USER MANUAL

For the

ChipWriter

PC Based Universal Device Programmer

DATA I/O

Data I/O Corporation
10525 Willows Road N.E., P.O. Box 97046
Redmond, Washington 98073-9746 USA

Acknowledgments:

Contents

6. Memory & Microcontroller programming

7. Programmable Logic

1. Introduction

Congratulations on choosing a Data I/O programmer. We are sure that you will find the programmer software very easy to use and full of features to support all of your device programming requirements, whether you are using it for development, field service, or low-medium scale production. With your new programmer you will be able to

- ✓ Read data from programmed devices
- ✓ Load data in from disk
- ✓ Edit data
- ✓ Program a device
- ✓ Verify the contents of a device

There are many other features of your programmer that will allow you to achieve the results you need. This manual explains these features and how to use them.

1.1 About this manual

This manual provides the instructions for using the ChipWriter programmer from Data I/O. It is organized so that it is possible to program a device quickly by following one of the *QuickStart* tutorials. A reference section follows which gives instructions in greater detail.

It is recommended that you read through the relevant sections of this manual before using the programmer, and in case of any difficulties you encounter during use.

1.2 Conventions

Throughout this manual, certain conventions will be followed in typefaces in order to make instructions clear.

For example, in the line

Type `EPROM <ENTER>`

`EPROM` : this font is used throughout the manual to indicate something that should be typed into the computer

`<ENTER>` : Any items encased in <> indicate the name of a key to press, i.e. in this case the ENTER, or RETURN key.

Menu selections are indicated by words separated by forward slashes, e.g. `FILE/OPEN` indicates the menu option FILE, followed by the sub-menu option OPEN.

Messages displayed by the programmer software are indicated in italics, for example:

`DEVICE PROGRAMMED AND VERIFIED`

1.3 Contents of the package

When you open the package you should find :

- ChipWriter 48-pin Universal device programmer
- A CD ROM containing the software for the programmer.
- User manual
- A power supply

- A D25 male - male parallel cable
- Free software may be included in the package at the discretion of Data I/O. These packages may assist you in product development. Please see the READ.ME files on the individual disk(s) for information on how to use these packages and who to contact for further information.

The programmer can be operated from batteries. These are not included as standard. Use 8 alkaline AA cells (standard or rechargeable).

1.4 Low Voltage Features

The ChipWriter programmer supports Low-Voltage devices, including 1.8V and 3.3V devices, as well as standard 5V devices. It is capable of programming and verifying devices down to 1.8 Volts. When using low voltage devices, the programming algorithm in the system will automatically be set to the correct voltages as specified by the manufacturer's programming algorithm.

2. Installing the Software

2.1 Latest software

Both device programming algorithms and system operating firmware for this new line of programmers is distributed via the BBS/WEB page. Since software algorithms are added on a regular basis, it is possible that the software provided with your new system has been superseded by a newer revision.

To insure that you have all the latest device programming algorithms an system software installed on your programmer, it is recommended that you contact the Data-IO BBS/WEB page.

Contact information is given in Appendix C of this manual.

2.3 Running the Install program

An install program is provided on the CD ROM. This software copies all of the necessary files onto your hard drive. We suggest that you use the following directories to hold the programmer software:

```
C:\DATAIO\CWRITER
```

Where C: is the name of your hard drive

So, if you are installing the software onto the C: drive, to run the installation program place the CD-ROM in the drive and log onto this drive by typing **E:** or **F:** as necessary. Now type:

```
E:INSTALL C:\DATAIO\CWRITER <ENTER>
```

After installation you should add this directory to your current PATH statement in your AUTOEXEC.BAT file. For example change

```
PATH=C:\DOS;C:\WINDOWS
```

to

```
PATH=C:\DOS;C:\WINDOWS;C:\DATAIO\CWRITER
```

Then reset the PC.

3. Connecting the Programmer

3.1 Computer requirements

The programmer will operate with any IBM compatible PC (386 and above recommended) with a standard IBM DB25 parallel interface (parallel printer port). This means that it is very easy to move between PCs. Users who prefer to have an internal card or who do not wish to use their existing parallel port can fit an additional parallel card (available from any PC supplier). Alternatively, a manual switch box may be used.

Please note that the programmer is designed to use the Standard Printer Port configuration (SPP). Users with port settings in their BIOS, or on Enhanced parallel port cards, should NOT select the EPP or ECP options.

The software is compatible with MS-DOS v.3.0 or later, and with MS-Windows, v.3.1 or Windows 95/98/ME.

3.2 Connection to the PC

STEP 1 - Connect one end of the parallel cable to the programmer. The other end of the cable should connect to the PC. The cable will only fit the DB25 Parallel port. Tighten the holding screws thus ensuring that the cable will not cause programming problems.

NOTE: The programmer does not have a power on switch. It will power up when a signal is received from the computer, and automatically power down when not in use.

STEP 2 - Connect the power supply provided to the programmer and plug into any standard electrical outlet. Alternatively, fit 8 alkaline AA batteries.

USE ONLY THE POWER SUPPLY ADAPTER PROVIDED

Replacements and adapters for different countries (240V, 220V, 100V, 110V) can be supplied by Data I/O or their distributors.

STEP 3 - Switch on the PC.

STEP 4 - Run the SELFTEST program.

3.3 Running the SELFTEST program

Your programmer has been supplied with self-testing software which can check that the programmer has been installed correctly and can be used for fault diagnosis. Before using the programmer for the first time, whenever moving it to a new PC or if any fault is suspected, run the **SELFTEST** program. Should you prefer, a simple batch file can be written so that the Selftest is performed before using the programmer software. This Selftest has two purposes :

- It confirms which parallel port is being used by the programmer
- It checks most of the programmer hardware

WARNING :
Always ensure that there are no devices in the programmer before running Selftest, as leaving a device in the socket could damage the chip and give false results

To run the program, connect the programmer to the PC and either mains or battery power, remove any devices and type:

SELFTEST <ENTER> at the DOS prompt.

The SELFTEST will report which port the programmer is connected to, the programmer's Firmware version number and run a communications test. It will then run through a series of hardware checks.

If any error is reported, then you will need to contact Data I/O Technical Support or your distributor. In order to assist with your problem, it is possible to obtain a printout of the Selftest results by using:

```
SELFTEST >PRN
```

where PRN is the name of the resident printer device (i.e.. LPT1).
or `SELFTEST >ERROR.LST`
followed by `PRINT ERROR.LST`

See Appendix C for Technical Support information

3.4 Operating from batteries

The ChipWriter programmer can be operated from an A/C power supply or batteries. To operate from batteries use 8 alkaline AA type cells. The battery cover is held in place with screws. When replacing the cover, take care to use only the screws provided, and not to over tighten the screws.

NOTE: Certain Bipolar devices require high currents in very short bursts, and therefore some batteries may not always be able to handle these devices.

3.4.1 Recharging the batteries

The programmer can be used as a battery charger. In order to start recharging the batteries the programmer needs to be connected to a PC and connected to the electrical supply using the adapter provided.

To charge the batteries, from the DOS prompt, type: `BCHARGER`
usage: `BCHARGER[port]CHARGEMODE[REPORTMODE]`
where:

`port` : parallel port to use (default = 1)
`CHARGEMODE TRICKLE/SLOW/FAST` : (default=slow)
`REPORTMODE START/REPORT` : (default=both)
`START` : starts the charging and does immediate return
`REPORT` : reports current stage of charging

When the batteries are charging the 'BUSY' LED on the programmer will flash red and yellow. Once charging is complete this LED will switch off.

Once recharging has commenced the programmer can be disconnected from the PC. Charging will stop once the batteries have been fully charged. If the PC is switched off battery charging will be interrupted for a few seconds. Recharging will resume automatically.

Warning: Do not attempt to charge non-rechargeable batteries

3.5 Programming non-DIP devices and devices with over 48-pins

Devices in packaging other than DIP, and devices over 48-pins require socket adapters. A range of adapters is available from Data I/O for most common devices. If you are unsure as to which adapter to use, please contact Data I/O or your distributor for advice.

Adapters from other sources which are simple pin conversions can usually be used successfully. However, for devices over 48-pins, pin mappings are not usually standard, and it is therefore necessary to use an adapter

specifically designed for the programmer in use. Where devices require non-standard adapters to be supported, these are available from Data I/O. All adapters available from Data I/O are standard across the ChipWriter range.

3.5.1 Adapter usage

Insert the adapter into the ZIF socket of the programmer with the bottom end of the adapter in the bottom pins of the socket. The text on the adapter boards should be facing up. Lock the ZIF socket handle in place. For PLCC adapters, the notched corner of the adapter socket will usually be in the top left hand corner. Devices are placed in the socket in the ‘live-bug’ position, i.e. with legs downwards for all sockets 28-pins and above. 20-pin sockets are "dead-bug" loading. Push the device down to lock in place, then push down the socket sides to release.

See figure 3.1 below.

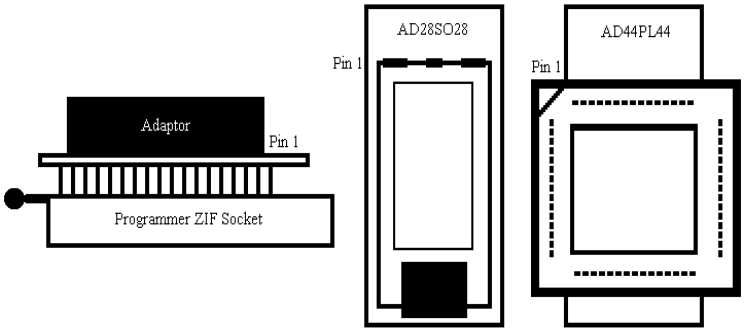

Figure 3.1 - Adapter Usage

4. Getting Started

We have provided some Quickstart tutorials to help you start using the programmer as quickly as possible. If you have not used a programmer before, these may help you to begin. However, these should not be taken as a comprehensive guide to the programmer's capabilities. The following tutorials are provided:

TUTORIAL 1. Programming an EPROM or Microcontroller from a master device.

TUTORIAL 2. Programming an EPROM or Microcontroller from a file on disk.

TUTORIAL 3. Programming a logic device from a file on disk.

These tutorials are provided to get you using the programmer quickly for simple tasks. However, we recommend that you read the reference sections of the manual if you have any difficulty and before attempting any more complex operations.

4.1 Before You Start

If you are not sure how to select menu items, please see section 5.3 for more information .

Figure 4.1 below illustrates the orientation of chips in the programmer. The chip is located at the bottom of the ZIF socket with the bottom pins next to the ZIF socket handle.

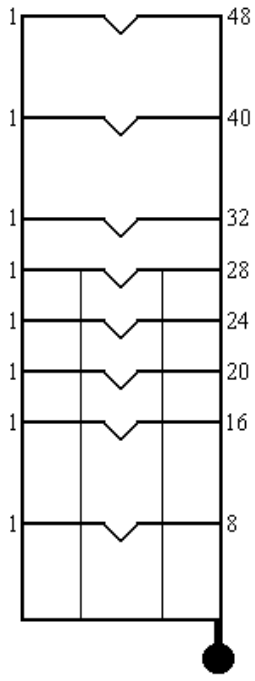

Figure 4.1 - Device Location

If you are using a socket adapter, please see the instructions with the adapter for how to place the chip in the socket.

4.2 Quickstart tutorials

TUTORIAL 1:

Programming an EPROM or Microcontroller from a master device.

1. Type CWEPROM <ENTER>
2. Select the Device you are using as a master, using DEVICE/TYPE. Select the device type you are wishing to use and press <ENTER>. You will then be given a list of manufacturers who support the selected device type. Select the appropriate manufacturer and press <ENTER>. You will then be prompted for the device name. Choose the correct device name from the menu you are given and press <ENTER> to accept. The details of the device you have chosen will now be displayed at the bottom of the screen.
3. Ensure that the data memory buffer size is large enough to hold the data to be read from the master device. You can change the buffer size using DATA/BUFFER-SIZE. See section 6.2.1 for more information.
4. Insert the master device in the programmer.
5. Select PROCESS/READ to read the contents of the chip into the data memory buffer. Press <F10> to accept the default addresses.
6. Select the device you are copying to if it is different (see 2.)
7. Now place the blank copy device in the programmer socket.
8. Select PROCESS/PROGRAM. Press <F10> to accept the default addresses.
9. The message: *Device programmed and verified* will be displayed.

TUTORIAL 2:

Programming an EPROM or Microcontroller from a file on disk

1. Type CWEPROM <ENTER>
2. Select the Device you are using as a master, using DEVICE/TYPE. Select the device type you are wishing to use and press <ENTER>. You will then be given a list of manufacturers who support the selected device type. Select the appropriate manufacturer and press <ENTER>. You will then be prompted for the device name. Choose the correct device name from the menu you are given and press <ENTER> to accept. The details of the device you have chosen will now be displayed at the bottom of the screen.
3. Ensure that the data memory buffer size is large enough to hold the data to be read from the master device. You can change the buffer size using DATA/BUFFER-SIZE. See section 6.2.1 for more information.
4. Select FILE/OPEN
5. Specify the path and filename where your data is located.
6. Select the file format used (HEX auto-recognition will detect any of the available HEX formats)
7. Press <F10> to accept the default settings and load the data. A checksum will be displayed on the screen.
8. Select the device you are copying to if it is different (see 2.)
9. Place the blank device in the programmer socket.
10. Select PROCESS/PROGRAM. Press <F10> to accept the default addresses.
11. The message: *Device programmed and verified* will be displayed.

TUTORIAL 3:

Programming a logic device from a file on disk

1. Type CWPAL <ENTER>
2. Select the device you are going to program using DEVICE/MANUFACTURER. Select the manufacturer and press <ENTER>. Choose the device name from the list on the screen by highlighting the correct device and pressing <ENTER>.
3. Select FILE/OPEN.
4. Specify the path and filename where your JEDEC file is located.
5. Press <F10> to accept the defaults. A checksum will be displayed once the file has loaded.
6. Place the blank device in the programmer socket.
7. Select PROCESS/PROGRAM. The device will now be erased (where relevant), programmed, verified and any test vectors present in the JEDEC file will be applied.
8. The message: *Device programmed and verified* will be displayed.

5. Using the Programmer Software

5.1 Basic software configuration

The operating software contains the following programs.

CWEPROM.EXE : Used to program the various memory and microcontroller device types.

CWPAL.EXE : Used for programmable logic devices.

CWCTEST.EXE: Used to identify and test TTL and CMOS logic devices.

These are menu driven programs which contain all the features required to program and test devices. The actual programming information for individual devices is held in library files, which allows easy updating of the software to accommodate new devices and algorithm updates.

In order for these programs to run correctly, the appropriate .INI files must be configured correctly. If the software has been installed as described in section 2.3, the default .INI files will contain the correct path and directory information required by each of the programs. If however, on running EPROM for instance, the message:

Cannot find database. Modify the ini file

is displayed, then the [Directory] section of the `CW_EPROM.INI` file is not pointing to the correct path. If either the PAL or EPROM libraries are not correctly mapped, after selection of a device within the software, the first selection off the `PROCESS` menu will display the message:

Bad or missing library file

Using any ASCII editor, edit the CW_EPROM.INI file. If the software has been installed on C:\DATAIO\CWRITER, the [Directory] section of the .INI file should read:

```
[Directory]

Memory Libraries=C:\DATAIO\CWRITER\Libs\Memory
Pal Libraries=C:\DATAIO\CWRITER\Libs\Pals
Pars=C:\DATAIO\CWRITER\Pars
```

Batch software is also included which allows the programmer to be operated from DOS batch files. Details of the commands available are shown in section 8.

For some programmable logic devices, separate programs are supplied to handle non-JEDEC standard files.

5.2 Screen layout

Each program displays information as follows :

Menu headings are displayed across the top of the screen
Device programming details are displayed in the bottom window.
Data memory contents are displayed in the main window.

The software and library version numbers are displayed at the top of the screen, just below the Menu bar. These are useful to note should you need to contact Technical Support or to check for new version numbers on the BBS or Internet.

5.3 General menu selection

The software has been designed to make the programmer as simple as possible to use. All functions are contained within the menu system. The menus are obtained by pressing the

<ESCAPE> key. The user can then move around the menus by pressing the cursor keys. To select an option press <ENTER>.

Each menu function is available by pressing the first letter of the function. For instance, the DEVICE pull-down menu is available by pressing the 'D' key. Some options can be selected by pressing **ALT+letter**. These are indicated by a triangle next to a letter on the menu display. For example, Program can be selected by pressing **ALT+P**. A full list of shorthand keystrokes is given in Appendix A.

5.4 Dialogue boxes

The menu system makes extensive use of dialogue boxes. Within these boxes the following prompt types may appear :

Prompt	**:**	requires you to type in some information
	>	will display a series of options when you press <ENTER>.

Inside the dialogue boxes use the cursor keys to move between options; use <ENTER> to select the contents of the dialogue box. Use <F10> to accept all chosen options within the window. If you wish to abort from the window, press <ESCAPE>.

5.5 Changing the settings

Before starting, you may wish to change any of the following. They are found in the OPTIONS menu. This will have to be done for both CWEPROM and CWPAL :

OPTIONS/PARALLEL PORT NO
Selects parallel port to be used. This auto-detects each time that the software is accessed when the port setting in the INI file does not agree with the actual port setting. The setting is then stored in the appropriate INI file on exit. Once the parallel port has been chosen it will be remembered when next using the programmer.

OPTIONS/COLOR MODE
To choose whether the display is MONO or COLOR

OPTIONS/SNOW PRECAUTIONS
Usually set to OFF but set it to ON if you have problems with snow on your monitor. (affects older PC's with CGA)

OPTIONS/HI-RES MODE
Select ON for 40/50 line mode. Effective only with EGA and VGA monitors.

5.6 Using a mouse

All three application programs can be mouse driven. However, your mouse driver software must previously have been installed. This is normally done automatically in your CONFIG.SYS file (by loading a device driver such as MOUSE.SYS) or in your AUTOEXEC.BAT file (by invoking a mouse driver program such as MOUSE.COM). If you are unsure how to do this please refer to your system manual or the manual supplied with your mouse. **CWEPROM.EXE, CWPAL.EXE** or **CWCTEST.EXE** automatically enable the mouse if one is installed (see later if you wish to disable it).

5.6.1 Left mouse button (LMB)

The LMB is used to select items in a list, such as a menu selection or a device name from the device list. Simply move the pointer to the required item and press the LMB. If the item is a menu heading, such as FILE, PROCESS etc., then this will open up that particular menu. If the item is a sub-menu or part of a list this will highlight the chosen item. To select the highlighted item Double Click on the LMB. Double Clicking performs the same action as pressing <ENTER>.

5.6.2 Right mouse button (RMB)

Pressing the RMB is exactly the same as pressing **<ESC>**.

5.6.3 Middle mouse button (MMB)

Pressing the MMB is exactly the same as pressing **<F10>**. When this key is a possible input, the window being displayed will have "F10" in the bottom border. Pointing at the text "F10" and pressing the LMB will also produce the same result (for two button mouse users).

5.6.4 Disabling the mouse

To disable the use of the mouse invoke each of the programs with the /NM option. e.g.

```
CWPROM /NM
```

5.7 Edit and view modes and scroll Bars

In order to observe data in the data buffer area without risking modification a VIEW mode is available in the DATA menu. This will only allow scrolling, paging etc. Mouse users can enter this mode directly by pointing at the buffer window and pressing the LMB. In order to make moving around the buffer much easier, Scroll Bars have been added to the window borders.

5.8 Running the software under Windows

All three programs can be run in a DOS window under Windows 3.1x or Windows 95. Example .PIF and .ICO files are included. To create an ICON for the ChipWriter - CWEPROM.EXE, for example on Windows 3.1, do the following:

- Select NEW PROGRAM ITEM in the Program Manager FILES/NEW menu.
- Change Description to CWPROM
- Change Command Line to: C:\DATAIO\CWRITER\CWEPROM.PIF
- Change Working Directory to:

C:\DATAIO\CWRITER

- Select CHANGE ICON and change the file in the CHANGE ICON prompt to:
  C:\DATAIO\CWRITER\CWEPROM.ICO

or `C:\DATAIO\CWRITER\EPROM1.ICO`

- Select `OK`

Repeat for CWPAL and CWCTEST. Use Optional Parameters if required. The mouse functions should work in a DOS window in 386 enhanced mode in Windows 3.1x or Windows 95, but if you have trouble see section 10 of the Windows readme file.

5.9 Using the programmer software over a network

It is possible to run the programmer software from a network hard drive with the programmer itself connected to the local computer. However, the appropriate .INI files will have to be modified in order to map the PAL and EPROM libraries, and PARS sub-directory to the correct full path name. The **CWPROM.EXE** and **CWPAL.EXE** files use their respective .INI files to locate the Devices menu and device libraries.

If the `PARS` sub-directory is not correctly mapped, on running the software, the message:

`Cannot find database. Modify the .ini file` will be displayed.

If either the PAL or EPROM libraries are not correctly mapped, after selection of a device, the first selection off the `PROCESS` menu will display the message:

`Bad or missing library file`

5.10 Using EMS software

The programmer is supplied with EMS software which will enable expanded memory to be accessed if it is available. This requires an expanded memory manager compatible with EMS 3.2 standard. It allows up to 8 Mb of memory and will run on any PC with an expanded memory board or a 286 or 386 with an expanded memory manager. The software will not give the EMS option if a driver such as EMM386.EXE is not installed correctly.

If the EMS function is working correctly, it will be possible to increase the data memory buffer size to 1024K without problem. If this is not possible, and the maximum it can be set to is 384K for instance, then the EMM386 driver has not been correctly installed. Check the CONFIG.SYS file in the root directory on your hard drive to check that the following line exists:

```
DEVICE=C:\DOS\EMM386.EXE
```

Also ensure that the **NOEMS** option has **not** been included on this line. DOS maybe replaced by WINDOWS if you are running under Windows 95. It is also possible to run MEMMAKER and select YES when prompted for EMS memory requirements.

5.11 Saving software settings

When exiting the programmer software, settings such as parallel port, the device chosen, etc. are automatically saved to avoid having to repeatedly set up the chosen options.

It is also possible to save the whole of a device environment, including full file load and programming options, in a **Task** file. See section 6.2.8

6. Memory & Microcontroller Programming

This section covers the Memory and Microcontroller devices which are programmed using **CWEPROM.EXE**.

To use this program, at the DOS prompt, type:

```
CWEPROM <ENTER>
```

A HELP screen is available by typing:

```
CWEPROM /help <ENTER>
```

6.1 Device Selection

In order to tell the programmer software which programming algorithm to use, it is necessary first to select the device which is being used.

6.1.1 To select a device

To do this, enter **CWEPROM.EXE** and select

```
DEVICE/TYPE
```

You will then be given a choice of device types to select. Choose between:

```
EPROM's
EEPROM/FLASH/NVRAM's
Serial (E)EPROM's
Microcontrollers
BPROM's
```

Highlight the type required or press the first letter, and press <ENTER>.

A list of manufacturers will then be displayed on the screen who support the selected device type. Use the up and down cursor keys to find the manufacturer of the device you are using. You can also press the initial letter of the manufacturer you need, e.g. After selecting EPROM's, pressing M will take the cursor down to MACRONIX. When the cursor is highlighting the required manufacturer, press <ENTER>.

The software then displays a list of all the supported devices of the manufacturer and type selected. Highlight the required device name and press <ENTER> again.

The details of the selected device will then be displayed in the bottom part of the screen. The appropriate device library file name will be displayed at the top of the screen, below the menu bar, together with the library version number.

6.1.2 Changing the device selected

Once a device has been selected, to select another device of the same manufacturer and type, press:

```
DEVICE/DEVICE NAME
```

To select another manufacturer of the selected device type, select:

```
DEVICE/MANUFACTURER
```

6.1.3 Which device?

Generally it is necessary to select the device name which is as close as possible to that stamped on the chip you are using. For example, a PIC16C54 microcontroller in windowed DIP package will be marked PIC16C54/JW, whereas an OTP version could be marked PIC16C54-XT/4. The software uses the correct nomenclature according to the manufacturer's datasheet. Speed and package indicators are only required when they affect programming algorithms.

6.1.4 Low Voltage devices

Low voltage devices are selected exactly as other devices. In the device information in the bottom window, you will notice that the Vcc voltages during program and verify (VccP & VccV respectively) will be a combination of 5V, 3.3V and/or 1.8V. As most programmers are still unable to provide 1.8V and 3.3V circuits, manufacturers still allow the devices to be programmed at 5V. The ChipWriter programmers can not only verify a low voltage device at its nominal operating voltage, they can also program with voltages down as low as 1.8V as well, giving much greater flexibility.

6.2 Opening data files

Files can be loaded from disk into the data buffer and saved to disk after editing using the `OPEN` and `SAVE` options in the `FILE` menu. See section 6.2.8 for more details on saving files.

6.2.1 Setting the data memory buffer size

It is necessary to ensure that the data memory area has been set to a size large enough to accept the contents of the file to be loaded. Do this using

`DATA/BUFFER SIZE`

Enter the required size in Kbytes. The maximum buffer size available is determined by the available memory of the PC. The programmer software uses approximately 150K plus the chosen buffer size. The EMS software will use expanded memory for the buffer before using conventional DOS memory. If the buffer

cannot be increased sufficiently see section 5.10 for information on installing EMS.

6.2.2 Opening a file

Select

`FILE/OPEN`

This then displays a number of different options:

`FILE NAME` : Enter the required filename
`FILE TYPE` : Select the required file format (see section 6.2.3)
`FILE WINDOW LOW ADDRESS` : Enter the required file start address. Default = 0
`FILE WINDOW HIGH ADDRESS` : Enter the required file end address. Default = FFFFFFFF
`FILE INCREMENT` : Select from Every Byte, Every 2nd Byte etc. upto Every 4th Word
`BUFFER INCREMENT` : Select from Every Byte/Word etc. upto Every 8th Byte/Word

BUFFER START ADDRESS : Enter the required address.
Default = 0
PRE-FILL BUFFER WITH FF : Select ON or OFF. ON will destroy all current buffer contents.

Pressing <F10> accepts the parameters and opens the selected file, displaying the checksum when loaded (See 6.2.5).

6.2.3 File format

The file type can be selected by pressing <ENTER> to see the options available. Choose from:

```
RAW/BINARY
ASCII-SPACE-HEX
ASCII-OCTAL
HEX-AUTO RECOGNITION
HEX-MOTOROLA S1, S2 OR S3 RECORDS
HEX-INTEL MCS86
HEX-TECTRONIX
HEX-EXTENDED TECTRONIX
HEX-TI TAG
```

Choosing HEX-AUTO RECOGNITION is generally the easiest way of loading a HEX formatted file. However some linkers put additional headers at the start of the HEX file which can confuse the loader when using this option. Hence the other options.

ASCII-SPACE-HEX embodies the formats of ASCII space HEX, ASCII ‘ HEX etc. The software assumes that the file consists of HEX digits separated by non HEX characters. Checksum information is not looked for and no separators are needed.

e.g.. If a file consisted of A BCD EF this would appear as

Address	0	1	2	3
Data	0A	BC	0D	EF

NOTE: When using the PICALC assembler to produce files for Microchip PIC microcontrollers, use the option -F INHX8M to save the file as an 8-bit Intel-Hex file, then load as HEX-Auto Recognition. Any line within your code which sets the format to anything else will NOT be overridden by entering the INHX command at the command line.

6.2.4 Handling large files and devices

In order to facilitate the programming of memory devices and the loading of files whose size is greater than the maximum buffer size the FILE/OPEN command (ALT-O) has the following parameters:

```
FILE WINDOW LOW ADDRESS (FWLA)
FILE WINDOW HIGH ADDRESS (FWHA)
FILE INCREMENT (FINC)
BUFFER START ADDRESS (BADD)
```

A byte is ONLY loaded into the buffer if its FILE ADDRESS is inside the window limits defined by the above (inclusive) range. Data at FWLA is loaded into the buffer at BUFFER START ADDRESS. The FILE INCREMENT works within this window. See Figure 6.1 below.

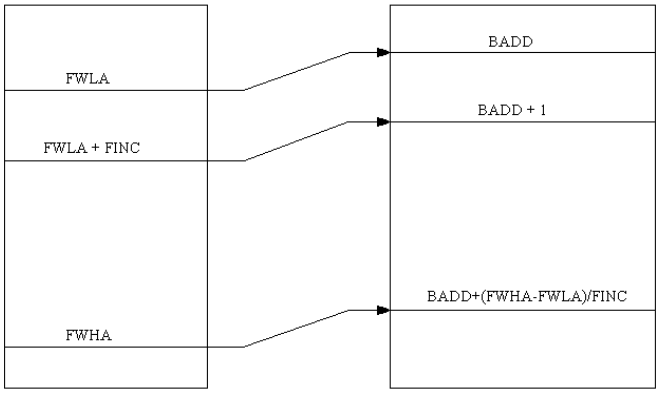

Figure 6.1 - Handling Large Files

If you have file loading problems, check the above parameters. Normally FWLA=0 and FWHA=FFFFFFFF (i.e. the whole file is to be loaded). One point to note is that these parameters are stored each time the software is quit. If in a previous session FWLA was changed to 10000 (HEX), then files with information in the range 0-FFFF will appear empty. Therefore, it is important to check all the load parameters each time a file is opened.

6.2.5 File checksum

When a file has been opened and loaded into the data buffer, a checksum is calculated on the contents of the file and displayed on the screen.

This is displayed as three figures:

a) The sum of unsigned bytes with the carry being added to the MSB of the checksum.
b) The complement of (a)
c) The arithmetic negative of (a)

NOTE: The checksum displayed on `FILE/OPEN` is the checksum of the FILE only, and does not necessarily include the whole address range of the selected device. A checksum of the device, or of a particular address range, can be calculated using `DATA/CHECKSUM` as explained in 6.3.5.

6.2.6 Wild card loading

Wild cards may be used in the filename to perform drive/directory searches. Wild cards consist of * and ?.

* matches to any number of characters
? matches to any one character

A list of matching filenames is displayed alphabetically (Sub-Directories first) and the user may select using the cursor keys or the first letter of the name.

6.2.7 Saving a file to disk

Data edited in the buffer area or read in from a device can be saved to disk in a number of different formats. It is recommended that frequently used files are saved in binary mode to save disk space and reduce loading time.

`FILE/SAVE AS`

Displays two options:

SAVE ALL DEVICE AREAS:
Saves the whole of the device address range and, where applicable, the configuration byte stored at the end of the device address range.

SAVE MEMORY BLOCK:
Saves a user defined address range from the data memory area.

Both options allow the file to be saved in a number of formats:

```
RAW/BINARY
ASCII-SPACE-HEX
ASCII-OCTAL
HEX-MOTOROLA (S2 format)
HEX-INTEL
```

To save any modifications to the currently loaded file, simply select:

FILE/SAVE

6.2.8 Saving configuration

Using the SAVE TASK command from the FILE menu saves a set of system parameters to a Task file for future use. The system parameters include the currently selected device, the open data file, the file parameters, and the device options.

If you are using microcontrollers, or memory devices with configuration options, all of the configuration bits can be saved in the Task file as an extra byte at the end of the data file, or at the location specified in the manufacturers programming algorithm. This means that the next time the device is used, all of the configuration options have been configured exactly as they were the previous time, saving time, and reducing the chance of operator error.

To implement this feature, choose your required device and open the required data file, and select PROCESS/PROGRAM. All of the programming options are then displayed. Select all of your required configuration options and press <F10>. Even with no device in the socket, the data buffer will be modified with the appropriate configuration byte. On a National Semiconductor COP878x device, this is stored in location #1000, with an AMD 29F040 Flash PROM, this would be stored at address #80000.

When you have the program configured the way you want it, select:

FILE/Save Task

Type in the required Task filename, prefixed with the appropriate drive and directory in which you want the configuration file to be saved. By default, the extension .TSK will be appended to the filename (IC Environment). Press <F10> to accept the filename.

To load a previously saved Task file, select:

FILE/Load Task

The Task filename is then entered. Press <F10> to accept, and the whole device configuration is loaded, including the data file and the whole of the program parameters. Alternatively, use:

EPROM TASKFILE.TSK <ENTER> from DOS.

6.3. Using the editing facilities

As well as loading from disk or reading a pre-programmed device, data to be programmed can be entered directly into the buffer from the keyboard. In addition, files can be amended using the DATA menu options.

6.3.1 Entering EDIT mode

To edit the contents of the data buffer, select

DATA/EDIT

This allows you to type in HEX or OCTAL code, or ASCII characters. Once EDIT is selected, the cursor will move to the HEX display in the buffer window. Use the <TAB> key to switch between HEX and ASCII. In HEX mode, only keys 0-9 and A-F are allowed. In OCTAL mode only keys 0 to 7 are allowed. The cursor keys can be used to move around the buffer.

6.3.2 Viewing without making changes

If you wish to view the buffer contents beyond the first screen without danger of making unintentional changes to the data, use

DATA/VIEW

This allows you to view the data contents as HEX or OCTAL code or ASCII characters. Once VIEW is selected, the cursor will move to the HEX display in the buffer window. Use the <TAB> key to switch between HEX and ASCII.

6.3.3 Quickly moving around the data

The <HOME> and <END> keys allow you to move more quickly. Press once to move to the beginning (end) of the line, twice to move to the beginning (end) of the page or three times to move to the beginning (end) of the file.

When you have finished editing or viewing the buffer, press <ESCAPE> to return to the main menu.

6.3.4 Quick data editing commands

DATA/FILL

Allows you to fill a defined area of the data area with information in byte, word or long word format. Enter the buffer start and end addresses and the information to be inserted. Use less than 3 characters for a byte, 3 or 4 for a word, more than 4 for a long word. Press <F10> to accept.

DATA/COPY

Copies a defined block of buffer memory to a new location.

DATA/SEARCH

Searches for data in HEX, OCTAL or ASCII format. Use the Up/Down arrow in the dialogue box to move between HEX, OCTAL and ASCII data entry.

DATA/JUMP TO

Jumps to a chosen address in the data area

DATA/SWAP BYTES

Swaps odd and even bytes within a defined range. Useful for 16 bit wide EPROM's.

6.3.5 Data checksum

The command:

DATA/CHECKSUM

Gives the option of performing a checksum on either the contents of a user defined range in the buffer, or the selected device.

Selecting MEMORY BLOCK CHECKSUM provides an option box:

BUFFER START ADDRESS : default to 0
BUFFER END ADDRESS : default to end of buffer

CHECKSUM TYPE : Select between:

```
Sum of Bytes
Sum of Words
Sum of Long Words
CRC 32 bit
```

Choosing DEVICE CHECKSUM just gives the option of entering a device START and END address.

Whichever option is selected, the checksum is displayed as three figures:

a) The sum of unsigned bytes in the memory range with the carry being added to the MSB of the checksum.
b) The complement of (a)
c) The arithmetic negative of (a)

NOTE: The memory area can be larger than the size of the opened data file. If there is a discrepancy between the expected and calculated checksums, recalculate specifying the start and end address of the file. Checksums for Microcontrollers are displayed as defined by the manufacturer's data sheet or algorithm.

6.3.6 Data display modes

The contents of the buffer can be displayed in 8-bit HEX, 8-bit OCTAL, 12-bit (16-bit storage, LSB = low address) or 16-bit mode (LSB = low address). To change the display, select:

OPTIONS/DISPLAY MODE

and select the display mode required.

6.3.7 Printing the data contents

A print-out of all or part of the data memory contents can be obtained by selecting:

DATA/PRINT

Specify the start and end addresses of the block to be printed, and the name of the output device (e.g. LPT1).

6.4 Reading and checking devices

This section explains how to read the contents of a programmed device, verify the contents against known data, check the device signature and identify a device from its signature, and perform illegal bit checks and blank checks on devices.

6.4.1 Reading a device

To read the contents of a device, select the correct device name from the DEVICE menu as described in section 6.1 and place the device to be read in the programmer socket. If you do not know the device name, you can use **DEVICE/QUERY EPROM** to identify the device in many cases.

Please note that some devices CANNOT be read, either because the chip type itself is designed not to be read for security purposes, or because the on-chip security features have been enabled when programming.

Before reading EPROM's, ensure that the buffer size is set large enough to contain the data (see section 6.2.1 on setting the buffer size). **NOTE:** A 16-bit device with a maximum device address of 128K will require 256K bytes of buffer space.

To read the contents, or part of the contents, of a device, select

`PROCESS/READ`

The programmer software then quickly accesses the library file for the programming algorithm required. With memory devices, a dialogue box appears requesting Device Start and End address, and Buffer start address:

`DEVICE START ADDRESS:`
Enter the first address from which you wish to start reading data from the device. Default = 0.

`DEVICE END ADDRESS:`
Enter the last address from which you wish to read data from the device. Defaults to the highest address available for the selected device.

`BUFFER START BYTE ADDRESS:`
Enter the buffer address where you wish to begin reading the data into. Default = 0.

Press `<F10>` to accept the options displayed and the device will then be read. If the selected device is a microcontroller, then the entire contents of the device are read and placed into the appropriate memory. The contents of the device will then be visible in the main data buffer, and a checksum will be displayed.

Once the contents of a device have been read, they can be edited using the `DATA/EDIT` option, and then saved to disk, or programmed into another device. See section 6.3 for more detail on editing data.

6.4.2 Verifying the contents of a device against a file

To verify the contents of the selected device against known data, first ensure that the buffer size is large enough to handle the file. Then open the file against which you need to verify the data. Place the device to be verified in the programmer and select:

`PROCESS/VERIFY`

If the data matches, the message ***`VERIFY OK`*** will be displayed. If there is a mismatch, the message will indicate the first address which has failed to verify.

6.4.3 Verifying the device type

Many newer devices contain Electronic Signature codes which enable you to electronically identify the device. It is possible both to verify a known device type against this ID code. The ID code is also used by your programmer to check that the correct device type has been selected before proceeding with any process that may damage a chip. However, not all devices contain this ID code, and the checks for electronic signature may themselves damage a device without a signature, or one of a different number of pins from that expected. This means that these checks cannot be foolproof and need to be used with some caution.

To verify the signature of a device, select

`PROCESS/VERIFY SIGNATURE`

If the database contains a manufacturer's code and device code they can be verified against that of the device.

6.4.4 Checking if a device is blank

In order to program a device, it must either be fully blank, or be capable of containing the information to be programmed into it in addition to the information it already contains. In order to test whether a device can hold additional data, open the required data file and select:

`PROCESS/ILLEGAL BIT CHECK`

This checks to see if a non-blank device can be programmed with the data currently held in memory. For example, with an EPROM whose unprogrammed state is #FF, #F0 may be programmed to #00 but not to #0F.

In order to test whether a device is completely unprogrammed, select

`PROCESS/BLANK CHECK`

This option checks if the device is blank (all 1's or all 0's depending on the actual part).

NOTE: It is possible for a device to appear blank even though it is not fully erased. If problems are encountered in programming a device, try erasing it for a longer period and then reprogramming.

Either the Blank Check or Illegal Bit Check can be performed when you select `PROCESS/PROGRAM`. This is selected from `OPTIONS/PROCESS SETTINGS`. See section 6.5.1.

6.5 Programming devices

IMPORTANT NOTE

Data I/O uses only the specified programming algorithms as provided by the semiconductor vendors. While every effort is made to ensure that programming algorithms are correct, the said algorithms are not guaranteed by the manufacturers and therefore cannot be guaranteed by Data I/O.

Users are advised to ensure that devices work correctly in their systems before programming large quantities.

6.5.1 Programming options

Prior to programming, there are a number of different options available which can be selected using `OPTIONS/PROCESS SETTINGS`. These are generally more applicable when using the programmer for larger quantity programming.

`SERIAL NUMBERS` : For production programming, it is often a requirement to have a unique serial number programmed into a designated location(s) on the selected device. The `PROGRAM` process implements the serialization options selected here. (See 6.5.2)

Select from:

```
None
Sequential Hex: LSB in low memory
Random Hex
Date, Weekday, Time in HEX:YYYYMMDDWDHHMMSS
Sequential BCD: LSB in low memory
Random BCD
Date, Weekday, Time in BCD:YYYYMMDDWDHHMMSS
```

`SN BUFFER ADDRESS` : Enter the required start address of the serial number.

`SN DIRECTION` : Select `LSB` or `MSB` in Low memory. For example: A 4-byte serial number starting at 0000, being incremented by 1, starting at buffer address 00. Selecting LSB will mean that address 0 is incremented to FF, with address 01 then being the overflow, etc. Selecting MSB will mean address 03 is incremented to FF, with address 02 then being the overflow, etc.

SN STORAGE : Select between NIBBLE, BYTE and WORD.
SN Buffer Increment: Default is 1.
SN LENGTH : Enter the number of bytes required for the serial number - up to 8 bytes.
SN (bytes 0-3) : Least significant 4 bytes (8 digits)
SN (bytes 4-7) : Most Significant 4 bytes (8 digits)
AUTO-DEVICE CHECKSUM : Select OFF, BEFORE SECURING or AFTER SECURING
If turned ON, after a device has been successfully programmed and verified, the device checksum is displayed, either before or after the security fuse has been programmed.
Device Checksum Ignores SN : Select ON or OFF
If a unique serial number is being programmed into each device, the overall checksum will change after each program cycle, invalidating the Checksum as a fail-safe checking procedure. If this option is turned ON, then checksum will be calculated using only the UNCHANGED data.
PROGRAM PRE-TEST :
Select None, USE ILLEGAL BIT CHECK or USE BLANK CHECK. This determines which kind of pre-testing is performed when the Program option is selected.
ASK FOR PARAMETERS ON REPEAT : Select ON or OFF
With this option set to ON, after each program cycle, the Device Programming Parameters box is displayed. If a large quantity of devices are being programmed with exactly the same information, then it is often desirable not to display the parameters box once the device has been configured correctly. Selecting OFF also ensures protection against accidental alteration to the programming parameters.
FORCE MARGINAL VERIFY : Select ON or OFF
After any program operation, an automatic verify pass is performed. This is at the nominal operating voltages as specified by the manufacturer which may be a combination of 5V, 3.3V and/or 1.8V. Selecting ON provides additional verify passes at +/-5% of each of the designated verify voltages. See VccV for the selected verify voltages in the Device parameters window at the bottom of the screen.

6.5.2 Programming operation

To program the selected device, use

PROCESS/PROGRAM

This displays a box which enables the required Programming Parameters to be selected. This information will differ depending on the type of device and the on-chip functions supported.

The most basic configuration will offer the following options:

DEVICE START ADDRESS : Enter the address where you wish the programming operation to start. Defaults to 0.
DEVICE END ADDRESS : Enter the last address of the device you need programming. Defaults to the device end address.
BUFFER START ADDRESS : Enter the address of the first byte in the buffer from where you want to start programming. Default is 0.

BUFFER INCREMENT : Default is Every Byte/Word

Should a Sequential Serial Numbers option have been selected in PROCESS SETTINGS (see 6.5.1), then the Serialization parameters will be displayed here. These can also be modified if required.

SERIAL NUMBER (bytes 0-3): Least significant 4 bytes
SERIAL NUMBER (bytes 4-7): Most Significant 4 bytes

Microcontrollers will generally offer a number of options in addition to the basic device parameters. See 6.6.

To accept the address selections or defaults, press <F10>.

The programmer will first check the device selection and position in the socket, then perform either an illegal bit check or blank check, depending on what has been selected in the PROCESS SETTINGS, and then proceed with programming. Non-blank Flash devices will always be erased during the Pre-Test stage.

While a device is being programmed, a message will be displayed in the center of the screen showing the address which has been reached. Following programming, an automatic verify will be conducted and if all is well, the message *DEVICE PROGRAMMED AND VERIFIED* will be displayed. If AUTO-DEVICE CHECKSUM has been selected, the device checksum will now be displayed.

If you need to program another device using the same configuration, simply press <F10> to repeat the process. If ASK FOR PARAMETERS ON REPEAT has been set to ON, then the programming parameters box will be displayed, which allows you to change any of the options. Again, just press <F10> to continue. If the repeat option is set to OFF, then the parameters box will not be displayed and a single key-stroke is all that is required to repeat the programming cycle.

6.6 Microcontroller programming

Microcontrollers generally have a number of specific features which make their programming different to that of memory devices. The ChipWriter programmers are specifically designed to handle a wide range of microcontrollers, and can generally support all of the optional features which are available on them.

6.6.1 Address locations

The programmer software handles address locations as they appear on the device. For example, default buffer and device start addresses are at the addresses where the data is held in the device, which is not always a single block of contiguous memory. It is not possible to edit device start and end addresses of microcontroller devices.

6.6.2 Security features

For microcontrollers, the security features are supported as part of another option within the Programming Parameters dialogue box through the PROGRAM process. Security status is performed automatically before a BLANK CHECK or PROGRAM. To manually check the security status of a pre-programmed device, use:

PROCESS/SECURITY STATUS

This gives the status of the device security features. The information displayed is representative of that on the device data sheet.

P	represents a security bit which is programmed
U	represents an unprogrammed security bit

IMPORTANT NOTE: Some devices cannot indicate security status when the security fuse(s) has been programmed and in these cases the device may look blank. Thus the device may fail programming attempts until it is erased. If any problems arise with devices that have on-chip security, please erase the device first and retry.

PROCESS/VERIFY ENCRYPTION

If encryption is in effect/relevant this will verify the encrypted buffer data with the contents of the device.

6.6.3 Device specific features

Where device specific features are supported, the options available will be displayed in the Programming Parameters dialogue box after selecting PROCESS/PROGRAM (See 6.5.2). Microcontroller programming options include:

```
Security
Encryption Array
```

```
Oscillator type
Brown-Out protection
Watch Dog Timer
RAM size selection
```

Please refer to the manufacturers datasheets for more specific information on these options. If there are any programmable options which are not included, please contact Data I/O for possible updates.

6.7 Erasing devices

For Flash memory, and some parallel EEPROM parts which support a device erase feature, use

PROCESS/ERASE

This electrically erases the data held on the chip.

To erase all other EEPROM's, as well as NVRAM's, fill the buffer with (FF's) and program the whole device.

7. Programmable Logic

This section covers the Programmable Logic device types which are programmed using **CWPAL.EXE**.

To use this program, at the DOS prompt, type

`CWPAL <ENTER>`

7.1 Device selection

In order to tell the programmer software which programming algorithm to use, it is necessary first to select the device which is being used.

7.1.1 To select a device

To do this, enter CWPAL.EXE and select

`DEVICE/MANUFACTURER`

A list of supported manufacturers will be displayed on the screen. Use the up and down cursor keys to find the manufacturer of the device you are using. You can also press the initial letter of the manufacturer you need, e.g. pressing `N` will take the cursor down to `NATIONAL SEMICONDUCTOR`. When the cursor is highlighting the required manufacturer, press `<ENTER>`.

The software then displays a list of all the supported devices of the manufacturer and type selected. Highlight the required device name and press `<ENTER>` again.

The details of the selected device will then be displayed in the bottom part of the screen.

Note: Some devices which are supported on the programmer will not appear on the menu. This is due to the fact that they do not use JEDEC standard files and therefore are programmed using a separate piece of software. At present this affects Altera and Xilinx EPLD's. See section 7.10 and 7.11 for details of how to use this software.

7.1.2 Changing the device selected

Once a device has been selected, to select another device of the same manufacturer, select

`DEVICE/DEVICENAME`

When a new device is selected which has a different fuse map from the current device, the fuse map displayed on the screen will alter.

7.1.3 Which device?

Generally it is necessary to select the device name which is as close as possible to that stamped on the device you are using. Selecting a different device, no matter how small the difference may seem, may result in the device being damaged. Sometimes, however, a number of devices with different suffixes will come under the same device name in the software. The programmer will check the internal electronic device signature and use the correct algorithm for the device you are programming.

When using an adapter to program a non-DIP package, select the device as normal. Any pin differences are taken account of by the adapter. Use Data I/O recommended adapters wherever possible, especially for devices where the non-DIP package is NOT a straightforward pin for pin conversion of the DIP version.

7.1.4 Low Voltage devices

Low voltage devices are selected exactly as other devices. The program and verify voltages will be a combination of 5V, 3.3V and/or 1.8V. As most programmers are still unable to provide 1.8V and 3.3V circuits, semiconductor vendors still allow the devices to be programmed at 5V. The ChipWriter programmers can not only verify a low voltage device at its nominal operating voltage, they can also program with voltages down as low as 1.8V as well.

7.2 Opening data files

7.2.1 Opening a file

To open a logic data file, first ensure that the correct device has been chosen, then select

```
FILE/OPEN
```

You will then be prompted for the name of the file. The software will expect the file to be in the correct format for the device selected. If there is a mismatch a warning will be displayed. Once the file has been opened, the screen will show the total number of 1's in the file, followed by the checksum. The JEDEC file loaded will include any test vectors that are present. These will be applied automatically when the device is programmed.

7.2.2 File format

Standard JEDEC files can be loaded into the fuse map area. These can be produced by packages such as PALASM™, OPAL™, CUPL™, PLACE™ etc. Contact device manufacturers or for details of programmable logic design packages.

In addition to JEDEC files, Altera POF format files and HEX files for Xilinx devices are accepted by the separate programs provided for these devices (See 7.11 and 7.12).

7.2.3 Wild card loading

Wild cards may be used in the file name to perform drive/directory searches. Wild cards consist of * and ?.

* * matches to any number of characters
* ? matches to any one character

A list of matching file names is displayed alphabetically (Sub-Directories first) and these may be selected using the cursor keys or the first letter of the name.

7.2.4 Saving a file to disk

Files edited in the buffer can be saved to disk using

```
FILE/SAVE AS
```

All current test vectors will be saved along with the fusemap.
If a file has been converted from an old PAL JEDEC format file to a newer GAL using the -conv options, the next time it is loaded it will be in the correct format for the GAL device.

To save any modifications to the currently loaded file, simply select:

```
FILE/SAVE
```

7.2.5 Saving configuration

Using the SAVE TASK command from the FILE menu saves a set of system parameters to a Task file for future use. The system parameters include the currently selected device, the open data file, the file parameters, and the device options.

If you are using logic devices with configuration options, all of the configuration bits can be saved as part of the Task file. This means that the next time the device is used, all of the configuration options have been configured exactly as they were the previous time, saving time, and reducing the chance of operator error.

To implement this feature, use:

FILE/Save Task

Type in the required Task filename, prefixed with the appropriate drive and directory in which you want the configuration file to be saved. By default, the extension .TSK will be appended to the filename (IC Environment). Press <F10> to accept the filename.

To load a previously saved Task file, select:

FILE/Load Task

The Task filename is then entered. Press <F10> to accept, and the whole device configuration is loaded, including the JEDEC file and the whole of the program parameters. Alternatively use:

CWPAL TASKFILE.TSK <ENTER> from DOS.

7.3 Reading and checking devices

This section explains how to read the contents of a programmed device, verify the contents against known data, and perform illegal bit checks and blank checks on devices.

7.3.1 Reading a device

To read the contents of a device, select the correct device name from the DEVICE menu as described in section 7.1 and place the device to be read in the programmer socket.

Please note that some devices CANNOT be read correctly, usually because the on-chip security features have been enabled when the device was programmed.

To read the contents of a device, select

PROCESS/READ

The programmer software then quickly accesses the library file for the programming algorithm required. The contents of the device will then be visible in the buffer, and a checksum will be displayed.

Once the contents of a device have been read, they can be edited using the DATA/EDIT option, and then saved to disk, or programmed into another device. See section 7.5 for more detail on editing data.

7.3.2 Verifying the contents of a device against a file

To verify the contents of the selected device against a known fusemap, first open the file against which you need to verify the data. Place the device to be verified in the programmer and select

PROCESS/VERIFY

If the data matches, the message *VERIFY OK* will be displayed. If there is a mismatch, an error message will be displayed.

7.3.3 Checking if a device is blank

In order to test whether a device is blank, select

PROCESS/BLANK CHECK

This option checks if the selected device is blank (all 1's or all 0's depending on the device).

NOTE: It is possible for a device to appear blank even though it is not fully erased. If problems are encountered in programming a device, try erasing it for a longer period and reprogramming.

7.4 Programming devices

IMPORTANT NOTE

Data I/O uses only the specified programming algorithms as provided by the semiconductor vendors. While every effort is made to ensure that programming algorithms are correct, the said algorithms are not guaranteed by the manufacturers and therefore cannot be guaranteed by Data I/O.

Users are advised to ensure that devices work correctly in their systems before programming large quantities.

7.4.1 Programming operation

In normal circumstances when you do not wish to retain any data which may already be in the device, first select the device required, ensure that the data in the loaded fusemap is correct, and then select

PROCESS/PROGRAM

Should the device have on-chip security features, you will be prompted to select the AUTO-SECURE option, select either ON (all devices will be secured) or OFF (no security on current device). The programmer will first check the device selection and position, then perform a blank check, followed by an Erase (should the device support electrical erase), then proceed with programming. While a device is being programmed, an activity bar will be displayed in the center of the screen. Following programming, an automatic verify will be conducted and if all is well the message *DEVICE PROGRAMMED AND VERIFIED* will be displayed.

7.4.2 Single key programming

If more than one device is required to be programmed in exactly the same way, simply press <F10> at the end of the programming cycle to repeat the process. If AUTO-SECURE is selected to OFF, you are given the option to secure the device before proceeding again by pressing <F10>.

7.5 Erasing devices

For GAL, PALCE, PEEL and other electrically erasable devices, use

PROCESS/ERASE

This electrically erases all data and security settings on the device.

7.6 Device security features

When a programmable logic device has been secured, it is usually NOT possible to check the security status. The device will generally read blank. However, with devices whose Security status can be read, select

PROCESS/SECURITY STATUS

To set the security on by default during all programming, select PROCESS/AUTO-SECURE

7.7 Using the editing facilities

Data to be programmed into programmable logic devices is usually created using a PLD development package such as OPAL™, ABEL™ etc. This takes your logic equations and translates them into a JEDEC file which contains the AND/OR array which will be programmed into the device. In general it will not be necessary to edit this AND/OR array, or fusemap. However, the programmer software provides the option to be able to do this, for example when entering simple designs without a logic compiler, or for educational purposes.

7.7.1 Editing the AND/OR array

When a device is selected, the buffer is automatically adjusted to the correct size and layout for the chosen device. The AND array and OR array (if applicable) are highlighted separately on the display. Product lines are represented horizontally and OR lines vertically.

Relevant keys:

1 - enters a 1 in the fuse map
0 - enters a 0 in the fuse map

In AND array:
`ALT 1` : changes the whole horizontal product line to 1
`ALT 0` : changes the whole horizontal product line to 0

In OR array:
`ALT 1` : changes the whole vertical OR line to 1
`ALT 0` : changes the whole vertical OR line to 0

`DATA/BLANK FUSE MAP`

Sets all fuses to 0 and deletes the test vectors.

`DATA/SET FUSE MAP`

Sets all fuses to 1 and deletes the test vectors.

7.7.2 Moving around the array

`DATA/GOTO`

Moves the cursor to the selected fuse number.

7.7.3 User's Electronic Signature (UES)

`DATA/EDIT UES`

Allows the user to enter the Users Electronic Signature (where relevant). This can be entered as either ASCII or HEX code and the number of bytes depends on the device in use. The UES can be used for stock control, labeling, revision numbers etc. The UES is completely separate from the main array, and in no way affects the operation of a device.

NOTE: Electrically and operationally equivalent devices do not necessarily have the same UES capabilities. For instance, an AMD PALCE22V10 has no UES, a Lattice GAL22V10 does. If your JEDEC file has been compiled for the AMD device, and you are using the Lattice part, a file incompatibility error will be displayed during a `FILE/OPEN` operation. In this instance, the error message can be ignored.

7.7.4 Viewing the fusemap

To view the contents of the fusemap without making any changes, select

`DATA/VIEW`

When viewing is complete press `<ESC>` to return.

7.7.5 Array checksum

The command

`DATA/CHECKSUM`

Performs a checksum on the contents of the buffer. This is displayed as two figures:

a) The number of '1's in the array
b) A simple 2-byte checksum

7.8 Converting PAL files for GAL devices

It is possible to use the programmer to convert old files which have been compiled for non-erasable PAL devices to work in GAL devices. To do this, select a `-CONV` device from the menu, load the file, and select

`DATA/CONVERT FUSE MAP`

This will convert the fuse map from the `-CONV` device to the generic device. This is performed automatically if the `-CONV` device is to be programmed.

For example, if an existing design in a PAL16L8 is to be converted into a National Semiconductor GAL16V8 :

1. `ALT+M` (manufacturer)
2. Select `NATIONAL SEMICONDUCTOR`
3. Select `GAL16V8-CONV` from devices list
4. Select `XPAL16L8` from conversion list
5. Open the file using `FILE/OPEN`
6. Select `DATA/CONVERT FUSE MAP`

Alternatively, select `PROCESS/PROGRAM` to convert and program in one operation. Once a file has been converted it can be saved and used again as a file for the device to which it has been converted, without the need for any further action.

7.9 Test vector editing and operation

Test vectors can be edited or created using the `BUFFER/EDIT TEST VECTORS` command. This will open a window with pin numbers and test vector numbers.

Relevant keys are:

`0`	Forces a logic zero onto pin
`1`	Forces a logic one onto pin
`L`	Expect a logic zero on pin
`H`	Expect a logic one on pin
`C`	(Clock) Apply all other signals with C=0 and then apply C=1 followed by C=0
`X`	don't care
`Z`	Expect high impedance
`N`	Power pin (Vcc or GND). (Sometimes used as don’t care but the test vector loader will issue a warning and

convert all non power pins to X)

ALT+T apply the test vectors.

Standard JEDEC Test Vectors can be loaded into the test vector buffer and applied to the device through the PROCESS/VECTOR TEST (ALT+T) command. Test vectors are applied to all 48 pins at the same time and are applied automatically when using PROGRAM. Any errors are highlighted, displaying exactly which pin in which vector fail. Skew will be determined by the capacitance of the actual device, with only a small skew may causing problems on fast logic devices (faster than 7ns) if asynchronous circuitry is implemented within the PLD.

7.10 Using .POF files for Altera MAX devices

Altera MAX devices do not use JEDEC format files and are therefore not implemented in PAL.EXE. Separate software enables the programmer to handle the .POF files which are used on these devices. If you do not have this software, please request it from Data I/O or download it from the bulletin board or Web Page (See Appendix C).

The software contains drivers for each of the devices supported. To program a device, type in the device name and the file to be programmed;

e.g. `EPM5064 file.pof -pN`

where N is parallel port number (default is lpt1)

POF file editing is not implemented as the POF format data refers to ELECTRICAL ADDRESSES and DATA. Without recourse to the chip architecture data the information in the POF file is meaningless to the end user.

7.11 Programming Xilinx EPLD's

Xilinx Epld's use HEX files rather than JEDEC files, and are not implemented in PAL.EXE. Separate software enables the programmer to handle the HEX files that are created by the Xilinx compiler. Each EPLD has a different .EXE file which operates in the following way. The example shown is for the XI7336. All Xilinx software uses the following syntax.

```
XI7336 [FILENAME] [options]
```

FILENAME is the INTEL HEX file to be used. Only INTEL HEX files generated by the XEPLD development system can be used.

The following options are available :

-READ This can be used to read the Xilinx device in the socket and save the file to disk as an INTEL HEX. This is the default option and can be used in the following ways:

```
XI7336  readfile.hex
```

or `XI7336 readfile.hex -READ`

-BLANK Useful for checking to see if a device is blank. No filename is required. The syntax is therefore:

```
XI7336 -BLANK
```

-VERIFY A verify will be automatically executed after programming but this option may be used to verify the device against an INTEL HEX file. The syntax is:

```
XI7336 vfyfile.hex -VERIFY
```

-PROG This is used to program and verify a device from the INTEL HEX file specified. If the security has been set in the file then this will automatically be programmed. If not then a separate

command with `-SEC` may be executed as shown below.

```
XI7336 prgfile.hex -PROG
```

`-DISPSIG` This allows the user signature to be displayed. This option does not depend on whether the security has been programmed and can be displayed in either case.

```
XI7336 -DISPSIG
```

`-DISPSEC` Allows you to read and display the security status of the device in the socket.

```
XI7336 -DISPSEC
```

`-SEC` This option allows you to program the security bits on the Xilinx device. Once this has been done only the signature can be read from the device.

```
XI7336 -SEC
```

`-Pn` If the programmer is not on Parallel Port 1 then the port number can be set using this option. The default is 1. For example, to read from port 2, you should type :

```
XI7336 testfile.hex -READ -P2
```

8. Command Line Batch Software

In addition to the EPROM menu driven software which is provided as standard with the ChipWriter, batch software is also included free of charge. The batch software enables the programmer to be operated using DOS batch commands, thus eliminating the need for the operator to use the menu software, and reducing the margin for error. In addition, time savings can be made by reducing a series of menu selections to a single command.

The batch software is initialized through the menu software. The procedure is as follows:

1. From DOS, type `CWEPROM <ENTER>`
2. Select the device to be batch programmed (see 6.1)
3. Open the required data file (see 6.2)
4. Program one device, thus configuring the appropriate programming options and device addresses (see 6.5).
5. Save the environment using `FILE/SAVE TASK` (see 6.2.8). This is stored with a .TSK suffix
6. . **At least one device must have been programmed in order for the Task file to save correctly.**
7. `Exit` to DOS
8. To program a device from the DOS command line, with the selected options, type: `EPROM TASKFILE.TSK -PROG`

The `PROCESS` options available are:

```
-PROG
-VERIFY
-ERASE
-BLANK
-BITTEST
-READ
-CHKSUM
```

Each option requires its own command line.

If `-READ` is selected, the file opened within EPROM before saving the Task file will be overwritten with the data read from the device. It is possible to create a blank dummy file which can store all data read from a device. This would also require a new associated Task file name.

Error levels are reported and can be used within a Batch file to branch to a particular option. For instance, a blank check will report either a blank or non-blank device, the results of this would then determine if the device needs to be erased or not before programming.

Batch software example:
This batch file blank checks a device, performs an erase cycle if necessary, then programs & verifies the part.

Device selected: AMD 29F040
File Opened: TESTFILE.BIN
Program options: Device address 0 - 7FFFF, Sector Protect = 0
Task File = AMDTEST.TSK

```
REM    Check that the device is blank
CWEPROM AMDTEST.TSK -BLANK
If errorlevel=0 go to program
REM    now erase a non-blank device
CWEPROM AMDTEST.TSK -ERASE
If error level = 1 go to fail
:program
EPROM AMDTEST.TSK -PROG
If error level = 1 go to fail
echo Device Programmed OK
go to end
:fail
echo Programming error
:end
```

9. Using CWCTEST.EXE

CWCTEST.EXE is a utility which allows a whole range of ICs to be tested. These include 74 series and 4000 series logic devices, SRAMS and DRAM's etc. To run the program type :

```
CWCTEST [P] [/NM] <ENTER>
```

where

`P` is the parallel port number (default = 1)

and

`/NM` disables the mouse

Functional tests can now be made on the more common 7400 and 4000 series devices as well as on static and dynamic rams. A search facility has also been included.

9.1 Testing a device

To test a device :

1. Select the logic family or device to be tested
2. For logic devices, select the device to be tested
3. Place the device in the ZIF socket and select TEST DEVICE.
4. The test vectors for the device will now be applied. If an error occurs then the failed vector will be displayed along with what was actually found when the device was tested. If no errors are found then a message will tell you that the test vectors were OK.

9.2 Identifying a device

If you are unsure of a device then insert the device into ZIF socket. Select one of the logic families from the main menu (It doesn't matter which family is selected, both the 74 and 4000 vectors will be applied). Select `FIND A MATCH`. The search will last for about 5 seconds. If all the vectors for a particular device in the library match those of the device being tested then the device number is shown at the bottom of the screen. In some cases there may be more than one match. The software displays all of the devices whose tested vectors match that of the device under test.

9.3 Adding devices to the user library

Users can create test vectors to test less common devices. This is done in the same way as defined for PAL.EXE (see section 7.9). Ground and Vcc pins can be defined with the G and V descriptors within one test vector. Ground Pins are currently limited (in software) to PIN 20 of the ZIF socket. This will be extended at a later date.

To add a new device to chiptest, select

```
EDIT TEST VECTORS
```

and enter the test vectors required.

Then select

```
ADD TO USER LIBRARY
```

9.4 CWCTEST restrictions

The CWCTEST software works by applying test vectors to the device to be tested. If the chip passes the tests, the device is assumed good, otherwise it is assumed bad. The search mode works by applying test vectors for all devices in the data base and recording all devices that are good.

Herein lies the problem. The chip is only tested at a frequency determined by the rate at which test vectors can be applied to the device. With the programmer hardware this is typically 500 - 3000 test vectors per second (depending on PC). Although this is much faster than most portable chip testers, it is generally much slower than the target hardware. The chip may pass the test vectors but may fail in the target system. Similarly, output loading may be completely different between system and tester.

NOTE: If you suspect a chip is causing problems in your target system replace it. The only test which is guaranteed is one with a negative result.

Testing of RAM's and LSI devices is slightly different. This is done by the microprocessor inside the tester interfacing to the chip directly. RAM tests are effectively performed on a 1MHz bus. These results should therefore be more reliable. However access times are not measured, hence please note the above warning.

9.5 CWCTEST Example

The CWCTEST software allows simple logic gates to be tested quickly and easily.

As an example suppose a batch of 74LS245 devices are suspect. To test the devices run the chiptest software by typing :

```
CWCTEST N
```

where N is the parallel port where the programmer is connected (default is 1 if this parameter is missed out). The main screen now appears listing the types of devices that can be tested. In this case, select the 74 series family which is the first selection in the menu. This can be selected by moving the cursor to the correct line and pressing <ENTER> or by double clicking on this selection with the mouse.

A second menu should now appear on the right hand side of the screen. Choose the SELECT DEVICE option. You will now be given the list of devices supported in the 74 series. Move to 74245 by pressing the cursor keys or by using the mouse to move up and down the scroll bar by the side of the window. Once this device has been selected then information you are ready to test the suspect devices.

Select TEST DEVICE from the left hand menu. The test vectors will now be applied. If the device passes all of the test vectors then a message will be displayed on the screen informing you that the test vectors were OK. If the device fails a particular vector test then the failed vector is displayed on the screen. Press a key to return to the left menu. You could also select FIND A MATCH to apply all of the test vectors to the device to see if the device matches with any in the library. The matches are listed at the bottom of the screen. To quit from the left hand menu press <ESC> or select the MAIN MENU option. This takes you back to the right hand menu where you can test a device from another family or choose QUIT to return to the operating system.

Appendix A: Shorthand Keystrokes

For quick operation of the ChipWriter Programmers the following shorthand keystrokes are available :-

	EPROM	**PAL**
Open	Alt+O	Alt+O
Save	Alt+S	Alt+S
Exit	Alt+X	Alt+X
Data Edit	Alt+E	Alt+E
Checksum	Alt+C	Alt+C
Fill Buffer	Alt+F	N/A
Goto Fuse	N/A	Alt+G
Goto Address	Alt+G	N/A
Edit UES	N/A	Alt+U
Buffer Size	Alt+Z	N/A
Read	Alt+R	Alt+R
Verify	Alt+V	Alt+V
Blank Check	Alt+B	Alt+B
Program	Alt+P	Alt+P
Vector Test	N/A	Alt+T
Manufacturer	Alt+M	Alt+M
Type	Alt+T	N/A
DeviceName	Alt+N	Alt+N
DOS Command	Alt+D	Alt+D
Display Mode	Alt+Y	N/A

Appendix B: Menu Options

This section gives a brief overview of each of the menu options available within PAL and EPROM and tells you in which section of the manual you will find more detailed information.

CWPROM.EXE

Menu Option	**Section**
FILE MENU	
Open Open a binary or HEX formatted from disk and load it into buffer memory area.	6.2.2
Save As Save an area of the buffer to a binary or formatted file on disk.	6.2.7
Save Save the currently loaded file.	6.2.7
Save Task Save the current device environment and configuration as a Task file.	6.2.8
Load Task Loads a previously saved Task file and restore device environment and configuration.	6.2.8
Exit Exit the software and save all settings.	5.11
DATA MENU	
Edit Edit the contents of the data buffer.	6.3.1
View View the contents of the data buffer without making any changes.	6.3.2
Fill Fill an area of the buffer with a specified value.	6.3.4
Copy Copy an area of the buffer to another area.	6.3.4
Search Search an area of the buffer for a specified value or string.	6.3.3
Go to	6.3.3

Go to a user specified area of the buffer.

Swap Bytes 6.3.4
Swaps odd and even bytes. Useful for 16-bit devices.

Buffer Size 6.2.1
Changes to size of the buffer.

Print 6.3.7
Prints out an area of the buffer

Checksum 6.3.5
Calculates a checksum over a given area.

PROCESS MENU

Read 6.4.1
Read the device in the ZIF socket and store the data in the buffer.

Verify 6.4.2
Verify the data in the device against the data in the buffer.

Verify Electronic ID 6.4.3
Checks the electronic signature of the device.

Blank Check 6.4.4
Checks to see if a device is blank.

Program 6.5.2
Programs data in the buffer into the device.

Erase 6.7
Erases electronically erasable devices.

Security Status 6.6.2
Checks the status of the device security bits.

Verify Encryption 6.6.2
Checks the Encryption status of a microcontroller

DEVICE MENU

Type 6.1.1
Selects the type of device to be programmed.

Manufacturer 6.1.1
Selects the manufacturer of the device to be programmed.

DeviceName 6.1.1
Selects the actual device to be used.

OPTIONS MENU

Parallel Port 5.5
Tells the software which parallel port the programmer is connected.

Screen Color 5.5
Selects either mono or color displays.

Snow Precautions 5.5
Overcomes snow problems on old CGA displays.

Hi-Res. Mode 5.5
Changes number of lines on EGA and VGA displays.

Display Mode 5.5
Switch between 8, 12, 16 bit display.

DOS Command 5.5
Enables a DOS command to be run from the software.

Programming Options 6.5.1
Selects user and device configuration options for programming

CWPAL.EXE

Menu Option	**Section**

FILE MENU

Open 7.2.1
Open a JEDEC formatted file from disk

Save As 7.2.4
Save the current fusemap to a JEDEC file on disk.

Save 7.2.4
Save the currently loaded file.

Save Task 7.2.5
Save the current device environment and configuration as a Task file.

Load Task 7.2.5
Loads a previously saved Task file and restore device environment and configuration.

Exit 5.11
Exit the software and save all settings.

DATA MENU

Edit 7.7.1
Edit the contents of the fuse map.

View 7.7.4
View the contents of the fuse map without making changes.

Goto Fuse 7.7.2
Goto a defined fuse in the fuse map.

Blank Fuse Map 7.7.1
Sets all the fuses in the buffer to '0'.

Set Fuse Map 7.7.1
Sets all the fuses in the buffer to '1'.

Convert Fuse Map 7.8
Used to convert old PAL files to GAL's.

Edit UES 7.7.3
Edits the Users Electronic Signature.

Edit Test vectors 7.9
Edit the test vectors for this file.

Checksum 7.7.5
Calculate checksum on the data in the fuse map.

PROCESS MENU

Read 7.3.1
Read the device in the ZIF socket and store the contents in the fuse map.

Verify 7.3.2
Verify the data in the device against the fusemap in the buffer.

Blank Check 7.3.3
Checks to see if a device is blank.

Program 7.4.1
Programs data in the fusemap into the device.

Erase 7.5
Erases electronically erasable devices.

Vector Test 7.9
Applies Test Vectors.

Security Status 7.6
Checks the status of the device security bits.

Auto Secure 7.6
Automatically secures all devices when programming

DEVICE MENU

Manufacturer 7.1
Selects the manufacturer of the device to be programmed.

Device Name 7.1
Selects the name of the device to be programmed.

OPTIONS MENU

Parallel Port 5.5
Tells the software which parallel port the programmer is connected.

Screen Color 5.5
Selects either mono or color displays.

Snow Precautions 5.5
Overcomes snow problems on old CGA displays.

Hi-Res Mode 5.5
Changes number of lines on EGA and VGA displays.

Display Mode 5.5
Switch between 8, 12, 16 bit display.

DOS Command 5.5
Enables a DOS command to be run from the software.

APPENDIX C: Technical Assistance, Warranty, and Repair

This appendix includes details about contacting Data I/O for technical assistance, repair and warranty services. The appendix also explains the Bulletin Board Service and Web page.

Technical Assistance

You may contact Data I/O for technical assistance by calling, sending a fax or electronic mail (e-mail), or using the Bulletin Board Service (BBS). To help us give you quick and accurate assistance, please provide the following information:

- Product version number
- Product serial number (if available)
- The firmware version : To find out which firmware version you are using run SELFTEST. The firmware version number is displayed near the start of the test.
- The results of the SELFTEST program. See Section 3.3
- Detailed description of the problem you are experiencing
- The software version and library version for the device you are programming: To find out which version of the library you are using connect the programmer and run EPROM or PAL. Select the manufacturer and device name of the device being programmed. The library version number is then displayed on the third line of the screen.
- Error messages (if any)
- Device manufacturer and part number (if device-related)

It will assist our engineers if you are next to the programmer when calling.

Data I/O Customer Support

United States
For technical assistance, contact:

Data I/O Customer Resource Center
Telephone: 800-247-5700
Fax: 425-869-2821

For repair or warranty service, contact:

Data I/O Central Dispatch
Telephone: 800-735-6060
Fax: 425-881-0561

Canada
For technical assistance, contact:

Data I/O Customer Resource Center
Telephone: 800-247-5700
Fax: 425-869-2821

For repair or warranty service, contact:

Data I/O (Canada)
6725 Airport Road, Suite 302
Mississauga, Ontario, L4V 1V2
Telephone: 905-678-0761
Fax: 905-678-7306

Japan

For technical assistance, repair, warranty service, contact:

Data I/O Japan
Osaki CN Building 2F
5-10-10, Osaki Shinagawa-Ku
Tokyo 141
Telephone: 3-3779-2207 (Operations)
3779-2203 (Other)

Germany

For technical assistance, repair, warranty service, contact:

Data I/O GmbH
Lochhamer Schlag 5a
82166 Gräfeling
Telephone: 089-858580
Fax: 089-8585810

Other European Countries

For technical assistance, repair or warranty service, contact your local Data I/O representative.

Other Countries World-wide

For technical assistance, repair or warranty service, contact the office below and ask for the number of your local Data I/O representative.

Data I/O International
10525 Willows Road N.E.
P.O. Box 97046
Redmond, WA USA 98073-9746
Telephone: 425-881-6444
Fax: 425-882-1043
Telex: 4740166

Calling

Calling the appropriate Data I/O Customer Support number listed at the front of this Appendix. When you call, please be at your programmer or computer, have the product manual nearby, and be ready to provide the information listed above.

Sending a fax

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

Sending E-mail

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

```
techhelp@data-io.com
```

Logging on to the BBS

The Data I/O Bulletin Board System (BBS) enables you to:

- Obtain a wide range of information on Data I/O products, including current product descriptions, new revision information, technical support information, application notes, and other miscellaneous information
- Access device support information
- Request support for a particular device
- Leave messages for the BBS system operator, Customer Support personnel, or other customers
- Download many DOS and Windows utilities

Online help files provide more information about the BBS and its capabilities.

Before you can log on to the BBS, you must have your telecommunications software and modem installed and operating. Refer to your telecommunications software manual and your modem manual for set-up procedures.

Have the following information available prior to contacting the BBS:

Name and Password
File transfer protocol (used by your telecommunication software)

Use the following procedure to log on to the BBS:

- To contact the BBS, call one of the following numbers:

  (Modem)

United States	+1-425-881-3465
Germany	+49-89-858-5880
	+49-89-858-5833
Japan	+81-33436-02055

  (Internet: Using, telnet, go to `bbs.data-io.com`)

- If this is the first time you have called, when the logon screen appears, type new at the prompt to create a new account and provide the requested information.

  Note: Be sure to write down your user name and password for future reference. This is your personal BBS account and should not be shared.

- Select the Library Files menu .
- Select the ChipWriter Library (CHIPWRIT)
- Search for the ChipWriter Gang files.

Using the Data I/O Web Page

The Data I/O Home Page on the World Wide Web (WWW) includes links to online information about technical products, information about Data I/O, a list of sales offices, and technical user information such as application notes and device lists.

To access the WWW, you will need an Internet account with Web access, and a Web browser such as Netscape or Internet Explorer. The address of the data I/O Home Page is:

`http://www.data-io.com`

You may also access the data files from an FTP at the following address location:

`ftp.data-io.com/dataio/chipwriter`

Click the (Programmer Device Support) image or go to Technical Information and then ChipWriter software.

- Once you have reached the ChipWriter Library, select the correct file and download the ChipWriter compressed file.
- The ChipWriter file comes in a self-extracting file format. Place the file on a floppy disk or temporary hard drive directory and then expand the file by running it. The following files should be created:

File Name	Description
DIO_PARS.EXE	Part lists and device names
DIO_EP.EXE	Main EPROM software
DIO_ELIB.EXE	EPROM.EXE library files
DIO_PAL.EXE	Main PAL software
DIO_PLIB.EXE	PAL library files
DIO_TEST.EXE	Chiptest software
INSTALL.BAT	Installation batch file

Note: It is recommended that the disk is labeled "ChipWriter" and add the version number to avoid mismatched software version numbers if it becomes necessary to use the disk again.

- At the DOS prompt, run the INSTALL.BAT file and follow the instructions for installing in to the correct directory.

Warranty Conditions

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 your distributor or local sales office.