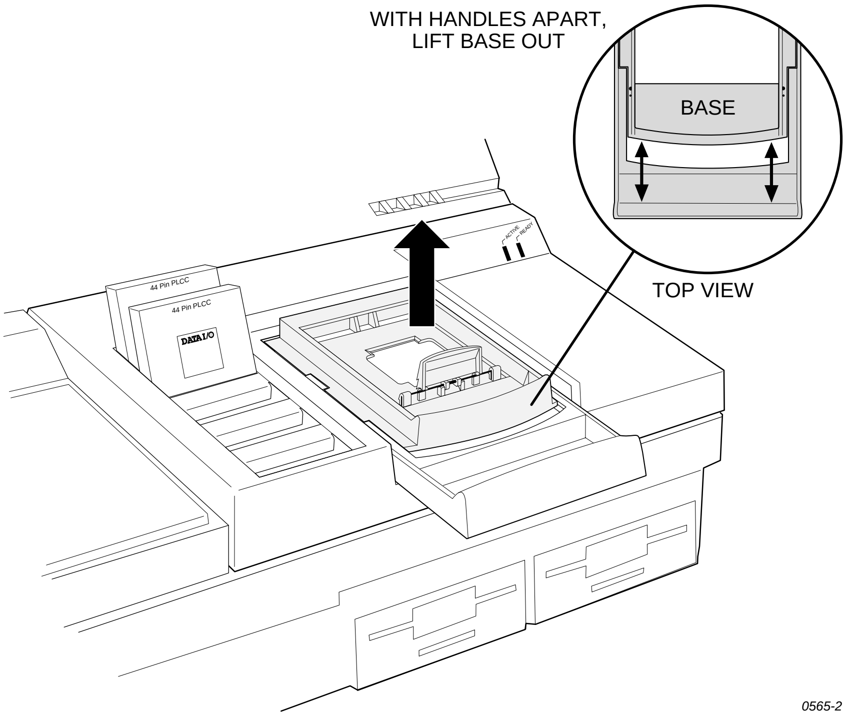
Removing a Base

To remove a Base, follow the steps below.

1. Separate the handles by pushing back on the Base handle with both thumbs and, at the same time, pushing forward on the sliding PinSite handle with your fingers. See Figure A-6.

CAUTION: You must apply even pressure while moving the handles apart. If you exert uneven pressure on the handles, they may jam in the tracks. Apply an even force to realign the handles.

Figure A-6
Removing a Base from PinSite



2. Lift the Base up and out of PinSite. Store the Base in the storage container that you received with your PinSite.

Installing a MatchBook in a Base

The MatchBook holds PLCC, LCC, and SOIC devices in place on the Base. When a device is locked in place, the conductive pad on the Base forms a conductive path between the pin drivers and the device in the MatchBook.

To install and remove a MatchBook, follow the steps below.

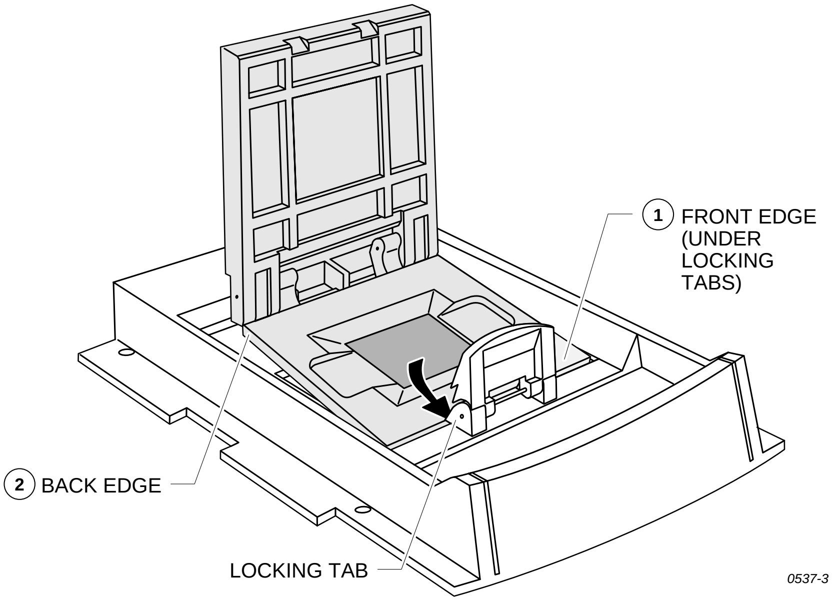
Note: Although the illustrations show a PLCC MatchBook, the procedure is the same for LCC and SOIC MatchBooks.

1. Select a Base and insert it into PinSite. Lock the Base into place.
2. Select the appropriate MatchBook for package type (PLCC, LCC, or SOIC) and pinout (20 pins, 28 pins). Select SOIC MatchBooks by device package width, such as .300 inches.

Note: We recommend that you use PLCC parts with the PLCC MatchBook only.

3. With the MatchBook open 90 degrees, insert it into the Base by setting its front edge onto the Base under the two locking tabs at the front edge of the Base opening, and then lowering the back edge of the MatchBook into place. See Figure A-7.

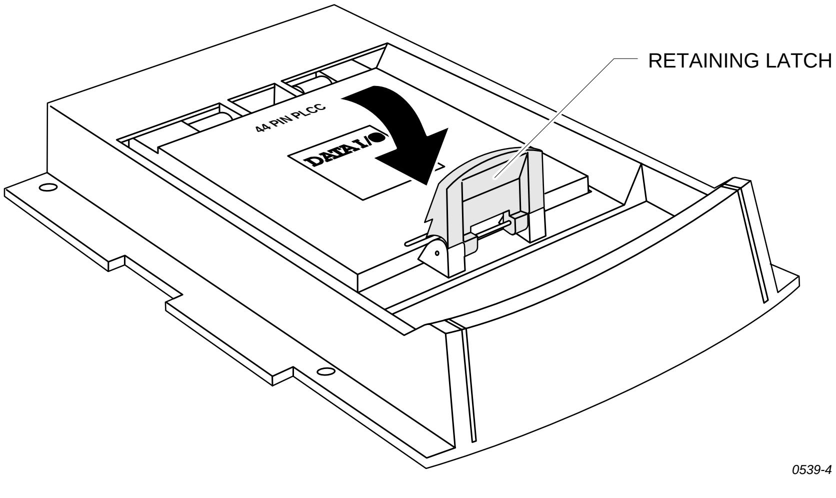
Figure A-7
Inserting a MatchBook into a Base



4. Insert a device into the MatchBook. Insertion procedures vary according to package type. For information on inserting the various device types, refer to the following sections.
5. Close the MatchBook by pressing the retaining latch forward with your thumb until the latch snaps into place. See Figure A-8.

CAUTION: Do not place excessive force on the top of the MatchBook, as this may cause premature wear on the conductive pad.

Figure A-8
Closing the MatchBook



Opening/Removing a MatchBook

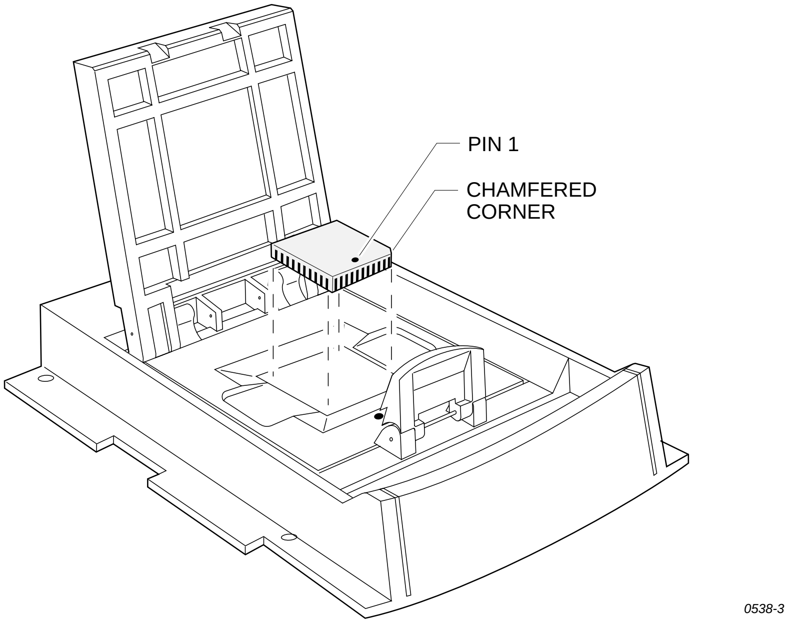
To open a MatchBook, unsnap the retaining latch. To remove a MatchBook, open the MatchBook and lift it out of PinSite.

Installing a PLCC or LCC Device in a MatchBook

Use the following procedure to insert a PLCC or LCC device into a MatchBook after the appropriate Base has been installed in PinSite.

1. Select and insert the appropriate MatchBook into the PLCC/LCC Base. For example, select the 44-pin PLCC MatchBook for a 44-pin PLCC device.
2. Position the PLCC or LCC device so that pin 1 is at the center bottom as you view it from the top. Figure A-9 illustrates the proper positioning of a PLCC or LCC device.

Figure A-9
Inserting a Device into the PLCC or LCC Base



3. Insert the PLCC or LCC device into the open MatchBook. Make sure that the center bottom locating dot on the device is properly aligned with the locating dot on the MatchBook.

Note: Position the device so that pin 1 is near to the retaining clip. There is a small dot molded into each MatchBook to help you align your device. Each PLCC MatchBook also has a beveled corner to help you align devices with a chamfered corner to indicate pin 1.

4. Close the MatchBook and press the retaining latch forward with your thumb until the latch snaps into place.

Installing an SOIC Device in a MatchBook

Choosing an SOIC MatchBook

The following table shows a list of SOIC MatchBooks and the maximum number of pins each MatchBook supports.

MatchBook	Maximum Number of Pins
.150 SOIC	20
.220 SOIC	32
.300 SOIC	32
.350 SOIC	32
.420 SOIC	44
.450 SOIC	44
.530 SOIC	44

The device support capability of the programmer varies depending on the model number of the SOIC Base and the number of pin drivers installed in UniSite-*xpi*.

SOIC Base	Pin Drivers	Pin Driver Cards	Max. #
715-2038-001	28	7	28
715-2038-001	32	8	32
PSBASE-0302	28	7	28
PSBASE-0302	32	8	32
PSBASE-0302	40	10	40
PSBASE-0302	44	11	44
PSBASE-0302	48	12	48
PSBASE-0302	52	13	52
PSBASE-0302	56	14	56

Selecting the SOIC Device and MatchBook

Even though the MatchBooks come in different widths and are named according to their size, the difference in size is small. It may be difficult to tell which MatchBook is the correct one to use with your device.

Some manufacturers will have several listings of the same SOIC device on the menu; the suffix distinguishes them by device width. For example, the MCT device 24C02 could have more than one suffix: /A-SN or / A-SM. We recommend using the following procedure to select the correct MatchBook.

Note: Refer to your programmer manual for complete device selection information.

1. Go to the Select Device screen and choose the device you want to use.
2. Pay particular attention to the suffix to avoid putting a device into the wrong size MatchBook.

   The correct MatchBook size for the device appears on the right-hand side of the message bar, near the top of the screen. For example:

   ```
   Matchbook: .300 SOIC
   ```

Inserting the SOIC Device into a MatchBook

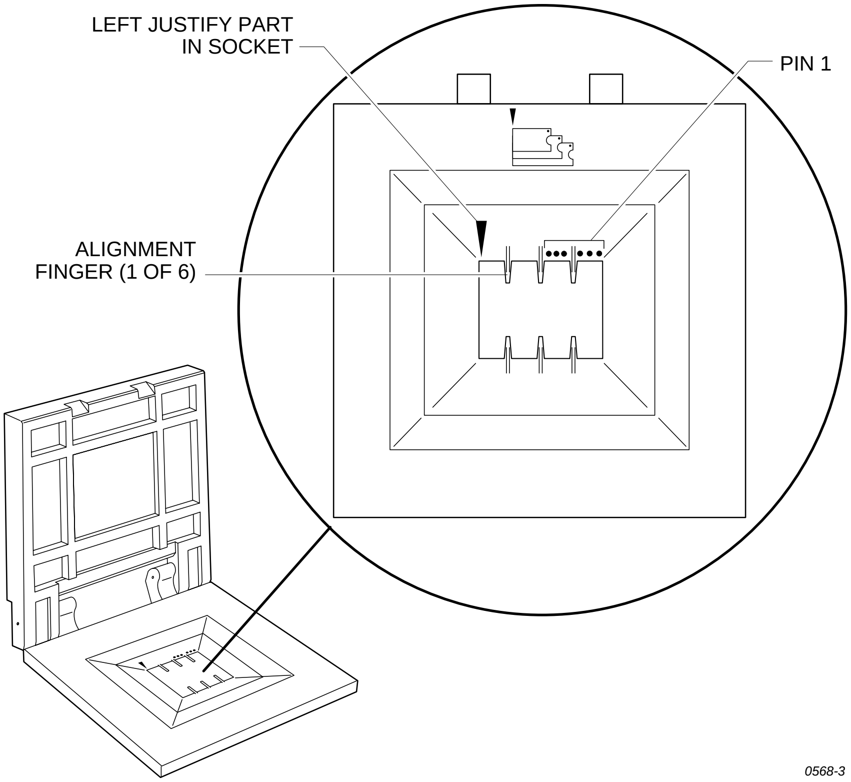
Use the following procedure to insert an SOIC device into the SOIC MatchBook. Before you go any further, make sure you have the SOIC Base properly installed in PinSite.

1. Position the SOIC device so that pin 1 is up and to the right as you view it from the top.
2. Insert the SOIC device into the open MatchBook. Make sure the SOIC device is flush against the left-hand side of the MatchBook. Also make sure that the device is positioned between the six alignment fingers and not on top of them.

   The device should be left justified. The unused portion of the socket will be on the right as you view it from the top.

   The small, round dots along the top of the opening, as shown in Figure A-10, indicate the location of pin 1 for the various sizes of SOIC devices the SOIC MatchBook will accept.

Figure A-10
SOIC Device Insertion



3. Finally, close the MatchBook and press the retaining latch forward with your thumb until the latch snaps into place.

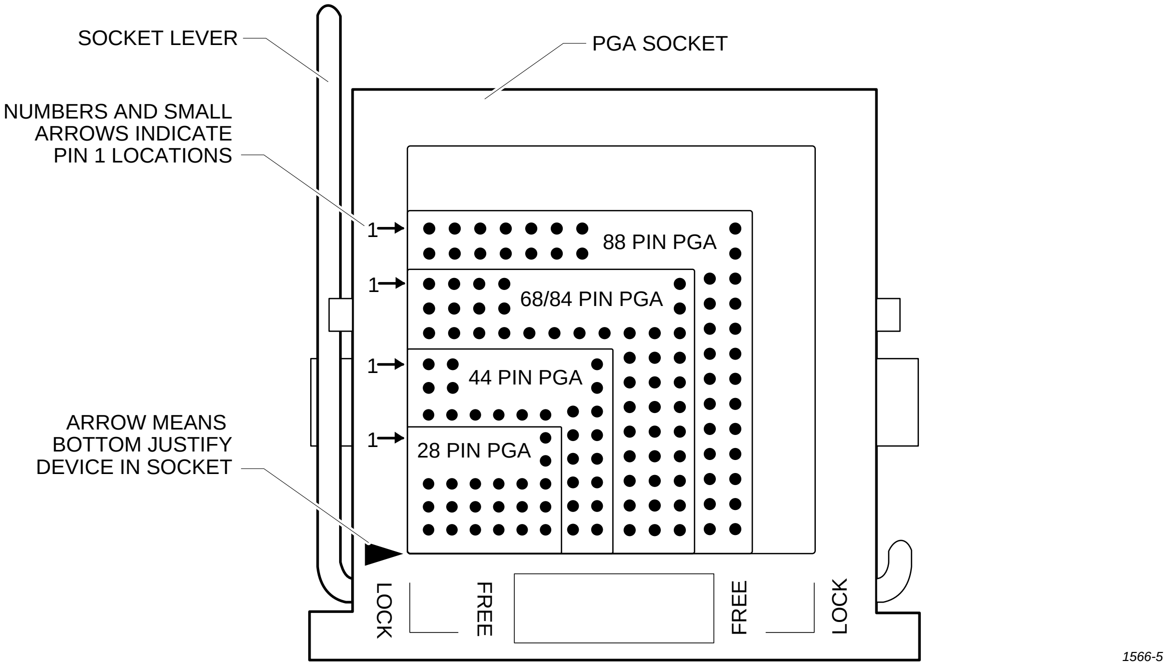
Installing a PGA Device in a PGA Base

The PGA Base can accommodate PGA packages up to 15 x 15 pin arrays. The PGA Base contains a ZIF (zero insertion force) socket to hold the PGA device and routes the signals from the compliant pins to the correct socket pins. The PGA Base supports 28-, 44-, 68- 84-, and 88-pin PGAs. Use the following procedure to insert a PGA device into the PGA Base.

Note: When you insert a PGA device into the PGA Base, pay particular attention to the orientation and positioning of the device.

1. Install the PGA Base into PinSite. Unlock the PGA socket by lifting up on the socket lever.
2. Insert the device into the PGA socket. Make sure the PGA device is bottom justified and that pin 1 of the device is against the left side of the socket. Figure A-11 shows the proper alignment of a PGA device.
3. Push the socket lever down to lock the PGA device into the PGA socket.

Figure A-11
PGA Device Orientation


SOCKET LEVER
PGA SOCKET
NUMBERS AND SMALL ARROWS INDICATE PIN 1 LOCATIONS
1
1
1
1
88 PIN PGA
68/84 PIN PGA
44 PIN PGA
28 PIN PGA
ARROW MEANS BOTTOM JUSTIFY DEVICE IN SOCKET
LOCK
FREE
FREE
LOCK
1566-5

Note: Figure A-11 shows the socket lever attached at the lower left. On some bases, the socket lever may be attached at a different point (such as the upper left or the lower right).

Using the PPI Base

The PPI interface system provides an easy exchange of programming sockets while maintaining the electrical integrity of the programming signals. The PPI system also allows support of devices with nonstandard pinouts or pincounts beyond the pin driver capability of the programmer.

About the PPI Base

The PPI Base has 108 pins configured in four rows of 27 pins each. These pins support up to 88 pin drivers with V_{CC} and ground capability.

The PPI Base supports the following applications:

- QFP (Quad Flat Pack)
- Device Carriers
- SDIP (Shrink Dip)
- SOIC (Small Outline Integrated Circuit)
- TSOP (Thin Small Outline Package)
- PGA (Pin Grid Array)
- Memory Card
- Non-JEDEC pinout — For devices not supported in standard bases
- Pin Driver Expansion — For devices that do not require a pin driver on every pin to perform a programming operation.

About the PPI Adapter

A variety of PPI adapters are available from Data I/O to accommodate the device pinouts of many package types. For example, a different PPI adapter is available for each lead pitch (the space between adjacent pins).

The PPI Base provides a durable electrical interconnect that allows adapters to be exchanged easily. A wide variety of sockets are available to support the large number of device package types.

For example, an 80-pin Quad Flat Pack (QFP) is available in many different package configurations. It is available in a square 20 x 20 pinout or in a rectangular 16 x 24 pin package. The QFP may also have different lead pitches and package dimensions.

Each adapter board routes universal pin drivers, V_{CC}, and ground to the appropriate device pins.

Installing a PPI Adapter into the PPI Base

Use the following procedure to install an adapter into the PPI Base.

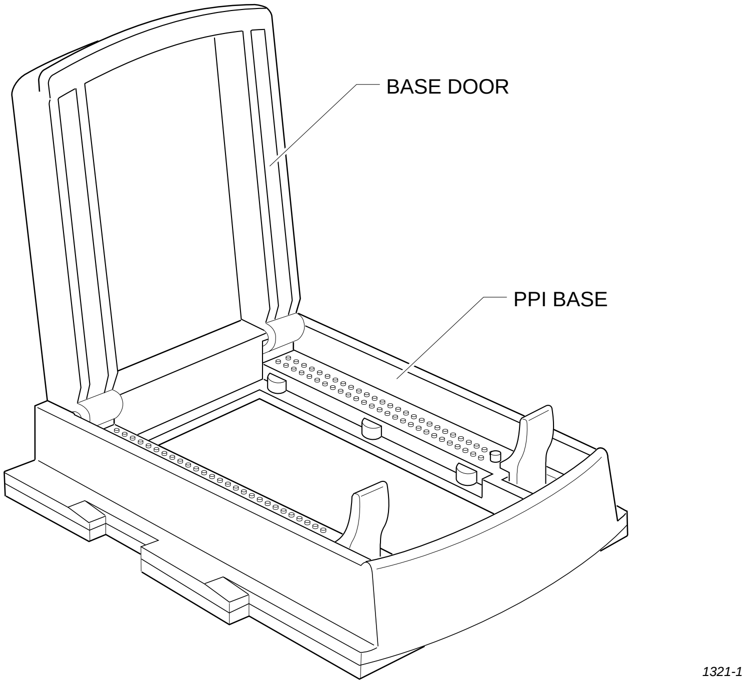
CAUTION: Do not touch the exposed pins or components on the base. Contamination could lessen the programming reliability of the adapter. Also, do not disassemble the base; doing so could cause the pins to drop out.

Installing a device into the adapter before the adapter has been securely placed in the base may damage the device.

1. Make sure the PPI Base is installed in your programmer.

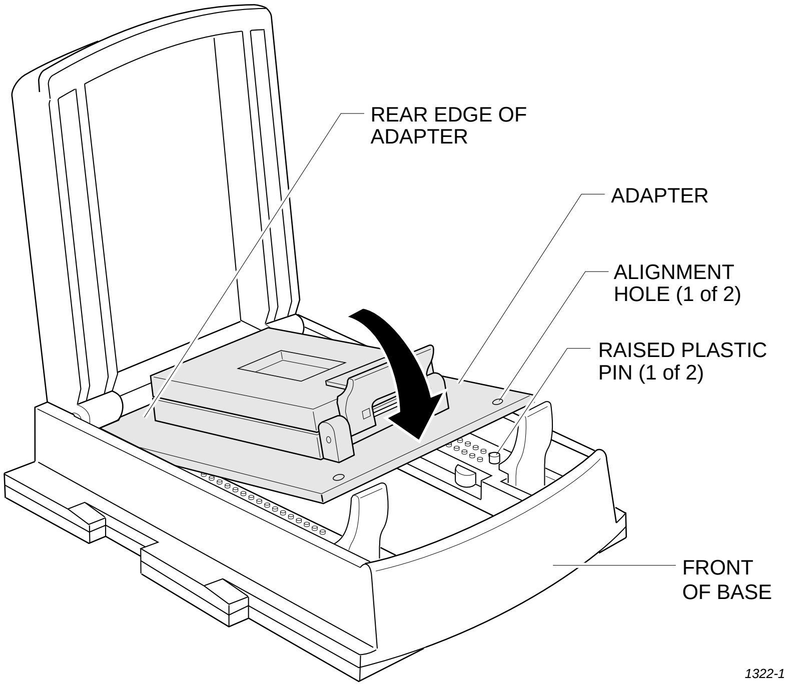
2. Raise the base door until it rests in the upright position against the back of the PPI Base as shown in Figure A-12.

Figure A-12
PPI Base



3. Pick up the adapter with the retaining latch end facing toward you, and slide the rear of the adapter into the back of the base as shown in Figure A-13. The alignment holes in the front corners of the adapter slide onto the raised plastic pins.

Figure A-13
Inserting an Adapter into the Base



4. Lower the removable base door over the PPI adapter.

Note: If the adapter socket has a high profile, you may need to remove the door and tilt the adapter upward to be able to lower the door.

5. While pulling back on the two latches, press down on the door until the latches snap into place over the door. See Figures A-14 and A-15. Make sure that both latches are securely in place.

Figure A-14
Lowering the Base Door Over the Adapter

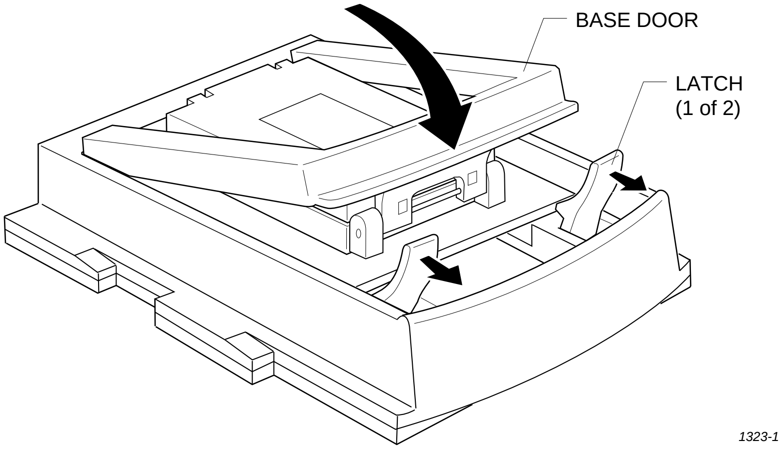
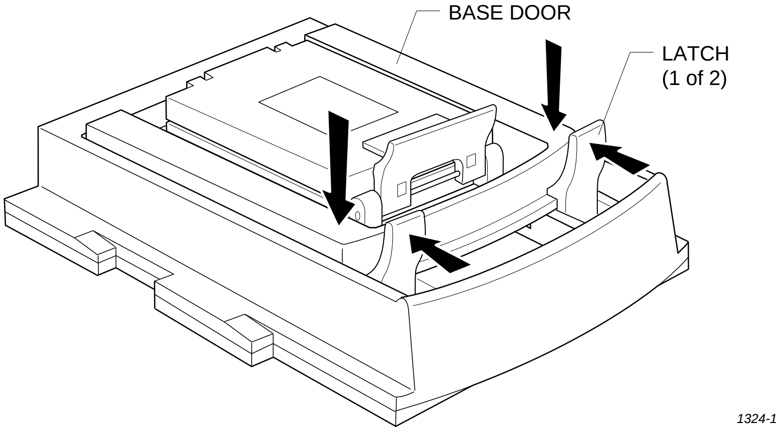


Figure A-15
Securing the Base Door

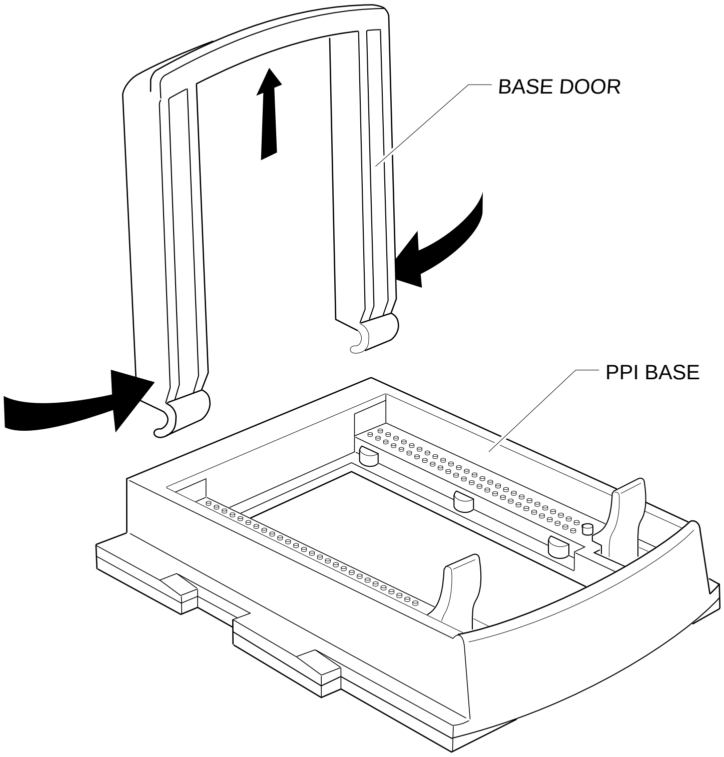


High Profile PPI Adapters

Some PPI adapters have sockets that sit particularly high on the circuit board. To lower the door over the adapter, align the adapter and the door as a unit. Use the following procedure and refer to Figures A-16 and A-17.

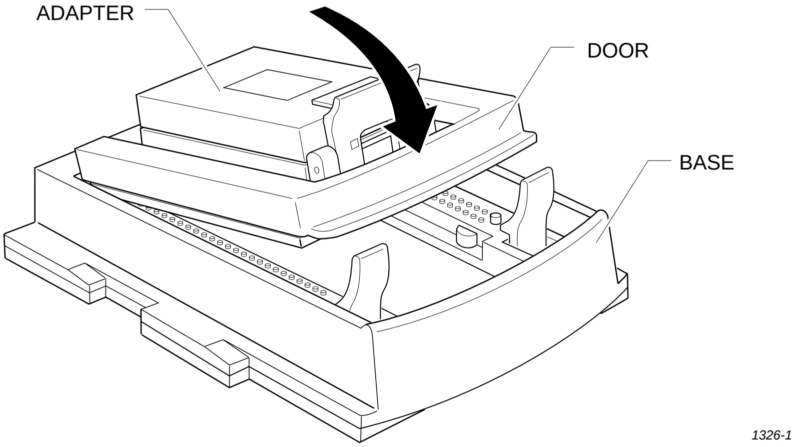
1. To remove the door from the PPI Base, lift the front of the door up and press in slightly on the sides of the door near the hinges. Remove the door as shown in Figure A-16.

Figure A-16
Removing the Door from PPI Base



2. Place the door over the adapter as shown in Figure A-17.
3. As a unit, guide the door hooks into the PPI Base and lower the adapter and door together onto the base.
4. Move the adapter from side to side until it rests on the base.
5. Make sure the holes on the adapter line up with the the raised plastic pins on the base.
6. Close the door.

Figure A-17
Aligning a High Profile Adapter



Installing Devices into a PPI Adapter

The PPI system consists of one base for each programmer, and exchangeable adapters. These PPI adapters accept a variety of device package types and pin configurations. The following figures show three common device packages and the proper device orientation in the adapter. Each adapter has an ICON showing the proper device orientation.

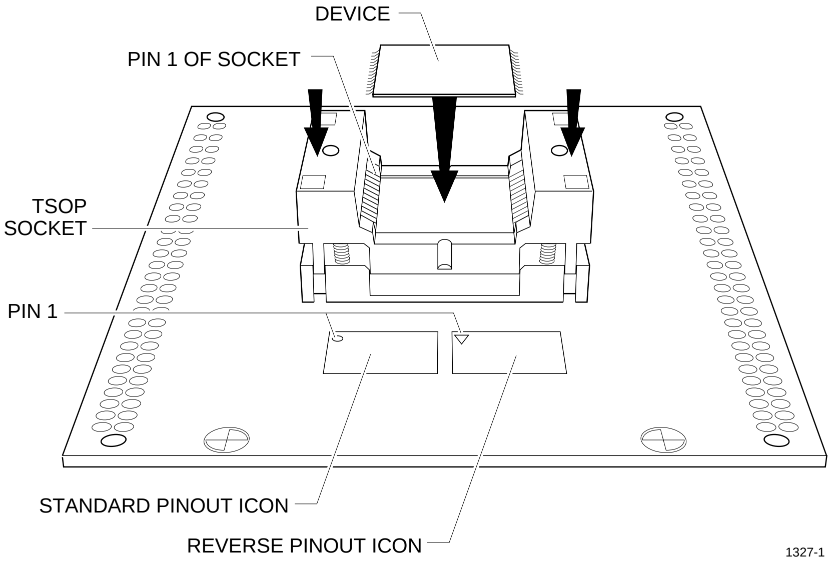
TSOP Device Orientation

There are two different pinouts for the TSOP device as shown in Figure A-18. The standard pinout, shown by the standard pinout ICON on the left in Figure A-18, shows Pin 1 indicated by a circle in the upper-left corner of the device. The reverse pinout, shown by the reverse pinout ICON on the right in Figure A-18, shows Pin 1 indicated by a triangle in the upper-left corner of the device.

1. Press down on the outer edges of the socket.
2. Drop the TSOP device into the socket while still holding the socket edges down.
3. Release the socket edges. The device should be secured.

To remove the device, press down on the socket edges and lift the device out.

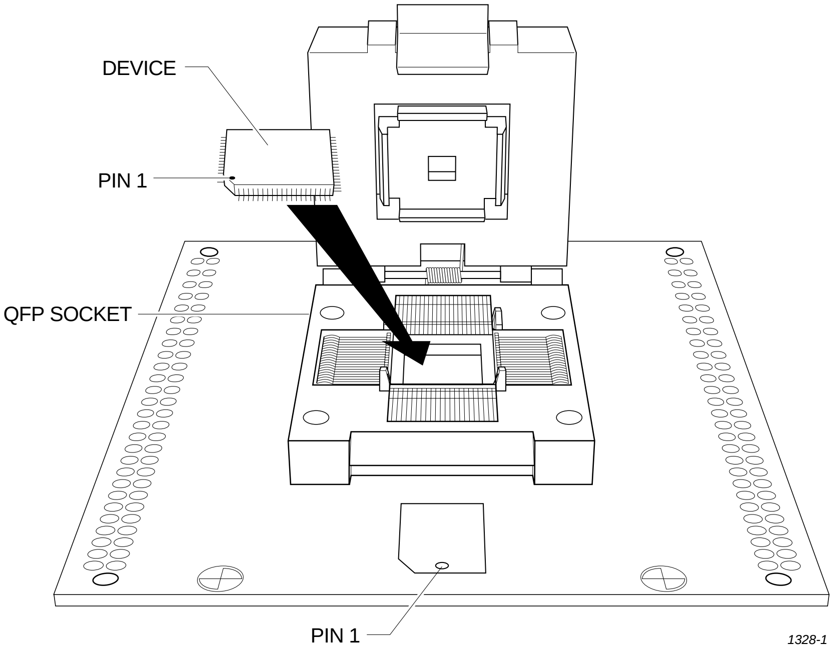
Figure A-18
TSOP Device Orientation



Quad Flat Pack Device Orientation

Pin 1 on the QFP device is usually at the front left of the device. Orient the QFP device according to the icon or indicator on the adapter.

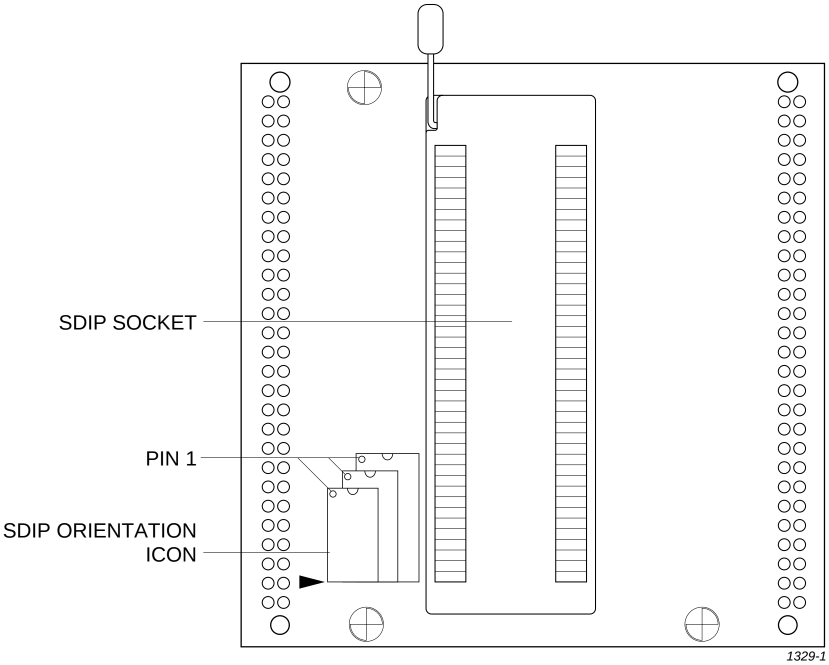
Figure A-19
QFP Device Orientation



Shrink Dip Device

Orient the SDIP device in the socket with pin 1 at the top left and bottom justified as shown in Figure A-20.

Figure A-20
Shrink Dip Device Orientation



Using PinSite

If you want to program devices that are programmable with the PinSite Module, the PinSite Module must first be installed in UniSite-*xpi*. See the section "Installing PinSite in UniSite-*xpi*" for installation instructions.

If the PinSite Module is installed in UniSite-*xpi* at powerup, one of the following messages appears on the first UniSite-*xpi* screen:

```
FSM: PinSite V01 w PLCC/LCC Base

FSM: PinSite V01 w PGA Base

FSM: PinSite V01 w SOIC Base

FSM: PinSite V01 w PPI Base

FSM: PinSite V01 w PGA PSBASE-0402 Base

FSM: PinSite V01 w PGA PSBASE-0302 Base

FSM: PinSite V01 w unknown Base
```

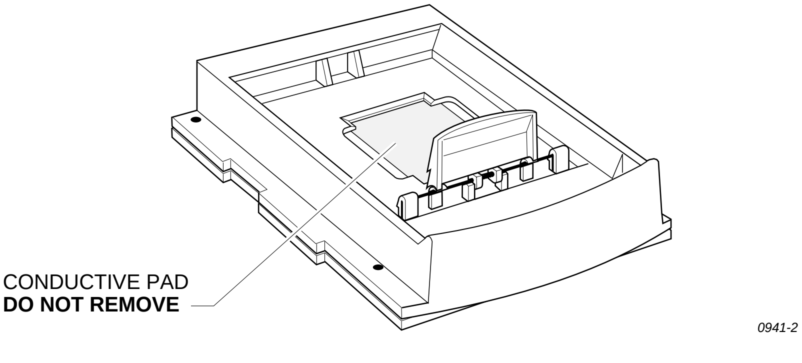
`Unknown Base` can mean either no Base has been installed or an unidentified Base has been installed. This may mean that your software needs to be updated to support a new type of Base or there is a problem with your PinSite.

Conductive Pad Care

The conductive pad, shown in Figure A-21, is a key element in the MatchBook technology. Keeping the pad free of dirt or contamination will help keep yields high and prolong the life of the pad. We recommend that you inspect and clean the pad at least once every 1000 insertions or once a month, whichever comes first.

Note: After a number of insertions you may notice an indentation in the middle of the conductive pad. The indentation is normal and does not degrade the performance of the MatchBook.

Figure A-21
Conductive Pad



The life of the pad is dependent on proper care as well as the pin count and package type of the device being used. Not all devices have the same tolerances, and each device type may result in different life cycles for the pad.

An increase in device insertion errors or continuity errors, or a sudden drop in programming yields may indicate that the pad needs to be cleaned or replaced.

Cleaning the Pad

To clean debris from the conductive pad, blow air over it. If you use compressed air, direct the air stream from the front or back of the Base. Avoid air streams from the side of the pad, which could lift the pad off the circuit board.

To further clean the pad and dislodge dirt, apply a small amount of isopropyl alcohol on a cotton swab and, with a rolling motion, gently wipe off the pad. Make sure that the pad is clear of any cotton filaments left from the cleaning process.

CAUTION: Do not use any petroleum or freon based products to clean the pad. These substances will cause premature deterioration of the pad material.

Replacement Pad Kits

The Base has been designed to allow you to replace the pads quickly and easily, to minimize downtime. Replacement pads are available from Data I/O. Contact Data I/O Customer Support as listed in the Preface for ordering information.

Introduction to SetSite (Legacy)

Data I/O's SetSite module allows you to perform set or gang programming operations with UniSite. (Set and gang programming are defined on the following pages.) SetSite programs MOS/CMOS EPROMs and EEPROMs in the DIP style package. See the device list shipped with each UniSite software update for a complete list of supported devices. Devices programmable with SetSite are listed in the Product Module column on the device list.

Note: SetSite is not supported in TaskLink for Windows software. Use TaskLink for DOS or HiTerm (terminal emulation software).

Included here are applications for using SetSite and explanations of memory allocation for set/gang programming.

Compatibility

Whether or not SetSite is compatible with your UniSite depends on the version of UniSite software you are using and your memory requirements. The SetSite module is fully compatible with any UniSite that has a V2.0 (or later) System disk. If you are using SetSite or PinSite and the part number of your UniSite is 901-0058-007 or less, it is recommended that you update UniSite to assure that SetSite will operate properly with all devices.

Memory requirements depend on the type of device, and the set size and word width you want to use for set programming. UniSite has a standard user RAM size of 128K x 8. Using this architecture, you could gang program up to eight 1Mbit devices (programming the SAME data into all eight parts). Set programming the same eight 1Mbit devices (programming so each device has a DIFFERENT block of data) would require 1MB of user RAM. Should your programming needs require it, Data I/O offers upgrades for additional RAM. Contact your local Data I/O Service Center for more information about the RAM upgrades.

Note: If the disk is being used as user memory (instead of RAM), the user memory you have available is limited by the free space currently available on the disk. The disks have a capacity of 720 Kbytes when newly formatted.

If you need to find out what your UniSite's current configuration is, remember that both the RAM size and software configurations appear on the upper-right portion of most of the UniSite screens.

Features

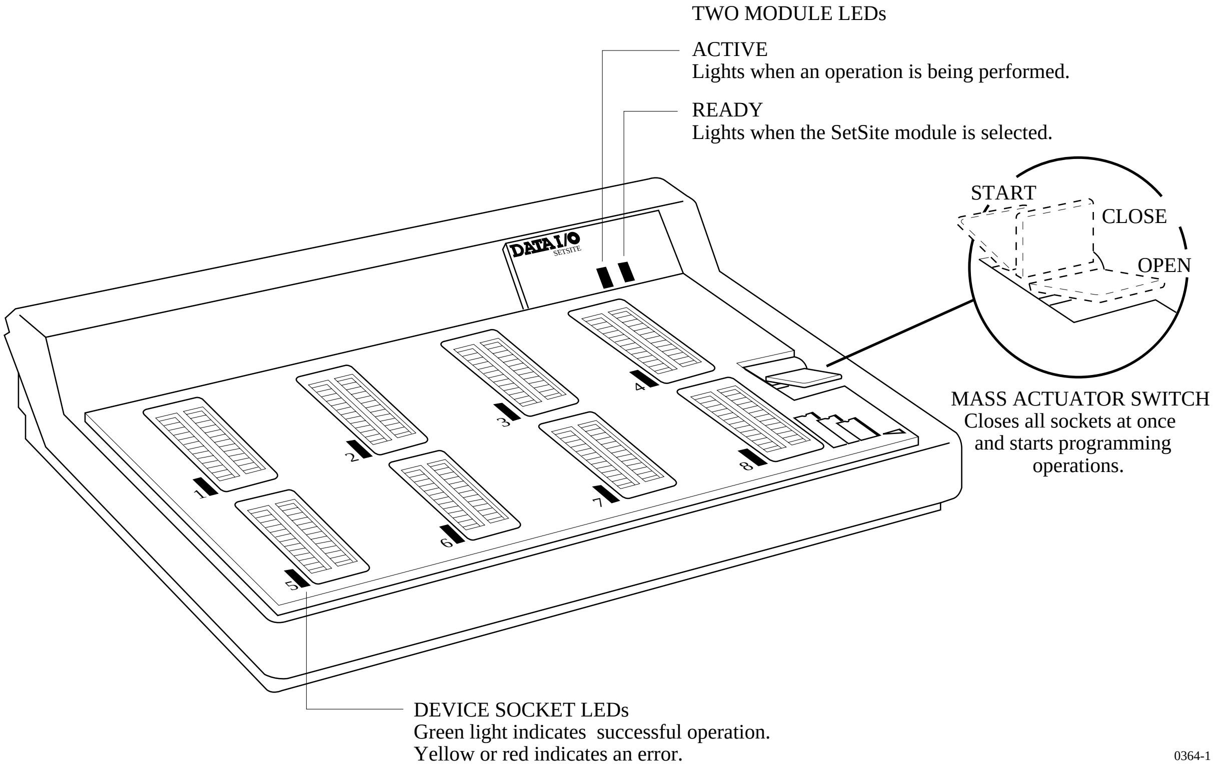
Start Switch

The SetSite module is shown in Figure A-22. The eight 40-pin DIP sockets have a "mass-actuation" feature: there is only one socket lever on the module. Moving this lever moves ALL the socket contacts.

The socket lever has three positions: open, close, and start. Because the lever is spring-loaded, it will automatically return to the closed (straight up) position after being pushed to the start position. The start position—lever pushed forward toward the ACTIVE/READY indicator panel—initiates Load, Program, or Verify operations, just like pressing **ENTER** on the terminal keyboard.

The socket lever also functions as an interrupt or **BREAK** key. If you pull the switch back (thereby opening the socket contacts) while an operation is in progress, that operation will be halted.

Figure A-22
The SetSite Module


TWO MODULE LEDs
ACTIVE
Lights when an operation is being performed.
READY
Lights when the SetSite module is selected.
START
CLOSE
OPEN
DATA I/O
SETSITE
1
2
3
4
5
6
7
8
MASS ACTUATOR SWITCH
Closes all sockets at once and starts programming operations.
DEVICE SOCKET LEDs
Green light indicates successful operation.
Yellow or red indicates an error.
0364-1

LED Indicators

SetSite has two module LEDs and eight individual socket LEDs. The two module LEDs indicate the current status for the entire unit. The “READY” LED lights yellow when SetSite is ready for use. The “ACTIVE” LED lights yellow whenever an operation (such as programming parts) is taking place.

CAUTION:** **When the “ACTIVE” LED is lit, programming voltages are being applied to the device sockets. Do not remove socketed devices from SetSite or the SetSite module from UniSite when the “ACTIVE” LED is illuminated.

LEDs next to each socket indicate status of that socketed part. The LEDs light green to indicate a successful operation; yellow or red indicate an error. A red LED signals a “fatal” error, meaning that the device cannot be programmed. A yellow LED signals a “non-fatal” error—the device can still be programmed. (For example, a “non-blank device” error is non-fatal because even though the device is found to contain programmed bits, those locations can still be programmed over.) The table below shows conditions and LED colors corresponding to each.

Condition	**LED Color**
Module	
SetSite READY for use	“READY” yellow
Operation executing	“ACTIVE” yellow
Socket	
Device programmed successfully	green
Device errors:	
Non-blank	yellow
Backwards device	red
Wrong silicon signature	red
Illegal bit	red
Programming error	red
Verify error	red
Empty socket	LED off

Using SetSite (Legacy)

Selecting SetSite

You may select the SetSite module in one of two ways: (1) in the Load, Program, or Verify menus, press the **PF4** key until the SetSite screen is displayed, or (2) use the "mode" parameter field in the **Manufacturer List** screen.

To select SetSite using the mode field, first enter the Select device menu (press **S** from the **Main** menu). The **Manufacturer List** screen will appear; the mode field is the reverse video block on the right. Move the cursor to the mode field. The space bar toggles the display between "Single device" and "Gang / set" mode. If you choose "Gang / set" mode, only the manufacturers and devices supported by the SetSite module will appear in the Manufacturer List and Part Menu for Manufacturer screens.

If you press **PF4** after selecting a Load, Program, or Verify screen, the appropriate gang / set screen will appear. Screens that are specific to the Set Site module operation are different from for other UniSite modules and display information on-screen for each of the eight device sockets.

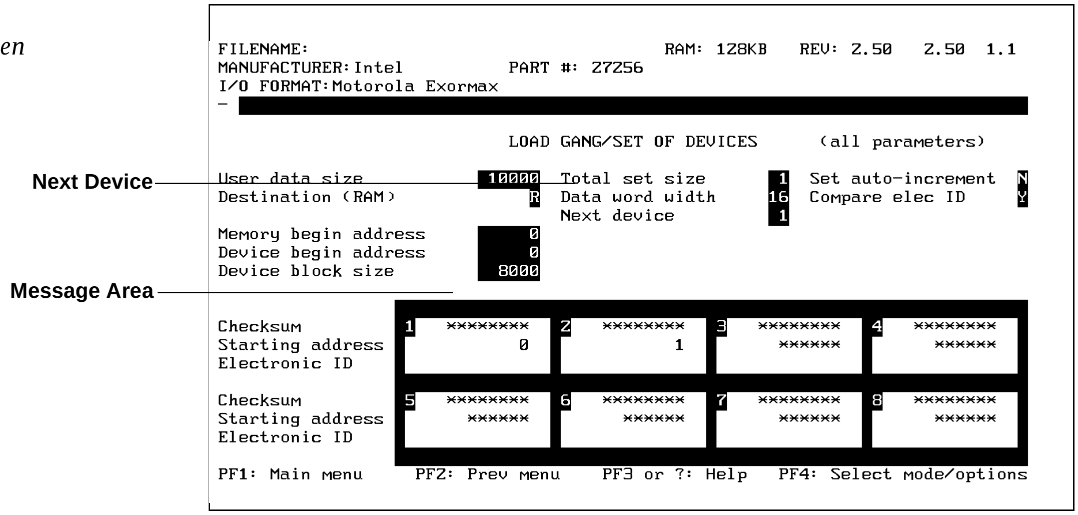
Pressing the PF4 key allows you to select between the "non-default" or "all parameters" screens for either the PSM or the FSM. These represent the PF4 key. UniSite will step through these four selections with each depression of the PF4 key. The "ready" LED will light for the module currently selected. Devices should be removed from any module that is not selected.

SetSite Screens

When the SetSite module has been selected, the SetSite screens will appear. These screens are different from the screens for other modules. This section describes the unique features of the SetSite screens.

Reverse video blocks at the bottom of the SetSite screen represent the eight SetSite sockets. Each block displays information for the socket it represents: error status, checksum, starting RAM address, and electronic ID information are displayed here.

Figure A-23
The SetSite Screen



```
FILENAME:                                  RAM: 128KB   REV: 2.50   2.50  1.1
MANUFACTURER:Intel         PART #: 27256
I/O FORMAT:Motorola Exormax
-

                           LOAD GANG/SET OF DEVICES      (all parameters)

User data size        10000 Total set size       1  Set auto-increment  N
Destination (RAM)         R Data word width     16  Compare elec ID     Y
                            Next device          1

Memory begin address      0
Device begin address      0
Device block size      8000

Checksum          1 ********  2 ********  3 ********  4 ********
Starting address           0           1      ******      ******
Electronic ID

Checksum          5 ********  6 ********  7 ********  8 ********
Starting address      ******      ******      ******      ******
Electronic ID

PF1: Main menu    PF2: Prev menu    PF3 or ?: Help    PF4: Select mode/options
```

Information shown in each of the blocks will differ depending on what kind of operation you are performing. If you are doing a Load operation, the eight blocks show how UniSite's user memory will be allocated. If you are programming devices, the reverse video blocks show how the user data will be organized into the parts.

After any device operation (such as loading or verifying data) has completed, messages will appear in the area above each block. The messages indicate what the device's status is. PASS will appear above all successfully programmed parts. Any programming errors that might occur will display above the block where the error occurred. A list of SetSite's status and error messages can be found at the back of this Appendix.

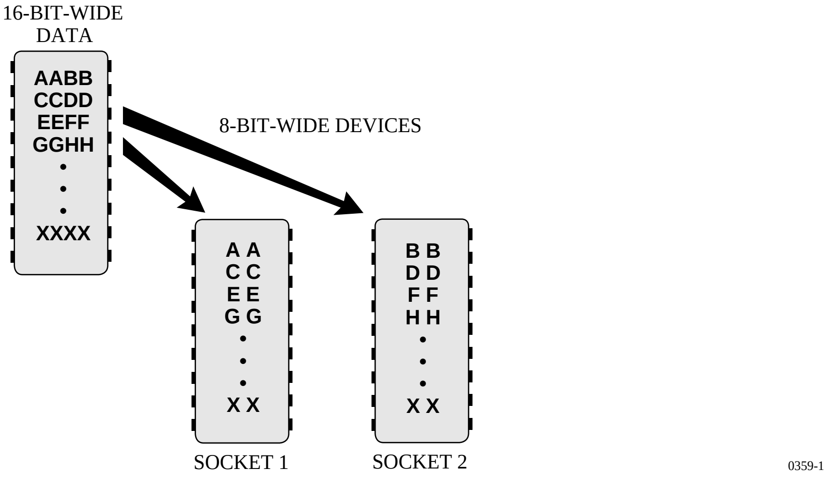
The "Next device" parameter on the screen can be used to visually scroll through the data organization. This feature is useful if you are programming partial sets of devices, or if you are programming a set of more than eight devices. For example, if you are programming a set of 20 devices and want to check status of organization in the ninth device of the set, move the cursor to the Next device parameter and type in **9**. UniSite will redisplay the status data showing device number 9's status data in socket number one.

Set and Gang Programming

Set Programming

A "set" is one or more devices that contain a unique block of data—regardless of how many devices that data is programmed into.

Figure A-24
Set Programming



SetSite can program any set size from one to 99. If you choose a set size such that the number of devices is less than or equal to 4, then multiple sets can be programmed on the SetSite module simultaneously. The way that data will be arranged in the devices programmed is determined by three factors: the device word width, the data word width, and the number of devices you want to have in each set. The general equation to use is:

$$\text{Set Size} = \frac{\text{Device word width}}{\text{Data word width}} \text{ x Number of devices}$$

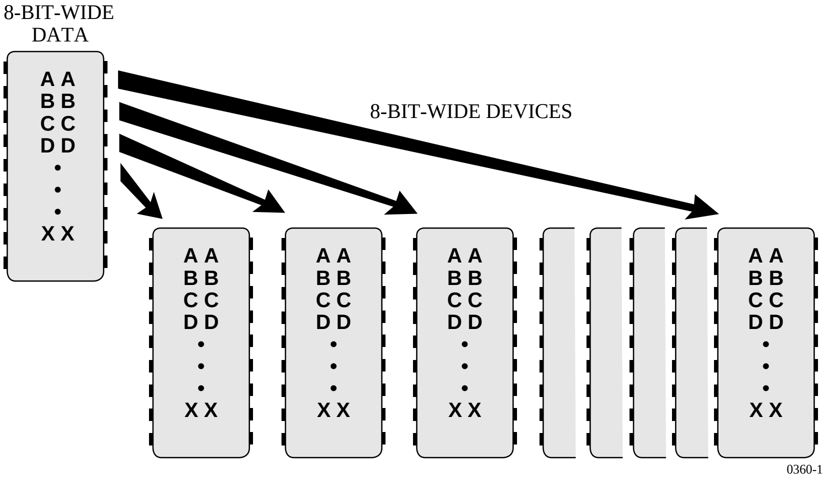
For example, if you are programming two 27128 devices, which are 8-bit-wide (device word width) parts, and your programming data is constructed 16-bit-wide (data word width) in the target system, the set size [8/16 x 2] equals one. The byte swap flag is enabled for this example.

Gang Programming

“Gang” programming simply means that the same data will be programmed into each device: the set size is therefore equal to one (and the device word width should equal the data word width).

If you want to perform **gang set programming** operations with SetSite, you will need to specify a set size of **one** in the Programming data entry screen.

Figure A-25
Gang Programming



SetSite Applications (Legacy)

This section includes examples of how UniSite and SetSite might be used in a programming environment. Three examples are given here:

- Gang programming eight 27128 devices—programming the same data into all eight parts.
- Set programming devices—programming devices so that each ends up with unique data.
- Programming a partial set of devices—using only part of the original block of data to program the devices.

Gang Programming

This example shows how to load data from a master (already programmed) 27128 device into RAM and then gang-program that data into a set of eight blank devices.

1. Power up UniSite with the SetSite module installed.
2. Using the Select device menu, select any 27128 EPROM from the manufacturers' and device type lists. When you have chosen a part, UniSite will return to the Main menu.
3. Type **L** to go to the Load device menu. The Load Gang / Set of Device screen should appear. If it does not, press the **PF4** key until the Load Gang / Set screen appears. The "READY" indicator will light yellow on SetSite's panel when SetSite has been selected.
4. The 27128 is a 16K x 8 part: there are 16K (or 4000 hex) addressable locations. Therefore, the device block size should be set at 4000. Check the device block size window and make sure that UniSite has set the device block size to 4000 and data word width to 8.
5. Check the User data size window. Make sure that the user data size is set to 4000.
6. Insert the master 27128 device into SetSite's socket number one, the socket at the upper-left of the module. The device should be inserted bottom-justified: no open socket pins below the device. Push the socket lever up to the closed position.
7. When you are certain the displayed screen parameters are correct, push the socket lever forward again to the START position (or press **ENTER** on the terminal keyboard) to start loading the data. The action symbol on the UniSite screen will rotate; when the operation is finished, a message will appear. Because the socket lever is spring-loaded, it will automatically return to the closed position.
8. The screen will now show the data just loaded. The total set size will be 1, meaning that the entire set of data can be programmed into one device. A "0" appears in socket number one next to the starting Address parameter, meaning that data was loaded from socket one beginning at user memory address 0. Notice that asterisks appear inside the other device socket boxes on the screen: it is not possible to "gang-load" data into RAM with the set size equal to 1 (gang mode).
9. Press **PF2** to return to the Main menu. Open the sockets and remove the master device. Data is now loaded into RAM. The next part of the procedure explains how to program that data into the eight blank devices.
10. Type **P** to go to the Program Gang / Set of devices menu.
11. Insert the eight devices into the SetSite sockets. Push the socket lever up to the closed position. Notice that starting address 0 appears in the boxes for all eight devices on-screen. This means that UniSite will take data from user memory address 0 to program each device, so they will all end up with the same data.
12. Push the socket lever forward to the START position (or press **ENTER** on the terminal's keyboard) to begin programming the devices.

13. The action symbol on the UniSite screen will rotate. When the operation is finished, a message will appear. “PASS” will appear on the SetSite screen above all sockets whose devices have been successfully programmed.
14. The devices are now programmed. Pull the socket lever to the open position and remove the eight devices.
15. If you want to program eight more devices with the same data, simply repeat steps 11 through 14.

Set Programming

This example shows how to load data from a master (already programmed) 27512 device (containing a whole 64K set of data) into RAM, and then set program that data into eight blank 27256 devices (4 sets). Because the 27256 has half the memory of the 27512, it will require two 256s to store an entire SET of data from a 512. The set size, therefore, equals two. If eight devices are programmed, four sets (of two 27256s) will be produced.

The 27512 is a 64K x 8 part: it has 64K (or 10000 hex) addressable locations. The 27256 devices have 32K (or 8000 hex) addressable locations. So that you do not have to worry about calculating and entering memory block parameters for two different types of devices, UniSite will automatically alter device parameters when the devices are ready to be programmed.

1. Power up UniSite with the SetSite module installed.
2. Using the Select Device menu, select any 27512 EPROM from the manufacturers’ lists. When you have chosen a part, UniSite will return to the Main menu.
3. Type **L** to go to the Load Device menu. The Load Gan / Set of Device screen should then appear. If it does not, press the **PF4** key to display the different Load screens until the correct one appears. The “READY” indicator on SetSite’s panel will illuminate when UniSite is ready.
4. Make sure the device block size window displays 10000, so that the device block size equals the selected device’s size.
5. Make sure the User data size window displays 10000 and the Data word width shows 8. The set size will be 1, meaning that the entire set can be programmed into one device. A “0” appears in socket number one, next to the starting Address parameter, meaning that data will be loaded from socket one beginning at user memory address 0. Notice that asterisks appear inside the other device sockets. It is not possible to “gang-load” data into RAM. Only one device’s data may be loaded at a time with a set size of 1.
6. Insert the master 27512 device into SetSite’s socket number one. The device should be inserted bottom-justified: no socket pins below the device. Push the socket lever up to the CLOSED position.

7. When you are certain the displayed screen parameters are correct, push the socket lever forward to the START position (or press **ENTER**). The action symbol on the UniSite screen will rotate. When the operation is finished, a message will appear.
8. Pull the socket lever back to the open position and remove the device.
9. Press **PF2** to return to the Main menu. Data is now loaded into RAM. The next part of the procedure explains how to program that data into the eight blank devices.
10. Type **S** to enter the Select Device screen. Select the 27256 device from the Select Device menu. UniSite will then load the correct programming algorithm for the devices.
11. Type **P** to go to the Program Gang / Set of Devices menu.
12. Even though you will be programming data into devices that are only half as big as the master, you will NOT need to change the Device block size parameters: UniSite does it automatically when you select the device.

Note: UniSite now displays “Set Size 2.” This means the entire set of data from the 27512 can be programmed into two devices. Notice also the memory allocation shown at the bottom of the screen: starting address 0 is shown for the odd-numbered sockets; address 8000 is displayed for the even-numbered sockets. This display means that data from UniSite’s RAM address 0000 to 7FFF will be programmed into half of the devices (those in odd-numbered sockets); data at addresses 8000 to FFFF will be programmed into the other four devices (those in even-numbered sockets).

13. Insert the eight devices into the SetSite sockets. Push the socket lever up to the CLOSED position.
14. When you are certain the displayed screen parameters are correct, push the socket lever forward to the START position (or press **ENTER**) to begin programming the devices.
15. The action symbol on the UniSite screen will rotate. When the operation is finished, a message will appear.
16. The devices are now programmed. Pull the socket lever back to the open position and remove the eight devices.

Programming Partial Sets of Devices

Some applications call for programming only part of a device set. UniSite and SetSite are designed to allow partial device operation with a minimum of effort on your part.

Reprogramming Devices

Say you are programming a set of eight devices, and two of the eight in the set do not program successfully. To reprogram the two devices, this is what you would need to do:

1. Move the socket lever back to the OPEN position and remove the six programmed parts and the two bad devices.
2. With NEW devices installed in the sockets positions of the two that failed, push the socket lever up to the START position (or press **ENTER**), initiating the programming operation.

It does not matter which two sockets the devices are in; the six LEDs adjacent to the empty sockets will remain off. You also do not need to change any of the data block limits: UniSite will automatically assume you want to attempt programming the SAME data into the socketed parts.

Programming Set of More Than Eight Devices

Programming using a set size of more than eight devices may also be considered “partial” set programming, since the set size is greater than the number of devices that SetSite can program simultaneously.

Devices in sets larger than eight can be programmed in two or more “partial” sets. UniSite can be viewed as being able to program a “window” of eight devices at a time, out of the total number in the set. You can position which eight devices to program using the Next Device parameter mentioned in the Introduction to SetSite. By setting the Next Device parameter to 1, 9, and 17, you could program successive windows of user data (eight devices at a time).

Note: You can accomplish the same thing using the Auto Increment feature. If Auto Increment is set to ***Y****, UniSite will increment the Next Device parameter for you.*

For example, say you want to program 20 devices. With Next Device set to 1, devices 1 through 8 would be programmed. After these first eight devices were programmed, you could set the Next Device to 9, and then program devices 9 through 16. The last four devices could be programmed after Next Device was set to 17.

The four empty sockets remaining when devices 17 through 20 were programmed would NOT cause an error to be generated; the socket LEDs would just remain off. If you use the Set Auto Increment feature, UniSite will reset the Next Device window to 1 after these last four devices are programmed.

Preventive Maintenance for Modules (Legacy)

Error Conditions

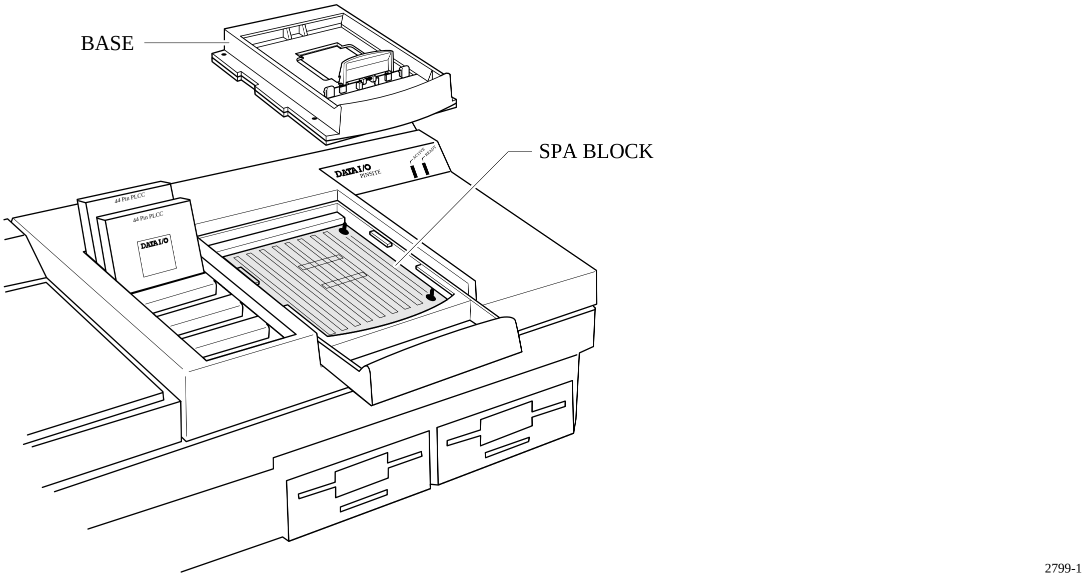
The following messages during device operations could result from dirt in the SPA block or base adapter.

```
ID Error
Continuity Error
Base Adapter not Installed
Device Insertion Error
Overcurrent Error
Base / Adapter Relay Failure
```

SPA Block

To avoid error conditions caused by dirty or worn SPA blocks, we recommend that you perform the following preventive maintenance procedures.

Figure A-26. SPA Block and Base



- Keep the SPA block covered when it is not in use.
- To prevent base adapters from contaminating the SPA block, store them in an uncontaminated area.
- Each time you remove or replace a programming module, clean the SPA block with a brush.
- Inspect the SPA block for dirt weekly. Perform the following steps to clean the SPA block:
    1. Blow filtered compressed air across the SPA block.
    2. Mildly dampen a small section of a lint-free cloth with a DeoxIT pen (Data I/O P/N 570-5500-901) and gently rub the dampened cloth across all the pins on the SPA block.
    3. Using a clean section of the lint-free cloth, gently wipe the surface again.

4. To ensure that all pins are properly positioned in their receptacles, push a programming module down on the SPA pins a few times and check that the pins spring up to their normal upright position.

Bases

Inspect the bases for dirt weekly. Clean the surface with compressed air.

To prevent base adapters from contaminating the SPA block, store them in an uncontaminated area.

Each time you remove or replace a PPI base, clean the spring pins.

Conductive Pad

After 1000 device insertions or once a month, whichever comes first, clean the conductive pads as described on page A-20.

B. Making Your Own Cable

Overview

You will need one 25-pin RS-232C serial cable for each piece of equipment you will connect to the programmer.

Normally, when you connect equipment to a programmer, you must match Data Terminal Equipment (DTE) to Data Communications Equipment (DCE).

The programmer is compatible with both types of equipment and automatically configures the Terminal and Remote ports to be compatible with the equipment connected to it.

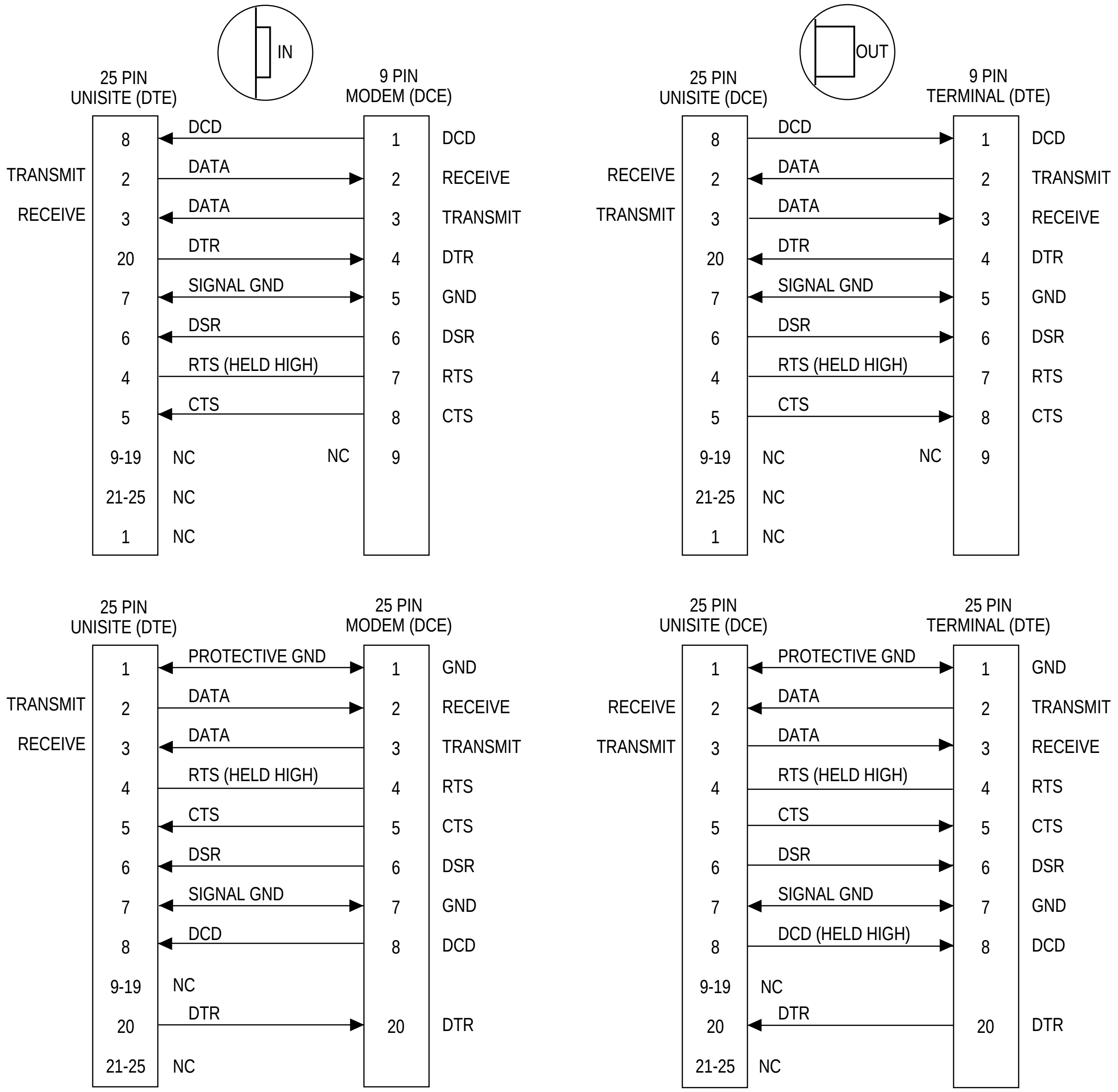
SmartPort

If you are not sure what type of equipment you have, don't worry. The programmer's SmartPort feature will automatically toggle between the two types until a connection is established. For now, just connect the cables. If you don't have 25-pin serial cables, or if you are not sure if they will work, you can use the cabling diagrams below to build your own cable.

Making Your Own Cable

The programmer receives commands and sends responses through an RS-232C port using a 25-pin D connector in two possible configurations: either DTE or DCE. The connections are shown in Figure B-B-1.

Figure B-1
Pin Designations for RS-232C Serial Port Connection



The minimum hookup includes Pins 2, 3, and 7.
Pins 1 and 7 are tied together.

0362-4

Switching Modes

As shown in Figure B-B-1, the switches on the programmer’s back panel toggle each port between DTE and DCE. For now, do not change the settings of the switches. For information on when to use the DTE/DCE switches, see “Install the Base” on page 2-15.

Pin Functions When In DCE Mode

The following table explains the function of the pins on the Terminal and Remote ports when connected to DCE equipment.

Pin	Function	Description
1	Ground	Provides a safety ground connection.
2	Transmit Data	Carries the transmitted data.
3	Receive Data	Carries the received data.
4	Request to Send	This line is held high by the programmer.
5*	Clear to Send	A high on this line enables the programmer to transmit data. (Used for hardware handshaking.) A low inhibits data transmission from the programmer.
6*	Data Set Ready	This line is held high when the remote source is ready to send or receive data. A low inhibits data transmission from the programmer.
7	Signal Ground	Provides a reference ground for all signals on the cable.
8*	Data Carrier Detect	This line is held high when the modem detects a carrier. A low on this line inhibits the programmer from transmitting data.
9-19	No Connection	
20	Data Terminal Ready	This line is pulled high by the programmer to indicate it is ready to receive data. This line is pulled low to signal the remote computer to stop sending data. (Used for hardware handshaking.)
21-25	No Connection	

* *If these lines are not connected, the programmer will consider them high and will function normally*

Pin Functions When In DTE Mode

The following table explains the function of the pins on the Terminal and Remote ports when connected to DTE equipment.

Pin	Signal	Description
1	Ground	Provides a safety ground connection.
2	Receive Data	Carries the received data from the DTE device to the programmer.
3	Transmit Data	Carries the transmitted data from the programmer to the DTE device.
4	Request to Send	This line is held high by the programmer.
5	Clear to Send	A high on this line from the programmer means that it is ready to receive data. (Used for hardware handshaking.)
6	Data Set Ready	This line is held high when the programmer is ready to transfer data.
7	Signal Ground	Provides a reference ground for all signals on the cable.
8	Data Carrier	This line is held high by the programmer.
9-19	No Connection	
20*	Data Terminal	A high on this line enables the programmer to transmit data. (Used for hardware handshaking.) A low inhibits data transmission from the programmer.
21-25	No Connection	

* *If this line is not connected, the programmer will consider it high and will function normally.*

Keep Current Subscription Service

The Keep Current™ subscription service keeps your programmer up-to-date with the latest features and device support. Periodic Update Kits, shipped up to three times a year to Keep Current subscribers, incorporate all changes since the previous update. In addition to receiving Update Kits, Keep Current subscribers have unlimited access to new and improved programming algorithms via the Data I/O Web site at **www.dataio.com**.

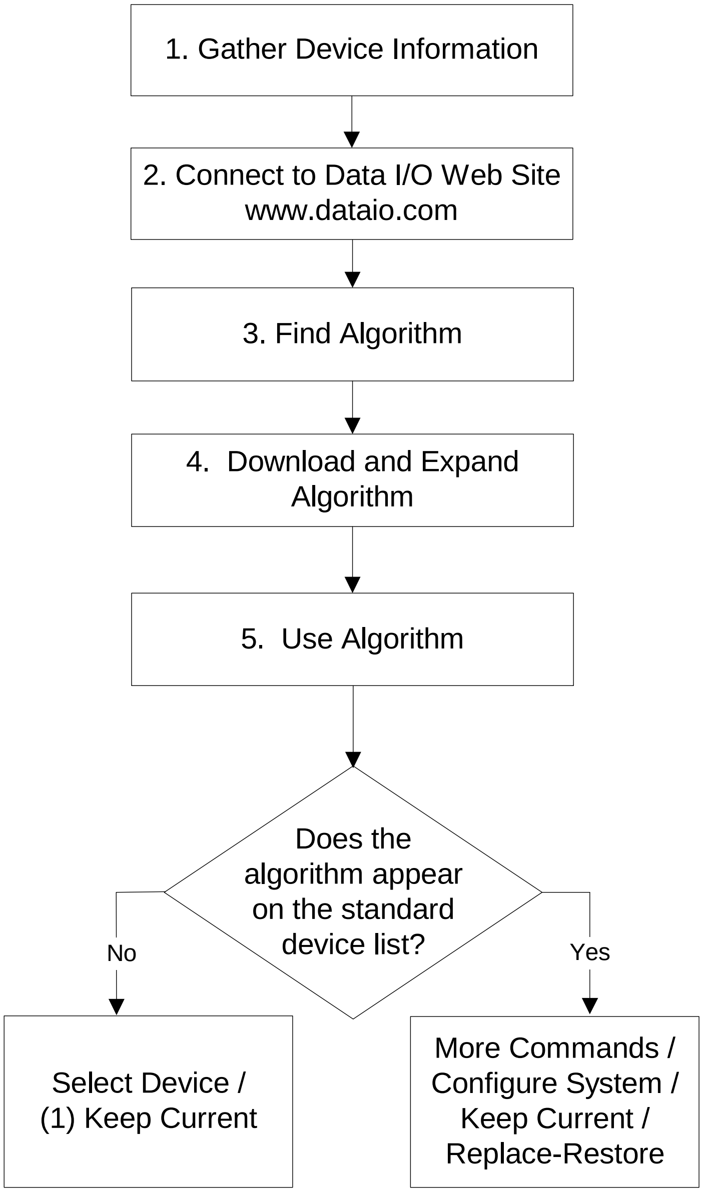
Semiconductor companies constantly introduce new devices and issue specification changes for existing devices. Incorporating these changes swiftly into your programming system ensures that you obtain the highest programming yields and best device reliability possible.

Computer Requirements

To access and download the Keep Current files, you need the following:

- The ability to create 3.5-inch DOS disks, 720KB.
- The ability to connect to the Data I/O Web site through the Internet.

Procedure Overview


1. Gather Device Information
2. Connect to Data I/O Web Site www.dataio.com
3. Find Algorithm
4. Download and Expand Algorithm
5. Use Algorithm
Does the algorithm appear on the standard device list?
No
Yes
Select Device / (1) Keep Current
More Commands / Configure System / Keep Current / Replace-Restore

1. Gather Device Information

Knowing the following information about the devices you will be programming enables you to find the correct algorithm once connected.

- Manufacturer (*example*: AMD)
- Device name (*example*: 27c1024)
- Package type (*example*: 48-pin PLCC)
- Current version of the programmer software (*example*: 6.7)

2. Connect to the Data I/O Web Site

The Data I/O Home Page is located at **www.dataio.com**. From the Home Page, click **Device Search**.

3. Find Algorithm

From the Device Search application, enter a generally descriptive part name in the “Device Part Number” field. For example, if searching for an algorithm for an AMD 29LV160BT-E, begin with “LV160” (without “ “).

Note: Searching with the full device name printed on top of the device may yield poor results. Begin your search with general device names. If your search returns too many devices, narrow your search by adding characters.

Select **UniSite** as the programmer for this search. Check “List devices with Keep Current algorithms only.” Click “Begin Search for Devices.“

In the list of devices returned, locate the exact device and click the link in the “Keep Current” column (*example*: KC6.6). The number following KC indicates the system software version at which this algorithm was released. Algorithms are designed to work with a specific version of system software. For example, KC6.6 works on a UniSite-*xpi* with system software version 6.6 installed.

System software and algorithm files are compatible if:

- All digits to the left of the decimal point are the same, and
- The first digit to the right of the decimal point is the same. (Any additional digits, if present, do not have to match.):

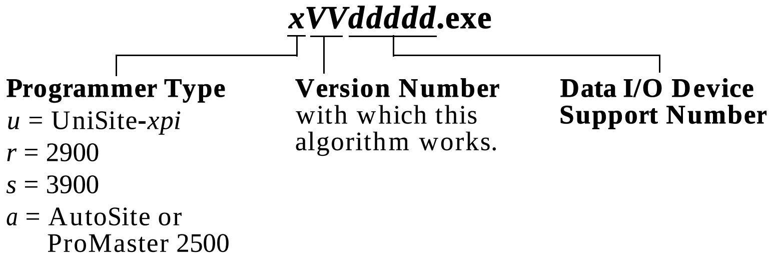
System Version	Algorithm Version	Compatible?
6.50	6.50	Yes
6.51	6.50	Yes
6.50	6.51	Yes
6.50	6.40	No

If a Keep Current file was available at one time but it isn't there any more, the algorithm has been incorporated into a newer version of software and you need to update your programmer.

4. Download and Expand Algorithm

Filenames

Keep Current filenames are represented as follows:



***xVVddddd*.exe**

Programmer Type
u = UniSite-*xpi*
r = 2900
s = 3900
a = AutoSite or
ProMaster 2500

Version Number
with which this
algorithm works.

Data I/O Device
Support Number

Files Created

Algorithms come in a self-extracting file format. When you download a Keep Current algorithm, the following files are created:

File Name	Description of File
xVVddddd.KCx	Algorithm
xVVddddd.txt	Instructions on use
adapters. sys*	System adapters
devfnote.sys*	Device notes

* *Optional file*

5. Use Algorithm

Before you save a Keep Current algorithm to a floppy disk, determine whether or not it appears on the programmer's standard device list and follow the directions in the appropriate section below.

On Standard List

If the algorithm is listed in the standard device list, in terminal mode use **More Commands / Configure System / Keep Current / Replace-Restore.** This command adds the Keep Current algorithm to the device list, ensuring that the latest algorithm is available to all programmer users.

From this point on, the operation is the same as using a regular algorithm.

Not on Standard List

If the algorithm is not listed in the standard device list, in terminal mode use **Select Device / (1) Keep Current**. The algorithm can be selected from the Select Device menu, but it cannot be added to the device list.

From this point on, the operation is the same as using a regular algorithm.

Save the File to Disk

Place the Keep Current file on a floppy disk that has been formatted on your programmer.

Label the disk **Keep Current** and specify the version number to avoid mismatched software version numbers if you need to use the disk again.

You can also create these files in a sub-directory and transfer the files to a disk using your programmer in terminal mode. Use the **More Commands/Transfer/Download** screen, select format **16** (Absolute Binary), and select **Disk** as the destination.

Sample Keep Current Scenario

The following example illustrates a typical Keep Current scenario:

1. In May, you update your system software to version *X*.8. At the same time, you enroll in the Keep Current Subscription Service.
2. In June, Cruft Technologies announces a new device, the Cruft 1263.
3. A week later, Data I/O announces support for the Cruft 1263 and places a Keep Current algorithm for the Cruft 1263 on the Data I/O Web site.
4. The next day you connect to the Data I/O Web site and download the new algorithm for the Cruft 1263.
5. In August, Data I/O releases version *X*.9 system software, complete with the new algorithm for the Cruft 1263.
6. You update your programmer to version *X*.9 system software. The algorithm for the Cruft 1263 is part of the system software.

Index