

PDS
IBM Editor Manual

The PDS is written by Andrew Glaister

Manual written by Fouad Katan

The assembler is the sole property and copyright of P.D. Systems Ltd. Revisions to the specification of the assembler may be made at the author's discretion. Users of the system, will of course be notified of these revisions, and downward compatibility will be maintained, where at all possible.

This manual is the copyright of P.D. Systems Ltd. No copying, duplication or any other such acts which may be interpreted as a breach of copyright may be performed without the authors' permission.

The contents of this document are copyright 1987.

2nd Edition, 1988.

Trademarks

IBM PC is a trademark of IBM.

MSDOS is a trademark of Microsoft Corp.

Amstrad PC is a trademark of Amstrad PLC.

Wordstar is a registered trademark of MicroPro International.

PDS is a trademark of P.D. Systems Ltd.

Contents

- 1 Getting started with the Programmers Development System
 - 1.1 System requirements of PDS.
 - 1.2 Installing the PDS interface.
 - 1.3 Running the PDS system.
 - 1.4 Connecting PDS to slave computer.
 - 1.5 Setting up CONFIG.SYS.

- 2 First steps in using PDS
 - 2.1 PDS keyboard use.
 - 2.2 The screen format.
 - 2.3 Files within PDS.
 - 2.4 Using the windowing system.

- 3 The PDS text editor
 - 3.1 Entering text into the editor.
 - 3.2 Cursor movements.
 - 3.3 Insertion and deletion.
 - 3.4 Placing and Finding markers.
 - 3.5 Block markers and Blocks
 - 3.6 Function keys in PDS.
 - 3.7 Normal Search and Replace functions.
 - 3.8 Special Search and replace functions
 - 3.9 Using the PDS help file.
 - 3.10 Printing with PDS.
 - 3.11 Accessing disk files from within the editor.
 - 3.12 Miscellaneous editor functions.

- 4 How to use the PDS escape mode
 - 4.1 How to use the escape mode.
 - 4.2 Escape mode functions.
 - 4.3 Escape mode macros.
 - 4.4 Examples of popular escape mode commands.

- 5 The Disk system
 - 5.1 The disk system format
 - 5.2 MSDOS disk handling
 - 5.3 Explanations of all the disk formats supported.
 - 5.4 Using the XMSDOS, PDS, and MSDOS menu systems.
 - 5.5 Using the Utility menu.

6 The Graphics editor

- 6.1 Entering the graphics editor.
- 6.2 The graphics editor display format.
- 6.3 Drawing using the graphics editor.
- 6.4 Special graphics editor functions.
- 6.5 Using the buffer to convert graphics formats
- 6.6 The output format of the graphics editor.
- 6.7 Using the graphics editor with character sets.

7 The monitor

- 7.1 Entering the Monitor.
- 7.2 Using the screen editor.

8 The assembler

- 8.1 Invoking the assembler.
- 8.2 Understanding error messages during assembly.
- 8.3 Using source code from other development systems.

9 The Configuration menus.

- 9.1 The configuration menu system.
- 9.2 The main configuration menu.
- 9.3 PDS run time options.

1 Getting started with PDS

The Programmers Development System is designed to allow fast, efficient, and easy assembly language programming. This manual is designed to familiarize the programmer with the System, and to give full instructions on the use of the PDS editor.

The editor is only part of PDS, albeit an important part. From within the editor you are able to activate other parts of PDS, namely the assembler, monitor, disk system and graphics editor. The disk system and graphics editor are all covered in this manual, but the assembler and monitor are covered in separate manuals provided with PDS. This manual will only discuss how to interact with the assembler or monitor, and its implementation on the IBM personal computer (or compatible).

Although this manual is specific to the IBM PC, most of its contents will translate perfectly to other versions. We have tried to maintain the same standards and feel for the system wherever possible.

No knowledge of MSDOS is required to use PDS, although some features do overlap with MSDOS, and these will be discussed in detail with reference to PDS.

1.1 System requirements of PDS

It is important that all the right equipment is used, and the System's specifications fulfilled.

First of all, the following should have been provided with the PDS:

- 1 IBM PC PDS interface
- 1 Editor manual (you're reading it!)
- 1 Disk containing the programs and download software.
- 1 One assembler/monitor manual, for each assembler.

You may also have bought an optional parallel interface, to allow fast interaction between the host and target computer for which you are working. This may take the form of a cable, or a boxed interface with a connecting cable, depending on the computer in question. The circuit diagrams for these parallel interfaces are included in the assembler/monitor manuals, in case you wish to build the interfaces yourself. However, PD Systems Ltd takes no responsibility for any damage caused when using non-PDS interfaces.

The following is the minimum configuration for use of PD System on the IBM PC:

- 1 IBM PC/XT/AT/PS2 or compatible, with any display card and monitor combination, and at least a single floppy drive. The drives may be of any capacity or type. The system must be running 3.0 or later versions of MSDOS.
- 1 Ram upgrade to 640k. Some PC's already come with 640k, while others need add on boards or some extra ram chips to be fitted. When the machine is switched on a ram test is performed and the ram size will be indicated on the screen.

Note that the system must have at least a CGA or EGA card (or compatible) to run the graphics editor, a mono card will run the system, but the graphics editor will not function.

If you find that your IBM PC system is lacking in any way, then contact your dealer immediately for assistance. If any part of PDS is found lacking then contact us immediately and we will resolve the problem as soon as possible.

1.2 Installing the PDS interface

Fitting the PDS interface is a simple operation, and will only require five minutes of your time. The operation varies depending on the type of IBM PC system being used. Therefore instruction can only be rough, so consult your PC specific manual for exact details.

Before you install the PDS interface to any PC, you must switch off the power to the machine. The next step is to locate your PC's expansion slots, you may need to open up the base units case, or remove some hatch to do this. Choose an eight bit slot (these have only one socket, while 16 or 32 bit slots have two sockets inline with each other), that is as far away from any other expansion cards as possible. Insert the PDS card into the slot, making sure that PDS connectors fall next to the casing. There should be a way of removing part of the casing, to give you access to the PDS connectors. The PDS card is now fitted to your PC, so replace any casing that was previously removed.

Switch on the PC, and make sure the normal startup procedure remains unchanged, ie the testing message appears, and the normal bootup prompts are displayed. If the machine refuses to follow the normal startup procedure, follow the installation instructions again and make sure the PDS interface is firmly in position, and that no part of it is touching any other expansion card. If problems persist, then contact PD Systems for assistance.

1.3 Running the PD System

At this stage you should be ready to execute the PDS program.

If you are using a hard disk system, allow the system to start or boot normally so that you are left with the MSDOS prompt. The standard MSDOS prompt will appear as 'C>'. Insert the required PDS assembler disk into the vacant disk drive. All the files on the PDS disk should now be copied onto your hard disk, but first of all create a PDS directory on your hard disk, by:

```
MD \PDS  
CD \PDS
```

The PDS files can then be copied by entering the command 'COPY A:.'. PDS is now installed onto your hard disk drive.

If you are using a non-hard disk system, then switch the machine on as normal, inserting the normal system disk. Once the system has booted, insert the PDS disk into drive A, or any vacant floppy drive.

You must then enter:

PDSZ80

To run the Z80 version of the PDS, or:

PDS6502

To run the 6502 version of the PDS. Make sure that the correct drive is selected before you do this, this is done by entering the drive name, followed by a colon, at the MSDOS prompt.

PDS will take a few seconds to initialize, after which the following or similar screen will appear:

```

>0< Line 00001 Chr 001 Mem 00000 <<< Z80 >>> <1> Line 00001 Chr 001 Mem 00000
;
; PDS Pc5.f :0000: (C) 1985-7 P.D Systems
;
;
; File "-----"
; Version          00.00
; By "-----"
;
; Created on Sat the 19th of Sep 1987
; Last update 15:52 on 19/09/87
;

```

If this does not appear, and you are left with a blank screen, then there is a problem with the PDS interface. Repeat the fitting instructions once more, and make sure the PDS interface is firmly in place. If you are still not unable to run PDS, then contact PD Systems immediately for assistance.

You are now in the editor, and can start using PDS.

1.4 Connecting PDS to the target computer

Now that the interface is installed, and PDS is running, you should complete the System by connecting PDS to the target computer. You should be able to see the PDS interface through the expansion slot gap. Both connectors will be visible, the top connector is referred to as computer1, the second, bottom connector, is known as computer2. Connect the 16 way ribbon cable from the target computer to computer 1, it will only fit in one way. If computer 1 is occupied, insert the cable into computer 2.

To complete the communication link between PDS and the target computer, you have to connect the interface to the target computer. This will normally connect to the expansion or user port of the target computer. Use the documentation given with the parallel interface, for detailed instructions. Download software will also be included with the parallel interface, this will be on tape or disk depending on the computer for which it is designed. Read the instructions on running the software, as the downloader will be used later.

1.5 Setting up CONFIG.SYS

When a PC boots up, one of the first things it does, is to load the system configuration from a file called 'CONFIG.SYS'. This file will always be in the main directory, and contains commands which allow you to customise your PC.

The two commands that concern PDS, are called 'BUFFER' and 'FILES'. These define the disk buffer size, and the maximum number of files allowed open at any one time. For PDS to work most efficiently, the commands should read:

```
BUFFERS=20  
FILES=20
```

To see what your current configuration is, then type 'TYPE \CONFIG.SYS' at the MSDOS prompt. The contents of the file will then be displayed. If 'BUFFERS' or 'FILES' are set to any value less than 20, or if they are not shown at all, then the 'CONFIG.SYS' file has to be updated.

The 'CONFIG.SYS' file is also used to install device drivers, these are programs that are incorporated into the MSDOS system to support special hardware or software devices. Each device driver will eat up some of the user memory, if too many drivers are installed or if a driver is particularly ram hungry then PDS may not function properly. These drivers must then be removed.

The configuration file is like any other text file, and can be edited in PDS or any other text editor.

If you have no 'CONFIG.SYS' file on your boot directory, then you should create one by typing 'COPY CON \CONFIG.SYS', you must then enter :

```
BUFFERS=20  
FILES=20  
COUNTRY=44  
^Z
```

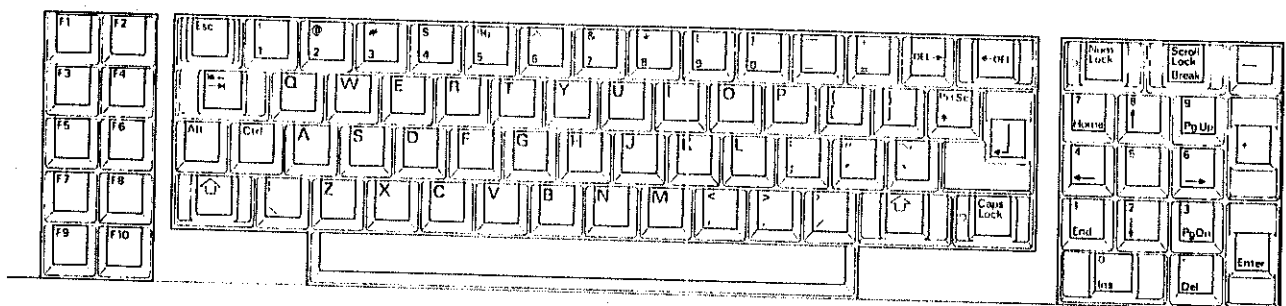
The '^Z' is generated by pressing the [CONTROL] and [Z] keys at the same time. The country command in the configuration file allows you to define which country MSDOS should be tailored to, 44 is for the UK.

2 First steps in using PDS

Once PDS is running, and the editor has been entered, you can start to use the system. This is best done once the PDS style and structure is understood, and you have familiarized yourself with the PC.

2.1 PDS keyboard use

PDS uses all the IBM PC's keyboard, with most keys performing three functions. Below is diagram of an 84 key IBM keyboard:



The keyboard is fairly standard, and has all the normal alphanumeric typewriter keys, and a numeric keypad. There are also a number of specialist editing keys, which are marked with various names, such as [HOME], [END] or [NUM LOCK]. Different PC's will have different keyboard layouts, and different editing keys, but this should make no real difference as all true PC's have a minimum of editing keys, and PDS will only use these. The only keys left are the function keys, all PC's have at least ten of these, and PDS uses all ten of them.

The numeric keypad can also be used for cursor related functions, this is decided by the [NUM LOCK] key. When num lock is on, the keypad will act like a normal numeric keypad, but when num lock is off, the keypad can be used as a cursor key cluster. Both uses should be indicated on the keytops.

The Ten function keys all perform particular functions, below is list of all their uses, and the name by which they will be referenced in this manual. It is important that you know all their uses, to operate PDS properly:

Function key	Name	Use
F1	[Assemble]	Assembles the program in memory.
F2	[Save/Load]	Saves or Loads all 8 files.
F3	[Graphics]	Enters the graphics editor.
F4	[Monitor]	Enters the Monitor.
F5	[Configure]	Enter the PDS configuration system.
F6	[Print]	Prints files.
F7	[Help]	Engages the help facility.
F8	[Repeat]	Repeats search and replace functions.
F9	[Disk]	Enter the PDS disk system.
F10	[Finish]	Ends current task.

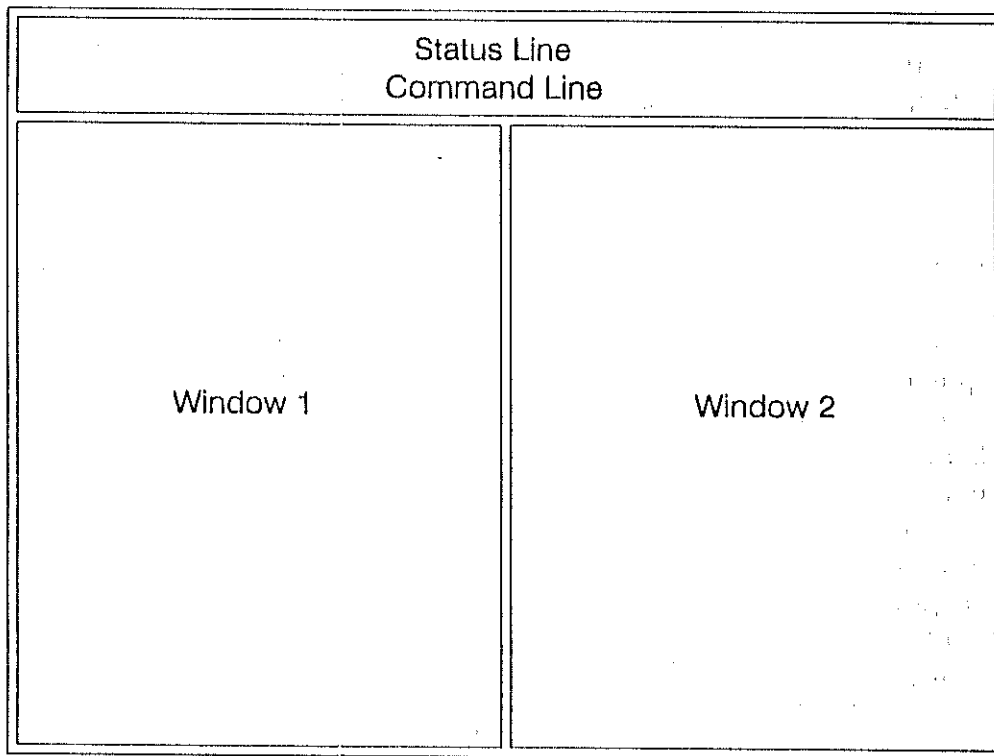
As new functions will be explained, their key operations will also be discussed. As you may have noticed, whenever a key is referenced to, its name is enclosed in '[' and ']' signs. If its name is in capital letters, then the key referred to will also have those letters on its key top. If on the other hand its name is in lower case, then the name 'describes' the keys appearance or function, ie a carriage return key is referenced as [carriage return], but a bent left arrow is shown on the key. This way you will learn key use, and location more rapidly. Wherever possible we have tried to make a key's names relevant to it's function.

2.2 The screen format

When PDS is first entered the display will be as shown in the diagram below, but this only shows one of the two possible editing windows. To see both windows on the screen at the same time, press the [CONTROL] and [SCROLL LOCK] keys together. A vertical line will appear in the middle part of the display, as shown in the diagram below:

<pre><0> Line 00001 Chr 001 Mem 00000 <<< ZBC >>> <1> Line 00001 Chr 001 Mem 00000</pre>	
<pre>; ; PDS Pc5.F :0000: (C) 1985-7 P.D Sys ; ; File "-----" ; Version 00.00 ; By "-----" ; ; Created on Sat the 19th of Sep 198 ; Last update 15:52 on 19/09/87 ;</pre>	<pre>; ; PDS Pc5.f :0000: (C) 1985-7 P.D ; ; File "-----" ; Version 00.00 ; By "-----" ; ; Created on Sat the 19th of Sep ; Last update 15:52 on 19/09/8</pre>

The screen will effectively be segmented as follows:



Windows 1 and 2 will show your source code and are both completely independent of each other, scrolling separately. Their sizes can also be varied, or either window can be hidden to display the other window completely.

The command line is used to enter special commands to the editor, or for PDS to convey special messages to the user. For example, if an error occurs then a bell will sound, and the error message will be displayed within the command line.

The final display area shows the current status of both display windows. The left half of the status line is for window 1, and the right half of the status line is for window 2. The status line is divided as below:

```
<F> Line L Chr C Mem M ** Z80 ** <F> Line L Chr C Mem M
```

'F' shows the current file that particular window is displaying. 'F' represents file numbers '0' to '7', so that all eight files can be portrayed by a particular digit. When both windows are visible on the display, the window which holds the cursor is the active window. If a single window covers the entire display, then file number for that window is enclosed by '>' and '<' characters, this indicates which window is the active window. The active window is the window to which all further keystrokes will be directed at. The inactive window will be enclosed by '<F>'.

'L' shows the current cursor line position and can be valued from 00001 to 15912. The first line in a file is obviously line 1, with line numbers incrementing down the file.

'C' shows the current cursor character position. This can take the value 001 to 120, as there are maximum of 120 characters in any single line. This is not the same column number as a tab character will move the cursor to the 10th column, but the new character position will be 002.

'M' shows the size of the currently selected file. File sizes can vary from 00000 to 31824 bytes. Try not to fill up files completely, as this may make them more difficult to work with.

2.3 Files within PDS

As already stated, there are 8 different files within each program. At the beginning of program development 8 files should suffice, which amounts to about 250K of source code.

Each file has its own cursor position, function keys, heading and other such information. When a particular file is selected all its information will be transferred to the current window's status line and the current display windows will show the contents of that file at the current cursor position.

At the beginning of each file there is a computer generated header which cannot be edited directly but provides you with useful information about the file. This header allows you to keep track of each file, its purpose and update condition. The header takes this format:

```
; PDS Pcn.n :SSSS: (C) 1985-7 P.D Systems
;
;
; FILE "-----"
; VERSION          VV.VV
; BY "-----"
;
; Created on DDD the DDth of MMMM YYYY
; Last updated 10:02 on dd/mm/yy
;
```

..... The file follows

The symbols 'n.n', 'VV.VV', 'DDD', 'DD', 'MMMM', 'YYYY', 'dd', 'mm', and 'yy', signify the following :

n.n	Version number of the PDS you are running.
SSSS	This is your copies serial number, each copy has its individual serial number.
VV.VV	Version number of the file, updated by 00.01 after each save.
DDD	The first three letters of the creation day (ie Sun). The creation date is when that file was first generated.
DD	The creation date, ie 21st, 10th, 3rd, etc. The creation date is when that file was first generated.
MMMM	The creation month in words, ie May, July, etc. The creation date is determined by the date when the file was first used.
YYYY	The creation year, ie 1988, etc. The creation date indicates when the file was first generated.
dd	The date of the last update date, ie 23, 05, etc. The last update date is set every

	time the file is saved.
mm	The month of the last update date, ie 12, 01, etc. This is updated when the file is saved.
yy	The year of the last update date, ie 87, 08, etc. When ever the file is saved, the update date is modified.

The dates and version numbers are all updated automatically, but the heading and author name are set via the configuration menu, see chapter 9 for details.

2.4 Using the windowing system

As explained earlier, each window can be set up to different sizes by pressing [ALT] together with the [4] key, to move the window partition left, or [ALT] and [6] to move the window partition to the right. If window 2 is moved over to the far left then it will obscure window 1. If window 1 is moved over to the far right, then window 2 will be hidden, and window 1 will fill the whole display area. The currently active window, the window with the cursor in, cannot be completely obscured, a single character width will always be visible.

The window can also be set to standard positions instantly, by pressing certain key combinations. To set the currently active window to its full size, where it fills the whole screen, then press [CONTROL] and [V] at the same time. To switch to the opposite window and increase that window to its largest size, where it fills display, depress [ALT] and [SCROLL LOCK] at the same time. For both windows to be displayed on the screen simultaneously, and in the same size, then depress [CONTROL] and [SCROLL LOCK] together.

The currently active window is the window to which all keystrokes and cursor movements are directed. This is determined by the position of the cursor, the cursor may be visible in window 1, window 2, or the command window. To swap between windows 1 and 2, press the [SCROLL LOCK] key. If the currently active window is set to its maximum size, and the [SCROLL LOCK] key is depressed to swap windows, then a 'BEEP' will sound; it is not possible to switch to a hidden window. The hidden window must first be re-introduced onto the display, by using the [ALT] & [4] or [6] keys or depressing [CONTROL] and [SCROLL LOCK] at the same time.

Get used to the window concept, as it will prove a very powerful facility, and can speed up programming significantly.

3. The PDS text editor

When the system is first entered, only window 1 is visible. A block cursor will also be visible at the current cursor position, at the beginning of file 0. Wherever the cursor block is placed is known as the 'active' window. The active window is where all keyboard functions will be directed. There are only three possible active windows: window 1, window 2 and the command window. Windows 1 and 2 can display twenty lines of text at a time, or you can say that one screen page is equal to 20 lines.

3.1 Entering text into the editor

The currently active window should be window 1.

Entering text into the editor is very easy: just start typing! Anything you type (as long as it's not control characters), will be displayed in the current window, and inserted into the current file. When typing in source code, insert any tabs or spaces in the normal way. If you reach the right hand side of window 1 while typing, then the line will be highlighted and will scroll to the left as each new character is inserted. This will continue until the character counter for window 1 reaches 120: this is the maximum number of characters allowed on any single line. As soon as you've finished a line, press the [carriage return] key. Once depressed, the cursor will be positioned at the beginning of the next line, setting the character counter to 1 and incrementing the line counter by 1. All text is entered in this way.

If you select window 2, by depressing [SCROLL LOCK], you can enter text in exactly the same way. You may also select different files by depressing [SHIFT] and any of first eight function keys ([F1] to [F8] will select file 0 to 7 respectively) on the left of the keyboard. Notice how each file has its own cursor position, so when you return to file 0 you will be positioned at the point at which you left it.

3.2 Cursor movements

Now some text has been entered in the editor, you will need to move through the text, making changes or corrections. The block cursor will always show your current text entry position. When the cursor is positioned on top of a character, then the character will be displayed in reverse graphics. To move the cursor, use one of the arrow keys. These are found on the numeric keypad, and will only function correctly if num lock is off. There are three types of cursor movement. The first of these, single, allows cursor movements in single increments and are accessed by depressing any of the arrow keys. Segment movement allows you to move the cursor in jumps, a jump may be a page or a tab position from your current position and are accessed by pressing [SHIFT] and an arrow key, or by using the [PG UP] and [PG DN] keys. Limits cursor movements will move the cursor through the extremities of the current file, to the start or end of a file or line by depressing [END], [HOME] or [CONTROL] and an arrow key, or [END] and [HOME].

If you depress the [left arrow] key, the cursor will move one position to left, on the current line. If you are at the beginning of line, then the cursor will be positioned at the end of the preceding line. If you are at the beginning of a file, then no movement occurs, but a warning sound will be made. If you depress [SHIFT] and [left arrow] together, then the cursor will move to the previous tab stop position, even if it's on the previous line. If [CONTROL] and [left arrow] are depressed, then the cursor will move to the start of the current line, known as character position 1.

When the [right arrow] key is depressed the cursor will move one character to the right, on the current line. If you are at the end of line, then the cursor will move to the start of the following line. If there is no following line, then you are at the end of the file, so a 'BEEP' will sound and the cursor will not be moved. If the [SHIFT] and [right arrow] keys are depressed, then the cursor will move to next tab stop position, this may be on the following line. If the [CONTROL] and [right arrow] keys are depressed, the cursor will move to the end of the current line. The end of a line is directly after the last character on that line.

If the [up arrow] key is pressed then the cursor will move up to the previous line. The cursor will be left at the same character position, which is not the same as the column position. If you try and move past the beginning of the file then a 'BEEP' will sound. If the [SHIFT] and [up arrow] keys or the [PG UP] key are depressed, then the cursor will move 20 lines back, the equivalent to a single display page. Again, if you try to move beyond the start of the file, a 'BEEP' will sound. If the [CONTROL] and [up arrow] keys, or the [HOME] key, are depressed, the cursor will move to the start of the first line of the file - line position 1.

If the [down arrow] key is pressed then the cursor will move down to the next line. The cursor will be left at the same character position, which is not the same as the column position. If you try and move past the end of the file then a 'BEEP' will sound. If the [SHIFT] and [down arrow] keys or the [PG DN] key are depressed, then the cursor will move 20 lines down within the file, the equivalent to a single display page. Again, if you try to move beyond the start of the file, a 'BEEP' will sound. If the [CONTROL] and [down arrow] keys or the [END] are depressed, the cursor will move to the end of the last line of the file, and the character and line positions will be updated accordingly.

If you press in [CONTROL] and [END] together, the current file is listed from the current cursor position. When this key combination is depressed, the current file will be scrolled through automatically from the current cursor position, and continues until any key is pressed or the end of file is reached.

The [HOME] key can be used to move the cursor to the beginning of the current file, line 1, character 1 (this performs the same function as [CONTROL] and [up arrow]). If [CONTROL] and [HOME] are pressed together, then the cursor in every file will be re-positioned at the beginning of the file, and the current cursor position will be moved to the beginning of file 0, window 1 (only if window 1 is visible).

There are many other commands that will influence the cursor position, but all of these perform another function primarily.

3.3 Insertion and deletion

As you've probably noticed, PDS is an insertion editor. This means when any text is typed, it will be inserted at the current cursor position. There are other ways to insert text, some of which are described in this section.

The [INS] key is used to insert spaces into text. When depressed, this will insert a space directly after the cursor. This is equivalent to pressing [space], then cursoring left. If [SHIFT] and [INS] are depressed together, then a line is inserted directly after the cursor. This is equivalent to pressing [carriage return], then cursoring left. This key has no function when used in conjunction with the [CONTROL] key.

So far none of the deletion functions have been discussed. The simplest of these is the [<-DEL], located directly above the [carriage return] key. When used, this deletes the character to the left of the cursor. Carriage returns can be deleted by pressing this key at the beginning of a line. This effectively joins two lines together, but their combined lengths must not be more than 120 characters, otherwise a 'BEEP' will sound. Trying to delete at the start of a file will also cause a 'BEEP' to sound. Pressing [SHIFT] and the other delete key, [DEL] will perform exactly the same function. Pressing [CONTROL] and [<-DEL] will delete to the beginning of the current line.

Pressing the [DEL] key, will delete the character currently underneath the cursor, and move the following characters in. Carriage return may also be deleted, and will join the current and following lines together. When used in conjunction with the [SHIFT] key, this will delete all of the current line, including a terminating carriage return. The cursor will then be left at the beginning of the following line. Any attempts to delete further than the end of the file will be stopped and a 'BEEP' will sound. Pressing [SHIFT] and [<-DEL] will perform exactly the same function.

Pressing the [CONTROL] and [DEL] will delete the characters to the right and underneath the cursor, until the end of the line, but a terminating carriage return will not be deleted. This key sequence will have no effect, when the cursor is at the end of the line. Pressing [CONTROL] and [<-DEL] at the same time will delete all the characters to the left of the cursor, until the beginning of the line is reached. If [CONTROL] and [<-DEL] are depressed when the cursor is at the beginning of the line, then nothing will happen.

If [ALT] is depressed together with any of the delete keys ([<-DEL] or [DEL]), then the line under the cursor is deleted, including a terminating carriage return character.

If the [CONTROL] and [ESC] keys are depressed together then the whole file can be deleted. Before any deletion occurs, the following warning will be displayed:

Delete whole file (Y/N)?

Pressing the [Y] key at this point, will perform the delete function, and all of the file will be deleted. Any other key will return you to the editor without performing a deletion. Once a file is deleted it cannot be recovered, and all of its header information is cleared with it.

3.4 Placing and finding markers

A common situation during programming is when a particular part of the source code needs to be referenced on many occasions. To save moving back and forth through the text, a marker can be positioned at that point.

Up to 10 markers can be placed anywhere within the source code, and at any point in time these markers can be referenced. To place a marker, just hold down the [ALT] key together with any of the 10 function keys, F1 to F10. When a marker is placed, any marker of the same number, anywhere in the text, is deleted. This prevents two markers of the same name being defined, so no confusion can occur. The marker name tag is then inserted at the current cursor position, the marker will be shown as the number 0 to 9 identifying the function key pressed (F10 will be shown as a 0), but will be displayed in reverse video to highlight it.

To find a marker just press the [CONTROL] key together with the relevant function key. The cursor will then be positioned at that marker position. If the required marker has not been found, then an error 'beep' will sound indicating that you have either tried to find the wrong marker, or the marker has simply not been defined. If the cursor is positioned at the marker, and you want to return to your original cursor position, just press the [5] key on the numeric keypad. This will move the cursor to its original position before the marker search was initiated. This key is also called the [undo] key.

Markers are completely transparent to the assembler, and even to certain parts of the editor, such as searches. In all other forms it is treated in the same way as a normal character, ie when you delete text around it, it will move in the normal editor fashion. To delete a marker without re-setting its position, just delete it like any other normal character, by moving the cursor directly to the marker's right and pressing the [< - DEL] key.

Most programmers use markers for frequently referenced source code, ie data areas, tables, routines, mainloop, etc.

3.5 Block markers and Blocks

Every once in a while, you find yourself needing to chop and change your code. Two pieces of code may be very similar, and to save typing, you may want to replicate that portion of the code. Following is just one of the many possible uses of block operations. You may need to move a block, delete a block, print a block, or as described earlier copy a block.

Before a block operation can take place, a block needs to be defined. In PDS a block is an area of text within any single file, enclosed by a start and end block marker.

To define the start of a block, move the cursor onto the first character of the block, then depress the [CONTROL] and [<] together. A '«' character in reverse video will be inserted at the cursor position to define the start of a block. Any earlier definitions of a start of block marker in any file, will be deleted. The start of block marker can also be defined by pressing [CONTROL] and [B] together (this feature is included for compatibility with Wordstar type editors and performs exactly the same task as [CONTROL] and [<]).

To define the end of block, just move your cursor further down in the same file, to the character directly after the last character in your block. Now depress [CONTROL] and [>] or [CONTROL] and [K], to insert the end of block character. The end of block marker is represented by a '»' character in reverse video. Any previous end of block markers, anywhere within your PDS source code will be deleted before the new block marker is inserted. Make sure that the end of block marker follows the start of block marker in the same text file.

Block markers are completely ignored by the assembler and certain editor functions, but are still regarded as normal character when normal cursor controls or insertion and deletion are used. To delete either block marker, simply move the cursor to right of the block marker and press the [`< - DEL`] key, in the same way you would delete any other character. You can delete both block markers with one keystroke, by pressing [`CONTROL`] and [`< - DEL`] together.

Once a block has been defined it may be deleted, moved, copied or printed. All these operations require that the block is defined in the correct manner. Make sure that both markers are in the same text file, and that the start of block marker is positioned before the end of block marker. Also move the cursor so that it does not fall between the two block markers - remember that characters are inserted to left of the cursor, so the cursor cannot be on top of the end of block marker either. If any of the block operations are attempted when the cursor is within the block boundaries, except when block deleting or printing, then an error message will be displayed.

In case you've lost track of their placing, there are facilities to find the current position of either of the block markers. To jump to the start of block marker at any time, simply press [`CONTROL`] and [`H`]. This will move the cursor to the character directly to the right of the start of block marker. To jump to the end of block marker at any time, simply press [`CONTROL`] and [`N`]. This will move the cursor to the character directly to the right of the end of block marker. If either block marker has not been defined when it was searched for, then the error bell will sound.

3.5.1 Block Deletion

To delete a block simply press [`CONTROL`] and [`D`] simultaneously. If the block is correctly defined, you will be given the prompt:

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

Any keystroke other than [`Y`], will cancel the function. If the response was [`Y`], then the area of text between the start and end block markers will be deleted, together with the block markers themselves. When a block is deleted, any location markers within the block will also be deleted.

3.5.2 Block Moves

Block moves are more complicated, as the function requires a destination. After the block has been defined as set out above, move the cursor to the desired location of the block, in any file. To initiate the move operation, press [`CONTROL`] and [`M`] together. Before the move can take place, a memory check is performed by the System, ie it ensures there is enough room in the destination file for the block operation to work.

If the move is successful then all the text between the start and end markers is moved to its new location, and deleted from the original location. The block markers will move with the block, making it possible to move the same block again and again without re-setting the block markers. Any location markers within the block will also be moved. The cursor is then placed at the end of the block.

3.5.3 Block copies

Block copies work in a very similar way to block moves, except the original block remains intact. The block markers have to be setup as described earlier, making sure the start of block marker precedes the end of block marker within the same file. The cursor then has to be placed at the point where the block is to be copied to, which can be any position in any file, except between the defined block markers. Once in position press [CONTROL] and [C] to copy the block of text. If the destination file has enough memory for the block then the operation will take place, and all the text between the block markers will be copied to the cursor position. The original text and markers all remain intact, allowing the same block to be copied again immediately. Any location markers found in the block, will not be copied, and left in position in the original block.

3.5.4 Block print

The block print operation is extremely useful, and allows the defined block to be sent to the printer, for a fast and easy print method.

First of all make sure the block is properly defined in the manner described, ie the start of block marker precedes the end of block marker within the same file. Also make sure the printer is connected, power on and on line, before printing the block. To print the block press [CONTROL] and [P] together. The block will then be sent to the print spooler. The spooler will start to print the block immediately in the background. If either the [Print] or [PRT SC] keys are depressed at this point, the current state of the printer buffer is shown. Either the number of bytes left to print is displayed, or you will be prompted to print your source code, in which case the block has been printed and you should press any key to return to the edit mode. If the [FINISH] key is depressed when the print status is being displayed, then the printing is aborted, and the print buffer is cleared.

When a block is sent to the printer, it can be preceded by a series of control bytes to initialise the printer to your requirements. This can be done by setting up the configuration menu accordingly (see chapter 9, printer setup).

Any number of blocks can be printed, but up to a maximum combined size of no more than 40K at any one time.

3.6 Function keys in PDS

Each file in PDS has its own set of function keys. A function key allows you to assign a list of characters to one key, so when that key is depressed, the list of characters are then inserted at the current cursor position. This is most useful when a line or piece of code is often repeated.

Each file has 10 of its own function keys, which in effect can give you a total of 80. These function keys are saved with your source code, if PDS or XMSDOS disk formats are used, and are therefore retained indefinitely. To define any function keys, just follow the steps outlined in chapter 9, the configuration menus.

To select any function key press [CONTROL] or [ALT] and a numeric key [0] to [9]. Only the numeric keys on the qwerty keyboard apply because the numeric keypad are used for other functions. Any of the ten function keys can be used, eg [1] is function key 1 and [0] is function key 10. When a function key is pressed, it simulates actually typing the string in. Therefore, the same rules apply when you enter text by function keys as when text is entered by the keyboard a character at a time. In other words, line length errors, or out of memory errors are still possible.

The configuration menu can be used to define all the keys globally, so that each file has the same 10 function keys.

3.7 Normal find and replace functions

Most editors have find and replace facilities, and PDS editor is no exception. Indeed, the PDS editor has a complete and specialized set of find and replace facilities. Searches are not limited to just word searches, but to full line searches, including carriage returns and tabs, with wildcards.

3.7.1 The find function

To perform a find, simply press [CONTROL] and [F] together. You will then be prompted :

Find :

At this point enter the series of characters to be located. Any character can be found, including carriage returns and tabs. Carriage returns will be displayed as '^M', while the tab key will be displayed as '^I'. The character '|' is a wildcard character. This means that any character in that position will be accepted in the search, ie a search for 'G|T' could locate the words 'GOT' or 'GET' among others. When the string has been inputted, press the [ESC] key to enter the search string. Now the following options will be presented to you :

Options : Backwards, Global, Label, Case, or File

There are five options available. To choose any particular option just press [B] for Backwards, [G] for Global, [L] for label, [C] for Case, and [F] for File. When an option is selected it will be displayed in reverse video - the options will be setup from your last find or exchange command. When the computer is first switched on only the Global option is selected. Global, Backwards, and File options can only be set at any one time. This is done automatically by the editor - when one of the three options are selected the other two are cleared.

If the backwards option is chosen then the search will begin from the current cursor position, and move back to the beginning of the current file only.

If the Global option is chosen then the search will start from line 1 of file 0 to the end of the file 7.

The File option will start the search from the beginning of the current file to the end of the current file.

If none of the three options are selected, then the search would commence from the current cursor position and end at the end of the current file.

The last two options are Case or Label. When these are selected they will either be turned on or off, like a toggle switch. They can both be on at the same time or be selected in any combination, with any of the other search options like Backwards, File or Global.

When the Case option is selected, it is displayed in reverse video, then when the search is initiated it will ignore case, so that 'a' will be treated the same as 'A'. If the option is not selected then the case in the search string has to match exactly for the search to be successful.

When the Label option is selected, it is displayed in reverse video, then the search will be made for a label. This means that any matching string must be preceded and followed by a non-label character. A non-label character is a character that cannot be used within a label definition, for example ' ', carriage return, '(', ')', '*', '/', '-', and '+' are all non-label characters. For a full list non-label characters, please see your assembler manual, as this varies from assembler to assembler. If the label option is not selected, then the match can be found in any context. For example, if a search is made for the label 'FRED', only the label 'FRED' will be found, while words like 'ALFRED' will be ignored.

Once all the options required are selected, then press any other key to initiate the Find operation. At this point if the cursor is placed at the beginning of the match found, then the search was successful and you are returned to the editor to take the appropriate action. If the bell sounds and the cursor is moved to the end of the search zone, then no match was found. You may abort the find operation at any time, by pressing [Finish].

If you need to search from the current cursor position, regardless of the current option settings, then press [CONTROL] and [T] at the options menu. This will perform the search globally from the current cursor position.

For example, if were searching for the word 'loop', with the case and label options on, all the following would be found :

Loop loop LOOP

But these would not be found, as they contain extra characters:

loop1 LOOPY sloop

3.7.2 The find and replace function

The find and replace option works in a very similar way to the Find option, except that the match found will be replaced by a given string. To perform a exchange, simply press [CONTROL] and [X] at the same time. You will then be prompted :

REPLACE :

At this point enter the series of characters to be replaced. Any character can be replaced, including carriage returns and tabs. Carriage returns will be displayed as '^M', while the tab key will be displayed as '^I'. The character '|' is a wildcard character. This means that any character in that position will be accepted in the search, ie a search for 'G|T' could locate the words 'GOT' or 'GET' among others. When the string has been inputted, press the [ESC] key to enter the search string. You will then be prompted:

WITH :

Now enter the replace string. This is entered in exactly the same way as the search string, and is terminated by pressing the [ESC] key. The replace string can be completely empty, this would delete the occurrence of the search string. Now the following options will be presented to you:

Options: Backwards, Global, Label, Case, or File

There are five options available. To choose any particular option, just press [B] for Backwards, [G] for Global, [L] for Label, [C] for Case, and [F] for File. When an option is selected it will be displayed in reverse video. The options will be setup from your last find or exchange command. When the computer is first switched on only the Global option is selected. Out of the Global, Backwards, and File options, only one of them can be set at any one time. This is done automatically by the editor, and when one of the three options are selected the other two are cleared.

All the options work in exactly the same way as they do in the find function, so see that section for more details.

3.7.3 Repeating a find or find and replace

Once a find or find and replace has been performed it will usually be necessary to repeat the search or replace. The simplest way to do this is to press [CONTROL] and [T]. This will repeat the last find or find and replace, whichever was last used, from the current cursor position, using the current search option setup. This will only repeat the function once and returns the same type of errors. If the bell sounds then there is an error, if no error message was displayed and the cursor has moved to the end of the search zone, then no match was found for the search string. Otherwise an error message will always be displayed if the bell sounds. This function can also be accessed by pressing [CONTROL] and [-] on the numeric keypad.

A more extensive repeat feature is also available. This can be used by pressing the [Repeat] key, one of the function keys ([F8]) on the left or top of the keyboard. When this is depressed the following prompt is given :

Once, Till end, Number of times or Question

These option are accessed in the same as the find and replace options were. To select the desired option press [O] for Once, [T] for Till end, [N] for Number of times or [Q] for Question.

Only one of the options can be selected, if you press any other key, then the Once option is selected as the default. All of the repeat options repeat the last find or find and replace function, whichever was last used, utilizing the current zone settings.

The Once option is exactly the same as pressing [CONTROL] and [T] - it will find the next occurrence from the current cursor position.

The Till end option will repeat the find or replace until the end of the search zone. If a find is being repeated, then the option is pointless, as it will only find the last occurrence of the search string within the current zone. If a find and replace is being repeated, then all matches will be replaced until the end of the current zone.

The Number of times option allows you to jump to the nth occurrence of a particular string, or will replace the next n occurrences of the search string with the replace string. You will be prompted for the number of times to repeat when you initiate the repeat, by pressing [carriage return].

The Question option is a toggle switch, ie when selected the option will turn on or off. If the option is on then you will be given this prompt for every repetition:

Again (Y/N)?

When a repeat find is being performed, then the question relates to "Again (Y or N)?". If you wish to find the next occurrence then answer by pressing [Y], otherwise answer [N] and you will return to the editor.

When a repeat find and replace is being performed, then the question relates to "Replace (Y or N)?". If you wish to replace this occurrence of the search string then answer by pressing [Y], otherwise answer [N], and the search will continue. Pressing [ESC] or [FINISH] will terminate the search and replace and you will return to the editor.

If no match was found by a repeat function then the error bell will sound and the cursor will be placed at the end of the current search zone. If the bell sounds for any other reason, then an error message will be displayed.

3.8 Special search and replace functions

Beyond the normal search and replace functions, there are other, more specialised functions, most of which are found only in the PDS editor.

As the PDS editor is intended primarily for use with assembler code, there are certain functions which are tailored for this purpose. The first of these allows you to find a label definition, anywhere in the whole program.

Position the cursor on top of a label. Once the cursor is positioned, press [CONTROL] and [G] together which initiates the search for that label definition. The search starts at the beginning of file 0, and will search for the first occurrence of the label definition, if there are multiple definitions, then only the first definition is located. A label definition is recognised when a label is placed at the beginning of a line.

The cursor is then moved to the start of the label definition. If no label definition is found throughout the program, then a warning 'beep' will sound. Once a label definition has been found, you can return to your original cursor position by depressing the [undo] key (this is key [5] on the numeric keypad). You may use the cursor keys, to move around the code, before the [undo] function is selected, and the correct return position will still be retained. The [undo] key will restore your cursor position to the point where the function was executed, which may even be in a different file. Although, if you enter any text, or perform any function, then the original undo position will be lost, and reset to the cursor position before the function was performed.

If [CONTROL] and [G] are pressed when the cursor is over a label definition, ie at the start of a line, then a global find is performed on that label, using the GLOBAL, LABEL, and CASE options. You can then repeat the search for that label using the [CONTROL] & [T] function.

The [CONTROL] [G] function is probably one of the most used functions in PDS. A typical example of use, is when you are about to call a particular routine, but are unable to remember the routine's exact use. All you have to do is to place the cursor on top of the routine name, and press [CONTROL] and [G] together. You can then examine the routine in detail, and when you are ready to continue coding, just press the [undo] key.

A search for label definition can also be performed so that the definition is displayed in the alternate window and makes it possible to look at the current code, and the code relating to the definition. Simply place the cursor on top of a label, then press [CONTROL] and [0] on the numeric keypad together. If a label definition is not found, then a 'beep' will sound.

If the alternate window is not currently visible on the display, then it will be brought onto the display at the central position.

A simpler search function allows you to search for a quote or apostrophe character. When [CONTROL] and [Q] are depressed together a search is performed for a quote or apostrophe, on the current line only, to the right of the cursor. If a quote or apostrophe is found, then the cursor is positioned directly to the right of the character. If the search is unsuccessful then the cursor will be positioned at the end of the current line. Remember that the search only operates on the current line, to the right of the cursor.

If you have the following line of text in your program:

```
DB "Hello", "Goodbye", "Goodmorning"
   1      2 3      4 5      6
```

And the cursor is located at the beginning of the line, when you press [CONTROL] [Q], the cursor will be positioned at the 'H' character in the word 'Hello', in the line above this position is marked with a 1. If the [CONTROL] [Q] function is selected for a second time, then the cursor is positioned on top of the comma character between the words 'Hello' and 'Goodbye', this position is marked with the number 2. The position of the cursor after six [CONTROL] [Q] functions are executed, are marked with the number 1 to 6.

3.9 Using the PDS help file

Because it is always essential to document source code, a special facility is provided to aid documentation. In most cases code is documented by using comments before each major routine, together with a general file documenting the overall program. This can be wasteful and time consuming, as it can involve commenting most things twice.

Using PDS you can setup a help file for your program, therefore keeping all the documentation together in one separate file, the control of which is maintained by PDS.

To access help simply position the cursor over a key word (any word containing the letters A to Z, not necessarily a valid label character), or subroutine name, anywhere in the program. When the cursor is positioned, depress the [Help] key (this is [F7]). At this instant the help file is loaded off disk, into a temporary buffer area. The file is then searched for the required key word. If the key word is found, then the help file will be displayed in the alternate window, with the specified key word at the top of the window. If the key word search is unsuccessful, then the very end of the help file will be displayed, so that you can make a new entry in the help file.

The help file remains in the alternate window, in the same way as any other PDS file, and it can be custom edited. If you enter the alternate window, you can see that you are able to cursor up and down the help file, looking at any part of it. If you wish to make a new entry in the help file, simply insert an asterisk (a '*' character) at the beginning of a line, followed directly without a space by the key word, followed by a carriage return. This way, key words are distinguished within the help file. When a key word in the help file is accessed, it will be displayed in the top line of the window, with the relevant comments following it. If you have multiple keywords defined, then the last keyword is displayed at the top of the screen. As you would expect, you can also edit previous help file entries just by deleting and inserting characters in the normal way.

When you have finished editing or examining the help file, then simply swap windows. As soon as the alternate window is selected, the help file is updated on the disk. The help file will then no longer be visible on the screen, and the alternate window will display the current file it is editing.

The System is supplied with micro-processor help files as standard. These are supplied to start you off with sample help files. They list all the micro-processor's opcodes, with explanations and timings.

The help file may be loaded off different drives and directories, in the XMSDOS format. Make sure that the System is setup for the correct format, filenames or drive numbers. If the file is stored in normal MSDOS (or ASCII) format, it will not be loaded. For more details see chapter 9, which covers the configuration menus.

3.10 Printing with PDS

If you need to print any text, other than a block, you must use the [Print] key, [F6]. When the [Print] key is depressed, the following options are available:

PRINT: Window, Printer form feed, File or All files?

You may either enter [W], [P], [F] or [A] to select an option, or press [ESC] or [Finish] to abort the print operation.

The [W] option will only print the contents of the current window. Pressing [P] will send a form feed character to the printer, this is a character 12. Selecting [F] will start printing the current file to the printer, including the file header. All the print options discussed so far utilize the printer buffer. This means you can carry on working while your documents are being printed, but if the [A] option is used, the computer will start printing all the files, starting from file 0, without any buffering.

If you had already been printing when the [Print] key was depressed, then the current state of the printer buffer is shown on the command line. A character counter showing the number of bytes left to print is also displayed. If you wish to abort the print, then simply press [Finish] when the printer status is being displayed, this will also clear the computers printer buffer as well. When the printer buffer is being used, some PDS functions may not be accessible, as they share the same work memory as the printer buffer.

3.11 Accessing disk files from within the editor

There may be times when you need to insert external files into your program, or alternatively save them on to disk.

To insert a file from disk into the current cursor position, press [CONTROL] and [R] together. The current path is then displayed, ending with a '\' character, this can be changed if you do not wish to use that path. You then enter the filename terminated by a carriage return. If the file does not exist then a "File not found" error message will be displayed in the command window. In this case repeat this process with the correct filename. The file will be loaded in MSDOS format from the drive and path given.

Once the correct file has been identified, it will be inserted at the current cursor position. Before the characters are inserted, they are checked to make sure they are valid characters: if they are not valid, then they will not be inserted. As the file is inserted, two problems may occur. The first being that if any line in the file exceeds 120 characters in length, it will be split into two separate lines. A '9' marker is placed at this point, so that you may locate any split lines and take the necessary action. If there are a number of extra long lines, then they will all be split and marked with '9' position markers. This is the only time that more than one '9' position marker can exist, and when [CONTROL] and [F9] are depressed, the first occurrence of the '9' marker is located only. So that '9' marker has to be deleted, before the next split line can be located. If this has occurred when a file is being read, a warning message will be displayed after the file has been read in.

The other problem that may occur is that the PDS file may run out of memory. In this case, the rest of the file is not read in, and the insertion will be aborted. In this case either make room for the file, or load the file into a different PDS file.

It is also possible to save part of your code to an MSDOS file. To do this, define the area to be saved by placing start and end of block markers at the beginning and end of the piece code to be saved. See section 3.5 on how to place block markers.

Once the block markers have been placed, simply press [CONTROL] and [W]. You will then be prompted for a name for file, terminated in the normal way by a carriage return. The block will then be written to the specified MSDOS file.

When reading or writing MSDOS files, you can specify the exact path for that file. A path is a route that has to be taken to locate or position a file in any particular directory. A complete path name consists, of a drive specifier at the beginning, followed by any directory names, then finally the actual filename is placed at the end of the path name. See the example pathname below:

```
C:\PDS\GAME\CAT.INC
```

If [CONTROL] and [R] are depressed, then this will load the file 'CAT.INC' from the 'GAME' directory, which is located in the 'PDS' directory on drive C. As you can see, to distinguish between a file name and a directory name, you use the '\' character. This has to proceed and follow a directory name. If the preceding '\' is omitted, then the specified directory is located within the current directory. Therefore the full path name need not be entered, only the path from the current directory has to be entered, and the drive need not be specified. For example, if the current path is set to 'C:\PDS\' , then to access the file 'CAT.INC', only the following path has to be entered :

```
GAME\CAT.INC
```

So far only blocks of text have been loaded or saved. You may require to load and save complete PDS files as well.

Loading a new file into the current file will delete all the text in the current file. Pressing [CONTROL] and [L] will allow you to load a new file. This will return the prompt :

```
Load from (Msdos, Xmsdos, 0-7 or Help.) ?
```

You will notice that the characters M, X, 0, 7 and H are highlighted. Pressing the key that corresponds to one of these highlighted characters selects that particular option.

Selecting the 'M' or 'X' options loads a 'MSDOS' or 'XMSDOS' file into the current memory file. Both these options will also prompt you for a file name. If the file exists, then the contents of the current file are deleted and the new file is loaded into the current file from the current drive and directory.

If any of the numbers 0 to 7, are selected, then the corresponding PDS format file will be loaded into the current file. All previous contents of the current file are lost.

Selecting the 'H' option loads the help file into the current file. The previous contents of the current file are lost. The help file is loaded from its defined drive and filename.

To save the current file to disk, press [CONTROL] and [S]. This will return the prompt:

```
Save to (Msdos, Xmsdos, 0-7, Help, Back to XXXXXX.) ?
```

You will notice that the characters M, X, 0, 7, H and B are highlighted. If the 'B' is not highlighted, then the file has not been previously loaded, and therefore no back option is possible. Pressing the key that corresponds to one of these highlighted characters selects that particular option.

Selecting the 'M' or 'X' options saves the current memory file to an 'MSDOS' or 'XMSDOS' disk file. Both these options will also prompt you for a file name. The file is then saved in the correct format to the specified disk file.

If any of the numbers 0 to 7, are selected, then the current file is saved to the corresponding PDS format file on the currently selected PDS format drive.

Selecting the 'H' option saves the current file in place of the help file. Remember that any old help file on the disk will be overwritten.

If the 'B' option is selected, then the current memory file is saved back to the file it was originally loaded from. This file is shown instead of the XXXXXX that was shown here in the options prompt. If the filename was preceded by a '*', then the file is an XMSDOS file. The old contents of that file are overwritten.

3.12 Miscellaneous editor functions

There are a number of functions that cannot be classified under any particular heading. All of these functions will be covered here.

One time-saving feature is automatic label generators. Pressing [CONTROL] and [A] will generate a label. If the cursor is positioned at the beginning of a line, then this will increment the label counter and generate a new label. If the cursor is not positioned at the beginning of the line, then a label is generated with the current label counter, allowing access to the last generated label. "Generate" means that a label name is automatically typed at the current cursor position. Each file has its own independent label counter, and when a file is saved in XMSDOS or PDS format, then the current label counter values are also saved. The format for a generated label is as follows:

```
LA*!!!!
```

The 'LA' signifies that it's a [CONTROL] [A] generated label. The '*' will actually be the file number the label is generated in. The '!!!!' represents the current label counter value for that file.

There is another type of generated label. This is accessed by pressing [CONTROL] and [Z]. This works in exactly the same way as the [CONTROL] [A] label generator, except that the first two characters of the generated label are 'LZ' instead of 'LA'. The 'LZ' and 'LA' labels use two completely independent counters.

Another useful function allows you to jump directly to any line in the current file. If you press [CONTROL] and [J] together, you will be prompted to enter a number. Once this number is entered, the cursor will move to the specified line number. If the specified line number does not exist, then a 'beep' will sound and the cursor will not move.

Many programmers mark the start of routines by placing the routine name or description in boxes. By pressing [CONTROL] and [I], a comment box is drawn up automatically, in the format shown below:

```
*****
*
*****
```

The cursor is positioned in the middle line, at the first tab stop position. The required comment is then typed in, and should be ended with a '*' character in the last tab position, at default this is the 40th column position. This will complete the box shape.

You may at some point require to date stamp a particular portion of code, independently from the file date stamp. If you press [CONTROL] and [ENTER] on the numeric keypad, the current date and time is inserted at the current cursor position. The format for the date stamp is time first followed by the date, ie:

```
00:00:00 on 10/10/10
```

The time is shown in the order of hours, minutes and seconds. The order of the date is day, month and year.

The last function to be discussed in this section is the [CONTROL] and [+] (the [+] on the numeric keypad) function. If the cursor is placed at the first character of a numeric expression, then that expression is evaluated using the assembler's expression evaluator. For details of the expression evaluation process, refer to the assembler manual. If your program has recently been assembled, then any labels in the expression can also be evaluated. Once the expression is evaluated, the result is displayed in the command window in decimal, HEX and binary. If the number is a valid character code, then it will also be displayed in ASCII.

For example if the following expression is evaluated :

```
(12*(1234+66))/200
```

The command line will show:

```
Value = $004E, 78, %00000000-01001110, 'N'
```

The format for the numbers are, hexadecimal, decimal, binary, and ASCII. This can be useful to find out where parts of your program are, by evaluating label definitions after assembly.

4 How to use the PDS escape mode

An important area in PDS, which as yet hasn't been mentioned, is the PDS escape mode. The escape mode allows you to use a line editor, rather than the conventional screen editor. VTECO is a line editor, which is used by many system programmers. PDS is based on this editor, so any VTECO users can start using PDS very quickly.

VTECO is a standard editor format which allows nearly all screen editor functions to be performed from a command line language. Once fully understood, you will realize the potential of the powerful escape mode. Because of its ability to setup special lists of editing functions and repeat them at will, the escape mode comes into its own when similar and repetitive editing functions have to be performed.

4.1 How to use the escape mode

The escape mode is entered by pressing the [ESC] key at any time when editing a file. A block cursor will then appear in the command window at the top of the screen, and any further text entered will be displayed in this window. The format for an escape command is:

`nCexpression[ESC]`

`n` is a numeric expression, used to show the number of times a command is to be repeated. If `n` is negative, then the direction of the command is reversed, ie `4L` moves four lines down, `-4L` will move four lines up. If `n` is not present then the command is executed only once. If only a negative sign is shown, then the command is executed backwards once. The one or two letters that follow define the command to be executed. The command itself may also require an optional *expression*, which depends on the command being used. A command is finally ended with an [ESC] character (seen in the examples as '[ESC]'). At this point another command can be added to the end of the command line using '[ESC]' as a delimiter between commands. The command line is not actually executed until two [ESC] characters are entered after each other. The [ESC] character is displayed in the command window as a happy face character.

The line below will define a block start marker:

```
BS [ESC] [ESC]
```

Using the same principle, you can define the next two lines at the cursor position as a block :

```
BS [ESC] 2L [ESC] BE [ESC] [ESC]
```

Commands can be given priority by enclosing them in '<' and '>' (more than and less than) signs. All the text within these characters is executed before the text outside the signs. These act in the same way as brackets do in normal arithmetic and they can also be nested in the same way. If a numeric expression directly precedes the '<' character, then the list of commands enclosed by the '<' and '>' signs, will be repeated the specified number of times. eg

```
100<Fbill[ESC]Iy[ESC]>[ESC][ESC]
```

This example will find 100 occurrences of 'bill' in current file, and insert a trailing 'y'. If there are less than 100 occurrences, then a 'Limit reached' error message will be displayed after all the occurrences of 'bill' have been found.

If the cursor keys are used while still in the escape mode at the beginning of the command line, then you will exit the escape mode. Otherwise the cursor is moved in the current window while you are still in the escape mode. To quit the escape mode, simply press the [ESC] key on an empty command line. This will return you to the current cursor position in the current file.

All the commands relate to the current cursor position. Most of the escape mode commands have an equivalent control key in edit mode.

4.2 Escape mode functions

Following is a list of all the escape mode commands in alphabetical order:

- B** Homes the cursor to the beginning of the file in the current window. The top of a file is line position 1, chr position 1.
- BS** Inserts a start of block marker at the current cursor position, in the current file. The start of block marker is shown as an inversed less than sign (<). The start of block marker has to come before an end of block marker. When a start of block marker is defined, any other start of block marker in any file is deleted. Both start and end of block markers **MUST** be in the same file. This function performs exactly the same task as the [CTRL] + [<].
- BE** Inserts an end of block marker at the current cursor position, in the current file. The end of block marker is shown as an inversed more than sign (>). An end of block marker has to come after a start of block marker. When an end of block marker is defined, any other end of block marker in any file is deleted. Both start and end of block markers **MUST** be in the same file. This escape mode function is the equivalent to pressing [CTRL] + [>].
- BR** Reads a block from an MSDOS file, and inserts it at the current cursor position. The file name has to follow the command, ie.

```
BRc:\pds\file.txt[ESC][ESC]
```

When a file is read in, any non ascii characters are filtered. If any line in the file is longer than 120 characters, then the line will be split, and over flow onto the next line. A '9' marker is also placed on all split lines, this allows you to examine all the split lines, so that take the necessary action.

- BW** Writes a block to an MSDOS file. The filename has to follow the command, ie.
- ```
BWfilename [ESC] [ESC]
```
- This allows you to save your source code in normal MSDOS ascii format, so that it can be used by another application.
- BC** Copies marked blocks from any file to the current cursor position. The block markers have to be correctly set before you can use this function. You cannot perform a block copy if the cursor is within the block to be copied.
- The following is a simple example that will duplicate the two characters following the cursor.
- ```
BS [ESC] 2C [ESC] BE [ESC] BC [ESC] [ESC]
```
- BD** Deletes marked block. Ensure that the block markers are correctly placed, otherwise an error will be returned. You are NOT prompted for confirmation, so be sure that you want to delete the correct block. Once a block has been deleted it is lost completely. You cannot delete a block if the cursor is located in the block.
- BM** If the block markers are correctly set, then this function will move the marked block from any file to the current cursor position. This function cannot be performed if the cursor is located within the block boundaries. The block markers are also moved with the block.
- The example here will swap the next two pairs of characters over, using the block move function :
- ```
BS [ESC] 2C [ESC] BE [ESC] 2C [ESC] BM [ESC] [ESC]
```
- C** Moves n characters to the right of the cursor. The value n is placed directly before the function name, as follows:
- ```
nC [ESC]
```
- If n is a negative number, then the cursor will move n characters to the left. If you try and move the cursor beyond the start or end of a file, then the following error will be displayed in the command line window:
- ```
Limit reached
```
- D** Deletes n characters to the left of the cursor. The syntax for this function is as follows:



nD[ESC]

Remember that 'D' will delete to the left of the cursor, while 'C' move to the right of the cursor. If n is a negative number then characters to the right of the cursor will be deleted. The error 'Limit reached' will be displayed in the command line window, if you try to delete beyond a file's limits.

**F** Finds the nth occurrence of the text following the command, starting at the current cursor position. The search will terminate at the end of the current file. Remember any text can be searched for, including carriage returns and tabs. Once a successful find has finished, then the cursor is placed directly after the end of the search string. This is the valid format for this command:

nFtext[ESC]

If n is a negative value, then the find is performed backwards, towards the beginning of the file, and the cursor will be left at the start of the search string after a successful find.

If the nth occurrence of a string is not found, then the message 'Limit reached' will be displayed in the command line window.

In the following example, all the label definition with the name 'LOOP', will be found and the character directly after definition will be replicated. This example could be used if you wanted to change the names of all the 'LOOP' variables, which are all followed by a single digit. This situation may occur if you had a multiple label definitions.

```
100<F^MLOOP[ESC]BS[ESC]C[ESC]BE[ESC]BC[ESC]>[ESC][ESC]
```

Note how a carriage return (^M) character was inserted before the 'LOOP' label, this makes sure that only label definitions are affected.

**I** Inserts the text following the command at the current cursor position. If an n value is present, then the text is inserted n times. The following example, demonstrates a very simple use for this instruction, which just saves a little typing :

```
6I^ILSR^IA^M[ESC][ESC]
```

Tabs and carriage returns are displayed as ^I and ^M, as used in the example above.

**J** Moves to line number n, remember n is placed before the command as seen below:

```
188J
```

This example simply jumps to line 188. If the line does not exist, then the 'Limit reached' error will be displayed in the command window.

**K** Deletes *n* lines up. It is important to remember that the first repetition of this function will delete to the start of the line, and only on the second repetition will the previous line be deleted. This will occur, unless the cursor was already at the start of the line before the first repetition, in which case the preceding line will be deleted in the first repetition. The cursor will always be at the start of the line after this function is executed.

The following example will delete all the characters to the left of the cursor, and the preceding line :

```
2K[ESC][ESC]
```

If a negative repetition rate is given, the function will delete *n* lines down the file, where the first repetition will delete to the end of the current line, unless the cursor is already at the end of the line.

**L** Moves cursor *n* lines down. This function will always move the cursor to the beginning of a line, after its execution. If a negative value is given for *n*, then the first repetition will move the cursor to the start of the current line, unless its already there, while any further repetitions will move the cursor to the preceding line.

**M** Defines an executable macro. All the text following the command is actually a list of the real escape mode commands. This may be an operation that has to be repeated often. To actually execute the macro definition, [CTRL] and [E] have to be depressed while in edit mode.

When a macro is defined, the text following the M command is inserted directly into the buffer, without any error checking. So only when the macro is executed, will you any errors become evident, in which case the errors will be shown in the command line window.

**N** This checks if the character under the cursor is equal to the character on the right of the command. If they are equal then the next escape command is executed. If they are not, then control will return to the editor.

This is the only 'conditional' escape mode command, and can be very useful when using slightly more complicated macros. The following example will delete how ever many spaces found at the cursor position, this saves space wasting :

```
120<N [ESC]-D[ESC]>[ESC][ESC]
```

Note that this is repeated 120 times, as this is the maximum characters on any single line. As soon as a space character is not found, the escape mode sequence will end. This sequence would probably best be defined as a macro, as it would have to be used at the programmer's discretion.

- QR** The line under the cursor, in the current file, is copied into the macro buffer. The old contents of the macro are lost. This is extremely useful if the macro function is used often to generate codes or tables, as the macro definition used to generate the code can be placed in a comment line, just before the routine. Another alternative is to place all your popular macro definitions at the start of the program, then select them when they are required.
- QW** The current macro definition is copied onto a new line, at the current cursor position. It will be preceded by a ';', so as it not interfere with assembly. This function allows macro libraries to be built up. You can place all your macro definitions at the start of your program, using them as you require.
- When a macro is written to the file, all [ESC] characters are converted to a '|' character, while carriage returns are displayed as '~' characters.
- U** Moves n words right. A word is distinguished by a leading and ending space, tab or carriage return. A negative value of n, will move the cursor n words to the left.
- W** Edits file n in current window. This allows you to change edit file, and is equivalent to pressing SHIFT or CONTROL and a function key, ie:
- `3W[ESC]`
- Will start editing file three in the current window. You cannot edit or exit the help file from this command.
- X** Searches and exchanges a string, starting from the current cursor position. Tab and carriage return characters can be entered as part of the search or replace strings. Tabs will appear as '^I' and carriage returns will be displayed as '^M'. The format is:
- `nXsearch[ESC]replace[ESC]`
- The search and replace strings follow the command, and are separated by an ESC character. This is repeated n times. If n is a negative number, then the search and replace is performed backwards. An example of its use is shown below:
- `12Xhello[ESC]goodbye[ESC][ESC]`
- This will replace the next 12 occurrences of 'hello' with 'goodbye'.
- Z** Moves the cursor to the end of the file in the current window.

## 4.3 Escape mode macros

Escape mode macros allow you to execute popular escape mode commands, with just one keystroke.

To define an escape mode macro, you have to use the 'M' escape mode command. First of all enter the escape mode by pressing [ESC] key. Now enter the escape mode list of commands, proceeded by the letter 'M'. When the command is entered, there will be no responses, as the command has not been executed, but the macro has been defined. If the command syntax was wrong, then an error will be returned and the macro will not be defined.

This is an example of a macro:

```
M10I*****^M[ESC] [ESC]
```

This will define macro that will insert 10 lines of '\*\*\*\*\*' into the text, at the current cursor position. Hardly a useful macro, but this explains its use simply.

To execute the macro simply press [CONTROL] and [E] together, this will insert the ten lines of '\*\*\*\*\*'. You can now execute this macro as often as you'd like.

It is even possible to hold a library of macro commands in you code. If you enter the escape mode and issue the command :

```
QW[ESC] [ESC]
```

This will insert the current macro definition on a new line at the current cursor position, proceeded by a ';' character as in:

```
;10I*****^M
```

You can also read in a macro definition from the text, by placing the cursor on the macro definition and executing the escape mode command :

```
QR[ESC] [ESC]
```

We recommend that you store all your macro definitions in one area of you program, so as not to lose track of them.

## 4.4 Examples of popular escape mode commands

Now that you know how to use escape mode macros, you may still wonder why you need to use them. In this section, we will demonstrate two complex examples of macros in detail. This will give you some idea of their use, and maybe help you think of your own uses.

The first macro, shown below, is used to convert hex data in 'DW' format, into faster, more compact, and readable 'HEX' format statements.

```
;XDW |HEX |20<-D|BS|2C|BE|2C|BM|BS|BE|2D|N,|-D|>| |
```

The macro definition is shown in the format that will be read in using the 'QR' command in the escape mode. Once the macro is defined, every time you press [CTRL] & [E] the macro will be executed. The macro will search for a 'DW' command, and replace it with a 'HEX' command. The hex expressions following the 'DW' will be converted into compact hex numbers without identifiers or spaces and commas. Any number of expressions can be converted, as long as they are separated by commas. For example, the line below :

```
DW &1234,&5678,&7856
```

Will be converted to :

```
HEX 341278565678
```

You can see that the low and high bytes have been swapped around. The macro will only work if the 'DW' command uses four digit hex expressions, that are proceeded by a hex identifier, and all the expressions follow each other directly with only a comma separating them.

This macro is useful if you are trying to make your source code more compact, and to achieve faster assembly times.

The second example is even more complicated, and is very specialised. The macro below, is shown in a format that can be read in using the 'QR' command

```
;This will take two shapes, both 16x8 in mono, separated by one line and merge them
```

```
;8<FHEX |20L|I HEX ~|2<-21L|5C|BS|2C|BE|21L|-C|BM|-
11L|5C|BS|2C|BE|11L|-C|BM||-20L|>| |
```

When you enter the above macro, type it in on one line, as it is only split over two lines here because it cannot be fitted on one line.

This macro will take two 16x8 shapes (8 lines of hex data, with 2 hex bytes on each line), separated by two lines, and merge them together into one shape. This is useful if you had a shape and its mask stored separately in your text, but you needed the them to be combined together, with 1 byte of shape, followed by one byte of mask. This would allow the routine that accessed them, faster and shorter. If your blocks of data looked like this :

```
;<16x08a>
```

```
HEX 1256
HEX 1256
HEX 1256
HEX 1256
HEX 1256
HEX 1256
HEX 1256
HEX 1256
```

```
;<16x08a>
```

```
HEX 3478
HEX 3478
HEX 3478
HEX 3478
HEX 3478
HEX 3478
HEX 3478
HEX 3478
```

If the cursor is positioned at the beginning of the first line of hex, and the macro is executed, then the resulting code will look like this:

```
;<16x08a>
```

```
HEX
HEX
HEX
HEX
HEX
HEX
HEX
HEX
```

```
;<16x08a>
```

```
HEX
HEX
HEX
HEX
HEX
HEX
HEX
HEX
```

```
HEX 12345678
HEX 12345678
HEX 12345678
HEX 12345678
HEX 12345678
HEX 12345678
HEX 12345678
HEX 12345678
```

The macro will also work with other sized data, but certain numbers in the macro must change. The number at the beginning of the macro, determines the number of lines in the hex shape. The 2 found before the second '<' and '>' expression, determines the number of hex bytes there are per line. All the 'L' (line move) commands, must then be recalculated, depending on the number of hex lines there are in a shape, and the number of lines that separate the shapes.

If you re-write and experiment with the macros, you will be able to understand them much more clearly.

## 5 The Disk system

The PDS disk system is a complete mini disk operating system. PDS caters for a number of disk formats to give you speed, compatibility and flexibility depending on the format used. We realize the importance of disk handling to ensure that disk mishandling is avoided as much as possible.

### 5.1 The disk system format

To enter the disk system, press the [Disk] function key, [F9], from the editor. Once entered, the display is cleared with only the top command window remaining unchanged. If you entered the system with your PDS booted of your hard disk drive (drive C:), then the display may look something like this:

```
>0< Line 00001 Chr 001 Mem 00000 <<< Z80 >>> <1> Line 00001 Chr 001 Mem 00000
```

| FILE | TITLE | AUTHOR | VERSION | SIZE | ORIGIN            |
|------|-------|--------|---------|------|-------------------|
| 0    | ----- | -----  | 00.00   | 00k  | New file          |
| 1    | ----- | -----  | 00.00   | 00k  | New file          |
| 2    | ----- | -----  | 00.00   | 00k  | Has not been used |
| 3    | ----- | -----  | 00.00   | 00k  | Has not been used |
| 4    | ----- | -----  | 00.00   | 00k  | Has not been used |
| 5    | ----- | -----  | 00.00   | 00k  | Has not been used |
| 6    | ----- | -----  | 00.00   | 00k  | Has not been used |
| 7    | ----- | -----  | 00.00   | 00k  | Has not been used |

```

PDSZ80.EXE PDS6502.EXE HELP.65 HELP.Z80 SPEC.DLO AMSTRAD.DLO
C64.DLO BBC.DL1 MAKEC64.DL1 SPEC.DL1 AMSTRAD.DL1 C64.DL1
C64.DL2 SPEC.DL2 AMSTRAD.DL2 MANUAL.IBM

```

```
MSDOS 12378k, PATH=C:\P-D-S
MSDOS FORMAT : 1[LOAD] 2[SAVE] 3[Move] 4[Path] 5[Exten] 6[More]
```

As you can see from the screen, the top half of the display contains information about the current program in memory, while the lower half of the screen contains information about the currently selected disk, in this case a directory of the current MSDOS path on drive C:.

The function keys, F1 to F6, now have new uses. Their new functions are displayed at the bottom of the screen. These will change again while you are using the disk system, but their current use is always displayed here.

Program information includes the current state of affairs of each file, its number, title, author, version number, length, and origin. These are the headings shown at the top of the information box, with the relevant information directly in the column below.

The file number is the normal number associated with that file. Whenever functions are directed at a file its file number is always used, so you should always be aware of a file's number as well as its title.

The file title is the title allocated to that particular file by you. The allocation would have been made via the configuration menus in chapter 9. If the title is not set, the title name will be blank. The file's author is also set by the configuration menus in chapter 9. In most cases this will tend to be the same for all the files, but can be varied.

A file's number is the current version of that particular file. This takes the form of a 4 digit number, with two decimal places, XX.XX. When a file is still new this is set to 0.00, but whenever a file is saved the version number will increment by 0.01 - ie the same version number that is found in the header at the beginning of each file.

The size is the current size of the file in K bytes. One K is exactly 1024 bytes, and helps you maintain an even source code level between files, making source code handling even easier. At the bottom of the size column is the total size of source code within the program, again in K bytes.

Finally the origin of the program is the area from which the file was originally loaded. This would normally give the file name, or PDS file name, or if the file is new the origin will be "New file".

## 5.2 MSDOS disk handling

It is useful to know basic MSDOS file and directory structures. Many users neglect to study these features, and generally end up with very confusing and cluttered hard disks. Most of the features discussed here, are most relevant to hard disks, but still apply to floppies.

When your system has booted, MSDOS will place you in the root directory. This is the main directory in your system, from which all files and directories can be accessed from. We recommend that you never have general files stored in this directory, other than 'AUTOEXEC.BAT', 'CONFIG.SYS' and other system related files. These have to be placed in the root directory. All other files should be split into their particular fields, and given separate directories.

When you first install PDS on your hard disk, we recommend that you make a new directory, specifically for PDS. This can be done by entering the following, at the MSDOS prompt:

```
MD \PDS
```

You could have also used the PDS disk system to do this, but this explained further on in this chapter. The 'MD' command, is used to Make a Directory, and the '\' character at the beginning of the following expression, means that the new directory, PDS, should be made from with the main directory. To actually use the directory we created, then the 'CD', Change Directory, command must be issued. To select the PDS directory, enter the following:



```
CD \PDS
```

If you now type 'DIR', you will notice that the directory is now empty, except for the ' <DIR >' and '.. <DIR >' directories. These will be explained later. Now that the PDS directory has been created, you should copy all of your PDS files into the PDS directory. Assuming your hardisk is labeled 'C:' and your floppy 'A:', then type in the following:

```
COPY A:*. * C:\PDS
```

Whenever a '\' is placed at the beginning of a file or path name, then, the files or directories are accessed from the root directory. If the file or directory can be found in the current directory, then the '\' character is not needed. You can also create more directories from within a directory, even if both directories share the same name. The best way to use PDS, is to define a directory for each program that you are developing, all from the PDS directory. This way you can easily move between your programs, using the 'PATH' option in the disk system. You can nest directories to any level, but too many may become unmanageable. If you wanted to create a directory called 'INVADERS' for your program, then simply type:

```
MD INVADERS
```

This will create the directory in the current directory, which should be PDS. If we then save a file called 'X0.PDS', in that directory, the file's whole name becomes:

```
\PDS\INVADERS\X0.PDS
```

Notice how the '\' character is used as a delimiter between directories and filenames. If we wanted to access the 'X0.PDS' file from the PDS directory, we could use the following name instead:

```
INVADERS\X0.PDS
```

The leading '\' is no longer required, as the file can be accessed through the current directory. If a directory or file was improperly accessed, or just does not exist, then an 'Invalid filename' error is returned.

It is important that you get to grips with directories, as they allow you to organise and categorise your files.

## 5.3 Explanations of all the disk formats supported

The Programmers Development System supports three different types of disk format. Each has a specific use, making it more efficient when used in particular contexts. The three formats are XMSDOS, PDS, and MSDOS.

XMSDOS is probably the one most commonly used because it works with standard MSDOS files, but is still reasonably fast and saves complete files. When a file is saved, all the header information, cursor position, function keys, and tab settings are saved with it. This adds 772 bytes to the length of the file. XMSDOS still uses the MSDOS operating system calls to save the complete file as a block of memory, so doesn't need special format disks. This does mean that other programs will not be able to access those files, and the file cannot be TYPED from the MSDOS prompt. Although they still use normal format MSDOS format disks. This is the only complete format save for hard-disks.

The PDS format is a disk format specifically designed for PDS. This only works on floppy disks, and can speed disk handling up by 3 times more than normal MSDOS. All the header information, cursor positions, function keys, and tab setting are saved with each file, as with XMSDOS. PDS has its own disk handling routines for this format, which are faster and more efficient than MSDOS equivalents. PDS format disks have to be formatted as PDS disks, and will not function properly as normal MSDOS disks.

The MSDOS disk format saves all files in normal MSDOS file formats, but this format is slowest, and does not save the header information, cursor positions, function keys, or tab settings. This allows other programs to access the PDS files normally, and is useful when files have to be sent by serial port, or have to be read in by word processors. When files are loaded or saved in MSDOS mode, they are accessed a byte at a time, with each byte being filtered, so that only valid ASCII characters are used. When an MSDOS file is loaded, any lines which are longer than 120 characters, are split over two lines, with a 9 marker being placed at the split point.

The default MSDOS and PDS drives can be setup in the configuration menu. Their setups vary depending on the system you're using, but the overall handling remains the same. If a hard disk drive is present, then that should be the default MSDOS drive. See the Configuration menu chapter, for details on changing default drives.

The bottom half of the display shows the directory of the currently selected disk format. More details follow in the next two sections.

## 5.4 Using the XMSDOS, PDS, and MSDOS menu system

When the disk system is first entered, the function key uses are reset. Their new setup is shown on the bottom line of the display, as discussed earlier. When selecting a particular function, the [ESC] key can be used to abort the selection. To return to the editor, press the [Finish] or [ESC] keys. This line shows the following information (the actual order that the menus are displayed in varies depending on the current configuration, this is just one of the possible order):

**XMSDOS : 1[LOAD] 2[SAVE] 3[Dir] 4[Path] 5[Exten] 6[More]**

The System is currently in XMSDOS mode. The function keys F1 to F6 are setup as, [LOAD], [SAVE], [Dir], [Path], [Exten], and [More]. The bottom half of the screen will display the directory of the currently selected MSDOS drive, and is defined in the configurations menu. The bottom line shows the disk space available in K bytes, and the current MSDOS path (XMSDOS and MSDOS use the same pathname).

Each option is explained below:

[LOAD]            The load function allows you to load a file from the MSDOS disk in XMSDOS

format. You will be prompted "Load which file (0-7, all)?" Type in the number for the file you wish to load. This may be any number key from 0 to 7. You will then be asked "From \*:", where '\*' is the current drive. Enter the filename of the XMSDOS file. XMSDOS files are normally numbered X0.PDS, X1.PDS, X2.PDS, X3.PDS, X4.PDS, X5.PDS, X6.PDS, and X7.PDS, (unlike the MSDOS files, which do not have a leading X in the filename) the number reflecting the file it holds. The PDS extension is the default extension, but can be changed by using the [Exten] key or the configuration menu. The file is loaded from the current path (as shown at the bottom of the display). If the [L] key was depressed at the original menu option, then all the files will be loaded from the current path, using the current extension with names X0 to X7.

**[SAVE]**

The save function allows you to save a file from memory to an MSDOS disk in XMSDOS format. You will be prompted by "Save which file (0-7, all)?" Type in the number for the file you wish to save to disk. This may be any number key from 0 to 7. You will then be asked "To \*:", where '\*' is the current drive. Enter the filename of the XMSDOS file. XMSDOS files are normally numbered X0.PDS, X1.PDS, X2.PDS, X3.PDS, X4.PDS, X5.PDS, X6.PDS, and X7.PDS, the number reflecting the file it holds. The PDS extension is the default extension, but can be changed by using the [Exten] key or the configuration menu. The file is saved to the current path (as shown at the bottom of the display). If the [L] key was depressed at the original menu option, then all the files will be saved to the current path, using the current extension with names X0 to X7.

**[Dir]**

When depressed, will prompt you "Create or Remove directory?". Answer by pressing the [C] or [R] keys. A directory is a folder holding files of the same description. Directories are most useful when using hard disks, as they allow neater handling of files. Instead of all files being in one single directory, they can be categorised into different directories. A directory is enclosed by '<' and '>' signs at the bottom of the display. Directories can be nested to any level. After you have answered [C] or [R], you will be prompted for the filename of the directory. This is a word of up to 8 characters, with no extension. When a file is accessed, through a number of directories, the list of directories is known as the path name.

**[Path]**

Lets you define the default path, this the drive letter followed by list of directory names, from which all files will be loaded or saved. As directories can be nested this is useful for retrieving work in a specific directory as if it were the only or root directory. If a path through a number of directories is defined then a '\' character is used in between each directory entry. You will be prompted for a new directory name.

**[Exten]**

Allows you to define the 3 characters extension on XMSDOS files. Every filename has an extension, and using this command defines the default extension that PDS will use. If the [Exten] key is depressed, you will be prompted to enter up to three letters.

**[More]**

Takes you to the next level of the menu system, which happens to be PDS. The bottom line of the display will show the new function key assignments and the current disk format in use, as seen below:

**PDS FORMAT : 1[LOAD] 2[SAVE] 3[Move] 4[Path] 5[Lock] 6[More]**

The bottom half of the screen will display the status of the PDS drive (as set by the path or the configuration menu), normally drive A:. If a PDS disk is in the drive, then a full rundown of each file is shown, in the same format as the upper half of the screen.

Each option is explained below:

- [LOAD] Loads any PDS disk file into memory. You are prompted to enter the destination file number, [0] to [7] or [L] to load all files. After which you are prompted for the source file, again [0] to [7] for PDS file or [H] to load the help file. If load all was originally selected then all files will be loaded, with no further prompts.
- [SAVE] Saves a PDS file to disk, in PDS format. You are prompted to enter the number of the file to be saved, [0] to [7] or [L] to save all files. After this you are prompted for the destination file, again [0] to [7] for PDS file or [H] to save the help file. If save all was originally selected then all files will be saved.
- [Move] Moves a PDS format file from the PDS drive to an MSDOS file on the MSDOS drive. You are first prompted for a file number ([0] to [7]) or [H] for the help file, then prompted for the destination MSDOS filename.
- [Path] Allows you to define the drive activated for all future PDS accesses. This may be answered by either pressing the [A] or [B] keys. Only floppy disks may be used as PDS disks.
- [Lock] Using this feature any PDS file can be locked or unlocked. Locking a file means that it cannot be overwritten. The lock command actually inverses the lock status, ie if it was locked, it will become unlocked and if unlocked it will become locked. You are prompted for the file number [0] to [7], [H] to inverse the lock status of the help file or A to inverse the lock status of all the files.
- [More] Activates the next level of menu systems for MSDOS format, menu 3. The bottom line of the display will now read :

**MSDOS FORMAT : 1[LOAD] 2[SAVE] 3[Move] 4[Path] 5[Exten] 6[More]**

The bottom half of the screen will display the status of the MSDOS drive (as set by path, defaults to drive C). If there is no disk, or a PDS disk in the drive, then the display will show this. If a MSDOS disk is in the drive, then a directory of the drive is shown, with disk space free and path name displayed at the bottom. The current MSDOS directory is only updated when the disk drive is accessed, ie when you enter the disk system, or change path.

Each option is explained below:

- [LOAD] The load function allows you to load a file from the MSDOS disk in MSDOS format. You will be prompted by "Load which file (0-7, all)?". Type in the letter for the file you wish to load. This may be any number key from 0 to 7. You will then be asked " From \*:", where "\*" is the current drive. Enter the filename of the MSDOS file. MSDOS files are normally numbered 0.PDS, 1.PDS, 2.PDS, 3.PDS, 4.PDS, 5.PDS, 6.PDS, and 7.PDS, the number reflecting the file they hold,

but can be called any legal filename. The PDS extension is the default extension, but can be changed by using the [Exten] key or the configuration menu. The file is loaded from the current path (as shown at the bottom of the display). If the [L] key was depressed at the original menu option, then all the files will be loaded from the current path, using the current extension with names 0 to 7.

- [SAVE] The save function allows you to save a file from memory, to an MSDOS disk in MSDOS format. You will be prompted by "Save which file (0-7, all)?" type in the letter for the file you wish to save to disk. This may be any number key from 0 to 7. You will then be asked "To \*:", where '\*' is the current drive. Enter the filename of the XMSDOS file. XMSDOS files are normally numbered 0.PDS, 1.PDS, 2.PDS, 3.PDS, 4.PDS, 5.PDS, 6.PDS, and 7.PDS, where the number reflects the file it holds. The PDS extension is the default extension, but can be changed by using the [Exten] key or the configuration menu. The file is saved to the current path (as shown at the bottom of the display). If the [L] key was depressed at the original menu option, then all the files will be saved to the current path, using the current extension with names 0 to 7.
- [Move] Allows you to move a MSDOS file from the MSDOS drive to a PDS file on the PDS drive. You are prompted for the MSDOS filename, and then for the PDS file, [0] to [7] or [H] for the help file.
- [Path] Lets you define the default path, this the drive letter followed by list of directory names, from which all files will be loaded or saved. As directories can be nested this is useful for retrieving work in a specific directory as if it were the only or root directory. If a path through a number of directories is defined then a '/' character is used in between each directory entry. You will be prompted for a new directory name. Note that MSDOS and XMSDOS menu options share the same path name.
- [Exten] Lets you define a 3 characters extension on MSDOS files. Every filename has an extension, and using this command allows you to define the default extension PDS will use. When the [EXTEN] key is depressed, you will be prompted to enter up to three letters.
- [More] Takes you to the next level of the menu system - the utilities menu.

## 5.5 Using the utility menu

This menu gives access to general utilities, and also displays the status of the MSDOS drive, as in menus 1 and 3. The function key status line will show :

**UTILITIES : 1[Bkup] 2[Format] 3[Delete] 4[Copy] 5[Rename] 6[More]**

- [Bkup] Copies all the contents of drive B to drive A. The process is started as soon as the space bar is depressed. This facility is only available on a twin floppy disk drive system.
- [Format] Allows a used or unused floppy disk to be formatted. You will first be prompted for the drive name, enter A or B. You will then be asked to format in either MSDOS or PDS format, answer [M] or [P]. If you have a double sided disk drive

system, you will also be prompted to format in Single or Double sided format.

- [Delete] Allows you to delete an MSDOS file from the current MSDOS path.
- [Copy] Allows copying of specific MSDOS files. You are prompted for the filename to be copied, then for its destination filename.
- [Rename] Allows the user to rename any MSDOS file to any other name. The current name has to be entered first, followed by the new name.
- [More] Returns to menu 1 (XMSDOS format).

---

## 6 Graphics Editor

---

### 6.1 Entering the graphics editor

The graphics editor can be entered by pressing the [Graphics] key, [F3]. At this point the System examines the text under the cursor. If the cursor is placed on previously generated graphics (these are a series of HEX statements, prefixed by a special header line), then the editor will use these graphics for editing. If the text cannot be converted into graphics then you will be allowed to create a new graphic shape.

To create a shape, just press the [Graphics] key when the cursor is not positioned on any graphics data. You will then be prompted to enter the size of the shape to be edited, in X then Y order. The X value must be a number from 1 to 64, any other value will not be accepted. The height of the shape has to be a number ranging from 1 to 48. The shape type will now have to be defined. There are four possible selections representing the following choices:

1. ABCDEFGH. This will select a format for any two colour modes, where the letters A to H signify the arrangement of the pixels. This is a universal mode, as the Spectrum, Commodore, Amstrad, BBC, Atari all have this type of high resolution screen mode, with 8 pixels to one byte.
2. abcdABCD. This will allow four colour graphics to be designed, each byte catering for four pixels. The arrangements of the bits signify where the pixels A, B, C and D are located. The uppercase letters are the high bits and the lower case letters the low bits. This mode will prove suitable for the Amstrad range of computers.
3. AaBbCcDd. This is also a four colour mode (as in type 2.), but the bits are arranged differently. This mode will prove suitable for the Commodore 64 and Atari range of computers.
4. ABCDabcd. This is yet another four colour graphics mode, with the bits arranged in a different order to those in modes 2 and 3. This mode will prove suitable for the BBC range of computers.

The final question asks you if you require the output to be in Row or Column format, enter R or C for the required format. When a shape is edited the bytes outputted into the HEX statements have to be placed in a suitable order for the user shape drawing routine. Two of the most popular formats have been catered for, bytes can be outputted left to right, from top to bottom, this is the Row format. The bytes can also be outputted top to bottom, from left to right, this is the Column format.

### 6.2 The graphics editor display format

At this point you will find yourself in the main graphics editor. If the sprite size chosen was 10 by 10, then the display should look like this:

In the top left of the screen a 10 by 10 grid is shown, and all sprites are edited in this grid. The edit cursor blinks at the current cursor position (at the top left hand side of the grid when the graphics editor is entered). On the right hand side of the screen, two boxes are displayed. The top box will always have a miniature version of the shape that is being edited in the main grid, referred to as the cell. This will give you some idea of what the shape will look like on the target computer. The box below is known as the buffer box and will be explained in detail later. Below the two boxes, the current drawing colour is displayed - any pixel drawn by the editor will be in this colour. Finally, below the current colour, your current X and Y positions relative to the top left of the shape are displayed constantly.

The bottom two lines show the new function key settings. These are:

|      |   |          |
|------|---|----------|
| [F1] | = | [Shift]  |
| [F2] | = | [Buffer] |
| [F3] | = | [Cell]   |
| [F4] | = | [Clear]  |
| [F5] | = | [Keep]   |
| [F6] | = | [Quit]   |



## 6.3 Drawing using the graphics editor

To move the cursor use the numeric keypad, pressing the any of the numeric keys surrounding the number five. The cursor will then move in the same direction as the key is relative to the [5] key. For example, pressing [4] on the numeric keypad will move the cursor left, while pressing [3] will move the cursor diagonally right and down. Holding down any of the keys will repeat the move, and when the cursor is moved off any of the grid edges, it will re-appear on the opposite edge.

If the [space bar] is pressed, the pixel under the cursor is set to the current colour, unless the pixel already contained the current colour, in which case the pixel is deleted and set to colour 0. Therefore, if the [space bar] is repeatedly depressed then the pixel will be set, and cleared repeatedly. The colour of the pixel can be incremented (or set to colour 3 in the two colour mode) by pressing the [5] key on the keypad, or decrementing the colour number by pressing the [0] key (or set to colour 0 in the two colour mode). Colours can also be selected directly by pressing [A], [B], [C] or [D] for colours 0, 1, 2 and 3 respectively. If you are working in the two colour mode then only colour 0 and 3 are available.

If the [SHIFT] key is held down while the cursor is moved, then a trail of pixels, set in the current colour, will be left at the cursor positions. If the [CONTROL] key is held down while the cursor is moved, then any pixels accessed by the cursor will be cleared to colour 0. These are all the basic controls allowing you to design any type of shape.

## 6.4 Special graphics editor functions

There are other useful functions to aid shape development. All the extra functions require additional keystrokes, involving the function keys. Once a function key is pressed, any option available will be shown at the bottom of the display. Each of the keys will be discussed individually:

**F1 [Shift]** This allows you to shift all the graphics in one of four directions, either Up, Down, Left or Right by pressing [U], [D], [L] or [R]. Pixels that 'fall off' the shape to any side, will re-appear at the opposite side. This option remains active until any other key is pressed.

**F2 [Buffer]** The buffer function allows operations to be carried out on the buffer box. The box is used to store and retrieve shapes, and remains intact until cleared or when you exit the PD System. Seven separate operations can be directed to the buffer box, each one is explained below (to select use the first letter of the explanatory word as the key):

**Get** Gets the contents of the buffer box and transfer it to the main cell. The contents of the buffer are clipped to the size of the cell before the transfer.

**Put** Puts the main graphics in the cell into the buffer box. All the previous contents of the buffer box will be lost. Note that only the exact size of the cell graphics is copied into the buffer area, so any surrounding data in the buffer will remain unchanged.

**Set** This logically OR's the cell shape with the buffer shape. For

every pixel that is set in the buffer box, the corresponding pixel in the cell box will also be set. If the pixel is clear in the buffer box then the corresponding pixel in the cell box is left intact.

- Reset** For every pixel that is clear in the buffer box, the corresponding pixel in the cell box will also be cleared. If the pixel is set in the buffer box then the corresponding pixel in the cell box is left unchanged.
- Xor** XOR's the cell shape with the buffer shape. For every pixel that is set in the buffer box, the corresponding pixel in the cell box will be reversed. If the pixel is clear in the buffer box then the corresponding pixel in the cell box is left intact.
- Clear** Deletes any shape in the buffer box, if the response to the "ARE YOU SURE (Y/N)" prompt is [Y].
- Exchange** Transfers the contents of cell box to the buffer box and the contents of the buffer box to the cell box.

**F3 [Keep]**

This leaves the graphics editor without creating any 'HEX' statements, but it will also leave the contents of the cell box intact. Therefore, the next time the graphics editor is used, the current graphics will be used, and any graphics under the cursor at the time are ignored. This is extremely useful if you need to pop out of the graphics editor briefly.

**F4 [Cell]**

This allows access to operations specifically for the cell box. The options are listed below (to select them enter the first letter of the explanatory word as the key):

- Invert** Inverts the whole shape. This will change all the colour 0 to colour 1 and all the colour 1 to colour 0 in the two colour mode. In the four colour mode colours 0 and 3 will be swapped, along with colours 1 and 2.
- Rotate** This function can only be used when the shape being edited is perfectly square (ie the size co-ordinates match). If this is the case then the entire shape will be rotated by ninety degrees clockwise.
- X mirror** This mirrors all the graphics on the X axis, ie the top half of the shape will be reversed and swapped with the bottom half of the shape.
- Y mirror** This mirrors all the graphics on the Y axis, ie the left half of the shape will be reversed and swapped with the right half of the shape.

**F5 [Clear]**

This will clear the cell box completely. Before doing so, your action is prompted with Y(es) or N(o).

**F6 [Quit]**

This leaves the graphics editor without generating any text whatsoever, losing the

graphics stored in the cell box.

To exit the graphics editor just press the [Finish] key, and will also save the current shape in HEX statements. The HEX for the shape will either be left at the current cursor position or, if you were modifying a previously defined shape, the HEX will be left directly before the old shape. If a label was found at the beginning of the original shape, then the same label will be placed at the beginning of the new shape. Either way, the block markers will be set to the start and end of the HEX, so you can immediately copy, move or delete the new graphics, or delete the old graphics using line delete ([CONTROL] and [DEL] held down together).

## 6.5 Using the buffer to convert between formats

At some point you may require to convert a two colour Spectrum shape into multi-colour Commodore 64 graphics. This can easily be done, by using the buffer.

Find the shape to be converted, and when the cursor is on top of any part of it, press the [Graphics] key. The shape will then be shown in the main grid. Select the [Buffer] option on the microscreen, and put the shape into the buffer. Now leave the graphics editor using the [Quit] key.

The contents of the buffer are still retained at this point. Find a good position for the new shape, making sure it does not interfere with any other shape. Press the [Graphics] key again, and enter all the parameters for the new shape. Finally select the [Buffer] option from the microscreen and get the shape from the buffer. You now have a converted shape in the cell box. When converting between colours see the table below:

| 2 Colour to 4 colour |       | 4 colour to 2 colour mode |       |
|----------------------|-------|---------------------------|-------|
| original             | final | original                  | final |
| 0                    | 0     | 0                         | 0     |
| 3                    | 3     | 1                         | 3     |
|                      |       | 2                         | 3     |
|                      |       | 3                         | 3     |

The [Finish] key can now be pressed to put the HEX for the shape into the source code. A shape's size can also be converted in this way.

## 6.6 The output format of the graphics editor

On exiting the editor, you can leave your shape embedded within your source code, which is achieved by using the 'HEX' format. The HEX for the shape has a specially defined format which includes a header line and a list of HEX statements. For a shape to be valid it must have the header line for the shape, this may look something like:

```
;<10x08a>
```

The semi-colon at the beginning of the shape header makes sure that it will be ignored during assembly. The vital shape header is then enclosed in greater than('>') and less than signs ('<'). The first number defines the pixel width of the sprite, and the next value defines the height of the shape. The two numbers separated by an 'x'. Note that even when these values are below 10 they still require a leading zero. The final letter is known as the shape identification. This tells the graphics editor which format the shape was generated in, be it a two colour mode in columns or a four colour mode in rows. The identifications byte may be any of the following:

| Identification code | Use                                            |
|---------------------|------------------------------------------------|
| a                   | Format 1, row mode                             |
| b                   | Format 2, row mode                             |
| c                   | Format 3, row mode                             |
| d                   | Format 4, row mode                             |
| A                   | Format 1, column mode                          |
| B                   | Format 2, column mode                          |
| C                   | Format 3, column mode                          |
| D                   | Format 4, column mode                          |
| .                   | Character set, followed by format code (above) |

If you have any trouble entering a previously defined shape into the editor, then look out for extra spaces, tabs or loose characters at the beginning or the end of a line. You should find that as soon as the shape is cleaned up the graphics editor should accept it readily.

## 6.7 Using the graphics editor with character sets

The final option in the table shown above, the full stop ".", is used to define a character set character. This has to be followed by a format type character and, as normal, can only be a, b, c, d, A, B, C or D. Each character can only take up one line, therefore the character shape is not broken up into different HEX statements, but brought together on one line. A character can be of any size and any format, with only one header byte required for a whole list of characters.

Shown below is a small four character, character table, which could be used on any computer with a hires 2 colour graphics mode:

```
;<08x08.a>
HEX 0001020408102040
HEX FFFFFFFF00000000
HEX FF00325687921347
HEX 675444BB56DE78FA
```

If the cursor was on the third character (the third line down), then that character will be transferred to the editor. If a new character is generated, it will be left directly before the original character. If a 16 by 4 character is required, it may look like this in format b:

```
HEX 0102040810204080
```

In format B the same character would be stored like this:

```
HEX 0104104002082080
```

See the difference between the column and row modes for yourself. If a character set option is required the header line has to be typed out by hand in the format shown.

---

## 7 The monitor

---

### 7.1 Entering the monitor

When entering the monitor by pressing the [Monitor] key ([F4]), the old monitor display which had previously been saved, is brought back. If you are entering the monitor after loading PDS, a help screen is displayed.

The monitor differs slightly in appearance, depending on the version of PDS running. Another difference is in the disassembler. The difference, however, is only skin deep, ie the code may differ, but the format and general style of disassembly is the same.

The screen is basically divided in two. The top two status lines now display the value of the machine's registers and current pointers. It is here that the monitors vary: obviously different processors have different register sets, and architecture.

### 7.2 Using the screen editor

The lower window is the display window, and it is set up as a full screen editor. The cursor can be moved around the screen using the four cursor keys. If you move the cursor left when you are at the beginning of a line, it will move to the end of the previous line. Each line can have one command, and cursoring to an old line, and pressing [return] will execute that line. The [backspace/delete] key will delete the character to the left of the cursor, and insert a space in its place. You cannot delete past the left edge of the screen.

To insert an extra space within a command line, just press the [INS] key. This will shift all the characters to the right of the cursor, right by one character position. If a line becomes too long, then the last character of that line is lost. Alternatively, the [DEL] key deletes the character under the cursor, and brings all the characters to the right of the cursor. The [HOME] key moves the cursor to the top left hand corner of the screen. If the [HOME] key is depressed together with the [CONTROL] key, it will clear the entire screen, and all the information on the screen will be lost. You can delete the current line by pressing [CONTROL] and [PG DN] together.

Most of the monitor commands are single letter commands, and must start at the beginning of a line. The monitor in the PDS is comprehensive and some of the operations it is performing need to be fully appreciated. First, it is important to remember that NONE of the commands affects the PC in any way. All the operations are either passed to the target computer, or emulated by the PC. All the other features of the monitor are explained in the assembler manuals, as these are specific to the micro processor being used.

---

## 8 The assembler

---

One of PDS's main features is the assembler. The assembler is completely integrated within the editor, thus allowing data to be shared between them. The assembler itself is not documented here, only the editor's link with the assembler and general source code handling is covered. For specific details about the assembler, please refer to the assembler manual supplied with the system.

### 8.1 Invoking the assembler

To assemble a program simply press the [Assemble] key, this is function key 1. You can also assemble by pressing [CONTROL] and [O]. It is important to remember that you cannot assemble when the printer spooler is printing: both parts of the system share the same data area and therefore cannot run at the same time. PDS will allow you to assemble as soon as you have finished printing, or the printer buffer is cleared. PDS will make this clear to you if the situation should ever occur.

Once the assembler is entered a message will be displayed showing current assembly status. This message takes a number of forms depending on the assembler in operation (refer to the assembler manual for more details). When assembly is complete the bell will sound to attract your attention.

At this point, you may be prompted to download your program. If this is the case, then respond accordingly, as described in the assembler manual. If a negative answer is given then control reverts to the editor. If the code is not to be downloaded then, any key depression will return you to the editor. Once in the editor carry on programming normally. During assembly the code can be listed to the screen which will not effect the editor in any way.

If the assembly stops before this point, then an error bell will sound, and an error message will be displayed.

At any point during assembly or downloading, you can abort the assembler by pressing the [Finish] key ([F10]). When assembly is completed, you will return to your original cursor position, unless an error has occurred, in which case you will be positioned at the line containing the error.

### 8.2 Understanding error messages during assembly

If the assembler returns an error during assembly, then the error bell will sound and the relevant error message will be displayed. Look and understand the error message. If you have a problem understanding the assembler error message, then refer to the assembler manual for details. When the message is understood, press any key to return to the editor, you will then be placed on the line the error had taken place.

While debugging you can return to the offending line by pressing [CONTROL] and [Y]. This will move the cursor to the line with the error, and the error message will be displayed in the command window. Note that if lines have been inserted or deleted in the same file in which the error occurred, then the cursor position set by [CONTROL] [Y] may be incorrect.

### 8.3 Using source code from other development systems

Once you have downloaded your source code from your original development system, it is a simple process to insert the code into the PDS editor.

The easiest way to download your code, is to go via your target machine. Load up the source code into the target machine, then upload the code via PDS into an MSDOS file (this done using the 'U >' command in the monitor). Once your code is in MSDOS disk format, you can treat it like any other ascii text.

To get the MSDOS files into the PDS editor, simply read in the files using the [CONTROL] and [R] keystroke combination, as described in section 3.11. Make sure that each disk file is read in, in order, and that the whole program is spread across a number of PDS text files. There is no point in cramming all of the code into one file as this will make it more difficult to handle, about 20k a file should be enough.

Once the code is entered, the first thing you should do is try document each file, with a file description, as shown in chapter 9. Also make sure that line 1 of file 0 contains a description or name of the whole program. This will help protect against accidental deletion overwrite. Prepare a disk using the format utility in PDS (see chapter 5.4), in the required format, then press [F2] to save the program, while you are still in the disk system. Make sure you use the correct save format for that disk (see chapter 5.3).

Once saved, start chopping and changing the code in the manner described in the assembler manual. Each PDS assembler tries to be as compatible as possible, but not every assembler on the market manages to keep to the manufacturers' assembler design specifications, so source codes from that system may require changes. Although most systems are fairly standard, and your code should require very little, if any changes.

If you find you have any major problems in getting original code to run, then call PD Systems, with full details of your problem.



---

## 9 The PDS configuration menu system

---

There are many features in PDS that can be altered, and many defaults that can be redefined. This is done in the configuration menu system. To enter the system, press the [Configure] key from the editor ([F5]). The configuration can be saved on to disk, and will be automatically loaded when the system is run in the future.

### 9.1 The main configuration menu

When the configuration menu system is first entered, the following options will be displayed.

1. The IBM hardware.
2. The PDS editor.
3. The PDS assembler.
4. The PDS disk system.
5. The PDS monitor and interface.
6. The PDS cold start presets.
7. Other areas of the PDS.
8. Saving and loading setup files.
9. Information on PDS.
0. Back to previous menu.

To select any of the options, just press any of the [0] to [9] keys to select that option. You will then enter a further menu, whose options can be selected in exactly the same manner. At any point the [0], [ESC], or [Finish] keys, will return you back to the previous menu, or back to the editor if you were in the main menu. Any re-configurations will only come into effect once you return to the editor (except cold start presets).

### 9.2 The configuration options

When an option is being entered, the current settings are displayed directly before the cursor prompt. If you press [ESC] or [Finish], this will abort the selection, and the option will remain unchanged. You would then return to the menu. Each menu option will now be discussed individually.

#### 9.2.1 The IBM Hardware

The following options are available in this menu option:

##### 1. The sound output

This will then show a list of further options:

1. Keyclick volume.

This allows you to adjust the volume of the keyclick sound that is made whenever a key is pressed. This may be of any value from 0, the loudest, to 15, where the click will no longer be audible. The default value is 12.

#### 2. Bell volume.

When an error occurs in PDS a bell sounds. The bell's volume can be adjusted using this option, from 0, the loudest, to 15, when it's off. The default bell volume is 11.

#### 3. Bell frequency.

The bell's frequency can be adjusted to suit your ears, with this option. The highest frequency being 0, and the lowest, 1023. The default is 60.

#### 4. Bell duration.

The length of time that the bell sounds for, can be set from here. The duration is defined in units of 20 ms, and can vary from 0 to 255 units in length. The default is 15.

## 2. The keyboard

A further menu is then displayed, allowing you to access these options.

#### 1. Key repeat.

When a key is held down, it will normally repeat, this can be de-selected using this option. Where 0 is off, and 1 is on. The default is 1, on.

#### 2. Repeat lead-in delay.

The amount of time a key has to be held down before it repeats, can be defined from this menu option. This may take a value of 0 to 255 units, where each unit is 20 ms. The default is 15.

#### 3. Repeat rate.

Once a key has been held down, and has started to repeat, the interval between repetition can also be defined. This can take a value of 0 to 255, and is set in steps of 20 ms. The default is 2, which is very fast.

#### 4. Auto caps lock control.

One of PDS's unique features, is its auto caps lock facility. When the auto caps lock feature is on, and the [CAPS LOCK] is locked (the LED is lit), every time the cursor falls into a line within a comment, or ascii expression, then caps lock is automatically turned off until the end of the expression, or the comment. If the [CAPS LOCK] key is depressed while auto caps lock is in effect, then auto caps lock is turned off, and all text will be entered in uppercase (auto caps lock can be turned on again if the [CAPS LOCK] key is depressed twice). If the [CAPS LOCK] key is depressed at any other time, then all text will be entered in

lower case. Entering 1 will turn this feature on, or 0 will turn it off and the [CAPS LOCK] key will be used conventionally. The default is 1, on.

#### 5. Different # and ' key enable.

This feature allows you to configure PDS for your keyboard. Entering a value of 0 will configure PDS for a normal 84 key keyboard, while entering 1 or 2 will set PDS up for different types of 102 key keyboard. The default is 0.

### 3. The screen

You will then be able to select any of these options.

#### 1. Paper Colour

The allows you to define the colour used for the background. You may enter 0 to 7. The default is colour 1.

#### 2. Ink Colour

The allows you to define the colour used for the normal text. You may enter 0 to 7. The default is colour 7.

#### 3. Cursor Colour

The allows you to define the colour used for the cursor. You may enter 0 to 7. The default is colour 2.

#### 4. Marker Colour

The allows you to define the colour used for the block markers. You may enter 0 to 7. The default is colour 4.

#### 5. Border Colour

The allows you to define the colour used for the border (Please note that not all graphics adapters allow you to change the border colour). You may enter 0 to 7. The default is colour 1.

### 4. The RS232 port

This will then present you with a new menu, with these options.

#### 1. Transmit baud rate.

This can be set to 1 to 15, where 1 = 50, 2 = 75, 3 = 110, 4 = 135, 5 = 150, 6 = 300, 7 = 600, 8 = 1200, 9 = 1800, 10 = 2400, 11 = 3600, 12 = 4800, 13 = 7200, 14 = 9600, and 15 = 19200. The default is 14, 9600 baud.

#### 2. Receive baud rate.

This can be set to 1 to 15, where 1 = 50, 2 = 75, 3 = 110, 4 = 135, 5 = 150, 6 = 300, 7 = 600, 8 = 1200, 9 = 1800, 10 = 2400, 11 = 3600, 12 = 4800, 13 = 7200, 14 = 9600, and 15 = 19200. The default is 14, 9600 baud.

#### 3. Bits per character.

The number of bits per byte can be set to 5, 6, 7 or 8. The default is 8 bits to one byte.

#### 4. Number of stop bits.

When a byte is transmitted, a number of stop bits can be sent to signal the end of a byte. Either 1, 1.5, or 2 stop bits can be sent, selecting 1, 2, or 3 will choose each of them respectively. The default is 1, one stop bit.

#### 5. Parity type.

At the end of a byte a parity bit can be used to error check the byte. The can be set to 0, no parity, 1, odd parity, or 2, even parity. The default is 0, no parity.

#### 6. Handshaking control protocol to be used.

To ensure that the two computers can keep up with each other, when data is being transmitted, they must perform some sort of handshaking process. You may enter 0, for no protocols, 1, to use XON/XOFF, 2, for RTS/CTS, or 3, for CTS/DCD handshaking. The default is 0, no handshaking.

#### 7. Nulls to be sent after CR.

When a carriage return is sent (character 13), the target device may need a few padding nulls, character 0, to be sent after it. You can send nulls in blocks of 10. From 0 to 255 blocks can be sent after a CR. The default is 0, where no nulls will be sent.

#### 8. Nulls to be sent after FF.

When a form feed character is sent (character 12), the target device may need a few padding nulls, character 0, to be sent after it. You can send nulls in blocks of 10. From 0 to 255 blocks can be sent after a FF. The default is 0, where no nulls will be sent.

### 5. Printer port.

The following printer related options are available.

#### 1. Serial or parallel printer.

A printer may be connected to the serial or parallel ports, 0 will select parallel, or 1 will select serial. The default is 0, parallel.

#### 2. Auto LF after CR.

Some printers require a Line feed, character 10, to be sent after every carriage return, character 13. Entering 0 will not send a LF, while entering a 1 will. The default is 1, to send LF's.

#### 3. When PDS is entered printer control bytes.

When PDS is first entered it can be configured to send certain control bytes to the printer. Some printers may need

this to reset, or initialise them. Up to 15 control bytes can be entered in decimal or ascii characters, all separated by commas. If a form feed is to be sent (character 12), then make it the first character in the list. If all the bytes are zero then nothing is sent to the printer.

4. When PDS is exited control bytes.

When PDS is exited, it can be configured to send certain control bytes to the printer. Some printer may need this to reset them. Up to 15 control bytes can be entered in decimal or ascii characters, all separated by commas. If a form feed is to be sent (character 12), then make it the first character in the list. If all the bytes are zero then nothing is sent to the printer.

5. Before a file is printed.

PDS can send up to 15 control bytes to the printer, just before a file is printed. You may need this to setup certain printer options at the start of a file. If a form feed is to be sent (character 12), then make it the first character in the list. As a default nothing is sent.

6. After a file is printed.

Once a file has finished printing, PDS can be made to send up to 15 control bytes. These are entered in decimal or ascii, and are all separated by commas. As a default a form feed (12) is sent.

7. Special change font or colour control bytes.

When a program is assembled to the printer, PDS will try and display non source code text in a special font or colour. The control bytes to select this special font can be entered from this option.

8. Revert to normal font or colour control bytes.

When a program is assembled to the printer, PDS will try and display non-source code text in a special font or colour, as defined in the previous option. Once normal source code is printed, PDS will send the control bytes from this option to return to the normal font or colour. It is important that this is defined, if a special font is used in the previous option.

9. Skip over end of form.

When printing on continuous paper, you may need PDS to skip over the tear off perforation. Entering 1 will select this option, or 0 will not use the option. If you use this option, make sure that your printer does not try to skip over end of form automatically as well. The default is 1, PDS will skip over end of form. You must setup the form length correctly for this feature to work properly.

## 9.2.2 The PDS editor

The following options are available from this menu option.

### 1. Tab settings.

When the [TAB] key is pressed, the cursor will move to the next defined tab position. This option will allow you to define the 4 tab positions. They will be prompted to enter each tab position separately. PDS will let you type in 'illegal' tabs, ie where tab stop 2 is located in front of tab stop 1, but PDS will then sort them into their correct order, ie lowest first. The default tab positions are 12, 18, 30, and 40.

### 2. Author's name.

This allows you to change the author name for the current file, which is displayed in the file header and disk system. This can be up to 18 characters in length, and is empty on default.

### 3. File name.

Each file may have a name, this name is displayed in the header file, in the disk system, and on printer listings. This option allows you to define the name for the current file, and which can be up to 18 characters in length. The default name is empty.

### 4. ^A label number.

The current label ^A, automatic label generator number, can be set to any 4 digit value. The default is '0000'.

### 5. ^Z label number.

The current label ^Z, automatic label generator number, can be set to any 4 digit value. The default is '0000'.

### 6. Function keys.

This option allows you to define any of the 10 function keys, for the current file. You will be prompted for the key to be re-defined, and you will then have to enter the new definition. The keys contain their key names as the default, ie function key 0 will be set to 'F0'.

### 7. Normal cursor up/down (1) or to character position (0).

When the cursor up and down keys are pressed, the cursor can either be moved to same screen position on the new line, or the same character position on the new line. Entering 0 will select character position movement, while 1 will select the more conventional screen movement option. The default is 1, normal screen up and down.

### 8. Auto indent.

When this option is on, a new line will be automatically indented by the same number of spaces as the previous line, if the [TAB] key is depressed (this is to allow labels to be inserted at the start of the line). Note that it will only indent spaces, not tabs. Entering 1 will set this option on, while 0 will turn it off. The default is zero, off.

### 9. Global options menu

This gives you a menu containing these four options:

1. Function keys

This will set all files to the same function key settings as the current file.

2. Tab settings.

This will setup all the tab positions to the same places, on all files. The files are setup to the current file settings.

3. Authors name.

All files will be set to the same author name as the current file.

4. File name.

All file names will set to the same name as that of the current file.

### 9.2.3 The PDS assembler.

The following options will be available from this menu.

1. Swap < and > in expression.

The assemblers use '<' and '>' characters to signify high or low byte selection, but different assemblers use them differently. Entering 0 will set '>' to mean high byte, and '<' to mean low byte. While entering 1 will set '>' to mean low byte and '<' to mean high byte. The default is 41.

2. Default list options.

There are a number of list options available when a program is assembled to printer. This may take a value 0 to 65535, but defaults to a value of 0.

### 9.2.4 The PDS disk system.

Here are the disk system options.

1. Default format for loading and saving.

PDS can save file in three different formats, PDS, MSDOS, and XMSDOS. You can enter 0 for PDS, 1 for MSDOS, or 2 for XMSDOS, to work as the default format. The normal default is 1, MSDOS.

2. Default MSDOS drive and path.

When MSDOS or XMSDOS files are accessed in PDS, they are accessed from a particular drive and path. Paths are explained in the disk system in great detail, so read it fully to understand this option. The default drive and path is the directory you ran PDS from.

3. Default PDS drive.

All PDS format files are loaded and saved to this drive. Simply enter the drive

letter, with the default being your first floppy drive.

#### **4. Where the help file comes from.**

The help file may come from either a PDS or XMSDOS format file. Enter 0 for PDS or 1 for XMSDOS. If XMSDOS is chosen, then you will also be prompted to enter the help file's file name, including the drive and path specifiers. The default is 1, where the help file comes from the current PDS directory.

#### **5. Verify after write on all operations.**

0 will turn this option on, while 1 will turn it off. When on, all disk saves will be verified to ensure against disk failures. This option will slow down write operations considerably. The default is 0, off.

#### **6. MSDOS file overwrite protection.**

When you save a file (using [CONTROL] & [S], [CONTROL] & [W], or the disk system), and a file already exists with the same name, this option will confirm the save with the user, before continuing. This however, does not apply to the save all function. Enter 1 for on, and 0 for off. The default is 0, off.

#### **7. Default MSDOS file extension.**

The MSDOS file name extension used by PDS (these are the three letters following the full stop after a filename), can be changed. You must enter the new file extension, upto three letters, when you are prompted. The default extension is 'PDS'.

#### **8. Backup file creation.**

When a save all is performed, and this option is on, then any files which are written will be renamed with a 'BAK' extension, ie X0.PDS will become X0.BAK. This means you will always be able to recover the code before the last save all. The default for this option is off, 0.

#### **9. MSDOS save all overwrite protection.**

With this option on, PDS will check that the first line of file 0 in memory and on disk are identical before the save all is initiated. If not, you would then be prompted to confirm the save all. Always make sure that the first line of file 0 contains a program identifier, ie ';This is Space Invaders' or any other distinguishable line. The default is 1, on.

### **9.2.5 The PDS monitor and interface**

Most of these options can also be set using the 'Q' command in the monitor, but they will be initialised to these defaults on power up. The options are as follows.

#### **1. Default interface in monitor.**

The monitor can work with three different types of connections, 0 will use the nibble format via computer1, 1 will use the byte format via computer1, or 2 will use the byte format via computer2. The default is 1, byte communication via computer1.

#### **2. Target computer type.**

The monitor can be setup to work with different target computers, and even different screen modes. The options are, 0 for the Spectrum, 1 for the



Commodore 64, 2 for Amstrad mode 0, 3 for Amstrad model 1, 4 for Amstrad mode 2, 5 for Amstrad in spectrum type mode. The default is 0, the Spectrum, in Z80 PDS or 1 for the Commodore 64, in 6502 PDS.

### 3. Radix for monitor expressions.

The can use base 2 to base 16 for the default radix in the monitor. If you enter 0, then the monitor will only use HEX, ie 'MABCD' can be used instead of 'M\$ABCD' or 'M0ABCDh'. The default radix is 10, decimal.

### 4. Lines in M command.

The 'M' command in the monitor will modify memory, displaying a certain number of lines each time it is used. The 'M' command can display from 1 to 255 lines, or if 0 is entered then 1 screen page is displayed at a time, you must press a key to continue, or cursor up to see the previous page. The default is 8 lines.

### 5. Lines is D command.

The disassembly command, 'D', will disassemble a number of lines. This can be set from 1 to 255, or 0, which will disassemble a page at a time. The default is for 8 line disassembly.

### 6. Disassembler flags.

This will allow you to define the default disassembler flags. This can take a value of 0 to 255, with 0 being the default. See the relevant assembler manual for more details.

### 7. Trace system flags.

This lets you setup the default for the trace system flags. This has a value of 0 to 255, with 0 being the default. See the relevant assembler manual for more details.

### 8. General monitor flags.

The general monitor flags can have the value 0 to 255, where 0 is the default. See the relevant assembler manual for more details.

## 9.2.6 The PDS cold start presets.

All cold start presets can be set, but the settings will only take effect once the setup has been saved, and PDS re-run. The following cold start presets can be changed.

### 1. Double width multi-colour graphics.

This will make the graphics editor display multi-colour sprites twice as wide. This allows you to see a sprite closer to its correct size when it would appear on some target machines. This option can be set on with 1, or off with 0. The default is 0, off.

### 2. When PDS is first entered control bytes.

PDS sends up to 15 control bytes to the printer when it is first run. These have to be separated by commas when they are entered. PDS will send nothing as default.

### 3. When PDS is exited control bytes.

PDS sends up to 15 control bytes to the printer when it is exited. These have to be separated by commas when they are entered. PDS will send nothing as default.

## 8.2.7 Other areas of PDS.

The options are.

### 1. Default printer width.

All printing is carried out to a specific line width, this can be set to 0 to 255, with 80 being the default printer width.

### 2. Default printer page length.

The number of lines that are printed per page, can be set to 0 to 255. This will obviously vary on the type of paper your using, and is needed by the program to format listings properly. The default is 66 lines to a page.

### 3. Tabs or spaces between operands.

Different programmers have different programming styles. Therefore PDS will cater for this, and will insert either spaces, 0, or tab, 1, between the opcode and operand, when it generates code, ie in the Upload command in the monitor or in the graphics editor. The default is 0, spaces.

## 9.2.8 Saving and loading setup files.

Once you have defined all your default options, it is important that you save to disk. When PDS is next run, it will load in the setup file and configure itself. Normally PDS will load the 'SETUP.PDS' setup file from the default directory if it is present, otherwise the normal default settings are used. Other setup files with different extension names, 0 to 9, can also be used. When PDS is run you would follow the PDS file name with a space and the setup number you require. More details about loading setup files can be found in the next section of the manual. These are the setup file options.

### 1. Save this configuration to SETUP.PDS.

This will save the current configuration to the SETUP.PDS. The file extension can be changed using option 3 from this menu.

### 2. Load in the SETUP.PDS file.

This will load the SETUP.PDS file, and use its configuration for PDS. The file extension can be changed using option 3 from this menu.

### 3. Change SETUP file extension.

When setup files are loaded and saved from this menu, they will be saved to a file called 'SETUP', the extension for this file can be defined using this menu option. A three character extension has to entered. The default extension is 'PDS'.

## 9.2.9 Information on the PDS.

This option will display the current version number and update date of the PDS you are running. This should be stated whenever any problems occur. Any recent changes or bug fixes may be mentioned here. There will also be details of how and who to contact in PD Systems, if you have any problems or enquiries.

### 9.3 PDS run time options.

When PDS is run, there are a number of possible options that can be used. If you want to load in one of your setup files, then simply follow the PDS filename with a space and setup file number, 0 to 9. This will search for the specified setup file, and use it if it exists. If no setup file number is specified, then PDS will load in the 'SETUP.PDS' file, if one exists in the current directory. The example below will load in file 'SETUP.3'.

```
PDS 3
```

There is one final PDS run time option, which must only be used in emergencies. If for whatever reason you did not save your source code before exiting PDS, it is possible to re-enter PDS without PDS clearing its memory. This must be done as soon as possible, the longer you leave it, the greater the chance of your source code becoming corrupted. To enter PDS without clearing memory, follow the PDS filename with '/C' or '\C', ie:

```
PDS /C
```

You must remember that because PDS does not clear memory, you find the PDS editor acting strangely when it encounter's any corruption. You should therefore save your source code as soon as possible, then re-run PDS without the '/C' option. If you save to MSDOS, then load from MSDOS this will remove all 'odd' characters and corruption, saving to PDS or XMSDOS would just save the file as it was, even if corrupted. This is a very 'dangerous' command, but we felt that it was important that you could recover code if you forgot to save it; please use it only if nothing else is possible.