PROTEXT

WORD PROCESSOR

MAIL MERGE AND SPELLING CHECKER

Amstrad PCW8256/8512, 9512
CPC6128

Issue 2a, 1988

The PROTEXT program was developed using the MAXAM II assembler and ARNOR BCPL compiler.

This manual was written using PROTEXT and printed from camera-ready copy produced by PROTEXT on a KYOCERA F1010 laser printer.

Manual written by David Foster.

Arnor Ltd., 611 Lincoln Road, Peterborough PE1 3HA

CONTENTS

MAIL MERGING MERGE

UTILITY PROGRAMS UTILITY

INTRODUCTION

1. ABOUT PROTEXT

The CP/M Plus version of PROTEXT is a fully integrated word processing system, comprising word processor, spelling checker, mail merge, disc utilities and file conversion utility, designed as a package which provides a complete environment from which all aspects of word processing and file management can be carried out without the need to load individual programs. Once PROTEXT is loaded, there is no need to leave it until the day's word processing is completed. Documents may be created, the spelling checked, and printed to any printer without leaving PROTEXT.

PROTEXT is a development of four separate programs with a pedigree dating back to the introduction of the Amstrad computers and is specially written to take full advantage of the Amstrad's features, unlike many programs which are designed to run on a variety of machines and as a result suffer from the inevitable compromises imposed by such a requirement.

A high degree of compatibility has been built into this version of PROTEXT and files created on other versions of PROTEXT can be loaded and edited as normal. This compatibility extends also to the controls and commands used, so that anyone used to other versions of PROTEXT will instantly feel at home with it and will only have to learn the new features incorporated into this version.

Protext provides speed and power far in excess of anything else available for the Amstrad computers, or for that matter, most computers, yet at the same time is designed in such a way that it is easy to learn. Once the basic principles have been mastered, with the aid of 'tutorial' files and the comprehensive manual, documents may be easily created and all the extra features which are a part of PROTEXT can be learnt one at a time as they are needed, or when the user feels ready to learn them. Unlike many programs, there is no need to learn it all at once.

2. ABOUT THE MANUAL AND TUTORIAL FILES

PROTEXT is supplied on a single 3" disc which contains all the program files, as well as various tutorial and help files on one side and the SPELL dictionary on the other side. These, together with this manual, provide all that is necessary to use PROTEXT.

IMPORTANT NOTE: With the exception of the first time that PROTEXT is used, to create a working copy of the disc, the original disc should NEVER be used. It must be retained as a back up and kept in a safe place.

If the original is used and is damaged, such as by accidentally formatting it, or even spilling a drink on it, you will not have any back up with which to create another disc.

Being human ourselves, we are only too well aware that manuals are items that come with programs, are intended to be used to prop the computer at a better angle and are only to be read when all else fails! We would however urge users to read the next chapter, which gives full details of how to create a working copy of PROTEXT painlessly and would hope that even experienced word processor users will make the effort to read further.

a). About the manual

The manual is separated into a number of sections, each of which covers one aspect of the PROTEXT word processing system. There is no need initially to read the whole manual. In fact there would be far too much information to absorb at one go. Ideally the sections on 'Getting started' and 'Word processing' should be read thoroughly and the remaining sections can be read when some new feature is to be learnt.

The manual is written in such a way that experienced users of word processors and in particular those familiar with other versions of PROTEXT can make use of the detailed appendices which are provided at the back of the manual and which give full technical information on all PROTEXT commands. These will provide most of the information necessary to get started and only when some feature is not understood, will it be necessary to refer to the relevant section of the manual.

For newcomers to word processing, it is recommended that the reverse process is followed and that the appropriate section of the manual is read and studied and full use is made of the tutorial file.

Throughout the manual it is assumed that the user has little, or no experience of word processing and the main sections have, as far as possible, been written in non technical terms, but inevitably it has been necessary to use some 'jargon' and a 'Glossary of terms' is provided at the back of the manual.

Each section of the manual is split up into smaller chapters, each covering an aspect of the subject and all commands and procedures are covered in detail, with examples where necessary.

The section on mail merging, in particular, should be read by anyone who needs to use mail merging, but also contains much of interest to everyone, as inevitably there is a degree of overlap between one section and another. The mail merging section also provides numerous examples, which start at the beginning with very simple examples and expands on these.

b). Page numbering

The manual is broken down into sections and each section is referred to by name. Each section is further broken down into chapters, which are numbered and each chapter has its pages numbered, starting at page 1.

When consulting the 'Contents pages', it will be noticed that a comprehensive list of all the separate parts of each chapter is also listed, with the page numbers on the right.

Every page of the manual is numbered on the outside top corner of the page and consists of a name and numbers. For example, this page is numbered as follows:

INTRO 2-2

This means that the section of the manual is the 'Introduction' and this is page 2 of chapter 2. The chapter is the first number and the actual page in the chapter is the number following the hyphen.

c). Version numbers, updates and README

Every Arnor program has a version number. The main PROTEXT version number is displayed on the command mode banner line. The individual utility programs display version numbers when they are used (usually at the top of the screen). Any queries regarding the software should be accompanied by the relevant version number, and your PROTEXT registration number must always be quoted.

Arnor have a policy of continual enhancement and improvement of software, and from time to time new versions of PROTEXT will be made available. Existing users qualify for a low cost upgrade to any new version, but only if the registration card has been returned to Arnor.

Inevitably the printed documentation cannot keep up with the changes to the software, and so a text file is supplied on the PROTEXT disc, called 'README' giving any updates to the documentation. This file should be loaded into PROTEXT to be read and printed.

d). Conventions used

PROTEXT uses a 'standard set' of commands which are common to all versions of PROTEXT as far as possible and these are used wherever possible to describe the function. In addition, all the 'special' keys on the PCW have been configured to function according to their normal uses. It is recommended that the PROTEXT standard controls are learnt, as it will then be possible to use any version of PROTEXT with little difficulty.

Throughout the manual a consistent form is used to describe the various commands, as follows:

(i) Edit mode commands

ALT-F - means the key marked 'ALT' and the key marked 'F'. Wherever a hyphen is used between them, it means that the first key should be held down whilst the second key is pressed. Most of the editing commands take this form.

ALT-V T - means that the 'ALT' and 'V' keys should be used as described above, then released and the 'T' key pressed. Note that there is no hyphen between the 'V' and the 'T'.

ALT-SHIFT-H - means that all three keys should be pressed at the same time. This sort of command that requires more than two keys to be pressed at a time is rarely used and at least two of the keys are always adjacent to each other.

ALT-(- means that the 'ALT' key and the key which has the '(' on it are pressed together. It does NOT mean that SHIFT is required as well. The '(' is merely being used for ease of remembering its function.

(ii) Command mode commands

Command mode commands are always shown in upper case, though when they are being entered into the computer, they may equally well be entered in upper or lower case. Similarly, when entering filenames to LOAD or SAVE a file, even though they may be shown in upper case, lower case is acceptable and PROTEXT will automatically convert them to upper case if required.

(iii) Stored commands

Stored commands are always shown with their two letter name in upper case, but again, upper or lower case is acceptable.

e). Key variations for the CPC6128

There are a number of differences between the keyboards of the PCW and the CPC computers. Throughout the manual the PCW key names are the ones which are used, rather than cause confusion by listing both keys on all occasions.

The following keys are direct equivalents:

PCW key	CPC key
STOP	ESC
EXIT	ESC
ALT	CONTROL
DEL→	CLR
←DEL	DEL

There is no direct equivalent to the EXTRA key, but for most purposes, CTRL-0 (zero) serves the same purpose.

Any other variations where, for example, there is no directly equivalent key on the CPC6128 are noted at the appropriate point in the manual.

The various appendices give full details of the commands for both the CPC6128 and the PCW8256/8512.

f). About the 'Tutorial files'

There are a number of 'tutorial files' for PROTEXT, available on request from Arnor. These have been made available separately due to lack of disc space, but a sample file 'TUTOR', covering basic word processing is included with PROTEXT. We recommend that newcomers to word processing read the next few chapters and then, following the instructions provided, load the tutorial file and follow it through.

Only editing commands are covered by the tutorials, but this is the basis of all word processing and by the time the user is conversant with the various commands which permit rapid and easy movement throughout the text, the remainder of the manual will make a lot more sense and progress will then be rapid.

g). The Tutorial disc

The tutorial disc contains several tutorial files covering most aspects of editing in a different way from the manual, and also includes a selection of sample mail merge documents and template files. It is available from Arnor for a small charge.

3. GETTING STARTED

This chapter explains how to create a 'start of day' disc from the supplied master disc and should now be read and the instructions followed carefully.
PROTEXT is supplied on one 3" disc, with the program and other files on one side and the master dictionary on the other side.

WARNING: All the programs, files and the dictionary provided on the master disc are subject to copyright laws and copies may be made of these files and the dictionary, for your own use on one machine only. It is an offence to give, hire or sell copies of copyrighted material to other parties.

PROTEXT operates under CP/M Plus and is suitable for use on the Amstrad PCW8256, PCW8512 and CPC6128 computers. It may be used with single drive computers, but will take full advantage of two or more drives.
Before PROTEXT is used, 'working' copies of the relevant files MUST be made, together with at least one copy of the dictionary. Make sure that the disc is 'write protected' before starting to create a 'start of day' disc. If there is any doubt about the use of the write protection tab, full details are given on the disc inlay card and also in the Amstrad computer manual.

A 'start of day' disc is one which, when created by following the instructions in this chapter, will contain all the necessary files from the CP/M Plus System Disc and the supplied PROTEXT disc. It will enable the computer to automatically load CP/M and PROTEXT when the disc is inserted at the start of the day, without the need to insert different discs. CPC6128 users should note that they will still have to type '¦CPM' and press RETURN after switching on.

At least two blank discs will be required, one for the dictionary and a second one for the program working disc.

a). Formatting a disc

The first thing that must be done is to format one of the new discs. There is no need to format the second disc.
Note: CPC6128 users must format both sides of the disc, but PCW owners need only format side A.
The computer should be reset by switching it off and then on again, to ensure that it is completely reset.

(i) PCW8256/8512 users

The reverse side of the original LOCOSCRIPT disc will be required. This disc is marked 'CP/M PLUS SYSTEM/UTILITIES/BASIC'. Insert this disc in drive A with side 2 to the left. The usual lines will appear on the screen as CP/M is loading. Eventually the following (or something similar) should appear:

```
CP/M Plus  Amstrad Consumer Electronics plc
v 1.4, 61K TPA, 2 disc drives, 368K drive M:
```

Type **DISCKIT** and press RETURN. The disc should be removed when requested and the on-screen instructions followed, to format a blank CF2 disc in drive A. When this is complete, exit the program as directed.

(ii) CPC6128 users

Take the CP/M system disc supplied with the computer. It is marked 'CP/M PLUS SYSTEM/UTILITIES'. Insert this disc in drive A.
Type ¦CPM
CP/M will load and eventually the following (or something similar) should appear:

```
CP/M Plus  Amstrad Consumer Electronics plc
v 1.0, 61K TPA, 1 disc drive
```

Type **DISCKIT3** and press RETURN. The disc should be removed when requested and the on-screen instructions followed, to format a blank disc as a system disc. When this is complete, the 'Y' option should be selected and Side 2 of the same disc should also be formatted before leaving the program as directed.

b). Creating a 'Start of day' disc

Note: The newly formatted disc should now be clearly marked

'PROTEXT - Start of day disc'

The other side should be marked:

'PROTEXT - Utilities'

Three discs will be needed to create the working disc.
(i) The CP/M SYSTEM/UTILITIES disc which was used to format the new disc.
(ii) The newly formatted disc (labelled 'PROTEXT - Start of day disc')
(iii) The master disc supplied with PROTEXT.

Check that the master disc is write protected before going any further. Side 1 of the master disc, marked 'PROGRAM' should now be inserted into drive A. On the PCW this means that the side marked 'PROGRAM' should be facing towards the centre of the computer. For CPC users, the 'PROGRAM' side should be facing upwards.

PCW users should type the following:

PROTEXT <MAKEPCW and press the RETURN key.

CPC users should type:

PROTEXT <MAKECPC and press the RETURN key.

Note: The '<' symbol is the 'smaller than' symbol, situated to the left of the RETURN key on the PCW and on the same key as the comma on the CPC6128 and is obtained in both cases by pressing SHIFT and the '<' key.

Note: CPC6128 owners with only a single disc drive will find that the MAKECPC file requires a large number of disc changes. In this case a number of possible alternatives may be used to reduce the number required. If the user has access to a CPC6128 with two drives, this can be used to create the start of day disc. Alternatively, if a PCW8256 or PCW8512 is available, a start of day disc can be created by following the instructions for the PCW, but using a CPC system formatted disc created with DISCKIT3 on the CPC, rather than one created on the PCW. A third alternative is to make use of one of the utility programs which are available for the CPC computers and which permit the easy transfer of files, to transfer the required files to the new start of day disc. Using the TYPE command from CP/M command mode will enable the file called MAKECPC to be viewed in order to see which files should be copied onto which side of the start of day disc.

The program will now start loading and messages will appear on the screen informing that certain files are being copied. These messages should be watched closely and when instructions are given to change discs, the appropriate disc should be inserted. Several requests to change discs may be received. The precise messages will vary depending on the machine being used, but after a short while activity will cease and PROTEXT will return to command mode and a flashing cursor. Note that some of the files will be copied onto side 2 of this disc because they will not all fit on side 1.

All the necessary PROTEXT files have now been copied onto the new disc. This disc must not be write protected, because it would not then operate correctly as a 'start of day' disc.

All that now remains is to make a copy of the dictionary disc.

Note: As soon as a degree of familiarity with the use of PROTEXT has been gained it is important to ensure that the default settings of a number of the PROTEXT features are correctly configured for the computer, disc drive and printer on which PROTEXT is being used. This is done with the CONFIG and SETPRINT utility programs (See Utility section of the manual for full details).

c). Copying the dictionary disc

Now that a working copy of PROTEXT has been created, PROTEXT can be used to copy the master dictionary onto the second new disc and the following procedure should be followed.

The new and unformatted disc should now be clearly marked:

PROTEXT - Dictionary disc

Just to prove that a start of day copy of PROTEXT has been created, any discs should be removed from the computer and the computer reset once more, by switching it off and on again.

The 'start of day' disc should now be inserted into drive A. PCW users will find that the program will automatically start to load and the usual lines will appear. CPC users must type **¦CPM** and press the RETURN key.

After a brief spell of activity, the screen will clear and be replaced by PROTEXT's normal screen. This is described in full in later chapters, but for the moment it is only necessary to know about the command mode prompt.

About two thirds of the way down the screen is a broad line in inverse video, containing certain messages. Immediately beneath this will be seen the PROTEXT command mode prompt, 'a>', followed by the cursor. Whenever this prompt and the cursor are visible, PROTEXT is in command mode and waiting for a command to be entered.

Type:

DCOPY and press the RETURN key.

After a further brief period of disc activity, the following message (or something similar) will appear:

```
ARNOR Disc Copier  v1.00
```

followed by instructions about which disc to insert into which drive. The precise message displayed will vary depending on the machine and the number of drives installed. Finally the message ends with:

```
Press S to start, C to cancel:
```

In this case the 'original disc' is the side of the supplied PROTEXT disc marked DICTIONARY (check that it is write protected!) and the 'copy' disc is the new disc onto which the dictionary is to be copied. At this point the appropriate disc(s) should be inserted, as instructed. Once certain that the discs are inserted correctly, the 'S' key should be pressed and copying will commence.

PROTEXT will indicate the number of the tracks that it is 'Reading' and 'Writing'. Depending on whether one or two drives are installed, instructions to change the disc may be issued as copying proceeds. With a two drive machine, the complete process will be carried out automatically.

When the copying process is complete, the message:

```
Copy another disc (y/n)?
```

will appear. In this instance, 'N' should be selected and a further message will appear, requesting that the PROTEXT disc is inserted. The PROTEXT disc referred to is the new start of day copy that has been created, not the original master disc or the master dictionary disc. A return will now be made to PROTEXT command mode and the 'a>' prompt.

The copy of the dictionary disc has now been created and at this point the original master disc should be put away in a safe place. The only time it may be required again will be if the working disc gets damaged and a new working disc has to be made, or to use the tutorial file.

Everything has now been done that is necessary to use PROTEXT, but there are a number of things which can and should be done at an early stage to configure PROTEXT to suit the printer that is to be used and to make the best use of the available drives. At some point it is worth giving consideration to the sort of work that PROTEXT will be expected to do for most of the time.

d). Configuration of PROTEXT

Whilst PROTEXT may now be used, it is extremely important to carry out a procedure to configure it, as soon as possible, to suit the precise arrangement of the computer with which it is being used. It is not essential to do this now, but it should be done before any serious work is carried out with PROTEXT.

In order to function as intended, it is important that PROTEXT should know the type of printer that is being used and the number of drives available.

It is recommended that the remainder of this chapter is read through and perhaps a small amount of experimentation with the use of PROTEXT carried out before returning to this chapter and carrying out the required configuration process as described below.

Note: When a configuration file (PROTEXT.CFG) or a new printer driver, has been created, it will be saved onto the 'Utilities' side of the start of day disc. PROTEXT.CFG and/or the new printer driver MUST be copied onto the 'Start of day' side of the disc. This is done with the COPY command and full details of how to use COPY are given in Word 4.

PROTEXT is supplied with two utility programs, CONFIG and SETPRINT, which are used to set many features to suit the individual preferences of the user.

(i). Configuring the printer with SETPRINT

PROTEXT requires a 'printer driver' if it is to take full advantage of the printer with which it is being used. A printer driver contains, amongst other details, all the information concerning the codes required by a specific printer and PROTEXT is supplied with three printer drivers:-

PCW.PTR - To suit the standard PCW printer.
EPSON.PTR - To suit any 'Epson' compatible printer with parallel (Centronics) interface.
JUKI6100.PTR - Specifically for the Juki 6100 daisywheel printer, but also suitable for use with a number of other printers with parallel (Centronics) interfaces.

If the printer to be used is covered by one of the above printer drivers, then no changes are required, except to set the 'default printer driver' in CONFIG (See below). If, however the printer is not one covered by the above list, then it will be necessary to create a printer driver using the SETPRINT utility program and the relevant chapter in the Utilities section of the manual should be consulted for full details of how to create a new driver.

(ii). Configuring the disc drives with CONFIG

PROTEXT makes use of a number of temporary files when text is being edited and it is necessary to specify which drive is to be used as the drive to hold temporary files. On the PCW computers this would normally be drive M, whilst on a CPC6128 with twin drives, it would usually be drive B. CPC6128 users with only a single drive have no alternative but to specify drive A. Additionally, it is possible to specify which drive is to be the default drive initially selected for the storage of text files. On the PCW 8512, this would usually be drive B, on the 8256, drive A and on the CPC with twin drives, drive B.

These two parameters may be set by using CONFIG. Full details are given in the Utilities section of the manual on how to use this program, but for the time being all that is required is to insert the 'Utilities' side of the start of day disc and type the word:-

CONFIG   and press the RETURN key.

After a second or two, the screen will clear and a menu will appear. There are many options, as most features of PROTEXT can be reconfigured, but for now it is only necessary to select option 1 to 'Set editing options'. The menu will be replaced with a list of options and the options, 'Default drive for text' and 'Temporary text drive' are the ones which require setting to the chosen drives. The ↑ and ↓ cursor keys should be used to move to the required option and then the appropriate letter typed in. When all changes have been made, the STOP key should be pressed, which will then recall the original menu.

Option 7, 'Set printer driver options' should then be selected from the menu and for the time being only the name of the required printer driver should be entered on the appropriate line. The name should be the name of one of the three supplied printer drivers, or the name of the new printer driver that it may have been necessary to create.

Option 9 should be selected and the configuration file saved by following the instructions. This will create a file called PROTEXT.CFG on the disc. CONFIG may now be exited by selecting the 0 option. The newly created PROTEXT.CFG file should then be copied onto the 'Start of day' side of the disc.

CONFIG may be used again at any later date to reconfigure other PROTEXT options if required.

e). The STARTUP file

PROTEXT is supplied with a file called 'STARTUP' which will have been copied onto the new start of day disc when it was created. This is a special file which is used by PROTEXT when it is first loaded, to carry out certain procedures, such as copying the 'Help' files onto drive M. STARTUP may be loaded into PROTEXT as a normal text file for editing if necessary.

There are a number of possible combinations of computer and disc drives with which PROTEXT can be used, starting with the CPC6128 with only a single drive and progressing to the PCW8512 with two drives and memory disc. These variations mean a great difference in the total disc capacity and memory available and depending on the prime use to which PROTEXT will be put, there may be advantages to copying certain of the files onto other drives when PROTEXT is first loaded (this is particularly true of the M drive on the PCW computers).

The STARTUP file may be loaded into PROTEXT for viewing and altering. Having made sure that the start of day disc is in drive A, type in the following at the PROTEXT command mode prompt:

LOAD STARTUP and press the RETURN key.

After a second or two the contents of the STARTUP file will appear in the upper part of the screen. STARTUP is a special type of file, called an EXEC file. For the time being, there is no need to understand exactly what an EXEC file is, merely that it is a file which is used by PROTEXT to carry out certain commands (full details of EXEC files are covered later in the manual).

Most of the lines will be seen to start with a semi-colon (;), which in an EXEC file PROTEXT understands to mean that the remainder of the line is merely a comment and is not to be treated as a command. These lines of text can be made into commands simply by deleting the semi-colon at the start of the line and depending on the decisions made, certain of the lines may be changed into commands, or new command lines added.

The following should be treated only as suggestions as to what may be considered the most suitable arrangement for the various combinations of computer and drives and users will soon find out what arrangement suits them best. At any time the STARTUP file may be reloaded and altered and then resaved.

f). STARTUP on the PCW8256 and PCW8512

(i) It is recommended that the help files are always copied onto the M drive.
(ii) If an unexpanded PCW8256 is being used and it is intended to only work with relatively small files, then it is suggested that PROTEXT.COM, PROTEXT.CFG, SPELL.COM, and the required printer driver are also copied onto the M drive. If large files are envisaged, then only the .HLP files should be copied and when the other files on the disc are required, then the start of day disc should be inserted into drive A.
(iii) With the PCW8512 or a PCW8256 with memory disc expansion, it is recommended that all the .COM files as well as the .HLP files and PROTEXT.CFG are copied onto drive M. It may also be convenient to copy any phrase files, template files and alternative printer drivers, onto drive M.

g). STARTUP on the CPC6128

As the CPC6128 does not contain a memory drive, it is not possible to use the STARTUP file in the same way.

It is recommended that with only a single drive, the .HLP files should be copied onto all discs which will be used for storage of text files, so that help is available at all times. Once the user is fully conversant with all the PROTEXT commands, then the .HLP files may be omitted, or removed, if desired.

With two drives, it is most convenient to leave the start of day disc in drive A for as much of the time as possible and to use drive B for the storage of text files.

h). Altering the STARTUP file

Once the STARTUP file has been loaded in the way described above, it is necessary to go into 'Edit' mode to remove the semi-colons. This is done by pressing the STOP key. The command mode line in the lower third of the screen will disappear and a flashing cursor will appear in the top left corner of the screen. The cursor may be moved down to the line which contains the semi-colon to be removed by pressing the down cursor key. This is the key with the downwards pointing arrow, to the right of the main keyboard. Once the cursor is positioned over the semi-colon it can be removed by pressing the key marked DEL→ once. If required, the cursor may be moved to other semi-colons, to delete them. When all the changes have been made, the STOP key should be pressed again and the PROTEXT will return to command mode.

The altered file can now be saved by typing SAVE and then pressing the RETURN key twice. Once the command mode prompt returns, the altered version of STARTUP will have been saved onto the start of day disc.

WORD PROCESSING

1. INTRODUCTION

a). What is 'Word processing'?

In its simplest form word processing is using a computer as a very sophisticated typewriter. Instead of typing a letter straight onto paper, the letter is displayed on the screen as it is typed.

Every word processor then allows you to:

(a) correct or change any part of the text.
(b) store the text on disc for later use.
(c) send the text to a printer.

An ordinary typewriter effectively only allows (c) because text cannot be changed or stored.
The advantages of even this simple form of word processing are enormous:

- it allows easy correction of mistakes, either immediately or at any later time.
- a number of similar documents can be produced with minimum effort.
- any number of copies can be printed without retyping the document.
- a disc can be given to someone else to use or add to.

Word processing offers far more than this, and PROTEXT provides many powerful features - yet all are designed to be easy to use. This manual introduces all these features gradually, so there is no need to read the entire manual immediately. This section of the manual restricts itself to the word processing aspects of PROTEXT, leaving later sections to describe the spelling checker, mail merging and utilities.

After a short description of the way to make use of the tutor file, the remainder of this introductory chapter provides an extremely brief description of the way in which PROTEXT is used, for the benefit of users who are impatient to have a try and feel that they have sufficient experience of other word processors to be able to cope without reading the remainder of this section.
It should also prove of interest to everyone, as it provides an insight into the principles of operation, rather than specific details of individual functions.

b). How to proceed?

At this point there are a number of ways in which the user can proceed, depending on the amount of experience already gained in the use of word processors.

Beginners to word processing are advised to read through this and the following chapters and to make use of the tutorial files, in order to make the quickest progress.

Experienced users may feel that they can manage without reading through the whole manual and for their benefit the appendices give full details of all the command mode and editing commands. In spite of this, PROTEXT has so many special features that many of them may never be discovered if the manual is not read through.

Finally, for the benefit of those people who either cannot bear to read manuals, or feel confident of being able to cope, the last part of this chapter gives a very brief description of the way PROTEXT is used.

c). How to use the Tutorial files

Once PROTEXT has been loaded and is in command mode, the 'PROGRAM' side of the master PROTEXT disc should be inserted into drive A (make sure the disc is write protected) and all that is required to start on the tutorial files is to type the following:

LOAD TUTOR and press the RETURN key.

After a few seconds, during which PROTEXT will load the tutor file, the disc drive will stop and the start of the tutorial file will appear in the upper part of the screen. From this point on, all that is required is to follow the instructions in the tutorial, which will then go through the most commonly used commands one at a time. Once the tutorial file has been used, the master disc should once more be returned to a safe place.

A disc containing additional tutorial files is available separately from Arnor. To start type:

LOAD TUTOR1 and press the RETURN key.

When the end of one tutor is reached, instructions on how to load the next are given.

Note: The tutors are not intended to be an exhaustive description of all of the functions available in PROTEXT and many of the lesser used commands are only covered in the manual, which the tutors are intended to be used in conjunction with. The object of them is to provide an easy and visual description of the most commonly used commands, together with the opportunity to try them out.

d). Can't wait?

The following description may be sufficient to enable experienced users to manage simple word processing, with only the aid of the technical appendices, but for other users it will provide a useful outline of the way in which PROTEXT is used and controlled.

A brief description of the typical procedures which might be followed to create and print a letter is given in an appendix at the back of the manual.

PROTEXT has two distinct modes of operation, 'Edit mode' and 'Command mode'. Edit mode is the mode used to create the text, whilst command mode is where all the saving, loading and printing is carried out. The STOP key is used to switch between Edit and Command mode whenever required.

(i) Edit mode

PROTEXT is a 'command driven' word processor, unlike some other programs, which are 'menu driven'. This means that the editing processes are carried out by the pressing of certain key combinations, unlike a menu driven program, where menus are selected and then further selections made from further menus. There are a considerable number of advantages to this, mostly in terms of speed and convenience of use. Whilst the idea of selecting options from a menu may seem a good idea and initially may be found convenient, once experience has been gained, menus tend to become irritating and time wasting. When help is required with PROTEXT, it is available through the use of the help command, ALT-H (that is, pressing the key marked 'H', whilst holding down the key marked ALT), which provides a panel of information at the bottom of the screen.

PROTEXT's editing commands are called by using certain keys on the keyboard, normally in conjunction with the ALT key, but also the SHIFT and EXTRA keys. They have been carefully chosen so that wherever possible, there is an association between the task to be performed and the key used to carry it out. For example: ALT-F to Format a piece of text - ALT-J to turn Justification on and off.

Certain commands, such as the ones to move forward or backwards a page, or a paragraph, or to the start or end of a document have been allocated the various types of opening and closing brackets, so that they are easily remembered. Opening brackets mean a move towards the start of the document, whilst closing brackets cause movement towards the end.

Full advantage is made of the cursor keys and when used in conjunction with SHIFT, or ALT, the effect becomes increasingly greater. For example: Using the right cursor key on its own will move the cursor one character at a time. Using it with the SHIFT key will move a word at a time, whilst with ALT, it will move to the end of the line.

Similarly, the commands to delete make use of the two DEL keys, which on their own will delete one character, but when used with SHIFT will delete a word and with ALT will delete to the beginning or end of the line.

Certain features, which determine the way the text will be printed are initiated by the use of 'Stored commands'. The number of lines to a page, where a new page break should be made, the width of the text and many other features can all be controlled with stored commands.

A stored command is an instruction which is inserted into the text and which is acted on when printing takes place, rather than being printed. Stored commands take the form of a 'greater than' symbol (>) in column 1 of the text, followed by two characters which define the command. For example: '>PA' will cause a new PAge to be taken and '>PL' will define the Page Length

Additionally, features of the printer, such as the style of printing, whether bold or Near Letter Quality, are controlled by the use of printer control codes. These are single characters which are inserted into the text and which PROTEXT will recognise when printing and send to the printer as the codes that the printer requires to initiate the function required. Printer control codes are inserted by pressing ALT-X and then a letter key. Printer control codes appear on the screen as the letter in inverse video. Again, the default settings chosen have been selected for ease of use. For example: 'i' is used to initiate italic printing and 'b' for bold printing.

(ii) Command mode

Command mode can be recognised by a broad inverse band about two thirds of the way down the screen and immediately beneath this is the command mode prompt 'a>', followed by the block cursor. Whenever the prompt and the cursor are visible, PROTEXT is waiting for a command. Commands are words which are typed in as instructions and may be followed by one or more parameters, depending on the command.

There are a considerable number of commands, which are covered in detail in the 'Command mode' chapter, but a few of the most commonly used ones will be mentioned here. Commands and parameters may all be entered on the same line, with the parameters being separated from the command by a space. If more than one parameter is specified, they may be separated either by a space, an equals sign (=) or a comma (,).

Any commands which NEED further parameters can be used just by typing the command and PROTEXT will prompt for the parameters. There are, however a number of commands which, optionally, may have parameters specified, in which case the parameter should follow the command, separated by a space.

Once the command has been specified, the RETURN key should be pressed and the command will be carried out and when completed, the command mode prompt will return.

Drives may be changed by just typing the drive letter, or alternatively, the DRIVE command may be used (DRIVE B). When a drive is changed, the command mode prompt letter will change to suit the new drive. Drives may be catalogued at any time by entering CAT, optionally with the drive letter as a parameter (CAT B will catalogue the contents of drive B).

Documents may be saved and loaded by use of the SAVE and LOAD commands, with the name specified as the parameter (see Word 4, 'command mode' for details of simple methods and current filenames). One file may be merged into another by using the MERGE command.

All printing is carried out from command mode and at its simplest consists of nothing more than typing PRINT, or its abbreviation 'P', which will have the effect of sending the text in memory to the printer. There are however a number of printing commands which enable documents or parts of documents to be printed, as well as documents to be printed directly from a file on disc. With most of the printing options it is possible to specify the number of copies to be printed as a parameter. Printing may also be done to the screen with the command PRINTS, or to a disc file with the PRINTF command. The chapter on Command mode and the part that covers printing commands is one that should be considered as 'essential reading', as it gives full details of the many ways that documents can be printed, which is really the main object of word processing.

One very important command is the command to clear a file from memory if it is no longer required. The command is CLEAR and is typed from command mode. If the document in memory has had any changes made to it since it was loaded and it has not been saved, a prompt, requesting confirmation that it should be cleared, is issued first. If a new document is loaded into memory, then any existing text will automatically be erased first.

In addition to the commands described above, there is a range of commands which allow 'housekeeping' of disc files, such as renaming, moving and copying of files, as well as formatting and copying of discs and these are all covered in detail in the relevant chapters.

WORD 1-6

2. EDIT MODE

Throughout this part of the manual, the standard 'PROTEXT' command keys are used to describe the commands. PROTEXT is available for use on a range of computers and all versions use the same commands as far as possible.

It is recommended that the standard PROTEXT commands are learnt, as versions of PROTEXT on other makes of computer will then be usable with little or no re-learning being required.

The PCW computers have a number of 'special' keys on the right hand side of the keyboard and these have been configured to correspond to their originally intended uses as far as possible, which in most cases is merely a duplication of the equivalent PROTEXT command. It should, however, be noted that due to the different methods used by PROTEXT for moving, copying and deleting blocks of text, there is some variation in the way that the CUT, CAN and COPY keys are used.

A complete summary of both types of command is given in an appendix.

Note: The 'NUM LOCK' facility provided with the PCW computer, obtained by pressing ALT-RELAY, is not available with PROTEXT. It has been disabled, as accidental use of the option, which changes the use of the cursor and other keys into numeric keys, can be very confusing, as the cursor keys would no longer function as expected.

a). Editing

Once PROTEXT has been loaded, three lines containing information about the state of the program will be visible at the top of the screen. These are the 'Status lines', the contents of which will be explained later. There is also a thin horizontal line, which always marks the end of the text and about two thirds of the way down the screen is another, broader, line containing further information.

At this stage the program is still in Command Mode, which is described in detail in the next chapter, but pressing the STOP key will put the program into Edit Mode, which is the mode used for all entry and correction of text. The line two thirds of the way down the screen will disappear, leaving the lower part of the screen clear. Pressing the STOP key at any time will return to command mode.

(I) On screen help

There are two distinct types of 'on screen help' available. One is for help whilst in edit mode and the other for help in command mode. Full details on the Command mode help are given in the chapter on 'Command mode'.

At the top of the screen, on the status line, will be found the message 'ALT-H for Help'. Pressing ALT-H at any time whilst in edit mode will turn on a help panel at the bottom of the screen. If required this may be left permanently on, but once some degree of familiarity with PROTEXT has been attained, it would normally only be turned on when information about a particular command was required.

The edit mode help is 'context sensitive', which means that the help provided will depend on the text that is being edited when ALT-H is pressed. For example, if the cursor is on a 'ruler line' at the time help is called, information on ruler lines will be displayed. Similarly if the line is a 'stored command' line, information on stored commands will be provided. If the line is a straightforward text line, then the start of help will be displayed. Pressing ALT-H subsequently will display further panels of help information. When the last panel is reached, the display will repeat from the start again. Pressing ALT-V B will move backwards through the help panels.

If help is turned on and ALT-X is pressed, in preparation for entering a printer control code, a list of the control codes and their functions will be displayed.

Pressing ALT-V H will turn the help panel off at any time and restore editing to the full screen.

Note: In order for help to be available, the files called EDIT.HLP and COMMAND.HLP must be available on one or other of the drives. In the case of the PCW, they would normally be present on drive M, having automatically been transferred to drive M as part of the 'Start up' procedure (See 'Getting started'). Other users would normally need to have copies of these files on their text disc, or the start of day disc in one of the drives. Suggestions for the best arrangement for the various combinations of drive and computer are given in the chapter called 'Getting started'.

(ii) Entering text

Once in edit mode a flashing cursor is positioned beneath the status lines and anything that is typed at the keyboard will appear on the screen at this position and the cursor will be moved forward one position.

As practice, a section of text should be entered by copying a paragraph out of this manual. There is no need to press RETURN, the words simply being typed in, one after the other. At the end of a line the cursor will automatically move onto the next line. When the end of a paragraph is reached, the RETURN key should be pressed.

Any mistakes made whilst typing, which are noticed at the time, may be corrected by pressing the <-DEL key, which will cancel the last key pressed.

The cursor can be moved around the screen by pressing the four cursor keys (those with arrows on them). By using these keys, text may be entered at any position. The cursor moves one line or column for each press of a cursor key. Holding a cursor key down will make the cursor move continuously; release the key and the cursor will stop.

The cursor cannot be moved past the end of text (the thin horizontal line on the screen). To position the cursor further down, the end of text must be moved down by positioning the cursor at the end of the text and pressing RETURN as many times as required.

(iii) Upper and lower case

Initially the letter keys produce lower case letters, unless SHIFT is pressed at the same time. If SHIFT LOCK, or CAPS LOCK, is pressed, upper case letters are always produced, and this is indicated on the status line.

Note: The Amstrad PCW computers are slightly unusual in having a SHIFT LOCK and no CAPS LOCK key. When SHIFT LOCK is on, all the characters on the upper part of those keys which have more than one character on them will also be selected. Caps lock is selected by pressing ALT-ENTER.

PROTEXT has two commands which change the case of a letter. To make a letter upper case, press ALT-/ when the cursor is on the letter. This command only affects letters, so the cursor can be moved quickly over a line to convert all letters to upper case by holding down ALT-/. Similarly, ALT-. (point) will convert upper case letters into lower case. (CPC6128 equivalent: ALT-\).

(iv) Deleting and inserting

The ability to move the cursor around, permits the correction or alteration of text anywhere on the screen. The cursor should be positioned on the letter to be changed and the DEL→ key pressed. This will remove the letter at the cursor position, and move the rest of the line to the left. As many letters as required can be deleted in this way. If the new letter is now entered it will appear on the screen and the rest of the line will move back to the right. Alternatively, pressing ←DEL will remove the character to the left of the cursor and the text will again move to the left to fill the gap. Repeated pressing of either DEL key will cause further characters to be deleted.

If extra text is to be inserted, the cursor should be positioned where the first new character is to be added and the new text entered.

To insert a new blank line into the text, ALT-I should be used. The cursor will remain where it is and all text from the current line to the end of the document will be moved down a line.

Just as a character can be deleted, so can a word. Pressing SHIFT and DEL→ when the cursor is at the start of a word will make the word disappear. If this is done when the cursor is in the middle of a word, only that part of the word at and to the right of the cursor position will be deleted.

Similarly, pressing SHIFT and ←DEL will remove the word to the left of the cursor, or if positioned in the middle of a word, the characters to the start of the word.

ALT-←DEL will delete all text from the character on the left of the cursor to the start of the line and ALT-DEL→ will delete all text from the cursor to the end of the line. ALT-E also deletes everything from the cursor position to the end of the line. (CPC6128 users: only CTRL-E is available).

ALT-CAN will delete the whole line. The line is removed from the document and the remainder of the text moved up a line. (CPC6128 equivalent: CTRL-CLR).

Note: Pressing ALT-←DEL followed by ALT-DEL→ will delete all the text from a line, but will not remove the empty line from the text, unlike ALT-CAN, which will remove the blank line as well.

(v) Swapping two characters

A common typing mistake, especially when typing quickly, is to type two letters the wrong way round, e.g. 'wrod' instead of 'word'. The ALT-A (Alternate characters) command will put this right. The cursor should be positioned on the first of the two offending characters (on the 'r', in the above example) and ALT-A pressed. The two characters will then be exchanged.

(vi) Un-deleting all or part of a line

PROTEXT maintains a buffer which always contains the most recently deleted section of text. If a line or part of a line, more than three characters long, is deleted, the deleted text will be saved in the buffer. If a section of text has been accidentally deleted, it may be restored by pressing ALT-U.

This command can also be put to good use for moving lines or parts of a line to a different position in the text, though this is not the purpose for which it is really intended. The text to be moved should be deleted using one of the word or line delete commands and the cursor moved to the position in the text where the deleted text is to be placed. Pressing ALT-U will then restore the text at the new location.

Note: Only the text removed by the last delete command will be stored in the buffer and any previous contents of the buffer will be lost. It is therefore only possible to un-delete a section of text until such time as any other section of text is deleted.

(vii) Insert and Overwrite mode

Initially PROTEXT, by default, is in insert mode and the word 'Insert' is displayed on the status line at the top of the screen to indicate this. This means that when text is typed, the rest of the text on the line is moved along to the right to make room. This is the mode that is preferred by most people for text entry.

Pressing ALT-TAB will change the status line to 'Overwrite'. Selecting overwrite mode can make certain editing jobs easier and the effect of using it means that if the cursor is positioned over an existing piece of text and new text typed in, the existing text will be replaced by the new text, unlike insert mode, where the existing text would be moved to the right.

If an extra character needs to be inserted whilst in overwrite mode (for example if replacing a word by a longer word), this can be done by pressing ALT and the space bar which will move the text to the right to make room.

(viii) Moving the cursor more rapidly

So far the cursor has been moved by a character at a time, but there are also various ways to move the cursor more quickly. These are as follows:

(a) Pressing SHIFT-→ or SHIFT-← will make the cursor jump a word to the start of the next (or last) word. This feature is useful for moving more quickly to a word which needs correction.

(b) Pressing ALT-← or ALT-→. This moves the cursor to the beginning or end of the line.

(c) Pressing SHIFT-RETURN or ALT-RETURN. This moves the cursor to the beginning of the next line, without causing a new line to be inserted, which would happen if the RETURN key was used on its own.

(d) Pressing ALT-↑ or ALT-↓. This moves the cursor up or down almost a full screen, but with a few lines overlap so that the context may more easily be followed. By holding down ALT-↑ or ALT-↓ the text can be rapidly scanned. Similar functions are performed by ALT-Q and ALT-Z, except that a full screen is scrolled each time, with no overlap of text.

(e) Pressing ALT-[or ALT-] moves the cursor to the beginning or end of the text resident in memory at that time. Pressing the same key a second time will move the cursor to the beginning or end of the complete document.

(f) Pressing ALT-< or ALT-> moves the cursor by a paragraph at a time.

(g) Pressing ALT-(or ALT-) moves the cursor by a page at a time. This is a page as it will be printed, not a 'screen'.

(h) Pressing ALT-@ [or ALT-@] will move to the opening or closing block markers, if set.

(i) Pressing ALT-@ L or ALT-@ R will move to the left or right margin on the current line.

Note: There is a difference between using ALT-@ L or ALT-@ R and ALT-← (which will always move to column 1) and ALT-→ (which will always go to the end of the text on that line, which may be somewhat less than the right margin).

(j) Pressing ALT-⊞ or ALT-⊟ will go to the next or previous marker in the document. See 'Place markers' and 'Multiple markers'.

(k) Pressing ALT-L moves the cursor back to the last position. This is particularly useful if the cursor has accidentally been moved to another part of the text by using an incorrect command. ALT-L will return the cursor to the position where it was before the incorrect move was made. It will only have any effect if the cursor has been moved with one of the 'jump' commands. Moving the cursor a single space at a time will not affect the use of ALT-L and it can still be used to return to the original position from which the last jump was made. With care, this facility can be put to good use, by permitting a jump to another part of the text, where one or two alterations or additions may be made, before pressing ALT-L to return to the original place in the text.

(ix) Moving to a specified page, line or column number

Pressing ALT-G will result in a message appearing on the status line, requesting 'P(age), L(ine) or C(olumn) number'. Entering 'P', followed by a number will move the cursor to the start of that page. Similarly, 'L' plus the line number will move to the line. Prefixing a number with 'C' will result in the cursor moving to the appropriate column. GOTO column is particularly useful when wishing to move to a column which may be a long way across the screen, say to create a wide ruler for some special reason, and is considerably quicker than holding the right cursor down.

If no letter prefix is used, PROTEXT will treat this as meaning a line number.

Note: The line number used by this command may be different from the line number displayed on the status line. This is because when in page mode, the line numbers shown take stored commands into account, whereas this command makes use of the absolute line number in the text. Typing ALT-P will switch 'page mode' off and the status line display will then match the line number used by ALT-G.

(x) Place markers

A place marker can be put anywhere in the text and is similar in use to a book marker. Ten place markers can be set, numbered 0 to 9. A place marker is set by pressing ALT-@ followed by the number. When a marker has been set, it will appear in the text as the number in inverse and will be shown on the status line, so that by looking at the status line it is easy to see which markers are available. Once a place marker has been set, it can easily be returned to at any time by repeating the ALT-@ command with the same number.

In addition to using ALT-@ and the number to find a place marker, it is possible to jump from one to the next in the document by using ALT-⊞ to move on through the document, or ALT-⊟ to move backwards. Using these commands will find the next or last marker in the text. All types of markers (place, multiple and block) will be found. They are not treated numerically, but are found in the order in which they occur in the document. (CPC6128 equivalents: CTRL-@ + moves to next marker, CTRL-@ - moves to previous marker).

As an example of the use of a place marker, suppose a long file is being edited and something needs to be added at the top of the text. A place marker can be set and ALT-[typed, to move to the top of the text, and after making the addition, ALT-@ and the place marker number used to move back to the place marker.

Note: Place markers are saved with the text and will be restored when the file is reloaded. If one file containing markers is merged into another, duplication of markers may occur. The duplicated markers can be removed in the normal way.

(xi) Multiple markers

In addition to the 0 to 9 place markers, PROTEXT also has another special marker. This is known as a 'multiple marker' and is inserted into the text in a similar way to the place markers, the difference being that whereas only one of each place marker can be set in a document, as many of the multiple markers as required may be set. ALT-@, followed by typing a question mark (?) will set a multiple marker and this will appear in the text as an inverse question mark. Multiple markers are saved with the text.

An example of how a multiple marker may be used would be where a standard document was created, which required different items of text to be inserted in a number of set places. If a document was created with a multiple marker everywhere that required insertion of text at a later date, when the file was loaded the next time, it would only be necessary to use ALT-⊞ to go to the first occurrence of a marker and enter the relevant text. Repeated use of ALT-⊞ would find subsequent occurrences. All that would then be required before printing would be to use the FORMAT command to reformat the document. If the multiple markers are positioned in the text by use of tabs, neatly tabulated lists of items are easily created.

(xii) Scrolling

When the text fills the entire depth of the screen, typing further text will cause the screen to scroll up. That is, the top line will disappear and the rest of the screen will move up one line to make room for a new line at the bottom of the screen.

In the same way the text will scroll if the cursor reaches the bottom of the screen but there is more text to come, or reaches the top of the screen when the text has previously scrolled. This is known as vertical scrolling, and is essential for editing text that is longer than a few lines.

PROTEXT has commands to force the screen to scroll either up or down at any time. This is done by pressing SHIFT-↑ or SHIFT-↓. The cursor will stay on the same line, but the whole text will scroll by one line. This feature is useful if a line is to be edited and it is desirable to see the text beneath or above.

There is another form of scrolling, called horizontal scrolling, which happens automatically when the cursor is moved beyond the right hand limit of the screen. If this is done the text will scroll to the left. This means that the text on the left of the screen will start to disappear as the cursor is moved further to the right of the screen. Horizontal scrolling allows text to be entered in lines that are longer than the screen width. This can be confusing at first and so is best avoided initially. If horizontal scrolling occurs, any of the commands which move the cursor to the left may be used to scroll the text back, or SHIFT and RETURN may be pressed together, which will return the cursor to the start of the next line.

Note: When horizontal scrolling takes place, the screen moves across a number of spaces at a time. This figure may be changed with CONFIG (See Utilities), but it should be borne in mind that the smaller the movement, the more it will slow down the operation of PROTEXT.

(xiii) Splitting and joining lines

Lines will often require splitting, or joining together. This is very easy in PROTEXT. There are two different methods of doing this, depending on whether 'Insert' or 'Overwrite' mode is in operation.

To split a line whilst in Insert mode, the cursor should be moved to the character which is to be the first on the new line and RETURN pressed. To join two lines, either move to the end of the first line and press DEL→, or move to the start of the second line and press ←DEL. The text on the second line will then move up and join onto the end of the text on the first line.

If in overwrite mode, ALT-* will split the line at the cursor and ALT-+ will join the next line to the end of the current line.

b). Cut and paste editing - BLOCK mode

This is where real word processing begins. PROTEXT allows any section of text to be moved or copied to any other part of the text. This is often called 'cut and paste' editing. PROTEXT has two different methods of using 'cut and paste' editing, block mode and box mode (sometimes referred to as 'column' mode).

This chapter will describe the ways to use block mode editing. A block of text is any continuous section of text. It may be of any length and may start at any position in the document and finish at any position. When in block editing mode, all text between these two points will be manipulated in whatever way is chosen.

(i) Defining a block

The first requirement is that the block of text is marked with block markers. The cursor should be moved to the start of the section of text and SHIFT-COPY pressed (alternatively SHIFT-⊞ can be used and may be found more convenient). This will set a block marker. The marker will be indicated on the screen by an inverse video square bracket. The cursor should then be moved to the end of the section and SHIFT-COPY pressed again, to set a second marker. The block has now been defined. An opening square bracket is the start marker, a closing square bracket the end marker. When markers are defined, this will be indicated on the status line, where the message 'No markers set' will be replaced by 'Markers [ ]', showing that both the start and end markers are set.

The markers can be set in either order, and can be at any position in the text. The first marker set will be displayed as an opening bracket, but if the second marker is positioned earlier in the text than the first marker, this will change to a closing bracket. If the marker is put in the wrong place, pressing SHIFT-COPY again while the cursor is still on the marker will remove it. Either or both block markers can be cleared at any time, by pressing ALT-K or CAN. Often a block will consist of a number of complete lines. To define a block like this, the first marker should be positioned at the start of the first line, and the second marker at the start of the line following the last line of the block.

If an attempt is made to set a marker when both are already set, a beep will sound and an error message will be displayed on the status line. Pressing STOP will resume editing and ALT-K can be used to clear the markers.

(ii) Moving or copying a block

Once a block has been defined, it can be moved to any point in the text simply by moving the cursor to the required position and pressing ALT-M or alternatively on the PCW, the PASTE key. The markers will move with the text. The cursor must not be within the block at the time; if it is an error message will be displayed on the status line. Pressing STOP will return to edit mode and the cursor can be moved to the correct position.

The block can also be copied, leaving the original text intact. This is done by pressing ALT-COPY or just COPY (PCW only). The markers will be moved with the block, which makes it easy to see clearly where the new copy of the block is and also to copy the block again if required. The cursor must not be within the block. (CPC6128 equivalent: CTRL-COPY).

(iii) Deleting a block

The section of text to be deleted must be defined in the usual way. Pressing the CUT key will delete the block. If the block is larger than a certain size (see below) a beep will sound and a warning message will be displayed on the status line, requesting confirmation that the block is to be deleted. The block will only be deleted if 'Y' is selected. (CPC6128 equivalent: CTRL-DEL).

(iv) Un-deleting a block

If a block of text is accidentally deleted, it may often be recovered by use of the ALT-U command. When text is deleted, PROTEXT retains the deleted block in a buffer and ALT-U will restore it to the document.

Note: By default the buffer will hold 512 characters, but this can be changed by the user through the use of the configuration program CONFIG.COM. Full details of how to alter the buffer size are given in the chapter on CONFIG.COM. If a block of text which is too large for the buffer is to be deleted, a warning will be given, with the option to continue. If 'Y' is selected, the block will be deleted and the buffer will be filled with as much of the text as it can hold and the remainder will be discarded.

Note: A block can only be restored until such time as further text is deleted, after which time the buffer will contain only the most recently deleted text.

c). Cut and paste editing - BOX mode

The block mode cut and paste facilities already described only operate on continuous sections of the text but the Box mode editing facilities described in this chapter extend this to operate on any section of text that can be defined by drawing a rectangle on the formatted text. A block defined in this way is termed a 'box'. This mode is also known as 'column mode'. There is one new command used for box mode.

ALT-B is used when in edit mode to toggle box mode on or off as required. Normally one would leave box mode turned off, except when needed.

(i) Defining a box

A box is defined by first typing ALT-B to enter box mode and the message 'BOX' will appear on the status line to indicate that box mode is in use. The cursor should be moved to the top left corner of the (imaginary) box and SHIFT-COPY or SHIFT-⊞ pressed to set a marker, in the same way as described in the previous chapter. The cursor should then be moved to the bottom right corner of the box and SHIFT-COPY pressed again, to set a second marker. Whereas in block mode only a closing block marker appeared, this time markers will appear along both the left and right hand sides of the box to mark its limits.

(ii) Moving or copying a box

A box can be moved or copied, using the same commands (ALT-M, ALT-COPY) that were used for block commands, to move or copy the box. The cursor should be positioned where the top left corner of the box is required and then the appropriate copy/move command used.

Note: When using box mode, the effect on existing text is always as if insert mode were in operation. If a box is positioned over existing text, that text will be moved to the right to make room for the box of text.

(iii) Deleting a box

Box mode delete can be used in one of two ways, depending on whether insert or overwrite mode is in operation at the time the box delete command is used. Once a box has been marked out for deletion, CUT may be used to delete the box. If insert mode is in operation, the effect of this will be that once the box has been deleted, any text to the right of the box will move across to the left to close the gap.

Alternatively, if overwrite mode is selected, using ALT-TAB, before the delete command is used, the box of text will be cleared, but the area where the box used to be will remain as a blank area.

Very effective layouts can be achieved by the use of box mode. One example would be to create a page with two columns of text, along the lines of a magazine page. This is simply done by creating the document, formatting it with a width of about 35 characters and then using box mode to move the second half up and to the right, positioning it alongside the first part of the text. This should be done immediately prior to printing, as any further reformatting might destroy the formatting of the two columns.

Another use for box mode which can save a considerable amount of time is to remove surplus columns of figures from text. A box can be marked round the columns which are not required and the box can be deleted using either overwrite or insert mode, depending on whether the remaining text should be moved across or not.

Note: When a box move is carried out and the box contains tabs, it may sometimes appear that the justification has been destroyed. The justification can be restored either by inserting an extra tab marker on the ruler line, or by replacing the tab, in the text, with spaces.

d). Rulers, tabs and margins

A ruler is a special line that marks out the margins and tabs. Margins are the left and right hand limits of the text, whilst tabs are markers used to position text in set columns on the screen and are particularly useful for lining up columns of words or numbers. Tabs work in a similar fashion to those of a typewriter, but are much more flexible. The line at the top of the screen, immediately underneath the two status lines, is the 'active ruler line'. When PROTEXT is initially loaded and there is no text present, this will be the 'default' ruler which is provided by PROTEXT when no other default ruler is found. Each '!' or '.' on the ruler marks a tab position and the 'L' and 'R' indicate the left and right margins.

(i) Ruler lines

Initially, with no document in memory, the default ruler supplied by PROTEXT is displayed as the active ruler. If a document is loaded which contains any ruler lines, then the default ruler for that document is considered to be the ruler line in the document which will be in operation **when printing commences**. This is not necessarily the first ruler in the text, but the last ruler before the start of the text that will be printed. This ruler line will be considered to be the 'default ruler' for the duration of the document, or until such time as it is replaced by another one at the start of the printed text. The reason for this is explained later in the chapter.

At all times the 'active ruler line' at the top of the screen is the one applicable to the line that the cursor is on.

New ruler lines can be created to suit any required layout. The new ruler line is typed in as a normal line of text. The line must have a 'greater than' symbol (>) in the first column. There is no need to type in the hyphens (-) along the line, though it may be preferred to make the ruler line more readily visible. Wherever a tab is required, a '!' should be typed in and the point '.' wherever a decimal tab is wanted. Columns 2 and 3 must contain either spaces, a hyphen, the letter 'L' or a '!' tab marker.

Note: The TAB key should not be used in a ruler line, when altering or creating a new ruler line since this will cause tabs to be set wrongly.

Any number of ruler lines are allowed in the text and each applies from the line beneath, until the next ruler line (or the end of the text if there are no more ruler lines). When the cursor is moved past a ruler line, in either direction, the first ruler line above the cursor will immediately replace the previously active ruler at the top of the screen. If the cursor is moved above the first ruler line in the text, the default ruler will be restored.

Ruler lines allow the formatting of text in any chosen way. Simply altering the ruler line and using the ALT-F command will reformat the text (alternatively FORMAT in Command mode will automatically reformat the entire document). If full use has been made of tabulating the text by inserting tab characters, moving the tab markers on the ruler line will cause the text to be repositioned immediately, using the altered ruler line. Thus it is easy to move columns of text/figures about or to experiment with different formats to decide which is best.

There are three editing commands connected with the use of rulers. The first of these is ALT-D, which can be used at any time to copy the default ruler to the current cursor position. This can be particularly useful to restore the text to a standard format after using a special ruler to create tabulated columns of figures, for example.

The second command is ALT-R which will restore **the previous ruler but one**. The most common use for this command is to restore the previous text format after, say, insetting the left and right margins of a piece of text to highlight it. One of the most common uses of ruler lines is to alternate between two different layouts and using ALT-R at the end of each one will automatically copy down the ruler appropriate to the previous layout. In this way it is particularly easy to switch between two layouts simply by the use of ALT-R.

A useful tip is to create an alternative ruler line at the start of the document, before any text is printed and before the first ruler line which will be used in the document. When the document is loaded, the second ruler line will become the default ruler, as it will be the first one used when printing and when the alternate ruler is required, ALT-R can be used to copy it down and used subsequently to alternately restore the default ruler and the alternative one.

The third command connected with rulers is ALT-V R. This command has the effect of 'hiding the active ruler line' at the top of the screen. Repeated use of this command will toggle the ruler line on and off. It only has any effect on the ruler line beneath the status lines and does not hide the ruler lines in the document and even when it is invisible, it still functions as normal.

(ii) Tabs

The main use of tabs is for aligning text or numbers in columns. PROTEXT has two types of tabs, the normal 'tab' and the 'decimal tab'. The decimal tab also serves the purpose of a 'right justification' tab. These tabs are inserted into a ruler line by typing in an exclamation mark (!) for a normal tab marker, or a point (.) for a decimal tab marker, wherever they are required. As many of either tab as necessary may be used on each ruler line.

When TAB is pressed (in insert mode) a special tab character is stored in the text. This character can be thought of as occupying several columns on a line, or as being one long, variable length space from the position where the tab was inserted, upto the column marked by the tab marker in the ruler line.

Note: It is recommended that full use is made of tabs when creating a document, as not only do they provide a simple means of aligning text, but they enable the easy re-arrangement of text (See chapter on Formatting) and are most important when proportional printing is being used.

(iii) Normal Tabs

Each '!' on the ruler line indicates a tab position. Pressing the TAB key during text entry will move the cursor to the column appropriate to the next of these positions.

If insert mode is in operation, pressing TAB will result in a tab marker being inserted into the text, giving the appearance that a number of spaces have been inserted in the line. Pressing ←DEL once, immediately after TAB, will remove the tab marker and all the apparent spaces and move the cursor back to its previous position. Since the cursor cannot be moved within the columns formed by this 'long' space, it will jump over a tab character if moved onto it.

Pressing SHIFT-TAB will move the cursor to the next tab position, but will not insert a tab character. If overwrite mode is selected, the functions of TAB and SHIFT-TAB are reversed.

(iv) Decimal (right justification) Tabs

This is a special form of tab which can be used in a number of ways. It permits the alignment of numbers so that the decimal points line up automatically. Any numbers entered after pressing TAB to move to a decimal tab will automatically be positioned to the left of the tab position until the decimal point (.) is entered, after which any further numbers will be placed to the right. If no decimal point is entered (as for a whole number) and TAB or RETURN is pressed, the number will be right justified in such a way that units, tens, hundreds etc. all line up.

This tab can also be used with words, as a right justification tab, in which case the words will be lined up in such a way that the end of the words are all in line. This can be very useful for entry of words like 'Total' and 'Subtotal' in conjunction with columns of figures.

Note: Some countries use the comma instead of the full stop as the decimal point. PROTEXT may be configured to accept the comma ',' instead of the full stop '.' by using the CONFIG utility (See Utilities.), but it should be noted that the full stop should still be used on the ruler line to mark the position of the decimal tab.

The right margin as a decimal or right justification tab.

The right margin (denoted by 'R' on the ruler line) also acts as a decimal/right justify tab. The TAB key should be pressed a sufficient number of times to move the cursor to the right hand margin of the document. Any text typed now will be right-justified at the margin. For example the address on a letter, or the date, may be usefully entered in this way.

Note: There must be sufficient space between the last tab and the margin for the text that is to be entered. otherwise part of the text will be forced onto a new line.

(v) Margins

The ruler line is also used to define margins. These are indicated by 'L' for left margin, and 'R' for right margin. These define the part of the screen within which text will be formatted. By default the left margin will be at column 1 and the right margin at column 70, but they can be set, just like tabs at any position (though of course the right margin must be to the right of the left margin!).

To set margins a ruler line should be created, as described above, the cursor moved to the column at which the left margin is to be set and the letter 'L' typed. Next the position of the right margin should be selected with the cursor and the letter 'R' typed. If the left margin is to remain at column 1 then the 'L' should be omitted.

To see the effect of changing margins try creating a new ruler line above a paragraph of text with, for example, a left margin at column 5 and a right margin at 60. The status lines show the current column number and serve as a guide to selecting the correct columns. Positioning the cursor in column 1 of the first line of the paragraph and reformatting it with ALT-F will rearrange the text to fit within the new margins.

(vi) Left margins and tabs - applications

1. Indentation of the first line of a paragraph

This is best done by setting a tab marker on the ruler line in the column to which the text is to be indented. At the start of each paragraph, pressing TAB, prior to typing the text, will position the cursor accordingly.

It is very easy to indent the text at any subsequent stage, simply by moving the cursor to the start of the paragraph, pressing TAB, and then ALT-F to re-format the paragraph. This is particularly useful where it is decided to split a paragraph into two paragraphs.

Example: a ruler line to give 57 column wide text with a tab at column 5 for indenting the first line of the paragraph.

```
>---!---------------------------------------------------R
```

2. Indentation of a section of text

The main use of left margins is for indentation of a section of text. When RETURN (or SHIFT-RETURN or ALT-RETURN) is pressed the cursor moves to the left margin, so, by setting an indented left margin, text can be automatically indented.

Note: A left margin should not be used to indent the whole of a document. This should be accomplished by use of the '>SM' side margin stored command (See Stored commands).

3. Marginal comments

When the text is formatted with the 'FORMAT' command any text that is within the left margin i.e. to the left of the left margin setting, at the start of a paragraph (the first line), is unaffected. If the formatting command, ALT-F is used from edit mode, in order to reformat a paragraph, the effect will be exactly the same, unless the cursor is positioned in column 1 at the start of a paragraph when the command is used, in which case the whole paragraph will be moved within the left and right margins.

This means that the left margin can be used to include marginal comments with the text at the start of a paragraph. To type a comment in the margin, ALT-← should be used to move the cursor to column 1. Tabs are allowed within the margin.

For example the following ruler line may be found useful for numbering indented paragraphs:

```
>---!---L-----------------------------------------------R

    1.  First paragraph.

    2.  Second paragraph.
```

Another use for this type of layout is for the creation of scripts and one of the advantages is that marginal comments can be added at any time without disturbing the main body of the text, as long as they are restricted to the area to the left of the left margin. Care must be taken since formatting the text will leave text in the margin on the first line of paragraphs, but integrate comments on later lines with the main body of the text. It is preferable not to reformat paragraphs of this sort.

Note: A command, FIXB, is provided which enables the format of a section of text to be 'fixed' (See miscellaneous commands in the chapter on Command mode). Once the text has been fixed, it will not be affected by subsequent reformatting.

e). Formatting

Formatting is the process which determines the layout of the text and takes two forms, automatic formatting and manual re-formatting.

Automatic Formatting

There are two main features which determine the final appearance of the document and which happen automatically as the text is entered.

(i) Word-Wrap

It will already have been noticed that when a complete line of text is typed, the cursor moves on to the next line, and any part-entered words are also moved with it. This operation is called 'word-wrap'. The status line will indicate that word-wrap is working. ALT-W toggles word-wrap on and off and the status line will change to indicate this. Setting word-wrap to 'off' and typing a line of text will show the difference. The cursor stays on the same line until RETURN is pressed. When the edge of the screen is reached, the display will scroll sideways. Pressing ALT-W a further time will turn word wrap back on again.

(ii) Right-Justification

It will probably also have been noticed that all text appears with a straight right hand edge, like a book. This is the default setting and the status line will indicate that right-justification is in operation.

PROTEXT lines up the right hand edge by automatically spacing out the words on a line. ALT-J may be used to turn off the right-justification, and the status line will change to indicate this. If a paragraph is now typed in, it will be found that the right hand edge is 'ragged', as if typed on a typewriter.

Note: The default settings of both Word-Wrap and Right-Justification may be altered to suit the user, by use of the CONFIG program (see Utilities section of the manual for full details).

Manual Re-formatting

If for any reason a line has been altered, some text deleted, moved, or a block of text copied, almost certainly the formatting of the text will have been spoilt. There are two ways to correct this.

(iii) Edit Mode Formatting

If the format of just a small part of the text has been destroyed, it is easier to re-format these small areas whilst still in edit mode. All that is required is to position the cursor on the first line that needs re-formatting and type ALT-F, which will cause the remainder of the paragraph to be re-formatted according to the justification, word-wrap and ruler line settings. A very useful sequence is ALT-< ALT-F, which will move the cursor to the start of a paragraph and then re-format the whole paragraph.

(iv) Command Mode Formatting

If considerable areas have been disrupted, or it has been decided to alter the number of characters on a line (see chapter on Tabs and Margins), the simplest way to re-format is to return to command mode, using STOP and then type in the command FORMAT. This will automatically re-format the entire text, from start to finish, in one go.

A variation of this command is FORMATB, which will format the part of the document selected with the Block markers (see 'Cut and Paste Editing'), but will continue past the end of block marker to the end of the paragraph.

(v) Centring text

It can be very useful to be able to position text, such as a title, exactly centred on the line. This can be done by moving the cursor to the line in question and pressing ALT-C. The text will be centred and the cursor will be moved to the left margin of the next line.

If the file is re-formatted with a different text width (as explained in the previous section), the line will no longer be centred. To ensure that the line is always correctly centred, the three characters '>CE' should be typed in the first three columns of the line. This is an example of a 'stored command' - a command that is put into the text to be used at some later time (In this case, during the course of printing).

(vi) Formatting whilst printing

PROTEXT also has a stored command, '>FP ON/OFF' which may be embedded into the text and which ensures that the document is reformatted during the course of printing. See 'Stored commands' for full details.

Note: When mail merging is in use, any paragraphs containing merged in text will automatically be reformatted.

(vii) Soft spaces, returns and hyphens

Soft characters are used by PROTEXT for formatting. When a paragraph is right-justified, spaces are inserted in the line. These spaces are treated differently from spaces inserted by pressing the space bar (these are called hard spaces). Hard spaces can never be removed by formatting, but soft spaces can.

There is a similar distinction between a hard and soft return (end of line). A hard return occurs where the RETURN key has been pressed, and this marks the end of the paragraph. A soft return occurs where the action of word-wrap has caused a new line to be started. Like soft spaces, soft returns can be removed by formatting.

A soft hyphen is slightly different in that it must be explicitly entered by pressing ALT and hyphen and will be displayed in inverse video. Soft hyphens can be inserted at points in a word where hyphenation is permissible. The formatter will then split a word at a soft hyphen and display a hyphen instead of moving the whole word onto the next line. A soft hyphen will only be printed on the printer if it is at the end of a line, i.e. where the formatter has split the word. This feature is especially useful if there is a long word that is just too long to fit on the end of the line, but which might subsequently be moved by reformatting of the text.

(viii) Non-break spaces and hyphens

When the text is formatted it can result in there being several spaces where only one was typed. This is sometimes undesirable and PROTEXT provides 'non-break spaces' which are not affected by formatting. A non-break space is entered by pressing ALT-N, followed by the space bar and will be displayed as a small triangular mark, so it can be distinguished from a normal space. Soft spaces will not be inserted between words linked by the non-break space and the words will not be separated at the end of a line.

Examples of use:

1. The space between 'Mr' and 'A' in 'Mr A.Smith'. This would look wrong if displayed as 'Mr A.Smith', so ALT-N space is used after 'Mr' instead of just pressing the space bar.

2. A string enclosed in quotes, that must be printed exactly as it was typed, e.g. 'T h i s s e n t e n c e i s s p a c e d o u t'.

Non-break hyphens are similar. Normally when a hyphen is used in a word, if that word comes at the end of a line, the word will be broken at the hyphen and the second part of the word printed at the start of the following line, as would be expected. Sometimes this is not the effect that is required and the two words should be kept together as one. An example of this is found many times in this manual, where editing commands are described. A command like 'ALT-F' would normally be split onto two lines if it came at the end of a line, which would look wrong. PROTEXT provides a non-break hyphen to take care of this situation. Instead of typing a normal hyphen, typing ALT-N, followed by pressing the hyphen key will set a non-break hyphen instead. This will be displayed as a 'heavy' hyphen on the screen, but when printed, will appear as a normal one.

(ix) Viewing the text without control codes

Control codes of various kinds (printer control codes, place markers etc) are normally displayed in the text in inverse video. This means that the formatting of the text (tabulated columns of figures, right-justification, etc.) may be unclear. To view the text without any control codes, the command ALT-V V should be used. Typing ALT-V V a second time will replace the control codes. The control codes affected by ALT-V are as follows:

Printer control codes (a-z), block and box markers ([,]), position markers (0-9 and ?), non-break spaces, and soft hyphens (-).

(x) Viewing the text with tabs, hard returns and hard spaces

The command ALT-V T toggles the display of tab characters and hard returns on and off. A tab is shown as a right pointing arrow, and a hard return as a cranked left pointing arrow. A second use of this command will turn the display off again.

The command ALT-V S will display all hard spaces (those that are typed in at the keyboard) as a small shaded pattern on the screen. Pressing ALT-V S a second time will restore the display to normal.

These options are primarily intended for the purpose of viewing where the various characters have been placed, particularly when sorting out complicated layouts and by default they are in the 'OFF' setting when PROTEXT is loaded. Some people prefer to have some or all of these options in operation at all times and the CONFIG program (see Utilities section) provides the option to adjust the defaults to suit the user.

f). Stored commands

Many of the features of PROTEXT are called into operation by stored commands, so it is essential that their use is fully understood before some of the advanced features described in later chapters are used.

The second part of the chapter lists all the general purpose commands which are not directly connected with mail merging. Commands connected with mail merging are covered in chapters devoted to the subject.

Note: It should be noted that there are many commands in the mail merging section, which, though their primary use may be in connection with mail merging, are also extremely useful in ordinary documents as well. Once a degree of familiarity with the following commands has been gained, it will be found useful to read the chapters on mail merging, particularly those concerned with the use of variables.

PROTEXT permits the controlling of various of the printing parameters in two ways, either through the use of the CONFIG command, or through the use of stored commands. CONFIG has the effect of changing the default settings of the various print options that are provided by PROTEXT. Full details of the CONFIG configuration program will be found in the section on Utility programs.

Stored commands can be used in PROTEXT for many things, including setting page length, margin sizes, page numbering, printing text at the top or bottom of every page and line spacing. By using stored commands within a document, it is possible to set the appropriate options relating to that individual document, and these will take priority over the default settings. This is a very flexible method, since different types of document may require quite different margin sizes, page lengths or line spacing, for example.

The ruler line that is used to define the width of the text and tab settings is a special example of a stored command. This can be used in conjunction with the stored command '>SM' (side margin) to ensure that text is printed centrally on the paper.

A stored command occupies a line by itself and is edited as normal text, but during printing it is obeyed and not printed. If a printed copy is required of the text, complete with stored commands, this can be done by going into Program mode and then printing. See chapter on 'Program mode'.

When printing is taking place, the program will make use of the default settings for the various printing options set by CONFIG until a stored command is found which redefines that parameter, in which case the stored command has the effect of overriding the defaults, until either printing is complete or the stored command is redefined again by another stored command. It should be noted that some of these commands take immediate effect, whilst others only take effect from the next page printed.

A stored command line is denoted by a greater-than sign ('>') in the first column. Columns 2 and 3 contain a two letter code defining the instruction. The remainder of the line may contain a parameter for the command. A semicolon may be used to terminate the command, allowing a comment to be put on the same line. Any characters after a semicolon are ignored (unless the semicolon is enclosed in quotes). Any line with '>' in column 1 that is not followed by a known command is taken to be a ruler line if columns 2 and 3 each contain one of the characters: 'L', '-', '!' or a space.

Whenever text is printed, PROTEXT performs a syntax check on the stored commands. If an unrecognised command is found, the syntax check stops and the error message 'Unknown stored command' is displayed. STOP should then be pressed to return to edit mode to correct the mistake.

Note: Some stored commands take immediate effect, but many only take effect at the next new page. Commands which take immediate effect are marked with an asterisk (*) throughout this chapter.

(i) Template Files

An important concept is that of creating a number of 'template' files, arranged for commonly used document formats, which are saved on disc and loaded when required. As an example, one template file might be for 'letter format', another for 'double spaced manuscript' and so on.

A template file would contain, amongst other things, a ruler line and a number of stored commands defining the page length, the margins, continuous or single sheet stationery, etc. In some cases the template might contain some standard text. For example, an invoice template could be a complete invoice with gaps to enter the name and address, details and the figures, or a letter template could contain the sender's address and telephone number.

When a new document is started, it is only necessary to load the appropriate template file and everything will automatically be configured to suit it. Another advantage of using stored commands is that the template will be saved with the document, ensuring that when the document is next loaded, all the parameters will be as before.

Example of a 'letter template'.

```
>CO Letter template
>SM 10        ; - Side margin 10
>PL 72        ; - Page length 72 (for 12 inch paper)
>CP OFF       ; - Continuous printing off (single sheet)
>----!-----------------------------------------R
```

Example of a 'manuscript template'

```
>CO Manuscript template
>TM 2         ; - Top margin 2
>LS 2         ; - Double line spacing
>CP ON        ; - Continuous printing
>PN 1         ; - Page number from 1
>HE           TITLE OF DOCUMENT
>FO                    Page %
>--------!--------------!------------------------------R
```

Note: On occasions a document may contain a considerable number of stored commands. One effect of this, which may be noticed, is that scrolling through a document may be considerably slower, as PROTEXT checks every stored command when a line is scrolled. If the ALT-P option is used to disable page mode, the speed of scrolling will be increased.

(ii) General purpose stored commands

The remainder of this chapter describes the stored commands which are not directly connected with mail merging or control of the printer. Full details of these will be found in the appropriate sections.

Rather than being listed in alphabetical order, the stored commands are grouped into a number of categories. A technical appendix in the back of the manual lists them in alphabetical order, with full details of syntax and usage.

Note: Full details of the use of variables in stored commands will be found in the chapter on 'Making the most of variables' in the Mail merging section of the manual.

Note: Comments may be placed on the same line as any stored command and can serve as a reminder of the function of the command. The semi-colon (;) is used to indicate tó PROTEXT that any further text is merely a comment and is not to be acted on.

In the lists of commands in this section the parameters are described by abbreviations:

n	A number less than 256.
nn	A number less than 65536.
(n)	An optional number less than 256.
text	A string of characters.

Note: Commands which have an immediate effect are indicated with an Asterisk (*). The remaining commands only become effective at the commencement of a new page.

(iii) Paper layout

The following commands are concerned with the dimensions and positioning of the area of the paper to be printed on and include the setting of margins, page lengths etc.

>BM n	Bottom Margin.
>EM n	Even side Margin.
>FM n	Footer Margin.
>HM n	Header Margin.
>OM n	Odd side Margin.
>PL n	Page Length.
>SM n	Side Margin.
>TM n	Top Margin.
>ZM	Zero all Margins.

Note: The number of lines of text printed on a page is given by the formula:-
PL-TM-BM(-HM if headers on)(-FM if footers on)
i.e. the page length minus the sum of the sizes of the margins.

BM Bottom Margin.

Example: >BM 10

Description: The bottom margin is the number of blank lines left at the bottom of every page, below the footer. The default is 3.

EM Even Margin.

Example: >EM 5

Description: Defines the number of spaces printed at the left hand edge of every even numbered page. The default is 5.

FM Footer Margin.

Example: >FM 3

Description: Defines the number of lines between the last line of text and the bottom margin. The bottom line of the footer margin may have the footer text printed on it (see 'FO', in next section). The default is 2, but only if footers are on.

HM Header Margin.

Example: >HM 4

Description: Defines the number of lines between the top margin and the first line of the text. The top line of the header margin may have the header text printed on it (see 'HE', in next section) The default is 2, but only if headers are on.

OM Odd Margin.

Example: >OM 7

Description: Defines the number of spaces printed at the left hand edge of every odd numbered page. The default is 5.

PL Page Length.

Example: >PL 72

Description: PL defines the actual size of a page of paper, and is different from the number of lines of text printed on a page. The default page length is 66, which is the correct setting for standard 11 inch paper.

SM Side Margin.

Example: >SM 10

Description: Defines the number of spaces printed at the start of every text line between the paper's edge and the first column of text. The default is 5. This is called 'side margin' to distinguish it from the left margin that can be set on a ruler line. The PROTEXT default ruler and side margin have been chosen for this reason.

Note: Setting the side margin is equivalent to setting both even and odd margins to the same value.

Note: Use this command to print text centrally on the paper, and leave the left margin of the ruler set at column 1 (except where text is indented).

TM Top Margin.

Example: >TM 0

Description: The top margin is the number of blank lines left at the top of every page, above the header. The default is 3.

ZM Zero all Margins.

Description Following this command all margins will be set to zero. This command has a number of uses, one in particular being when used in conjunction with PRINTF, to create a pure Ascii file. If PRINTF is used without this command, all margins and page breaks will also be sent to the file.

Note: Any margins required can be reset after the use of this command, using the appropriate stored command.

Reminder: PROTEXT provides each of the above margin settings with a default value. These default values may be changed by using the CONFIG utility (See Utilities.). The default values will automatically be used when printing takes place without the need to use the stored commands in the document. Any of these values may be changed in the document, by the use of the appropriate stored command, which will then override the default settings.

(iv) Page Formatting

These commands are concerned with how the text is presented on the area of paper defined by the previous commands and cover items like headers, footers, page numbering, line spacing and formatting.

*	>CE text	CEntre line.
	>CP ON/OFF	Continuous/Single sheet Printing.
	>EA nn	End At page.
	>EF text	Define Even Footer text and turn footers on.
	>EH text	Define Even Header text and turn headers on.
*	>EP (n)	Even Page throw (can be conditional).
	>FF ON/OFF	Form Feeds enabled/disabled.
	>FO text	Define FOoter text and turn footers on.
	>FO ON/OFF	Turn FOoters on/off.
*	>FP ON/OFF	Formatting whilst Printing on/off.
	>HE text	Define HEader text and turn headers on.
	>HE ON/OFF	Turn HEaders on/off.
*	>LS n	Line Spacing.
	>NC n	Number of Copies.
	>NP ON/OFF	Enable/disable New Page at end of printing.
	>OF text	Define Odd Footer text and turn footers on.
	>OH text	Define Odd Header text and turn headers on.
*	>OP (n)	Odd Page throw (can be conditional).
*	>PA (n)	PAge throw (can be conditional).
	>PE ON/OFF	Print Even numbered pages only.
	>PN nn	Page Number.
	>PO ON/OFF	Print Odd numbered pages only.
*	>PS ON/OFF	Paper Sensing on/off.
*	>RJ ON/OFF	Right Justifying on/off.
	>SA nn	Start At page.

CE * CEntre text in line.

Description: This command is used to centre the line during re-formatting and during printing.

CP ON/OFF Continuous/Single sheet Printing.

Examples:

```
>CP ON
>CP OFF
```

Description: "ON" selects continuous printing, "OFF" selects single sheet printing. The default setting is "ON".

EA End printing At page.

Example: >EA 47

Description: Sets the last page number to be printed.

EF Even Footer text definition.

Example: >EF This is printed at the foot of even pages only

Description: Permits the printing of the specified text at the foot of even numbered pages only. Page numbers may be included (see section on Headers and Footers).

EH Even Header text definition.

Example: >EH This is printed at the top of even pages only

Description: Permits the printing of the specified text at the top of even numbered pages only. Page numbers may be included (See section on Headers and Footers)

EP * Even Page throw.

Example: >EP 10

Description: Similar to the normal page break command '>PA' except that if the new page to be printed has an odd number, a complete blank page will be printed. The text is therefore made to start at the top of an even numbered page. If a parameter is given, a new page will be started only if the number of blank lines remaining on the current page is less than or equal to the number specified.

FF ON/OFF Form Feeds enabled/disabled.

Examples: >FF ON
>FF OFF

Description: When enabled a form feed character (12) is printed at the end of each page. The default setting is "OFF".

Note: This command is rarely required. If '>FF ON' is to be used, it is important to make sure that the correct page length is set, if necessary using the 'OC' command to send the correct page length codes to the printer.

FO FOoter text definition.

Example: >FO Continued on next page

Description: Defines the footer text which is to be printed at the bottom of every page. The text is printed on the bottom line of the footer margin.

Note: Page numbers may be included (see section on Headers and Footers).

FO ON/OFF Turn **FO**oters on or off, without affecting the footer text

Examples: >FO ON
>FO OFF

Description: Used to determine whether the footer text is to be printed or not. It does not affect the footer text contents.

Note: The text to be printed must have been previously defined with the >FO 'text' command.

FP ON/OFF * Formatting whilst **P**rinting on/off.

Examples: >FP ON
>FP OFF

Description: When enabled, the text is formatted during the printing operation. The command '>RJ' controls whether right justification is carried out or not. '>FP' should always be used in conjunction with '>RJ ON/OFF', but in the event that '>RJ' is not specified, formatting of the document will take place according to the current setting of edit mode justification when the file is printed.
Even when disabled, any paragraph that includes a variable reference will be re-formatted as a matter of course (see 'Mail merging'). The default setting is "OFF".

HE HEader text definition.

Example: >HE Chapter 1

Description: Turns headers on and defines the header text which is to be printed at the top of every page. The text is printed on the top line of the header margin.

Note: Page numbers may be included (see section on Headers and Footers).

HE ON/OFF Turn HEaders on or off without affecting the text.

Examples: >HE ON
>HE OFF

Description: Used to determine whether the header text is to be printed or not. It does not affect the header text contents.

Note: The text to be printed must have been previously defined with the >HE 'text' command.

LS * Line Spacing.

Examples: >LS 2
>LS 1.5 (the half symbol may also be used)

Description: Defines the number of line feeds to be printed after each line of text. Half line feeds are permitted, but the appropriate codes must be specified in the printer driver by using SETPRINT. The default is 1, that is single spacing.

Note: Do not use >LS 0 to prevent double line feeds - this would work but would have the side effect of printing the whole text without any page breaks. Unwanted double line feeds should be corrected by altering the dip switch settings on the printer (see Appendix on Printer and disc problems).

NC Number of Copies.

Example: >NC 3

Description: Sets the number of copies to be printed. Number must be between 1 and 255.

Note: This command is ignored if a data file is defined.

NP ON/OFF Enable/disable New Page at end of printing.

Examples: >NP ON
>NP OFF

Description: When enabled the paper is moved to the top of the next page at the end of printing the document. When disabled the paper is stopped as soon as the last line of text has been printed. The default setting is "ON".

Note: It may be useful to turn this setting off, in order to add 'personal' post scripts to letters, by using Typewriter mode (See Command mode).

OF Odd Footer definition.

Example: >OF This is the bottom of an odd numbered page

Description: Defines the footer text which is to be printed at the bottom of every odd numbered page.

Note: Page numbers may be included (see section on Headers and Footers).

OH Odd Header definition.

Example: >OH This is the top of an odd numbered page

Description: Defines the header text which is to be printed at the top of every odd numbered page.

Note: Page numbers may be included (see section on Headers and Footers).

OP * Odd Page throw.

Example: >OP

Description: Similar to the normal page break command '>PA' except that if the new page to be printed has an even number, a complete blank page will be printed. The text is therefore made to start at the top of an odd numbered page. If a parameter is given, a new page will be started only if the number of blank lines remaining on the current page is less than or equal to the number specified.

PA * PAge throw.

Examples: >PA
>PA 6

Description: In its simplest form (>PA), this command forces a new page to be started and printing of the text will continue from the top of the next page. If the PA command occurs at a natural page break then it will not force an extra new page.
If a parameter is given a new page will be started only if the number of blank lines remaining on the current page is less than or equal to the number specified.

PE ON/OFF Print Even numbered pages only.

Example: >PE ON

Description: Only pages with even numbers will be printed when printing takes place. '>PE OFF' causes all pages to be printed.

PN Page Number.

Example: >PN 1001

Description: The page number can be printed at the top or bottom of every page, or specified pages. (see stored commands EF, EH, FO, HE, OF, OH and FO). PN sets the page number, which can be set to any permitted value at any point in the text. The largest number allowed is 65535.

Note: If no PN command is used, the first page will be number 1 by default and numbers will increase by 1 for each new page.

PO ON/OFF Print Odd numbered pages only.

Example: >PO ON

Description: Only pages with odd numbers will be printed when printing takes place. ’>PO OFF’ causes all pages to be printed.

PS ON/OFF * Paper Sensing on/off.

Examples: >PS ON
>PS OFF

Description: This command will only work some printers. The effect of using it is to put the printer ’off line’ at the end of a page, to allow changing of single sheet paper when background printing is in use. The chapter on ’Background printing’ gives full details. The default setting is "OFF".

Note: This command should not be used with the PCW printer.

Note: This command does not work with the Kaga Taxan, because paper sensing only works with continuous paper on this printer.

RJ ON/OFF * Right Justifying on/off.

Example: >RJ OFF

Description: This command is used to select whether right justification is carried out, when the formatting whilst printing option ’>FP’ has been selected. The default setting is "ON".

Note: Unless ’>FP’ is set to ON, this command will have no effect.

SA Start printing At page.

Example: >SA 41

Description: Sets the first page number to be printed.

(v) Headers, footers and page numbers

By default, no headers or footers are printed. A header or footer definition command is required. There are six commands which may be used to define header and footer text and each of them takes a string of text as the parameter. This is the string of text that will be printed at the top or bottom of the pages, such as a title, or a page number. The text that is printed starts in column 5, so one space must be left after the command name and the remainder of the line of text should be typed exactly as it is to appear.

'HE' and 'FO' are the two commands to define header and footer text which is to be printed on all pages. Sometimes it is necessary to print different text on the even numbered pages from the odd numbered ones and if so, '>EH' and '>EF' will define the text for even pages and '>OH' and '>OF' will define text for odd pages. With all these commands, which are described fully in 'Page formatting' above, the act of defining the text also has the effect of turning the headers or footers on.

Sometimes it will be necessary to turn headers and footers on or off at some stage in the printing. This can be done with the two stored commands, '>HE ON/OFF' and '>FO ON/OFF'. Turning headers or footers off has no effect on the header or footer text, other than the fact that it will not be printed, and they may be turned on again at any time. There is no restriction on the number of different headers and footers that may be used in a document and they may be changed at any time by using the appropriate header or footer definition command to define the new text.

Note: Header and footer commands must be inserted into the text before the start of the page to which they are to apply as they will only take effect when the next new page is printed.

Note: Since the first four columns are occupied by the stored command name, the text will be printed four columns to the left of where it appears on the screen. In order to position the header or footer correctly, relative to the rest of the text, the string should be entered before the stored command is inserted into the left hand columns. Similarly, if the text is to be centred, this should be done first, with ALT-C and then the cursor moved to column 1 and the appropriate stored command inserted.

Page numbers can be printed anywhere in the header or footer line. This is done by typing a percent symbol ('%'). The page number will then be printed starting in the column occupied by the percent symbol in the text.

By making use of odd and even headers or footers it is therefore possible to have page numbers printed on the outside edge of pages and different text on odd and even pages as well, in a form that would frequently be used for a book.

If headers or footers are turned on with the ON/OFF commands without specifying any text, page numbers will automatically be printed, centred, unless alternative text is given with the header or footer commands. Using the command with no text will suppress the page number.

(vi) Pagination

Pagination is controlled by many factors. PROTEXT is supplied with a set of default values which are suited to use with either A4 single sheet paper, or continuous 11 inch paper, but any of the parameters may be altered by use of stored commands. The values of all the margins may be changed, as well as page lengths and line spacing (see 'Paper formatting' above). PROTEXT automatically calculates the number of lines that will actually be printed on a page. There is no need to print a page to see where the page breaks will occur, as PROTEXT constantly displays particulars of the page and line numbers on the status lines at the top of the screen. The line number shown is always the number of the line as it will actually be printed, having made any allowances necessary for headers and footers.

By making use of the editing commands, ALT-(and ALT-) to move forward and backwards a page at a time it is very easy to see where page breaks are going to occur. If the position in the text is unsuitable, a page break can be forced at any time by inserting the '>PA' stored command into the text at a more suitable location. Alternatively '>OP' and '>EP' can be used to ensure that the next page to be printed has an even or an odd number.

The stored command '>SM' is used to determine how many spaces are printed at the start of every line, to ensure that the text is printed in the correct position across the paper. In addition, it is possible to specify different side margins for odd and even pages, using the '>EM' and '>OM' commands. This can be useful to ensure that a larger margin is left on odd numbered pages and a smaller one on even numbered pages, so that text is not printed too near to the fold of a book.

'>PO' may be used to ensure that only odd numbered pages are printed, in order that the paper may be turned over at the end of a print run and the even numbered pages printed on the reverse side, using the '>PE' command.

Full details of all the stored commands concerned are given in the descriptions earlier in this chapter and a detailed appendix at the end of the manual gives a full syntax description for each command.

Note: The PRINTS command will give a visual display on screen of how the printed page will appear.

Note: When full proportional printing is to be used for printing, the line and page numbers shown at the top of the page can only be used as a guide to page breaks, as proportional printing will always cause the page to be reformatted during printing and will usually print more words on a page. It is usually preferable to force page breaks when required in this way.

(vii) Messaging commands

Most of the following commands are concerned with the display of information, either in the text, or on the screen during the course of printing, either with or without a pause in the printing.

* >CO text	COmment line.
* >CS text	Clear Screen and display message.
* >DM text	Display Message.
* >ST text	STop printing and display text.
* >WM text	Write Message to output file.
* >WT text	WaiT and display text.

CO * COmment.

Example: >CO These lines do nothing except permit you
>>> to put comments and reminders in the text.

Description: This command allows the entry of notes about the document and may be used anywhere in the text. When printing is taking place all comment lines are ignored.
'>>>' is also permitted as an alternative to '>CO'.

Note: It is also possible to put comments at the end of any stored command lines, by preceding them with a semi colon (;). Anything after the semi colon will be ignored.

CS * Clear Screen.

Example: >CS Enter Information when requested.

Description: The screen is cleared and any message is then displayed on the screen.

DM * Display Message.

Example: >DM Daisy wheel will soon require changing to Courier 10

Description: The message is displayed on the screen.

ST * STop printing.

Example: >ST No further text to be printed.

Description: Printing stops immediately. Any message is displayed on the screen. Further text is not printed.

WM * Write Message to output file.

Example: >WM This message will be sent to the disc file.

Description: This command is used in conjunction with the ’>WF’ commands and is used to send messages to the output file opened with the ’>WF filename’ command. See WF in miscellaneous commands for full details.

WT * WaiT.

Example: >WT Change Daisywheel now and press any key to continue.

Description: Printing pauses and the message is displayed on the screen. Printing resumes when a key is pressed. Pressing STOP will abandon further printing and return to command mode.

(viii) Miscellaneous commands

There is one command in particular which cannot readily be classified under any of the other headings, ’EX’, and the remaining commands, ’IN’ and ’WF’ which are covered in detail in the mail merging chapters, but are included here for completeness. The mail merging chapters should be read for full details of their uses.

* >EX text	EXternal command.
* >IN filename	INsert file into text.
* >WC	Write file is Closed.
* >WF filename	Writing to File - open disc file with filename.
* >WF ON/OFF	Write to File, on and off.

EX * EXternal command.

Example: >EX GROUP 7

Description: This permits commands that are normally used from command mode to be executed during the printing of the text. This facility is provided for flexibility. The above example would change the Group number during the course of printing, enabling files stored in Group 7 to be read and printed when used in conjunction with the '>IN filename' stored command.

IN * INsert file into text.

Example: >IN filename

Description: This command may be used at any point in the text of a document and the named file is read from disc and the contents printed. When the end of the file is reached, printing of the original file continues at the line after the IN command. The file is not loaded into memory, merely printed. Any number of 'IN' commands may be issued.

Note: Drive letters may be specified preceding the filename, in the normal fashion.

Another use of 'IN' is to permit the continuous printing of a number of files as one document. A text file consisting of the following entries will, when printed, result in 'file1' being printed, followed by 'file2' and finally 'file3'.

```
>IN file1
>IN file2
>IN file3
```

One of the advantages of this is that page numbering, headers, footers, or margins set at the start of the document will follow through correctly, unless altered by any stored commands in the files being printed. In this way, it would be possible to print out a whole book, from beginning to end, in one run, with headings and page numbers running correctly through.

WC * The file being Written to is Closed.

Example:
```
>WC
```

WF * Writing to File - open file with filename.

Examples:
```
>WF filename
>WF filename A
```

Description: A file of the specified name will be opened for writing. The '>WF ON/OFF' commands determine whether the text is written to the file or not, when printing takes place. See below and mail merging for full details.
The A option causes data to be appended to the file. If no A is given a new file will be created.

WF ON/OFF * Turn on and off Writing to the File.

Examples:
```
>WF ON
>WF OFF
```

Description: When printing takes place, if writing is turned on, all text is sent to the file, including stored commands, printer control codes, rulers etc. at the same time as the document is printed normally. The file when closed is saved as Protext document and can be edited and printed as a normal document. It is principally intended for use with mail merging documents, where it provides the ability to save individual copies of any documents printed. See mail merging for more details.

(ix) Printer control commands

This group of commands is principally concerned with the sending of control codes to the printer, to change the various settings of the printer. The commands ’CC’ and ’RC’ are included mainly for completeness and the best way to alter control codes and character definitions is with the SETPRINT program (see section on Utility programs), in order to create a special printer driver.

* >CC a {n} (;{n})	Define printer Control Code.
* >OC n {n}	Output Codes to printer.
* >PR filename	Load PRinter driver of that name.
* >RC a {n}	Redefine Character.

CC — Redefine Control Code.

Examples:

```
>CC "h",27,82,6 ; 27,82,3
>CC @
```

The second example stops the reset code being printed. This should be used on a printer that is not Epson-compatible and has no reset code (or the reset code is not known).

Description: Any printer control code (@, a-z) may be defined. The first item after ’CC’ is the control code letter, followed by the sequence of codes for ’on’, followed (optionally) by a semicolon and then the codes for ’off’.

OC — Output Codes.

Examples:

```
>OC 27,"3",12      ; change line spacing
>OC 27,J,20
>OC 27,"l",5       ; set left margin (use >SM 0 as well)
```

Description: This command takes a list of codes which are sent directly to the printer. It is particularly useful for defining character matrices on the printer when a long string of codes are required. The printer driver could be used, but can only hold a limited number of characters.
The codes may be sent as decimal numbers, hex numbers (prefix with & or #) or ASCII characters. If an item is not a number it is sent as an ASCII character. To send the ASCII code of a decimal digit enclose the digit in quotes. Codes may be separated by commas or spaces.

PR Load **PR**inter driver.

Example:

```
>PR taxan.ptr
```

Description: The named printer driver is loaded. All the control code and character translations are set to the new values.

RC Redefine Character.

Examples:

```
>RC £ 27 ">" 1 27 "="
>RC "!" = "!" 8 "."
```

Description: The first item after 'RC' is the character to be redefined, this is followed by the sequence of codes to be printed whenever the character occurs. The codes may be entered in the same way as for the 'OC' command, above.

g). Printer control codes

Printer control codes are used to control the special features of a printer, such as underlining, different sizes and styles of print, subscripts, superscripts and selecting different character sets.

Each make and model of printer has its own control codes and PROTEXT needs to know what printer control codes the printer in use requires to carry out the different type styles and effects. This is done by means of 'printer drivers'.

(i) Printer drivers

PROTEXT is supplied with three printer drivers to suit a wide range of printers.

PCW.PTR - specifies the control codes for the PCW internal printer.
EPSON.PTR - specifies control codes for a standard Epson compatible printer.
JUKI6100.PTR - specifies the codes for a Juki 6100 daisy wheel printer and many other daisy wheel printers.

PROTEXT also contains the details of a 'simple printer', i.e. a printer which recognises no control codes except the backspace code and can use this to provide underlining and **bold** facilities. In the event that no printer driver has been loaded into PROTEXT, this printer driver will be used.

Many dot matrix printers use 'Epson' codes, in which case the 'EPSON.PTR' driver may be used. There is much less conformity with daisy wheel printers, but the JUKI6100.PTR driver is suitable for use with a number of printers.

Note: It should also be appreciated that many of the special printing effects that are possible on a dot matrix printer may not be available on a daisy wheel printer. It should also be noted that not all dot matrix printers have all the facilities, even if they are 'Epson' compatible.

Note: If the printer does not have compatible codes, it will be necessary to create a printer driver, containing the correct codes to suit the printer. A utility program called SETPRINT is provided for the purpose of creating printer drivers and full details will be found in the section of the manual covering 'Utilities'. See also the section on 'Printer drivers' under 'Command mode'.

When PROTEXT is loaded from the start of day disc, it will automatically load the printer driver specified in the PROTEXT.CFG configuration file. By default, on the PCW computers, it will load the PCW.PTR file and on the CPC6128 if there is no printer driver specified, it will assume the 'simple printer' driver. If for any reason the specified or the default drivers are not found, PROTEXT will automatically configure itself for a simple printer.

Different printer drivers may be loaded at any time (other than when printing is actually taking place!), which means that if both a daisy wheel and a dot matrix printer are connected to the computer, it is possible to print out a draft copy on the dot matrix printer and then do the top copy on the daisy wheel printer. Printer drivers may be loaded either from command mode using the PRINTER command (See the section on 'Printer drivers' in the Command mode chapter) or from within the document by using the stored command '>PR' (See chapter on 'Stored commands').

(ii) What are printer control codes?

PROTEXT's printer control codes are special single characters, which are embedded into the text whilst editing. These are recognised as such by PROTEXT when it is sending text to the printer, at which point it substitutes the sequence of numbers appropriate to the code and sends those to the printer instead. The result of this is that once a suitable driver has been created (if necessary), inserting a single code into the text can have the effect of sending a complete sequence of codes to the printer.

(iii) How to use printer control codes

Printer control codes may be entered anywhere in the text. This is done by typing ALT-X followed by a letter, to define the code. Any letter from 'a' to 'z' can be defined to represent any code, but as standard, several have been allocated uses covering the most common features:

@	printer reset code
b	bold (emphasised)
c	condensed print
d	double-strike
e	elite (12 cpi)
i	italics
l	enlarged
n	normal (pica) (10 cpi)
p	proportional
q	near letter quality
s	subscript
t	superscript
u	underline

When one of these codes is entered, the letter appears in the text in inverse video display. In most cases the first occurrence of the code turns on the appropriate feature and the second occurrence turns it off.

For example: To underline a single word, the cursor should be moved to the start of the word and ALT-X pressed, followed by U. The cursor should then be moved to the end of the word and the same sequence repeated.

The codes 'e' and 'n' are different, having no meaning for 'off'. They are mutually exclusive styles of print, so turning one off is done by selecting the other. On certain printers (including the Kaga Taxan) the q code also works in this way, but on most printers the NLQ print must be turned off with a second 'q' code. The default codes defined by PROTEXT in the EPSON.PTR driver cater for both of these types of printers.

Each printer code occupies one column on the screen, but PROTEXT takes this into account and when the text is printed, justification will be correct. Any tabulation or formatting will appear visually wrong when printer codes are visible on the screen, so a command is included to overcome this problem and allow the text to be viewed without the printer codes. This is ALT-V V. Pressing ALT-V V a second time restores the display of printer codes.

(iv) Character pitch

Printing takes place in one of three pitches, pica, elite or proportional and each of these is exclusive. It is not possible to have pica and elite in operation at the same time. As a result, these two control codes do not have 'off' settings, as selecting one automatically turns off the other. Proportional is slightly different in that turning it on instructs the printer to go into a different style of printing altogether.

Note: Some printers will only commence printing in the new pitch on the first new line following the use of the control code.

In addition to the above pitches, it is possible on many printers to use further control codes to implement condensed or enlarged styles of printing and with most printers this has the effect of reducing or enlarging the typeface currently in use. In other words, 'condensed elite' and 'condensed pica' are both possible and will result in different size characters.

Note: On the PCW printer it is not possible to implement 'condensed elite' and if attempted, it will result in 'condensed pica'. Many printers can however manage this style. Generally, it is necessary for elite to be selected before condensed is used.

The following list gives an indication of the the numbers of characters printed per inch when the appropriate control codes are specified. By default, PROTEXT will print in draft quality with a pitch of 10 characters per inch.

Pica. -	If pica is in use:		
n	pica	10	
l	enlarged pica	5	
c	condensed pica	17	
lc	enlarged/condensed	8.5	
Elite. -	If elite is in use:		
e	elite	12	
l	enlarged elite	6	
c	condensed elite	20	(Not available on PCW Printer)
lc	enlarged/condensed	10	

Proportional.

This figure will vary with different printers and according to the different character widths used for each character, but generally the average may be considered to be equivalent to the elite pitch of 12 characters per inch.

h). Proportional Printing and Microspacing

When printing takes place normally, each character occupies a fixed width space on the paper. When PROTEXT right-justifies a line, extra spaces are spread out along the line as evenly as possible. Proportional printing is the process where individual letters occupy different widths (so for example, 'W' takes more space than 'i'). This produces a better appearance, and when combined with right-justification gives the method used for typesetting books. It can only be used with printers which are capable of producing proportionally spaced printing.

Microspacing is a means of simulating proportional printing to a degree. Microspacing evens out the spaces between the words on a line, and so gives a better appearance to the text. It does not adjust the widths of the characters and would normally only be used with a printer that was not capable of proportional spacing. There is a cost - printing is often much slower when microspacing is being used on a dot matrix printer, and for this reason is much more suited to daisywheel printers.

(i) Proportional Printing

Proportional printing can be considered a two part process. One part, where the actual character width that will be printed is decided, is controlled by the printer. The second part, which decides how many characters it is possible to fit on a line is controlled by PROTEXT.

The first part, where the printer does all the work can be used on its own, as long as the printer has the capability of doing proportional printing and this is brought into operation either by manually changing a switch on the printer, or by sending the required printer control code at the start of the document. The printer drivers supplied with PROTEXT are set so that the printer control code 'p' will set proportional printing.

The problem with using proportional printing in this way is that even though right justification might have been specified in the document, due to the way that the printer closes up the excess spaces between narrow characters, the end result will have a ragged right margin.

The solution to this is to let PROTEXT calculate how many characters the printer is going to be able to get on a line and for it to pull words up from the following line to fill any space at the end of the line. This is done by the use of a stored command in the document.

Note: When proportional printing is being carried out in this fashion, the ruler line determines the width of the line, but in the form of a measure, rather than the number of characters, as is normally the case. Each column of the ruler may be considered to be 1/10". A 70 column wide ruler would therefore result in text 7" wide being printed (7 X 10 = 70). The ruler can be changed to 1/12" spacing by use of the stored command >CW 10. See later in chapter.

* >PP ON/OFF Turn **Proportional Printing on and off.**

Note: This command is primarily intended for use with a proportional daisywheel printer and, though it may be used with the proportional mode of a dot matrix printer, it is not really recommended as the speed of printing is dramatically reduced. It should not be used with a fixed pitch daisywheel or a non-proportional dot matrix printer.

Note: The printer must also be put into proportional printing mode, either by switching it manually on the printer, or by use of the 'p' printer control code.

It is necessary that the printer driver in use has been correctly configured and each character requires its relative width specifying in the 'Set character widths' part of SETPRINT and also that the correct 'Microspacing codes' have been set. The JUKI6100.PTR and PCW.PTR drivers supplied with PROTEXT already contain all the necessary details

Proportional printing only affects the way that text is printed, not the appearance on the screen. When '>PP ON' is used PROTEXT will then re-format the text as printing takes place, moving as many words as possible onto a single line, using the character widths contained in the current printer driver (see SETPRINT). If right-justification is on as well, then extra spaces will be added between the words, as in microspacing (below). This requires that the microspace codes have been correctly defined.

The re-formatting will typically fit in a few extra words per line, moving them as necessary from the following line, destroying the original natural pagination. If the page breaks are to be retained, then >PA should be used to force each new page break in the appropriate place.

Note: Use of the PRINTS command to print the document to the screen will show where the new natural page breaks will occur.

Note: Leading spaces in a line can cause problems with alignment of the text and this can be avoided by making full use of the tabs to create lists and to inset the start of paragraphs.

Note: To ensure correct centring of text, the 'CE' stored command should be used instead of ALT-C to centre text. In order to centre text in headers or footers, it will be necessary to calculate the number of spaces required to centre the text, as it is not possible to use >CE in a header or footer definition.

(ii) Microspacing

This is designed to be used with characters of fixed width, for example, a fixed pitch daisywheel or a non-proportional dot matrix printer.

* >MS ON/OFF	MicroSpacing on/off.
* >MC n {,n}	Define Microspace Code sequence.
* >CW n	Define microspace Character Width.

On Epson-compatible printers, using microspacing is very easy. The stored command, '>MS ON', is inserted at the start of the text and microspacing will be enabled. '>MS OFF' will turn it off, so it is possible to print a document using microspacing on only part of the text.

Note: Right justification must NOT have been disabled with the '>RJ OFF' stored command.

The above will work only with normal size print (pica). If different size characters are to be used, the CW command must be used to define the width of the microspace character. This is done with a number that is the width of the character in multiples of 1/120th of an inch. The values that should be set for the different print sizes are as follows:

Pica	12 (this is the default setting)
Elite	10
Condensed	7

If the text is being printed enlarged, then these values should be doubled. (These figures are given for Epson printers).

On other printers, PROTEXT must be given a code sequence to print a microspace. This is the sequence of codes that moves the print head by the smallest possible amount. The way this is done will vary for different printers (and it may not always be possible). As an example the following definition could be used for Epson printers (though it is not necessary since microspacing already works on Epsons):

```
>MC 27 "L" 1 0 0
>CW 12
```

The codes 27, "L" put the printer into 'double density graphics bit image mode' (!). The next two bytes are the number of bytes of data to be printed, low byte first (i.e. 1 byte of data). The last zero is the one byte of data. The effect is that the print head is moved 1/120th of an inch to the right.

The definition of the character width setting, as being in multiples of 1/120th of an inch, assumes that it is possible to move the print head by 1/120th of an inch. If the smallest possible print head movement is different from this, and the microspace code has been defined to move the printer by this amount then the definition of character width is 'The number of microspace codes that need to be printed to move the printer by the width of a character'.

Note: Once the correct values for MC and CW are known, SETPRINT should be used to store them in the printer driver (see Utilities).

3. FIND AND REPLACE

Note: CPC6128 users should note that there is some variation in the use of keys in this chapter, as the CPC6128 does not have the special ⊞ and ⊟ keys. The COPY key serves the same purpose as the ⊞ key and CTRL-@ @ is the substitute for ⊟. SHIFT-f2 and CTRL-f2 generate REPLACE and FIND respectively instead of the PCW EXCH and FIND keys.

Two functions, FIND and REPLACE are provided, which permit searching through text for any string of characters and, if specified, replacing them with a second string.

Pressing FIND or EXCH whilst in edit mode will cause PROTEXT to enter command mode, with a request for the 'String to find'. Alternatively, typing FIND or REPLACE from command mode will have the same result.

The string to find is requested first, followed by the replacement string (if the REPLACE option was selected). After entering the string or strings, one or more of a number of options may be selected by typing the appropriate letters one after another (in any order). Each option is either a single letter abbreviation or a number (these are listed on the screen). Pressing RETURN on its own will cause no options to be selected.

The options available are as follows:

G Global search. If selected the whole text is searched from the start, otherwise only the text from the current cursor position to the end of the text.

C Case specific search. If selected all letters will only match letters that are the same case, otherwise either capitals or lower case letters will be treated as being the same.

W Find string only if it appears as a complete word. For example to find occurrences of the word 'and' without finding 'hand', 'England' etc.

B Search backwards. Searches from the end of the document to the beginning.

A Find or replace all strings automatically. REPLACE will change all occurrences of the string with the new one, without requesting confirmation and return a figure of the total number of replacements made. In the case of FIND being used, it will simply return the total number of occurrences of the string.

n Find or replace the nth occurrence. n should be a number between 1 and 255. This option has a number of uses, but a simple example might be to check that every set of quotation marks has a matching closing set, in which case FIND would be used to find '"' and '2G' would be specified as options, to search globally for every second occurrence.

If no options are selected the search will be forwards, from the current cursor position to the first occurrence of the string, ignoring the case of letters, finding the string even if it occurs as part of a longer word, and asking for confirmation before replacing a string.

Any number of wildcards are allowed in the string. A wildcard is a character that matches any character in the text, except the return character. It is entered in the string by typing a question mark (?).

A tab character may be entered simply by pressing the TAB key. It is displayed as a right pointing arrow.

There are various characters that cannot be entered directly, but that it may be useful to include in a search string. These include return characters and printer control codes. Provision has been made for including these in a string, by means of an 'escape character'. The 'escape character' (!) should be typed in, followed by a symbol, number or letter, as appropriate. Thus to search for a printer control code, type '!', followed by the code letter.

The full list of characters that are entered by this means is:

printer control code	!<letter>
question mark	!?
exclamation mark	!!
hard return	!.
soft hyphen	!-
non-break hyphen	!<underline>
non-break space	!<space>
search for specified code	!<number>

a). Using FIND

Once the string and any options have been selected, edit mode is entered and the cursor placed on the first character of the first occurrence of the string. To find the next occurrence of the string, the ⊞ key, positioned to the left of the space bar, should be pressed. This need not be done immediately. Editing can be carried out first and when complete, the search may be continued by pressing ⊞. At any stage, ⊟ can be used to search back towards the beginning, if necessary.

As with other commands, FIND can be used by typing the string on the same line as the command name, followed by any options. Thus the command 'FIND word GWC' will search for the string 'word' from the start of the document, selecting only those occurrences where it is a complete word with all letters in the same case as specified. If no options are specified, the default options will be used.

If the A option is selected, PROTEXT will return the total number of occurrences found when the search is complete.

b). Using REPLACE

The cursor will be positioned on the first character of the string and a message, 'Replace (y/n)?',will be displayed on the status line. Pressing 'Y' will replace the string with the new one and the cursor will move to the next occurrence. Pressing 'N' will leave the string untouched and move the cursor to the next occurrence. Alternatively STOP may be pressed and normal editing resumed. At a later time, ⊞ may be pressed to resume the find and replace operation. Alternatively ⊟ may be used to resume the search in the reverse direction, which may be found useful if an occurrence of the string is passed over by pressing 'N' in error.

If option A is selected then all occurrences of the string are replaced without prompting and the program will remain in command mode. When complete a count of the total number of changes made will be displayed.

Examples

1. To find all occurrences of the word 'text' in lower case only, starting at the cursor position.

```
FIND string: text
Options: CW
```

2. To convert all occurrences of 'rom' or 'Rom' to 'ROM', asking for confirmation of each replacement.

```
FIND string: rom
REPLACE with: ROM
Options: GW
```

3 To find the 12th 8 letter word.

```
FIND string: ????????
Options: WG12
```

4. To replace bold codes, changing the on and off codes to different typesetting commands. Note that the quotes are needed in the first command because the comma would otherwise mark the end of the replace string.

```
REPLACE !b '<nr5><ps10,12>' AG2
REPLACE !b <nr6><ps12> AG
```

A useful tip

The find and replace commands open up a whole range of ways of making word processing easier. It is not possible to list them all here, but one example should give some idea of the possibilities.

Suppose a letter is being written in which the words 'word processor' appear many times. A lot of typing time can be saved by typing a single character (say '@') for each occurrence of 'word processor' and then the REPLACE command used when the document is finished, as follows:

```
FIND string: @
REPLACE with: word processor
OPTIONS: AG
```

Use FORMAT, from command mode, to re-format the letter and the job will be complete even though 'word processor' was only typed once. This is an alternative to using the 'Phrase' functions (See Word 5).

4. COMMAND MODE

All entry of text is carried out in Edit mode, but in order to carry out operations such as saving, loading or printing, 'command' mode must be entered. This can be done at any stage of editing simply by pressing STOP. Pressing STOP a second time will return to edit mode.

When STOP is pressed, the bottom part of the screen will be cleared and the command mode banner line will appear, displaying the PROTEXT version number. The cursor will be positioned next to a '>' symbol. This symbol is the 'command prompt' and indicates that commands may be entered. The currently selected drive is indicated by the letter prefixing the > and if any 'group' other than group 0 is selected, this number will also be indicated.

The output of all commands will be displayed in this window at the bottom of the screen. Many commands produce more output than will fit in the window in which case the screen will automatically scroll as necessary.

Command mode HELP is available at any time. Typing HELP will give a list of the available subjects, such as FILES, PRINT, DISC, and EXTERNAL. Typing HELP, followed by the subject will list all the commands relevant to the subject. For example: HELP DISC, (or H D) will list all disc utility commands.

a). General information

Before studying the individual commands in detail, there are a number of points connected with the entry of commands which are of general interest and are listed below.

b). Command entry

PROTEXT has a special feature which permits the entry of commands in a simplified fashion. For example, to save a text file it is only necessary to type 'SAVE' and PROTEXT will prompt with 'SAVE filename:' and wait for entry of a name for the text file.

Alternatively, the parameters of a command may be entered on the same line as the command name, e.g. 'LOAD report', 'SAVE letter'. In this way the commands may be used without the prompts for the parameters appearing, which is often more convenient when familiar with the syntax of the commands.

Note: All commands which require a parameter will prompt for them if the command is used on its own. Commands which have optional parameters require these to be entered at the same time as the command.

PROTEXT provides a sophisticated line editing facility which is in operation whenever commands are being typed in. If a mistake is made the cursor can be moved back and the mistake corrected in the same way as in edit mode. The following editing commands are available in command mode:

PCW8256/8512	CPC6128	
←	←	Move cursor left one character.
→	→	Move cursor right one character.
ALT-←	CTRL-←	Move to start of line.
ALT-→	CTRL-→	Move to end of line.
DEL→	CLR	Delete at cursor.
←DEL	DEL	Delete before cursor.
ALT-A	CTRL-A	Alternate characters.
ALT-←DEL		Delete to beginning of line.
ALT-DEL→	CTRL-E	Delete to end of line.
ALT-TAB	CTRL-TAB	Switch between insert and overwrite modes.
CAN	CLR	Clear screen (if cursor at the start of a line)
STOP	ESC	Abandon entry of current command.

Pressing COPY or ⊞ when the cursor is at the start of a line recalls the last command line used that was 4 or more characters in length, and positions the cursor at the end of the command. This has a number of uses, such as carrying out multiple saves of the same file, or repeating a load command which failed because the wrong disc had been inserted. Short commands such as 'A', 'CAT', 'CO' do not affect the command recalled.

In addition, PROTEXT provides a 'copy cursor' facility when in command mode. If the SHIFT key is held down and one of the cursor keys used, a second cursor will appear and move according to the cursor keys. If this is positioned over a piece of text, the SHIFT key released and the ⊞ or COPY key pressed, the characters which are underneath the 'copy cursor' will be copied down to where the original cursor is positioned. Text may be copied from anywhere on the screen. This is a very convenient method of recalling commands which have been used previously, but can not be recalled by the method described in the previous paragraph because subsequent commands have been issued.

c). Abbreviations

Many of the commands can be abbreviated. For example, there is no need to type 'LOAD' in full, typing 'L' will serve the same purpose. Similarly 'S' for 'SAVE' and 'P' for 'PRINT'. A full list of the commands, abbreviations and their parameter syntax, is given later in this chapter and in Appendix 1.

d). The current filename

After a file has been loaded, or once a piece of text has been saved, the name of the file will be displayed on the status line. This becomes the 'current filename' and is remembered by PROTEXT until changed, either by saving with another name, by use of the NAME command, or by loading a new file. Once a file possesses a current filename the name may be omitted when saving a file. Entering the SAVE command, and just pressing RETURN when the prompt 'SAVE filename:' appears, will save the file with the current filename. Care must be taken to ensure that it is indeed the correct name, to avoid accidentally erasing something else. If SAVE is typed and the name displayed is incorrect it can be edited as described above ('Command entry').

e). Commands

The remainder of this chapter gives full details of the commands available in command mode. Details of the syntax used and what the command does are given together with any optional extensions to the basic command. They are split into groups in a similar fashion to the way they are broken down in the HELP summary, but for simplicity of explaining their function, they are further split down into smaller groups of connected or similar commands in this chapter.

Many of the commands allow the use of ambiguous filenames. An ambiguous filename is one which contains 'wildcards'. PROTEXT has two types of wildcards, which may be used in the same way as with CP/M commands.

?	may be used to mean 'any single character'.
*	may be used to indicate 'any number of characters'.

For example:

DATA?.TXT	- All filenames beginning with 'DATA' and having one further character (which may be blank), with the suffix 'TXT'.
B*.*	- All filenames beginning with 'B', of any length and any suffix.
.	- All files.

Note: Only one '*' may be used in each part of the filename and suffix.

The main sub headings are:

File handling.	- Details of file saving, loading etc.
Printing.	- Printing and printer control options.
Disc management.	- Drive selection and file movement etc.
Miscellaneous.	- Miscellaneous commands.
External commands.	- Additional programs called from PROTEXT.

f). File handling

Note: Letters in brackets, following a command name, are abbreviations or alternatives to the command and may be used instead. Extensions are optional parameters which may be specified to perform variations on the simple command.

This section can be divided into commands which affect the type of file to be created and commands to load, merge and save documents.

When any file is loaded into PROTEXT, document mode will be entered by default, unless the file was originally saved as a program file from program mode, in which case PROTEXT will be in program mode. This may be changed by use of the CONFIG utility, so that program mode is entered by default, unless the file was originally saved as a document.

DOC

Description: Selects document mode. This is the normal mode used when creating documents with PROTEXT. When PROTEXT is loaded initially, it is the mode selected. When document mode is in use, all stored commands, ruler lines and printer control codes are obeyed at the time of printing or formatting.

PROG

Description: Selects program mode. When program mode is in operation, any stored commands and ruler lines are printed and NOT obeyed. This provides a means of printing out a text file complete with stored commands, rather than them being acted on. Tabs are set at every eighth column and there is no right margin or formatting. See chapter on 'Program mode'.

Note: Program mode is particularly useful for writing source code for assemblers and compilers, and for editing data files.

LOAD (L)

Syntax: LOAD <filename>

Description: A document will be loaded into memory from a disc file of the specified name. A warning message will be given if the text currently in memory has not been saved. Press 'Y' to confirm that this text is to be discarded.

Note: If only the command name is entered, PROTEXT will prompt for a filename. Once loaded, the specified filename will become the 'current filename'

MERGE (MER)

Syntax: MERGE <filename>

Description: This is similar to LOAD but whereas LOAD clears any existing text from memory and then loads the file in, MERGE inserts the new file into the existing text at the current cursor position.

Note: Care should be taken to ensure that the cursor is in the required position before using this command.

Note: The current filename is NOT changed.

NAME (N)

Syntax: NAME <filename>

Description: Permits the name of the document in memory to be changed. The new name becomes the 'current filename'.

PRINTF (PF)

Syntax: PRINTF <outfile>

Description: The text in memory will be sent to a file on disc with the specified name, instead of to the printer. Tabs are expanded to spaces and page breaks and margins are shown as spaces. Each line is terminated with a carriage return. If these are not required, the stored command '>ZM' should be inserted at the start of the file, to set all margins to zero. The file will be a 'pure' Ascii file, suitable for use by other programs, or for transmission by 'Electronic Mail'.

Extensions: An input filename

Syntax: PF <outfile> <infile>

Description: The specified input file will be read from disc, converted and written back onto disc with the output filename as an Ascii file. This option permits the easy conversion of a PROTEXT file into an Ascii file.

Note: The file is not loaded into PROTEXT, merely read from one file and written to another, so the contents of any text in memory will not be lost.

Note: If no filename is specified, then both an input and an output filename are prompted for. If RETURN is pressed in response to the 'Input filename' prompt, the text in memory will be sent to the file.

Note: Filenames may be prefixed with the drive letter in the normal fashion. For example: 'PF B:outfile A:infile' will create an ASCII file called 'outfile' on drive B from a file on drive A called 'infile'.

PRINTFB (PFB)

Syntax: PFB <outfile>

Description: Only the section of text defined by the block markers will be sent to the file, but all stored commands in the text are taken into account.

SAVE (S)

Syntax: SAVE <filename>

Description: The complete document in memory will be saved to a disc file with the name specified.

Note: If only the command name is entered, PROTEXT will prompt for a filename. If the file already has a 'current filename', then pressing RETURN will result in the file being saved with the same name. Alternatively, a new name may be specified, which will then become the current filename.

SAVEB (SB)

Syntax: SB <filename>

Description: This is the same as SAVE except that only the text within the block defined by the block markers is saved.

Note: The current filename is NOT changed.

SPOOL (SPON)

Syntax: SPOOL <filename>

Description: All output to the screen will also be sent to a file on disc with the specified name until the file is closed with the SPOOLOFF command.

SPOOLOFF (SPOFF)

Description: Cancels the echoing of all screen output to a file, having first closed the file.

SWAP (SW)

Description: Swaps between two documents in memory. All settings of the files and cursor, block markers etc are retained. See 'Two file editing' for full details.

TYPE (T)

Syntax: TYPE <filename>

Description: Used to 'type' the contents of a PROTEXT or ASCII file to the screen. The file is not loaded into memory, merely the contents displayed on the screen. All stored commands and rulers will be displayed and not acted on. This can provide a convenient means of viewing the contents of a file without loading it into memory. Whilst the file is being typed pressing STOP will pause the display. Pressing STOP a second time will cancel the command and any other key will resume.

Note: In some respects TYPE may be thought of as duplicating the PRINTS command, but this is not really true, the difference being that TYPE just lists the contents of a file on the screen, whereas PRINTS displays on the screen what would normally be sent to the printer. TYPE will display all stored commands and control codes.

Note: Printer control codes may appear as different characters from when viewed in PROTEXT. This is because PROTEXT interprets them before display, whereas TYPE merely displays them.

g). Printing

Commands in this group have been split up into those which control the printer and those which carry out the printing. In this context, printing does not only mean printing text on a printer, but includes printing text on the screen and to a disc file as well, which is essentially the same process, but with the output sent to a different device.

(i) Printer control

These commands control which of the possible printer ports are used when printing takes place. If an Amstrad CPS8256 RS232/parallel interface (or equivalent) has been installed on the PCW, then printed output can readily be redirected to either the PCW printer, a serial or a parallel interfaced printer, simply be issuing the appropriate command. This makes it particularly easy to print out a draft copy on the dot matrix printer and a final copy on a daisywheel printer connected to one of the other ports.

INTERNAL (INT)

Description: Resets the printer output to the normal printer supplied with the PCW range. The printer driver PCW.PTR is loaded, if present on one of the discs, otherwise 'simple printer' is selected. It is ignored on the CPC6128.

PARALLEL (PAR)

Description: Selects the parallel (Centronics) printer port for the output of all printing, and selects the simple printer driver.

PRINTER (PR)

Syntax: PRINTER <filename>

Description: Loads a printer driver (see below) to configure PROTEXT for use with another printer. The '.PTR' suffix in the filename need not be typed. This may cause either parallel or serial output to be selected, depending on the printer driver contents. Printer drivers are created to suit different printers by using the SETPRINT utility program (see 'Utilities').

SERIAL (SER)

Description: Redirects all printed output to the serial interface, for use with a serial interfaced printer, and selects the simple printer driver.

(ii) Printer drivers

A printer driver is a collection of codes that a particular printer uses, including the codes for underlining, bold, etc. With one exception ('the 'simple printer driver') all printer drivers are stored as files with the suffix '.PTR'. There are four printer drivers supplied with PROTEXT.

The first is known as the 'simple printer'. This is built in to the program and is designed to work with virtually all printers, since it uses no special codes, but just assumes that the printer is capable of backspacing. The simple printer driver only supports underlining and boldface print. This is used by default on the CPC6128.

The second is for the PCW printer. This is a file called 'PCW.PTR' and is loaded automatically when PROTEXT is used on the PCW. If the file is not found, the simple printer driver is used.

The third printer driver is for Epson and Epson-compatible printers. It is called EPSON.PTR. Users of Epson or Epson-compatible printers should use CONFIG to select this as the default printer driver to be loaded automatically when PROTEXT is used, because it provides more facilities than the simple printer driver (see 'Utilities').

The fourth driver is called JUKI6100.PTR and is for use with the Juki 6100 daisy wheel printer and a number of other printers which use the same codes. If this printer is being used, then CONFIG should be used to select this as the default printer driver (see 'Utilities').

Printers which are not 'Epson' or 'Juki 6100' compatible will require their own printer driver, which can be created with the SETPRINT utility (See Utility 3) and CONFIG (See Utility 2) should be used to select the new printer driver as the default driver.

Controlling the standard printer (PCW8256/8512 only)

The printer controls may be accessed at any time by pressing the PTR key. The information appears at the bottom of the screen in the normal fashion. The PCW8256/8512 operating instructions should be referred to for details of how to use the printer controls.

(iii) Printing options

The following commands determine the form that printing will take and include options to print to a disc file and to print to the screen, as well as to send output to the printer in the normal fashion. Options are available which allow printing in NLQ (Near Letter Quality) mode as well as draft (though all styles and types of printing can be controlled through printer control codes in the text as well). Many of the options allow the number of copies to be printed to be specified as a parameter. Further variations permit the printing of selected pages only and most of the options permit the printing of files directly from disc, without the need to load them into PROTEXT. This last option means that normal editing can continue whilst printing of the disc file takes place as a background task.

Note: The chapters 'Background printing' and 'Summary of printing facilities' give more information about printing.

Note: Any commands which refer to NLQ will only work correctly if the printer is capable of printing in NLQ mode and the codes are set correctly in the printer driver (See SETPRINT, Utility 3).

PRINT (P)

Syntax: PRINT (num)

Description: This command permits printing of the document in memory. By default, the printing will be draft quality unless the appropriate NLQ printer control codes have been used in the text. All stored commands will be obeyed if document mode is being used, but will be printed if program mode has been selected. Optionally the number of copies to be printed may be specified.

Extensions: A filename.

Syntax: P <infile> (num)

Description: Optionally a filename may be specified after the PRINT command, in which case printing will take place directly from the file of that name on disc and a number may be specified for the number of copies to be printed. It will not be loaded into PROTEXT and will not corrupt any text in memory.

PRINTB (PB)

Description: Only the section of text defined by the block markers will be printed, but stored commands in the text are taken into account.

PRINTF (PF)

Syntax: PF <outfile>

Description: See the FILES section above for description.

PRINTFB (PFB)

Syntax: PFB <outfile>

Description: See the FILES section above for description.

PRINTP (PP)

Syntax: PP (num)

Description: Prints text, but permits pages to be skipped. Optionally, the number of copies may be specified. See 'Background printing' for full details.

Extensions: A filename.

Syntax: PRINTP <infile> (num)

Description: Prints text from a file on disc, but permits pages to be skipped. Optionally the number of copies to be printed may be specified.

PRINTPQ (PPQ)

Syntax: PPQ (num)

Description: As PRINTP, but printing takes place in Near Letter Quality mode, permitting pages to be skipped. Optionally the number of copies to be printed may be specified.

Extensions: A filename

Syntax: PPQ <filename> (num)

Description: As above, but prints text in Near Letter Quality mode from a file on disc. Optionally the number of copies to be printed may be specified.

Note: This command will work correctly with the standard PCW printer and Epson printers, but if other printers which do not use the standard printer control codes are used, it will be necessary to use the SETPRINT utility program to reconfigure the printer control codes, before this command can be used. See ’Utilities’ for full details of SETPRINT.

PRINTQ (PQ)

Syntax: PQ (num)

Description: The effect of PRINTQ is to cause printing to take place in ’Near Letter Quality’ (NLQ), without the need to alter SETPRINT, or to insert the NLQ control code into the text. Otherwise it is exactly the same as PRINT. Optionally the number of copies to be printed may be specified.

Note: This command will work correctly with the standard PCW printer and Epson printers, but if other printers, which do not use the standard printer control codes are used, it will be necessary to use the SETPRINT utility program to reconfigure the printer control codes, before this command can be used. See ’Utilities’ for full details of SETPRINT.

Extensions: An input filename.

Syntax: PQ <infile> (num)

Description: Printing will take place directly from the file of that name on disc. It will not be loaded into PROTEXT and will not corrupt any text in memory. Optionally the number of copies to be printed may be specified.

PRINTQB (PQB)

Description: Serves the same purpose as ’PRINTQ’, but only on a block of text selected by use of the block markers.

PRINTS (PS)

Description: The file in memory will be printed to the screen in the same format that it would be printed on the printer. Stored commands are obeyed. This option provides an easy way to see how page breaks, headers and footers etc. will appear.

Note: It is also particularly useful when used in conjunction with a mail merging file, to ensure that the expected results are being obtained, before printing.

Extensions: An input filename.

Syntax: PS <infile>

Description: This command allows printing to the screen directly from a file. The file will be printed directly from disc. It will not be loaded into PROTEXT and will not corrupt any text in memory. It provides a convenient method of reviewing the contents of a disc file without the need to load it into PROTEXT.

Note: Pressing STOP once will pause printing. Pressing STOP a second time will abort further printing, but pressing any other key will continue printing. If printing is aborted by pressing STOP twice during a PRINTS of a file in memory, on returning to edit mode the cursor will be positioned at the point that had been reached in the text. This makes it easy to correct mistakes noticed whilst printing to the screen.

PRINTSB (PSB)

Description: Only the section of text defined by the block markers will be printed to the screen, but all stored commands in the text are taken into account.

PRINTPS (PPS)

Description: This option is similar to the PRINTS option described above, the only difference being that the option to view each page, or skip pages is available, as in the PRINTP option.

(iv) Printer buffer control

PROTEXT provides a printer buffer, so that printing of a document can be carried out at the same time as editing of another document. The following commands control the printer buffer and allow background printing to be stopped, abandoned or continued at any time.

Note: Background printing can either be disabled or made optional by use of the CONFIG utility. With single drive CPC6128 computers or printers which contain their own large buffers, this may be desirable.

ABANDON (AB)

Description: Abandons printing and discards the contents of the printer buffer completely.

Note: After a STOP command, the buffer will still contain any unprinted text and ABANDON should always be used if the remainder of the text is not to be printed.

Note: Some printers contain their own internal buffers, in which case printing may continue after the ABANDON command has been issued. The printer manual should be consulted about how to empty its own buffer. Frequently this will consist of nothing more than switching the printer off.

Note: To empty the buffer on the PCW printer, the PTR key should be pressed and the cursor moved to the RESET position and the ⊞ key pressed, before using EXIT to return to PROTEXT.

CONT (CO)

Description: Resumes printing of the contents of the buffer following the use of STOP to pause printing.

STOP (ST)

Description: Causes printing to be stopped, but retains the contents of the printer buffer. Printing can be resumed with the CONT command, or abandoned completely with the ABANDON command.

Note: If further printing is not going to be carried out, the ABANDON command must be used, in order to empty the printer buffer, otherwise when further printing is carried out, the remaining contents of the buffer will be printed first.

(v) Sundry printing commands

PRINTON and PRINTOFF provide the facility to echo all screen output to the printer, whether generated by one of the print commands or any other command. TYPEWRITER provides the facility to use the printer directly from the keyboard in a similar fashion to a typewriter.

PRINTON (PRON)

Description: All output to the screen will also be echoed to the printer, after this command has been used, until PRINTOFF is used to cancel it. One particular use is to provide a printed copy of a disc catalogue.

PRINTOFF (PROFF)

Description: Cancels the echoing of screen output to the printer, which has previously been initiated by use of the PRINTON command.

TYPEWRITER (TW)

Syntax: TW

Description: This command puts the computer into direct typing mode. Everything that is typed is printed as soon as the RETURN key is pressed. Normal edit mode commands may be used to edit the text prior to pressing RETURN. Pressing STOP returns to PROTEXT command mode.
There are a number of possible uses for this command, including the easy addressing of envelopes, or adding postscripts to the end of letters.
If Near Letter Quality or any font other than normal draft mode is required, the printer must be set up correctly before entering TW. If a document has just been printed in NLQ, then TW will continue in NLQ. Alternatively, many printers have methods of easily switching into NLQ.

Extensions: A number equivalent to the number of spaces that the left margin should be inset.

Syntax: TW <number>

Note: This command defaults to zero margin, meaning that if text is normally printed with a side margin of 5, then it will be necessary to specify 5 when entering TW, to maintain the same left margin.

Note: It is not possible to use this command whilst background printing is in operation, for obvious reasons!

h). Disc management

These commands are principally concerned with disc management and cover such things as drive selection, file protection, movement and copying of files from one disc to another, as well as utilities to format and copy discs from within PROTEXT.

(i) Drive selection and cataloguing

These commands control selection of the disc drive and group numbers and the cataloguing of files on any drive or group.

A: (A)

Description: Select drive A as the currently selected drive. Optionally, the colon may be omitted.

B: (B)
C: (C)
D: (D)

Description: Select drive B, C, or D as the currently selected drive. Optionally, the colon may be omitted. Only valid on machines with the requested drive fitted.

M: (M)

Description: Select drive M as the currently selected drive. Optionally, the colon may be omitted. Only valid on the PCW computers.

DRIVE (DR)

Syntax: DRIVE <drive letter>

Description: Selects the specified drive. This command will accept drives between A and P, and an error message will be given if the requested drive does not exist, or if it does exist but there is no disc in the drive.

Note: If any special drives are installed, such as a hard disc, which use a drive letter other than A, B, C, D, or M, then this command may be used to select the drive.

GROUP (USER)

Syntax: GROUP <number> or USER <number>

Description: Selects the specified group number as the one which will be used by CAT, LOAD, SAVE etc. Alternatively, it is possible to use 'GR <number>' or 'U <number>', for convenience.

CAT (DIR)

Description: Performs a catalogue of the files on a disc. By default, with no parameters, it will catalogue all the files on the currently selected group of the currently selected drive.

Extensions: Filenames, drive letters and group numbers.

Syntax:
CAT <ambiguous filename>
CAT <drive letter>
CAT <group/user number>

Description: Either another group OR another drive may be specified. Alternatively a filename may be specified using wildcards, optionally with a drive letter prefix.
Example: CAT B:*.LTR will catalogue all the files with a LTR suffix on drive B and group 0.
The files are listed in alphabetical order with the size of each file shown. The amount of free disc space is also shown. If this last figure becomes too small it will be often be necessary to erase backup files in order to save a file. The catalogue also displays a symbol by certain files:

* indicates a protected file (see PROTECT, below)
+ indicates a program file (saved from program mode)

(ii) Disc copying and formatting commands

The following commands enable the formatting of discs whilst in PROTEXT, eliminating the possibility of having a full disc and not having a formatted disc to save a text file onto. All single sided discs may be copied onto another disc. The contents of any files in memory are not lost when these options are in use.

DCOPY

Description: Calls a PROTEXT utility program which copies the contents of one disc onto another. This command will copy single sided single density (CF2) discs only. See 'Utility Programs' for full description.

Note: The original contents of the disc to which the files are being copied will be erased.

Note: In order to copy CF2DD double sided discs, as used in drive B on the PCW8512, it is necessary to leave PROTEXT and use the DISCKIT program which is on the System Utilities disc supplied with the computer.

DFORM

Description: Formats a disc to either CF2 or CF2DD format, depending on which drive is selected. On the PCW computer a disc in drive B will be formatted to CF2DD format and a disc in drive A to CF2 format. On the CPC6128 data format is always used.

Note: Both sides of a CF2DD disc are formatted at the same time, but when formatting a disc in A to CF2 format, it is necessary to format each side separately.

DFORMD

Description: This command will format a disc as CPC6128 Data format.

Note: PCW users should use this command if the disc will also be used on a CPC6128. On the CPC6128 this command has exactly the same effect as DFORM.

(iii) File manipulation

The following commands allow the copying, renaming and erasing of files on any of the valid drives. In many cases wildcards may be used, enabling the transfer of several files at a time from one disc to another.

COPY

Syntax:

```
COPY <oldname> <newname>
COPY <ambiguous filename> (<group>) (<drive>)
```

Description: There are two variations of this command, the first of which will copy a file giving it a new name. The filenames may be prefixed with the drive letter to copy a file from one drive to another with a different name.
The second variation permits the use of 'wildcards', but the names cannot be changed. This allows the transfer of one or a number of files with some common feature, from the current group on any drive to any group on any drive. Either <group> or <drive> or both may be specified.

Examples:

```
COPY B:OLDNAME NEWNAME
COPY *.TXT 1
```

Copies all files with suffix 'TXT' into group 1.

```
COPY B:*.* 2M
```

Copies all files on drive B (current group) to drive M group 2.

Note: Any existing file with the same name, in the destination drive/group, will be renamed with a '.BAK' suffix.

Note: Copying does not erase the original files, so if they are no longer required, ERASE must be used after the copying process.

ERACOPY (ECOPY)

Syntax:

```
ECOPY <oldname> <newname>
ECOPY <ambiguous filename> (<group>) (<drive>)
```

Description: This is the same as COPY in every respect but one - if a file already exists with the same name as the file being copied, this file is erased before copying, whereas COPY renames this file as a backup file.

Example:

```
ECOPY B:*.* A
```

ERASE (ERA)

Syntax: ERA <ambiguous filename>

Description: All files which meet the criteria of the filename will be erased. Wildcards are permitted and the drive letter may be specified as a prefix to the filename.

Note: This is a potentially destructive command and should be used only with care. One very useful version is to use ERASE *.BAK to erase all back up files from the disc in the current drive. ALT-f7 on the PCW (and CTRL-f9 on the CPC) will perform 'ERA *.BAK'.

RENAME (REN)

Syntax: RENAME <newname> <oldname>

Description: This command renames files on a disc. It does not move or change the file, merely renames it.

Note: If a file requires moving to another disc and renaming, the COPY command should be used and then the original file erased with ERASE.

(iv) File protection

Files may be protected from erasure by use of the PROTECT command. ACCESS is the reverse process, making it possible to erase previously protected files.

ACCESS (ACC)

Syntax: ACCESS <ambiguous filename>

Description: Sets the status of a file or files to ’Read-write’. Wildcards are permitted. See PROTECT for details of the reverse operation.

PROTECT (PROT)

Syntax: PROTECT <ambiguous filename>

Description: Sets the status of a file or files to ’Read-only’. Wildcards are permitted. Files which have read only status can not be overwritten by subsequent files of the same name. An error message will be given if an attempt is made to do so. Protected files are indicated in the catalogue by an asterisk following the filename. See ACCESS for details of the reverse operation.

Note: PROTECT cannot stop files being erased if the disc is reformatted, or a complete disc is copied onto the disc, either with DISCKIT or the DCOPY command.

i). Miscellaneous commands

(i) Text manipulation and formatting

Two commands are provided to permit flexible finding and replacing of parts of the text, two to 'fix' text and two commands provide fast re-formatting of part or all of the text. A further two permit the insertion and removal of line numbers. One command is for the purpose of removing any text from memory.

CLEAR

Description: Clears the text currently in memory. A request for confirmation is made before this is done. On the PCW computer the same effect is achieved by typing ALT-SHIFT-CAN in edit mode.

EXEC (X)

Syntax: EXEC <filename>

Description: This command causes the contents of the specified file to be treated as if they were being typed in at the keyboard (See Chapter on EXEC Files for full details).

FIND (F)

Syntax: FIND <text> (<parameters>)

Description: The document will be searched for the first occurrence of the specified text, according to any parameters specified and the cursor positioned on the first character.
See chapter on Find and Replace for full details.

REPLACE (R)

Syntax: R <text> <newtext> (<parameters>)

Description: The document will be searched for the first occurrence of the specified text, according to any parameters specified, and the cursor positioned on the first character.

FIX

Description: When this command is used, all tabs and soft spaces are converted into hard spaces. Soft returns are converted into hard returns.

FIXB

Description: As FIX, but only affects the currently defined block. This command is particularly useful for 'fixing' sections of text that it is important will not be disturbed when formatting takes place. Other uses for it are in conjunction with BOX moves, where sometimes the inclusion of tabs can cause problems with the layout when a block is moved. If the section of text is 'fixed' first, all tabs are removed. A third use for this command is in conjunction with text which is to be typeset and which it is important retains the precise layout.

FORMAT (FT)

Description: This command will reformat the entire document according to the margins, rulers, tabs and stored commands within the document.

FORMATB (FTB)

Description: As FORMAT, but only affects the currently defined block.

NUMBER (NUM)

Description: The purpose of this command is to add line numbers to, or remove line numbers from, the beginning of every line of text. This command will prompt for whether numbers are to be added or removed from the document. If the choice to add line numbers is selected, a starting line number and the value by which each subsequent number is to be incremented will be requested.

This provides a convenient method of writing BASIC programs, amongst other uses, using the editing facilities of PROTEXT, and finally adding the line numbers.

NUMBERB (NUMB)

Description: This command is similar to NUMBER, but only adds or removes numbers within the marked block.

(ii) Phrase and symbol commands

LPHRASES (LP)

Description: Lists all the currently defined phrases between A and Z. See the chapter on 'Phrases' for full details.

PHRASE (KEY)

Syntax: KEY <letter> <phrase>

Description: Following the command should be a key letter (A-Z) which is the key that will be used with the EXTRA key, to recall the phrase. This should be followed by the phrase, which may consist of simple text or may be a combination of text, commands and Escape codes. See the chapter on 'Phrases' for full details.

SYMBOL (SYM)

Syntax: SYMBOL <code> <n1> <n2> <n3> <n4> <n5> <n6> <n7> <n8>

Description: This command redefines the symbol that is shown on screen for a particular character. <code> is the ASCII code of the character, and <n1> to <n8> are 8 single byte numbers defining the character matrix, starting at the top row and moving down.

Note: This command would normally be used as part of an EXEC file. See the chapter on EXEC Files for full details.

(iii) Miscellany

The remaining commands cannot readily be categorised and cover a range of functions.

CALC (CA)

Description: A simple numeric calculator. A prompt for an expression will be issued. Operators +, -, *, / are valid and calculation is from left to right. Brackets to force priority are NOT supported and calculations are made from left to right, not in any operator priority.

Note: The result of a calculation can be inserted into the text at the cursor location by pressing space when requested.

COUNT

Description: This command will count the number of words in the text, excluding stored command lines and their contents.

COUNTB

Description: As COUNT, but only counts words in the section of text defined by the block markers.

GOTO (G)

Syntax: GOTO (option) <number>

Description: Moves the editing cursor to the specified line, column or page in the text. The number should be prefixed by the letter 'P' for page, 'L' for line or 'C' for column. If the letter is omitted, the default is line.

Note: If line is selected, the line is the 'absolute' line in the text, not the 'page' line number, which ignores stored command lines.

Note: If the page option is selected and PROTEXT is not in page mode, the cursor will not be moved.

HELP (H)

Syntax: HELP
HELP <subheading>

Description: The command, 'HELP', used on its own will produce a list of the subheadings in which the commands are grouped. Entered with the appropriate subheading (subheadings may also be abbreviated to a single letter), it will list all commands relating to that subject, together with any abbreviations. The file COMMAND.HLP must be present in order to use HELP with a subheading.

PAUSE

Syntax: PAUSE

Description: This command is primarily intended for use in an EXEC file. See chapter on 'EXEC files'.

QUIT (Q)

Description: Quits PROTEXT and returns to CP/M command mode. If a document is in memory and any changes have been made to it since it was loaded or last saved, a caution will be issued, warning that the document has not been saved and asking for confirmation of the desire to continue.

j). External commands

External commands call other utility programs from disc. The program files specified must be available at the time the command is used. They may be on any disc drive - PROTEXT will search all drives to find the file. The following utility programs are designed so that on completion of their task, a return is made to PROTEXT with any text that was in memory at the time the command was called still intact.

Note: It is possible to call other programs from within PROTEXT command mode and they should be prefixed with an asterisk (*). When this is done, the text in memory will be saved to a temporary fiie and the name passed as a parameter to the program being called. For example, typing '*BCPL' will call the compiler called BCPL (i.e. the program file BCPL.COM), which will then compile the source code which was in memory. Unless these programs have been written specifically for the purpose, they will not return to PROTEXT automatically and this must be done by typing 'PROTEXT' from CP/M command mode. The temporary file will automatically be loaded back into PROTEXT.

CONFIG

Description: This command calls the PROTEXT configuration utility program CONFIG.COM. This allows many of the default settings of PROTEXT to be altered to suit the user. The file, CONFIG.COM must be available. See 'Utility programs' for full details.

CONVERT (CV)

Description: Calls a conversion utility program for use with non-PROTEXT files, providing facilities to modify these files in a number of ways, as well as to convert them into PROTEXT documents if required. The file CONVERT.COM must be available. See 'Utility programs' for full details.

SETPRINT (SP)

Description: A utility program to create suitable printer drivers to enable any printer to be used with PROTEXT. The file, SETPRINT.COM must be available. See 'Utility programs' for full details.

SM

Description: Calls up the main spelling checker menu, enabling use of all the options, unlike the SPELL command, which simply does a 'Single Pass Check' of the text. On completion of the check, normal operation of PROTEXT will resume. See the section 'Spelling Checker' for full details.

Note: The SPELL.COM utility program must be available.

SPELL

Syntax:
SPELL
SPELL <filename>

Description: Spell carries out a single pass check on the document in memory unless the optional filename is specified, in which case the file specified will be checked from disc. See the section 'Spelling checker' for full details.

Note: The SPELL.COM utility program must be available.

5. PHRASES AND FUNCTION KEYS

Note: The CPC6128 does not have an EXTRA key, or any equivalent key. CTRL-0 (zero) serves the same purpose as the EXTRA key. It should, however, be noted that CTRL-0 is not available in command mode and any phrases which may be required from command mode should be allocated to keys which are duplicated by the function keys, which may be used from command mode.

Phrases are sections of text which can be stored and used at any time with a single key press. The keys used to recall phrases are the keys marked 'A to Z' on the main keyboard when used in conjunction with the EXTRA key. Function keys are essentially the same, but use the special function keys on their own and in conjunction with the ALT, SHIFT and EXTRA keys.

PROTEXT has 31 expansion tokens which by default are allocated to the keys EXTRA-A to EXTRA-Z and a number of other keys. Several of these tokens are also allocated to the function keys, duplicating a number of the letters. A number of these tokens are pre-defined and cannot be changed, leaving 26 tokens which are user definable. By default, many of these tokens are pre-defined to give a variety of European characters, such as 'C, cedilla' and 'AE diphthong', but may be redefined by the user if not required for that purpose.

In theory, each of these tokens can be allocated a string of text or codes up to 255 characters long, but by default PROTEXT has a buffer of about 200 characters size (in total), in which all these token strings are stored. This buffer size can be redefined with the CONFIG program to provide a buffer of whatever size is required, but it should be remembered that the bigger the buffer, the less space will be available for text, so it is advisable not to make the buffer larger than is necessary. In addition, it is possible to redefine the keys to which the tokens are allocated with CONFIG, which means that any token may be allocated to any key on the keyboard (See Utilities).

a). Pre-defined tokens

The following tokens are pre-defined by PROTEXT and may not be changed. Each of these selects command mode and executes a command.

PCW	6128	Definition
f1	CTRL-f3	CAT
ALT-f7	CTRL-f9	ERA *.BAK
FIND	CTRL-f2	FIND
EXCH	SHIFT-f2	REPLACE
EXTRA-ENTER	CTRL-ENTER	EXEC EXFILE

A full list of all the expansion tokens, including the keys they are allocated to, and their default definitions, is given in an appendix at the end of the manual.

b). Phrases and function key definitions

As far as PROTEXT is concerned, there is no difference between phrases and function key definitions. They are both merely strings (of text or codes) and any difference would be in the use to which they were put, rather than their format. For example, function keys would probably be used to carry out tasks or functions, whereas phrases would be used to store text to be incorporated into documents, though there is no reason why they should not be used for other purposes.

A string has a maximum length of 255 characters, subject to the total buffer size and the free space remaining in it. It may contain any characters and control codes. Any normal text may be typed in from the keyboard as usual, but in order to be able to enter control codes, an escape code must be used to inform PROTEXT that the characters which follow constitute a control code. The escape code used by PROTEXT is the upwards pointing arrow (↑). This is obtained by pressing EXTRA-: (colon) on the PCW. It is used to allow entry of the following:

↑ <number> ↑ Inserts the code specified by the number. The code may be entered in decimal, eg. ↑13↑, or hexadecimal, in which case it must be prefixed by either # or &, eg. ↑&0D↑.

↑ <letter> is translated as a control code between 1 and 26 eg. ↑A would be translated as Ascii code 1, ↑B as 2, etc.

↑ ↑ is translated by PROTEXT as a single up arrow. This must be used if an arrow is required in the string.

Note: When specifying a code as a number, it must be both prefixed and suffixed with the escape code character (↑), but in other cases, it is only necessary to prefix the character with the escape code. This is because a number could consist of from one to three characters.

As an example of how one would use a control code, if a key was to be defined so that when it was pressed it automatically did a catalogue of drive A followed by a catalogue of drive B, the following string would be used:

```
CAT A↑13↑CAT B↑13↑
```

'13' is the code for a carriage return, which would normally be given when the RETURN key is pressed. As CAT requires the RETURN key to be pressed, the codes are inserted into the string. Alternatively ↑M could be used instead of ↑13↑.

Details of the most useful codes are given in an appendix at the end of the manual, but in the unlikely event that a full key translation list is required, this is available from Arnor on request.

c). Phrase commands

There are two commands which are directly connected with phrases and are used from command mode:

PHRASE (KEY)

Syntax: PHRASE <letter> <string>
KEY <letter> <string>

Description: PHRASE and KEY are alternative names for the same command. This command allows temporary strings to be created at any time. The command is used from command mode and the letter must be a letter between A and Z, followed by the string of text or codes, which should be wrapped in quotation marks.

Note: If the phrase is entered as a single command, including the PHRASE command and the phrase on the same command line, the maximum number of characters which may be entered is 128, including the command. If a larger phrase is required, the command should be used on its own and the key letter and phrase entered when prompted. It will then be possible to enter upto 255 characters in total.

Note: If it is required to cancel a key definition, this can be done by using a null string ("") following the key letter in the parameters. This may prove useful when a number of phrases have been defined and the buffer is too full to take any further definitions. Any phrases which are no longer required can be discarded in this way.

LPHRASES (LP)

Description: Use of this command will list the contents of all the defined phrases between A and Z. Where a phrase contains a code between 0 and 31 or between 192 and 255 this will be displayed in escape code form, e.g. ↑9↑.

d). Storing phrases for regular use

The command PHRASE, which enables temporary phrases to be defined has already been described and is very useful for quickly defining a phrase during the course of editing a document, but once the computer is switched off, these phrases will be lost and would require re-entering the next time that PROTEXT was used.

PROTEXT has another method of defining phrases and function key definitions, which enables users to keep one or more files of definitions on disc and to load them as and when required. This is done through the use of an EXEC file and is covered in detail in the next chapter, which describes the uses of EXEC files and how to create them.

A set of phrases should be saved with an appropriate name. Only those keys required and their definitions need to be in the phrase file and any existing phrases will not be changed or deleted unless redefined by the new ones. When they are required, it is only necessary to go into command mode and use the following command:

```
EXEC <filename>
```

where <filename> is the name of the file containing the phrases. This will automatically allocate them to the specified keys.

Note: Depending on the size set for the buffer which holds the phrases, it may be that too many characters are specified in the phrase file, particularly if some phrases have already been defined. If this happens, the excess phrases will be discarded. It may be useful to define any unused phrases as a 'null' string in the phrase file, in order to cancel any previous definitions. This is done by specifying two consecutive single or double quotes for the phrase.

e). Using phrases and function keys

Once a phrase or function key has been defined by either of the above processes it may be used at any time by pressing the appropriate key. Any of them may be used either when in edit mode, or command mode. The most convenient arrangement would probably be to use the function keys for commands which would be used in command mode and the keys 'A' to 'Z' for strings of text to be used in documents.

Note: Most of the function keys are allocated the same tokens as the keys A to Z and redefining one will also change its equivalent. It is not possible to have different contents in each of them. To set up a function key the KEY command is used with the corresponding letter. A table listing the function keys and the corresponding letters is given in an appendix.

Phrases are called by pressing EXTRA and one of the letter keys between 'A' and 'Z', which gives 26 different possibilities. The function keys may be used either on their own, or in conjunction with SHIFT, ALT, EXTRA and SHIFT-ALT, which gives 20 possible combinations on the PCW.

When a phrase key or function key is pressed, the contents of the string will be entered into the document (if in edit mode), or the command line (if in command mode), as if it had been typed in at the keyboard, and any control codes will be acted upon.

On the CPC6128 the 10 function keys f0 to f9 may be used either with SHIFT or with CONTROL. The phrases are obtained in edit mode by typing CTRL-0 (zero) followed by a letter. Only the function keys may be used in command mode, but these are set up to provide the various European symbols. See the 'Options' section for details.

Note: The PCW computers and the PCW.PTR printer driver are capable of printing all the characters between 160 (A0 hex) and 254 (FE hex), as defined in the PCW User Manual, in addition to the normal range of ASCII characters. These characters cannot normally be typed from the keyboard and this is overcome by defining a phrase key to reproduce the character. The code of the character required should be looked up and entered as described above.

In addition, codes above 192 (C0 hex) must be prefixed with a code value of 159 to avoid conflict with functions of the same number.

For example: The 'double shafted left arrow' has a code value of 253 and would be entered as:-

```
KEY A "↑159↑ ↑253↑"
```

From then on, pressing EXTRA-A would give the double shafted arrow symbol and if the PCW.PTR driver was being used, it would also be printed.

Note: The code 255 (FF hex) must not be used in this way as it serves a special purpose in PROTEXT.

WORD 5-6

6. EXEC FILES

EXEC files are a special kind of file, created with PROTEXT, to carry out special tasks.

There are a number of uses for EXEC files. One example, which will already have been encountered, was the file called MAKEPCW (or MAKECPC) which was called when PROTEXT was loaded for the first time and a 'start of day' disc was created. When this was done, it was the EXEC files, called MAKEPCW and MAKEPCW2, which contained all the commands to copy, erase and give information on when to change discs. It can therefore be seen that EXEC files can be made to automate many processes.

A second use of an EXEC file is as a file to contain phrases which may be loaded into PROTEXT's phrase buffer when required.

a). What is an EXEC file?

An EXEC file is a file that may contain text, commands and codes and which, when called with the EXEC command, will be read by PROTEXT and the contents treated and acted on, as if they had been typed in at the keyboard.

They are created in just the same way as any other text file, but what makes them different is the content of the file and the way it is used later.

b). Creating an EXEC file

Creating an EXEC file is extremely simple and is done by typing the required text in, as would be done with any document, but there are a number of special features which permit codes to be inserted into the text, which PROTEXT will understand to be instructions to do certain things.

In addition to ordinary text, any of PROTEXT's command mode commands may be used as well as any valid code between 0 and 254. Each of PROTEXT's edit mode commands, such as ALT-F to format a paragraph and ALT-<, to jump to the start of the paragraph, have their own code values, as do all the characters which appear on the screen when a key is pressed. Details of the most useful of these are given in the 'Key translation' appendix at the end of the manual.

Codes must be entered in a special way, otherwise PROTEXT will consider them to be ordinary text. A special 'Escape character' is used to tell PROTEXT that the character(s) which follow is/are a code and the escape character used is the vertical bar (¦). This is obtained by pressing EXTRA-. (full stop) on the PCW. The escape codes are exactly the same as used in phrase definitions except that the bar (¦) is used instead of the arrow (↑). See the previous chapter for details and examples of escape sequences.

The easiest way to describe the sort of uses to which an EXEC file might be put, is to give one or two examples. The examples given are intended to show the sort of things that can be done, rather than be particularly useful:

Example to change every occurrence of a certain word to another word in a number of files.

```
L file1
R "Protext" "PROTEXT" GA
S¦13¦
L file2
R "Protext" "PROTEXT" GA
S¦13¦
L file3
R "Protext" "PROTEXT" GA
S¦13¦
```

In the above example, ’L file1’ is the command to load a file called ’file1’ and ’R "Protext" "PROTEXT" GA’ is the REPLACE command used with the options to make it global and automatic. ’S ¦13¦’ is the SAVE command. The process is then repeated for ’file2’ and so on.

Note: When a new line is used, PROTEXT takes this to mean that a carriage return character (CR) is required, as would normally be given by pressing RETURN after entering the command. In the case of the lines concerned with saving, escape characters have been used to insert an extra CR code into the file. The reason for this is that if a file is to be saved with the same name, then RETURN is pressed once after entering the ’S’ and again to confirm the same filename.

Note: The file should be saved as a program file and should not contain any formatting features, such as word-wrap or right justification. It is permissible to create the file as a document, for ease of text entry, but before saving, any margins and justification should be removed. One way to do this is to create a temporary ruler at the top of the document, when the file has been created, say 1000 columns wide, using the ’>’ sign in column 1, followed by ALT-G and ’c1000’and entering the ’R’ marker. Using FORMAT will then remove all the soft returns and spaces and create a single line of text for each paragraph. Finally, the temporary ruler should be deleted and the file saved with an appropriate name. Entering ’EXEC <filename>’, from command mode will automatically execute the file of that name whenever required.

c). Creating a phrase file

In the previous chapter on 'Phrases and Function keys', it was mentioned that EXEC files could be used as a means to store regularly used phrases and function key definitions for loading back into PROTEXT. This is easily done by using the PHRASE (KEY) command in an EXEC file:

Example of a phrase file, to be used to define phrases and function keys in PROTEXT:

```
KEY C ""
KEY G ""
KEY B "This is a text phrase which can be inserted into the text"
KEY D "The EXTRA key and the appropriate letter should be pressed"
KEY F "Dear Sir,↑13↑ ↑13↑ ↑9↑"
KEY A "CAT A ↑13↑CAT B↑13↑"
```

In the above brief example, keys B and D are straightforward examples of text to be inserted when the appropriate key is pressed. Key F is a combination of text and commands, in this case, two carriage return followed by the code for a tab. The result of this would be that when EXTRA-F was pressed, the words 'Dear Sir,' would be printed, followed by the cursor returning to the start of the line, moving down two lines and then to the first tab marker, ready for the letter to be started.

Keys G and C are defined as null strings. This will have the effect of removing any existing definition from keys C and G. This may be desirable if a second phrase file is being loaded, when a number of keys are already defined, otherwise the phrase buffer may become full before all the new definitions are loaded.

Key A is an example of the sort of definition which would be used for a function key and in this case would perform a catalogue of drive A, followed by drive B when SHIFT-f1/2 was pressed.

It should also be noted that '↑' has been used, rather than the '¦' symbol, as these commands are simulating entry of the phrases at the keyboard and phrases require the '↑' escape character.

Note: Because the EXEC file is executing a command to define a string, it is necessary to specify the CR at the end of the command if one is required when the function key is pressed, as the CR which is implied at the end of each line of an EXEC file will serve only to execute the KEY command.

Note: As mentioned in the previous chapter, it is only possible to enter 128 characters in total, when specifying the phrase as part of the command line entry. If a phrase is required which is longer than 128 characters, then the 'KEY' command should be used on one line, followed by the key letter and phrase on the following line. This will simulate typing in 'KEY', pressing RETURN and then entering the phrase when prompted.

For example:-

```
KEY
A "This would be a string of text longer than 128 characters."
```

It is recommended that phrase files should be saved with a suitable filename extension to identify them, say '.PHR'. Users of the PCW might find it convenient to insert a command into the file 'STARTUP', to copy any '.PHR' files onto drive M as part of the start of day routine, so that they are always available.

d). Using EXEC for redefining symbols

The SYMBOL command is used in the same way as described above for KEY if it is necessary to redefine the screen symbols (for example to produce Greek letters or scientific symbols). An EXEC file should be created containing all the required SYMBOL commands. Alternatively the SYMBOL commands could be added to the STARTUP or EXFILE files so that the characters are defined each time PROTEXT is used.

Note: Whilst SYMBOL will redefine the screen appearance of a character, it will still be necessary to redefine the printer codes with SETPRINT if the character is to be printed. The chapter on SETPRINT in the Utilities section of the manual should be studied and the printer manual consulted for the codes required to carry this out.

e). Commands related to EXEC files

The following commands are directly concerned with EXEC files in particular, but it should be remembered that almost any command may be used within an EXEC file to simulate keyboard input.

EXEC (X) Execute a file

Syntax: X <filename>

Description: The file specified will be opened for reading and the contents read will be treated as if they were input from the keyboard, until the end of the file is reached, at which time normal operation will continue.

PAUSE Cause PROTEXT to go into a 'waiting' condition

Syntax: PAUSE

Description: When this command is read by an EXEC file, the program will halt until a key is pressed. Optionally a message will be displayed. This is useful during the course of an EXEC file being executed, as it will permit discs to be changed and messages to be displayed before continuing execution.

PHRASE (KEY) Define a phrase. See previous chapter on phrases for full details of syntax.

SYMBOL (SYM) Redefine a screen character

Syntax: SYM <code> <n1> <n2>...<n8>

Description: Code is the ASCII code of the character and n1 to n8 are each single byte numbers, defining the character matrix, starting at the top row and moving down.

f). Using EXEC files

EXEC files may be used at any time by typing EXEC from command mode, followed by the name of the file to be executed.

Note: It is not possible to call an external command or program from an EXEC file and continue with the EXEC file once the external program has returned to PROTEXT.

Note: It is not possible to 'nest' EXEC files within other EXEC files. It is, however, possible to call an EXEC file from within another. This causes any remaining lines in the original file to be ignored.

PROTEXT is supplied with a file called 'EXFILE', which is automatically executed whenever PROTEXT is loaded. It may be convenient to include certain KEY or SYMBOL definitions into this file, so that they are automatically loaded every time. CONFIG may be used to give a new name to the file that is to be auto-executed, if a different name is preferred.

If a file called 'EXFILE' is present on the currently selected drive, it may be executed at any time by pressing EXTRA-ENTER (CTRL-ENTER on the CPC6128).

g). Using an EXEC file when loading PROTEXT

Each time PROTEXT is loaded, it is possible for an EXEC file to be executed and by default the file called 'EXFILE' will be executed. In addition, it is possible to specify another EXEC file, when typing in the command to load PROTEXT. This feature has already been encountered when MAKEPCW was used to create a working disc. If the 'less than' symbol, '<' is used to prefix an EXEC filename when PROTEXT is called, it will be taken to mean that all input should be taken from the specified file, until the end of the file is reached.

For example:-

```
PROTEXT textfile <letter.phr
```

would load a text file called 'textfile' into memory and then execute the EXEC file called 'letter.phr', which in this case would load pre-defined phrases into PROTEXT, but could, if required, carry out operations on the text file, such as replacing text or reformatting the document.

7. LARGE FILES

Note: It is important that CONFIG is used to specify the drive to be used for temporary files, before any large file editing is carried out. See the section on Utilities for full details of CONFIG.

a). Editing large files

PROTEXT is capable of handling large files very efficiently and the only limit on the size of the files which can be edited is the capacity of the disc drives. It must be remembered that under CP/M, large files cannot be totally loaded into memory at one time, and as editing continues and progress is made through a long document, PROTEXT will automatically save parts of the document as temporary files.

As a result, it is preferable to start editing a large file with as empty a disc as possible. With the PCW computers, drive M is normally used as the drive on which these temporary files are stored. On the CPC6128, which does not have a memory drive, the temporary files are saved onto whichever drive has been specified for temporary files with CONFIG. On a CPC6128 with twin drives, this would usually be the drive not containing the text file and it is desirable to have as empty a disc as possible in this drive.

Note: it should also be remembered that there must be sufficient space on the text disc to save the file when editing is completed.

In the event that the document becomes so large that there is no room left for the temporary files to fit, a 'Disc full' message will be issued. If this happens, it will usually be possible to delete one or two files from the disc or drive to make room, before continuing. For example the disc might have copies of the help files on it, in which case deleting these would give more space. Once this situation has been reached, it is worth considering whether the document should be broken down into smaller parts if further editing is required.

Other than the points mentioned above, editing of large documents is exactly the same as editing any other document. It should also be remembered that the ALT-[and ALT-] commands move to the start and end of the text in memory, not the start and end of the whole document. With a small document this will be the same thing, but if the start or end of a long document is required, then ALT-[or ALT-] should be pressed a second time.

b). Important notes on large file editing.

1. It is important that the setting of the temporary drive is adjusted to suit the arrangement of computer and disc drives being used before large files are edited, using the CONFIG program (See Utility 2).

2. If any drive other than the PCW M drive is specified as the temporary drive, it is important to ensure that a disc is present in the selected temporary drive at all times and that it is not changed for another disc during the course of editing the document.

3. Protext saves temporary files with various names commencing with 'PROTEXT'. Under NO circumstances must any of these files be deleted. When the document is completed and saved, PROTEXT will automatically delete the temporary files which are no longer required.

c). Are large documents necessary?

Even though PROTEXT can handle 'unlimited size' files, this is perhaps a suitable place to consider whether it might be more convenient and efficient to work with a number of smaller files. Rarely is there any NEED for a long document to be in one piece. For example: A book is broken down into a number of chapters, which would normally make suitable places to split the manuscript.

Whilst it may appear that there are advantages to being able to work on one long document, so that it can all be viewed and edited at the same time, there are a number of points which should be considered.

1. In the event of a catastrophe, such as a power failure, or accidentally deleting a file from a disc, if the text is in one long document, the complete document may be lost.

2. Due to the limited amount of memory available under CP/M, it is not possible to have the whole of a large document in memory at the same time and as progress is made forwards and backwards through the document, parts of it have to be saved to temporary files and other parts loaded. PROTEXT has specially written routines which do this more efficiently than other programs, but it can still take a short time to jump from one part of a document to another, whereas with a smaller document this will to all intents and purposes be instantaneous.

3. It is usually easier to locate specific sections of text in a smaller file.

4. Usually only a relatively small part of a document will actually be worked on at a time and it is considerably quicker to load, save and format smaller files. Spelling checking is also more efficient, as only the files in which any alterations have been made need to be re-checked.

5. PROTEXT has many features, such as 'two file editing', 'printing to the screen' and the ability to link files together whilst printing, which make it easy to view, move sections of text around and print a number of files as one file.

In the event that it is decided that existing large files could more conveniently be handled as smaller files, PROTEXT has an option in the CONVERT utility program (see Utilities 4), which will split large files into a number of smaller files of a size specified by the user. This will usually be more convenient than loading the large file and saving it as a number of separate files using the SAVEB command.

WORD 7-4

8. TWO FILE EDITING

PROTEXT provides the facility to work on two documents at the same time. These documents are maintained quite separately and are loaded and saved individually. Any operation can be carried out on one document without affecting the other, the cursor location and all markers being maintained separately for each document. Blocks of text can be copied between one document and the other.

This is an extremely powerful function and is controlled by only three commands, one of which is used from command mode and the other two from edit mode.

SWAP (SW)	: Command mode	- Swap between two documents in memory
ALT-O	: Edit mode	- Copy block over from the other document
ALT-Y	: Edit mode	- same function as SWAP

To load a second document, 'SW' should be entered from command mode and the current document will be switched, leaving an empty document. The second document should be loaded in the normal way. Switching between the two documents will cause the information on the status lines to change to suit the current document, enabling easy recognition of which document is being worked on.

In edit mode, ALT-Y performs exactly the same purpose as 'SW', enabling quick switching between documents.

The ALT-O (letter o) command is extremely useful, as it enables any part of the text of either document to be copied over to the other.

Before a block of text can be copied over, the block should be marked out using the markers in the normal fashion. Typing ALT-Y will swap files and the cursor should then be positioned where the text is required. If ALT-O is then pressed, the block will be copied across.

If the original text is no longer required, ALT-Y should be pressed again, to return to the original document, followed by pressing the CUT key, to delete the original text.

Two file editing is also very convenient as a means of keeping notes, for later attention, during the course of editing a document. Press ALT-Y, make the note and ALT-Y again, to return to the original document.

Another use for ALT-O is for transferring text from one file to another - load the first file, type SWAP, load the second file and use ALT-O to copy the blocks required into the first file, before re-saving it. This is quicker than using SB (save block), loading the other file and merging the saved block of text into the document and finally resaving it.

WORD 8-2

9. BACKGROUND PRINTING

Background printing means that it is possible to continue editing another file whilst text is being printed. Background printing is used automatically whenever any of the PRINT commands are used. It is possible, however, to disable background printing completely using CONFIG, and also to have selective background printing - each PRINT command asks 'Background printing (y/n)?'. These options are provided principally for CPC6128 users who may find that there is insufficient disc space for effective background printing and for those users who have printers which contain their own large buffers.

a). How background printing works

When any of the print commands are used to print text, instead of the text merely being sent straight to the printer it is first put into a 'printer buffer'. The printer buffer is held partly in memory, and partly in temporary files. On the PCW computer these temporary files will normally be on the memory disc and so accessing them will be very fast. Whenever there is text in the buffer, it will automatically be pulled out and printed, so printing starts immediately. The benefit of this feature comes from the fact that text is put into the buffer at a much faster rate than it is pulled out for printing. When all the text to be printed has been put into the buffer the command mode cursor will return and normal editing can be continued.

Printing will then continue at the same time and will not be affected by anything else (unless, something drastic is done, like switching the computer off or changing the printer driver whilst printing is taking place!).

Note: It is not possible to continue background printing outside the main PROTEXT program, so care should be taken not to use any of the external commands, such as SPELL or DCOPY whilst background printing is taking place. If this is accidentally done, further printing of the buffer contents will be abandoned.

If a large document is being printed, it may sometimes appear that background printing is not working. This is not the case, it is simply that the amount of text to be printed is greater than the available capacity of the buffer. If the buffer is full, text is inserted into it as soon as text is removed from the other end for printing. Eventually the end of the text to be printed will be loaded into the buffer and then further editing can be carried out whilst background printing continues.

The size of the printer buffer depends on the amount of available memory and disc space.

b). Stopping background printing

Three commands control background printing and allow it to be stopped, resumed or completely abandoned. They can be used from command mode at any time during printing.

STOP	(ST)	Stop printing but leave buffer intact.
CONT	(CO)	Continue printing remaining contents of the buffer.
ABANDON	(AB)	Abandon printing completely and clear the buffer.

Typing ST from command mode merely pauses printing, whilst CO causes printing to re-commence. AB can be used either whilst text is being printed, in which case printing will stop and the buffer will be cleared, or after an ST command, in order to clear the buffer.

If the printer is on 'STOP' when the program attempts to send more text to the buffer, a message 'Re-start (y/n)' will appear. 'Y' or the RETURN key will restart printing, whereas 'N' will put text into the buffer, but not restart printing. Pressing the STOP key will cancel the attempt to send more text to the buffer.

Note: After an ST command, the buffer will still contain any unprinted text and ABANDON should always be used if the remainder of the text is not to be printed, as this will clear the buffer and make the space available for other purposes.

Note: Some printers contain printer buffers of their own, in which case printing can appear to continue after ST or AB have been used. This is because the printer's own buffer still contains text to be printed. The printer manufacturer's recommended method of clearing this buffer should then be used as well. This often consists of simply switching off the printer.

c). Background printing and single sheets

If printing is being done on single sheets of paper, the printer must stop after each page to allow the paper to be changed. There are three methods that can be used to do this. The first method will only work with printers which support paper sensing, i.e. the ability to turn themselves offline at the end of a page if they are out of paper. The second method will work with all printers, and the third is specifically for the PCW printer. The first method is to be preferred, if supported, as it permits continuous editing without interruption, except to insert the new paper when required. If the printer being used supports paper sensing, the appropriate codes should be inserted into the printer driver with the SETPRINT utility (See Utilities 3).

(i) Method for printers with paper sensing

(including Epson printers, some Epson compatibles and the Amstrad DMP2000).

This method uses the stored command PS (paper sensing). Include the command '>PS ON' near the start of the text (see chapter on stored commands). Single sheet printing (>CP OFF) need not be selected.

When the end of a page is reached the printer will automatically be turned off line. Simply inserting new paper and pressing the on line button on the printer will resume printing. This operation does not affect any other actions the computer may be performing.

Note: The paper sensing codes are stored in the printer driver, and will have to be changed using SETPRINT if they are not the same as the standard Epson codes, namely ESC "8" and ESC "9".

(ii) Method for all other printers except PCW

The '>PS ON' stored command should **NOT** be used, but >CP OFF should be used to cancel continuous paper printing.

This method should be used with all printers that do not support paper sensing. This includes some printers that are mostly Epson compatible, such as the Kaga Taxan.

When the end of a page is reached, the message:-

'Insert paper: press COPY. STOP to stop'

will appear on the screen, interrupting whatever work is in progress. The paper should then be changed and the COPY key pressed to restart printing, or the STOP key to stop printing. If printing is abandoned with STOP, the buffer will still contain any unprinted characters and the ABANDON command should be used to clear the buffer, for the reasons explained earlier.

Note: If the printer contains its own substantial printer buffer, then it may be found that the 'Insert paper' message appears some considerable time before the end of the page is reached. It may be possible to disable this buffer and the printer manual should be consulted. It is not a fault of the program, but a result of the printer's buffer absorbing characters to be printed, but not yet printed. If the buffer cannot be disabled, it may be found more convenient in this case to turn background printing off with CONFIG.

(iii) Method for the PCW printer

Single sheet printing must have been selected with >CP OFF. When the end of a page is reached, the message 'Waiting for paper' will appear at the bottom of the screen. Simply insert a new sheet in the normal way and press EXIT to continue printing.

The PTR key functions in exactly the way described in the CP/M part of the PCW manual and this should be referred to for full details of the use of the PTR key and the associated status control line which appears at the bottom of the screen when this key is pressed.

Note: To clear the buffer on the PCW printer, the PTR key should be pressed and the printer reset in the normal way (See the PCW operating manuals for full details).

10. SUMMARY OF PRINTING FACILITIES

This chapter introduces no new commands or facilities and is provided for the purpose of giving an overall picture of the various commands which determine the final appearance of a document and the order in which they take precedence.

PROTEXT has a natural order in which it uses commands. If PROTEXT is loaded with no printer driver specified, then the default 'simple printer' driver will be used. If a printer driver to suit the printer is loaded, then this will override the simple driver and the codes specified in the printer driver will be used if any printer control codes are inserted into the document. If any new or different printer control codes are specified within the document, using the '>CC', '>RC' or '>OC' stored commands, then these in turn will take precedence over the codes provided by the printer driver.

Similarly, if a document is created and contains no stored commands, then the default settings provided by PROTEXT for such things as page length, line spacing etc. will be used. If PROTEXT has been reconfigured to alter the default settings, then when the PROTEXT.CFG file is loaded, the original default settings will be replaced by the new ones provided in the configuration file. These settings may be modified, in turn, by the use of stored commands in the text.

The appearance of a document is therefore determined by a number of factors. Starting at the very beginning, the appearance will be governed by the settings of rulers and tabs in the document and whether right justification is turned on or off when the text is entered. The appearance can be further altered by the use of stored commands, which may be used to change the default settings for the sizes of the various margins, the page length, whether single or double line spacing is to be used, headers or footers printed on each page and many other features. Each of these stored commands will override the provided or reconfigured default settings. In addition to these features, it is possible to use printer control codes to specify the type style which is to be used in different parts of the text and, assuming a suitable printer driver has been loaded, these will override the simple printer driver.

Assuming a document has been created which contains the text in the desired format, there are now a further set of options which may be used to determine how and where the document will be printed. These commands are carried out from command mode and are the various print commands.

PROTEXT contains many different commands which may be used for printing, but for most purposes only one or two will be used. All the printing commands begin with the word 'PRINT', followed by further letters in many cases. These letters describe exactly what the commands do.

'PRINT' used on its own will print a document in draft mode, unless any printer control codes in the document specify otherwise.

Print commands with the letter 'Q' added, such as 'PRINTQ' and 'PRINTPQ' will automatically ensure that the document is printed in Near Letter Quality, without the need to use the printer control code in the document. This is particularly useful as it means that a draft copy can be printed first, using the simple command, then the 'top' copy produced with the same command, but with the 'Q' suffix. For example, PRINT, followed by PRINTQ.

A further group of printing commands have the 'P' suffix, such as 'PRINTP' and 'PRINTPQ'. These commands allow the selective printing of Pages and when used, confirmation is requested as to whether each page is to be printed.

A third group of printing commands use the letter 'B', as in 'PRINTB' or 'PRINTQB'. These commands will print only a Block of text marked with the block markers, rather than the whole document.

As can be seen from the above, in many cases combinations of the extra letters are used, such as 'PRINTQB'.

In addition, two further parameters may optionally be specified with all except the block commands. The first parameter is a filename and if this is specified, then the file of that name will be printed from the disc, rather than the file in memory. Finally, the number of copies to be printed may be specified as a number.

All the normal printing commands which send output to the printer have now been covered, but in addition there are two further groups of printing commands which send print output to other places. The 'PRINTS' and 'PRINTSB' commands send the output to the Screen instead of the printer. This provides a convenient method of checking on page breaks and the visual appearance of the document, before doing a proper print. It is also particularly useful for checking that a mail merging file is giving the expected results. The optional parameter of a filename can be used to print a disc file to the screen and this is useful as a means of checking the contents of a disc file.

The 'PRINTF' and 'PRINTFB' commands function in a similar fashion to the PRINTS commands, except that instead of sending the output to the screen, it is sent to a disc as an ASCII file. In this case it is necessary to specify a filename with the command and this will be used as the name that the file will be saved as. Optionally, a second filename may be specified and in this case, a file of that name will be read from disc and sent to a file with the name of the first filename specified, as an ASCII file.

That covers all of the normal printing commands, but there are still one or two more of a specialised type which are described in detail in the chapter on Command mode.

11. PROGRAM MODE

By default, PROTEXT always goes into 'document mode', which is the mode normally used for editing and allows all the formatting facilities to be used, unless the file being loaded was originally saved as a program file from program mode, in which case 'program mode' will be selected. Program files are displayed in a disc catalogue with a plus sign (+) next to the filename.

Note: The setting of PROTEXT which causes it to default to document mode may be altered by use of the CONFIG program. See 'Utility programs'.

a). What is 'Program mode'?

Sometimes, particularly if a program (BASIC, assembler, or anything else) is being edited, the formatting facilities are not required (and can be a nuisance). Program mode is selected by the command PROG, entered from command mode. The word 'Program' will replace 'Document' on the status lines and there will no longer be a ruler line displayed beneath the status lines. Tabs are set at every eighth position and cannot be changed and there is no right margin as there is no formatting. The DOC command may be used at any time to revert to document mode.

Any files saved by PROTEXT are 'marked' with the type of file, so that when they are subsequently reloaded, PROTEXT will automatically go into the mode that they were in when the text was saved. When any files created by other programs are loaded, PROTEXT will, by default, go into document mode, but this may be altered by use of the CONFIG utility.

b). Editing commands available in program mode

When program mode is in use, most of the normal editing commands may be used in the same way that they are used in document mode, but it should be noted that because there is no formatting, any of the commands concerned with formatting of the text, rulers and pages, will have no effect.

Note: Appendix 1 gives details of which commands are not available.

c). Printing in program mode

The PRINT command operates differently in program mode. Stored commands are printed and not obeyed, and the print options are ignored. Instead, the text is printed continuously with no margins and single line spacing. This is suitable for printing program listings, and is also useful to print a copy of the text with the stored commands shown. In the latter case enter the PROG command before printing and the DOC command afterwards. Printer control codes and character redefinitions are both effective.

Note: It should be noted that whilst any control codes in a program file will be acted on by the printer, it is not possible to insert them into the text whilst in program mode. If any control codes are required, they may be inserted by going into Doc mode, entering them in the normal way and then reverting to Prog mode. The most common reason for wanting to insert codes into a program file would be to print the document in a more condensed typestyle.

SPELLING CHECKER

1. INTRODUCTION - WHAT CAN SPELL DO?

a). Introduction

SPELL can do several things. First and foremost it will proof read a document and find most mistakes. This includes simple typing mistakes as well as words that are incorrectly spelt. SPELL will draw attention to these mistakes. It will not, and cannot, automatically correct them. In doing this, SPELL has three major advantages over other spelling checkers. It will check the text very quickly and it provides convenient methods for correcting the mistakes. SPELL can be used without having to leave PROTEXT and load another program.

SPELL works in a way that is fundamentally the same as would be used to check spellings using a traditional dictionary. This dictionary, however, is contained on a disc, and whereas normally only words about which there was some uncertainty would be looked up, SPELL looks up every single word in the dictionary, at a speed that could not be approached manually.

Although SPELL cannot automatically correct mistakes it does have one feature that goes a long way towards this. In many cases it is able to find the correct word in its dictionary. This is possible because most typing and spelling mistakes can be categorised by a few general rules.

One unavoidable limitation to be aware of is that if a word is misspelt, but the misspelling is also a valid word, then SPELL will not recognise this as an error. A program that did this would be a "context checker", and development of such programs is included in the research into artificial intelligence being carried out at universities throughout the world.

b). How SPELL decides what is a word

A word is any group of characters starting with a letter, and containing letters and possibly apostrophes. Foreign characters are recognised, though initially no words containing foreign or accented characters are incorporated into the dictionary.

There are additionally a few special cases as follows:

Single letters: These are always ignored, and so the words "a" and "I" are not stored in the dictionary. This means that when letters are used to label a list of items they will not be listed as an error. For the same reason the Roman numerals up to xii (12) are contained in the dictionary.

Hyphens: Hyphenated words are treated as two words.

Possessives: Words ending with an apostrophe followed by s ("'s") are checked up to the letter before the apostrophe and the possessive ending is ignored. Single apostrophes at the end of a word are also ignored. Words which contain an apostrophe that is not followed by a solitary letter 's' can be stored in the dictionary and will be checked as one word. eg. 'Won't'.

Words containing numbers: If the word starts with a number then the whole word is ignored. This means that, for example, dates such as "17th" are not listed as an error. If a word starts with two or more letters and is followed by numbers, then just the numbers are ignored. For example "PCW8512" is checked as "PCW" (which is in the dictionary).

c). What sort of errors will SPELL find?

SPELL will pick out any words which do not match with a word in the dictionary. There are two reasons why the word may not be found.

The word may be misspelt and therefore require correction, or alternatively the word may be a correctly spelt word which is not contained in the dictionary.

SPELL will also pick out any words which contain letters in a mixed case, with the exception of words which start with an upper case letter. For example, 'tHat', or 'wHEN' would not normally be considered correct and will be selected for possible attention, whereas 'The' would frequently be a quite acceptable use of an upper case letter and will not be singled out by SPELL.

Another special feature of SPELL is its ability to check for an unknown word separated from a solitary letter by a single space. For example, 'tha t' would be singled out for attention as 'tha' would not be recognised as a correctly spelt word and the letter 't' is separated by a single space. In this case the word and the separated letter are both treated as one unknown word and offered for attention. The reason for this is that this is one of the most frequent forms of typing error, as opposed to spelling error and is simply corrected by removing the offending space.

2. THE DICTIONARY DISC

IMPORTANT - The dictionary disc MUST be copied before use. The original must not be used and should be retained as a back up.

Please note that the dictionary disc is subject to copyright laws, and you are expressly permitted to make as many copies as you require, but only for your own use with PROTEXT's SPELL program.

The dictionary is on the reverse side of the program disc. This should be copied onto a blank disc, checking first that the write protect tab is pushed down on the original (to avoid any possibility of an accident), by following the instructions in the introductory chapter of this manual, called 'Getting Started'.

It is advisable to take at least two copies of the disc for added security. When the disc has been copied, one copy should be kept as a backup in case of accident. Periodically, as words are added to the dictionary disc (see later), backup copies should be updated. The write protect tab should always be open before taking a copy.

Data is stored in the dictionary in a special form and the disc does not use the directory in the normal way. If the dictionary disc is catalogued the message "0K free" will be given. This does not mean that the dictionary is full and new words may still be added to the dictionary by SPELL at any time, but is arranged in this way to ensure that it is not possible to corrupt the dictionary by accidentally saving text files onto the disc. It is, of course, still possible to erase the dictionary by copying a whole disc onto it, or re-formatting it, so be careful!

SPELL 2-2

3. SPELL CHECKING A DOCUMENT

A few general remarks

Virtually all of the options may be stopped at any time by pressing the STOP key. Pressing the STOP key once pauses the program, at which point pressing STOP a second time will stop it completely and return to the relevant menu, or any other key will resume the program.

Dictionaries are identified by name, and the name of the dictionary currently being used is always displayed at the top of the screen.

The dictionary should always be inserted in the drive indicated by the message at the top of the screen and when instructed by SPELL. For PCW users, this will always be drive A.

Using SPELL

The spelling checker can be used in three different ways:-

a). SPELL can be used from within PROTEXT to check the document in memory, or a file on disc, in a simple single pass check.

b). The main SPELL program can be called from PROTEXT, using the command 'SM' (short for Spell Menu), giving access to the full range of options available, before finally returning to PROTEXT.

c). SPELL can be used as a 'stand alone' program, called directly from CP/M command mode.

a). Simple checking of a document from within PROTEXT.

Note: This option provides the means to carry out a simple spelling check with one dictionary only. If more than one dictionary is required, or a two pass check (See later) is required, then one of the other methods of checking a document should be used.

This is the option which is used to check a document in memory.

The command 'SPELL' should be entered from PROTEXT command mode. After a slight pause, instructions will be given about which disc to insert. These messages vary according to the model of computer and the number of disc drives installed. A new status line will appear, giving details of the version of SPELL, together with information on the drives selected and the editing options available in SPELL.

Checking will start immediately and the first thing that will happen is that messages will appear advising that SPELL is ’Loading’, ’Scanning’ and then ’Sorting’. The words will first be sorted alphabetically. Once sorted, the words can be checked more quickly than would otherwise be possible.

SPELL will then start checking the document by comparing the words in the text against the dictionary. The number of words checked is displayed as a running count at the bottom of the screen. Any word that is not in SPELL’s dictionary will be listed on the screen, with a question mark (?) after it and the checking will stop.

Any words listed will generally fall into one of two categories, and the action to be taken will depend on this:

A word that is incorrectly spelt, whether because it was mistyped, or is a spelling mistake.
A word that is correctly spelt but is not included in the SPELL dictionary.

There are four possible courses of action which can be taken once a word has been picked out, each of which is selected by typing the appropriate letter:-

C	Change spelling of word
S	Store word in dictionary
I	Ignore word
L	Look up correct spelling

(i) C - Change spelling of word.

If the word is obviously incorrectly spelt, then C should be selected to correct the word. Part of the text will be displayed with the word highlighted, and then the word in question will be displayed with the cursor on the first letter.

If there is any doubt whether the word is incorrectly spelt, without seeing it in context, then C should be selected and a few lines of the text will be displayed. If the word is correct and is not to be changed, pressing STOP twice will return to the select option stage allowing one of the other options to be selected. Alternatively, the RETURN key may be pressed, but this will result in the word being included in the figure for the number of words changed in the final summary.

The word may now be edited using the PROTEXT line editing functions as in command mode. See the section ’Command mode’ for details of these.

If the same word was spelt incorrectly more than once, then when RETURN is pressed to indicate that the correction is complete, the option to change all occurrences is offered. If 'Y' is selected then they will all be changed and listed in context, with the corrected word highlighted and the spelling checked again.

If 'N' is selected, the remaining occurrences will be offered for correction, until all of them have been corrected or ignored.

If the word only occurred once, then after correction the text will be re-displayed in its new form, with the corrected word highlighted and the spelling of the word will be re-checked.

Note: It is possible to replace the word by two or more words but if this is done, only the first word will be highlighted and only the first word will be checked again.

The dictionary contains most commonly used words but there are many other words in the English language. If the word is correct, but is not one that the dictionary recognises, there are two possible courses of action that may be taken.

(ii) S - Store word in dictionary.

If the word is correct and it is one that should be incorporated into the dictionary, then selecting S will store it in the appropriate section. The most obvious examples of words that would be added to the dictionary are technical terms or the words making up the user's name and address.

(iii) I - Ignore word.

If the word is correct, but not one that should be stored in the dictionary, then pressing 'I' will cause the word to be ignored, together with all further occurrences of the same word in that document. If the word appeared more than once, the number of times it appeared will be displayed on the screen. An example of such a word would be 'ALT' which occurs on many occasions in this manual and is correct in this context, but would not normally be considered a word.

Finally there is one further option.

(iv) L - Look up correct spelling.

L is the most interesting option. If a word is highlighted as unrecognised and there is some doubt as to whether the spelling is correct, selecting L will get SPELL to attempt to find the correct spelling in the dictionary. It will search the dictionary and any likely words will be listed. If the correct word is recognised, the C option can then be used to change the word.

This is by no means guaranteed to find the correct spelling, as the correct word may not be in the dictionary, but in many cases the intended word will be found.

Note: The 'Look up' option does not work instantly, as it has to search the dictionary for similar words. Also the correct word may be found and yet the search will continue because SPELL has no way of knowing that the word found was correct. In many cases several possible words will be listed. If, however, the correct word has been listed, pressing STOP twice will cancel the search and return to the select option stage.

At the end of the checking operation

Once checking is complete, a number of statistics will be displayed. These are:

1. The number of words checked.
   Note that this ignores single letters, numbers, etc.
2. The number of different words.
3. The number of unrecognised words.
4. The number of words added to the dictionary.
5. The number of corrections made.

If the length of any word has been changed, this may have disrupted justification and the text may need to be re-formatted, so a warning message will be displayed as a reminder of this.

Finally SPELL will issue the message, 'Press a key to resume:' and on pressing a key, a return will be made to PROTEXT, which will now contain the corrected document.

Any reformatting necessary should then be carried out by the normal methods, usually the FORMAT command, before the document is saved and printed out.

Note: If the document is to be saved to a disc file, this must be specifically carried out from PROTEXT command mode. The references to 'Saving...' which occur during the course of spell checking do not mean that the document has been saved as a normal file.

b). Using the main SPELL program from within PROTEXT.

The full facilities of SPELL can be obtained from within PROTEXT by using the command 'SM', which will call up the main 'Spell Menu'. Before entering SPELL, PROTEXT will temporarily save any existing text in memory and then automatically load the SPELL program.

Once SPELL is loaded, the following options will be listed on the screen, together with the instruction to 'Select Command'.

S	-	Single pass check
T	-	Two pass check
C	-	Catalogue files
D	-	Drive select
G	-	Group select
B	-	Backup file erase
P	-	Printer on/off
U	-	Utilities
Q	-	Quit SPELL

Large files

SPELL is capable of handling any size file, but large files will not not be checked in one go. In these cases the file will be checked in two or more sections. When the first section has been checked the next section will be loaded in and checked.

Note that if a word appears in two sections of a document it will be listed twice, even if "I" is pressed the first time. The unrecognised words count will therefore count the word twice in this case.

(i) S - Single pass check

This is a simple single pass spelling check and is essentially the same as the simple check described in the previous section, the main difference being that instead of checking the document in memory, it checks a document saved as a file on disc.

When S is selected a filename is requested and the name of the file to be checked should be entered. From this point on, the spelling check will proceed in exactly the same way as described in the previous section until the check is complete, when pressing any key will return to the main menu rather than PROTEXT.

(ii) T - Two pass check

This option is an alternative to "S", and may be more convenient with long files. This option must be selected if more than one dictionary is to be used to check the document.

The general principle on which the spelling check is carried out is the same as for the single pass check and all editing and correcting of words is done in exactly the same way.

The difference is that once a filename has been specified, SPELL will load the file to be checked and will then do a first pass through the file, compiling a list of all unrecognised words and listing them on the screen but without stopping at each word.

When the first pass is complete the option to check with 'Another dictionary (y/n)' is offered. The subject of multiple dictionaries is covered later in this chapter.

If the 'N' option is selected, the second pass will commence. On the second pass each of the unrecognised words will be displayed one after another and the words may be edited in the same way as described for the simple check.

There are several reasons why the two pass check might be preferred:-

(i). There is no need to watch the screen for the whole time the check is taking place. It is only necessary to edit the words one after another at the end. This is most useful with longer documents.
(ii). If the 'Printer on' option is selected, a list of the unrecognised words will be listed by the printer for checking at leisure.
(iii). Multiple dictionaries. This allows the checking of a document with as many dictionaries as required.

Multiple dictionaries:

For specialised or technical uses it may be considered desirable to use different dictionaries containing any specialised words, whilst retaining the main dictionary for normal words.

The two pass check allows the text to be checked against any number of dictionaries in turn. Only the words not found in any of the previous dictionaries are checked in subsequent dictionaries.

At the end of the first pass the question "Another dictionary (y/n)?" is displayed. selecting Y will result in a request to insert the next dictionary and only the unrecognised words will be checked against the new dictionary.

Full details of methods of creating new dictionaries are given in the next chapter.

(iii) C - Catalogue files

This option catalogues the files on the text drive, listing the names of the files, together with the size of each file and the amount of free space on the disc.

Note: Only files in the currently selected group will be displayed and for full details of how to select other groups, the section, 'Group select', should be consulted (See below).

(iv) D - Drive select

This option allows the selection of which drives are to be used during spell checking. The available choice depends on the computer and the number of disc drives fitted. Repeated use of this option will cycle through the available combinations.

(i) PCW8256, 1 drive.
The default setting is 'Text A Dictionary A'. This can be changed to 'Text M Dictionary A'.

(ii) PCW8256, 2 drives; PCW8512.
The default setting is 'Text B Dictionary A'. The other possibilities are 'Text M Dictionary A' and 'Text A Dictionary A'. The last option allows SPELL to be used as if only one drive was fitted, but to make the most of the second drive, the text files should be held on a CF2DD format disc in drive B.

(iii) CPC6128, 1 drive.
There is no choice, this command will do nothing.

(iv) CPC6128, 2 drives.
The default setting is 'Text B dictionary A'. This can be reversed.

Note: When SPELL is entered from PROTEXT, the currently selected drive will be passed to SPELL as the 'Text drive' and SPELL will select the correct drive for the dictionary.

(v) G - Group select

This option enables any of the groups to be selected. If a document is stored in group 3, then group 3 must be selected before performing the spell check. If all documents for spell checking are stored in group 0, there is no need to be concerned with changing the group number.

Note: The catalogue of files only shows files in the currently selected group.

(vi) B - Backup file erase

This option erases all files in the currently selected drive/group directory with the suffix '.BAK'.

When a spelling check is performed the original (uncorrected) version of the file will remain on the disc with the filename extension changed to 'BAK'. For example a file called 'LETTER.3' will be renamed to 'LETTER.BAK'. The new version will be given the original name.

Note: SPELL will give a warning before checking commences if there is insufficient space available on the disc to save the corrected document. This can often be cured by erasing all the back up files on the disc. If this fails then the document should be copied onto a fresh disc before checking it.

(vii) P - Printer on/off

Selecting "P" will cause the display at the top right of the screen to change to "Printer on". Pressing it again will revert to "Printer off". When "Printer on" is selected most of the commands will produce printed output. When used with the Single pass 'S' option, a complete record of the unrecognised words and responses will be printed.

The 'Printer on' option is particularly useful when used in conjunction with the 'T', two pass check option. The words will be printed on the first pass so that a list of unrecognised words, printed in 4 columns, is available for verification, before correction on the second pass.

(viii) U - Utilities

Pressing "U" selects the Utilities menu. This menu lists various options generally concerned with the dictionary rather than spelling checking itself. See the next chapter for full details of Utilities options.

(ix) Q - Quit SPELL

Selecting this option will leave SPELL and return to PROTEXT.

c). Using SPELL as a 'stand alone' program, from CP/M command mode.

Both the previous methods of using SPELL have been carried out from PROTEXT, but it is also possible to use SPELL on its own from CP/M command mode and either of the previously described methods may be used.

If the command, 'SPELL' is used on its own, from CP/M, the main menu with all its options will be available for use and will function as described above for the 'SM' option. On exiting SPELL, the program will return to CP/M command mode.

Alternatively, a file name can be specified after the SPELL command, in which case the file specified will be checked with the 'S' option, without access to the main menu, in a similar fashion to the simple check described above. In this case, once the corrected file has been saved, the program will return to CP/M command mode.

For example, entering 'SPELL letter1', from CP/M command mode, will load and perform a simple spelling check on a file on disc called 'letter1', save the corrected version back to disc and finally return to CP/M command mode.

Note: The default settings used when SPELL is entered from CP/M may be changed by making use of the utility program, 'CONFIG'. See 'Utilities' for full details.

SPELL 3-10

4. DICTIONARY UTILITIES

The Utilities menu provides facilities to carry out management of the dictionary disc as well as several options which search for a word, or words, in the dictionary.

As with the main menu, options are selected by pressing the appropriate letter and STOP may be used to leave any of the options whilst they are in progress. STOP may be used from the utilities menu to return to the main menu. The options available are:-

E - Edit Dictionary
L - List dictionary
F - Find words
A - Anagrams
P - Printer on/off
W - Word count - dictionary
N - Name change - dictionary
I - Initialise new dictionary
B - Build dictionary
C - Catalogue files

a). E - Edit dictionary

The dictionary editor provides a simple means of correcting any errors in the dictionary. For example, if a misspelt word has accidentally been stored in the dictionary. After pressing "E", a word is requested and the word to be found should be entered.

SPELL will then search the dictionary for this word and one of two messages will then be displayed. If the word is found, the message 'Word in dictionary. D to delete, any other key not to:' will appear and pressing D will delete the word. Alternatively, if the word is not found, the message, 'Word not in dictionary. S to store, any other key not to:' will appear and selecting S will store the word. In either case, pressing any other key will leave the dictionary unchanged.

When editing is complete, pressing STOP will return to the menu.

b). L - List dictionary

This simply lists the words in the dictionary onto the screen (and printer if selected), starting at any point. A word at which the listing is to start can be specified, or by pressing RETURN, the whole dictionary can be listed from the start.

Note: Due to the method used to store words in the dictionary, the listing will usually start a little before the word selected.

Note: Although the dictionary is supplied with the words in alphabetical order, words that are added later will not appear in their strictly alphabetical place, but a little further on. This does not affect the operation of SPELL in any way.

c). F - Find words

A word pattern is requested and SPELL will search the dictionary and list words matching that pattern. A word pattern can consist of letters, apostrophes and two special "wildcard" characters. This can be used to great effect when solving or compiling crosswords.

?	stands for a single unknown letter
*	stands for any number of unknown letters (can be zero). Only one "*" may be used in a word pattern.

This is most clearly explained with examples:

Word pattern	Words listed
BA???	5 letter words starting with "BA"
A?A?U?	6 letter words with "A" in the first and third positions and "U" in the fifth position
BA*	All words starting with "BA" (any length)
S*K	Words starting with "S" and ending with "K"
*ALE	Words ending with "ALE"
????????????	12 letter words
*	All words in the dictionary!

d). A - Anagrams

This is very similar to Find words in use. It will look for and list all anagrams of the letters typed as the word pattern. Again wildcards can be used.

Examples:

Word pattern	Words listed
NAME	Anagrams of "NAME", e.g. "AMEN", "MANE", "NAME"
NAME?	Words containing the letters "N", "A", "M", "E", and one other, e.g. "NAMED", "MEANS"
II????	6 letter words containing two Is
KK*	Words containing 2 Ks (any length)
??????????*	Words of 10 or more letters

e). P - Printer on/off

This serves exactly the same purpose as in the main menu and is simply provided for convenience.

f). W - Word count - dictionary

The number of words stored in the dictionary is counted, and a running total displayed. The total number of words will finally be displayed, together with an indication of the percentage of the disc already used.

Note: The "Disc usage" figure does not accurately indicate the number of extra words that can be added. The method of dictionary storage allows many words to be added without using any more space, if the words are similar to words already present.

g). N - Name change - dictionary

This option allows the dictionary name to be changed at any time. The old name will be displayed and a new name requested. Pressing STOP, once this option has been selected, will leave the name as it is and return to the menu.

h). I - Initialise new dictionary

This option will create a formatted dictionary disc, with no words in it.

Note: This command will destroy the entire existing contents of a disc, so care should be taken to ensure that either a new disc is used, or that there is nothing of any importance on the disc.

Once the disc has been initialised, a name will be requested for the dictionary (which might reflect the particular use, or the name of the person using it).This name is displayed whenever the dictionary is accessed by SPELL, so that it is always possible to see which dictionary is in use.

i). B - Build dictionary

This option can be used to add words to a dictionary, to create new dictionaries, or to create sorted lists of words in a file.

When this option is selected, the message "Dictionary or File?" will appear (Except on a single drive CPC6128). Selecting "D" will store the words directly onto the dictionary disc, whereas "F" will send them to a file.

Note: If the words in the file are already sorted into alphabetical order, SPELL's sorting phase will take significantly longer. This apparent paradox is a consequence of the sorting algorithm used by SPELL, but since documents are rarely in alphabetical order this is not normally a limitation in practice.

(i) Directly onto the dictionary disc

This option is used to build up a dictionary from lists of words that are already known to be correct.

Operation is similar to the spelling check option, with the exception that all unrecognised words are automatically stored in the dictionary without pausing. Once the name of the file containing the words has been requested and entered, the process is completely automatic.

One variation in the use of this option is to select "Printer on" so that a list of words that have been added is printed. The printout can then be checked at leisure and the "Edit dictionary" option used to remove any words not required. This may sometimes be quicker than using the spell check option and storing the words individually.

(ii) To a file on disc (Requires 2 drives on the CPC6128)

This option requires two drives since the words are checked against the dictionary disc and written out to a file on the text disc. On the PCW8256 with 1 disc drive, the memory drive may be used to store the file but it must, of course, be copied onto a disc later.

When the D option is selected, a request is made for the 'text file', which is the file containing the words to be checked. This is followed by a request for an 'output file', which is the filename that is to be given to the file which will contain the unrecognised words.

Once input and output filenames have been specified, SPELL will proceed to carry out a spelling check as described above, but in this case any unrecognised words will automatically be sent to a file on disc, rather than be incorporated into the dictionary.

There are a number of possible uses for this command, but one in particular is to create a sorted list of all the different words used in a document. This can be done quite easily by creating a new dictionary disc that contains no words, with the 'Initialise dictionary' option and using this as the Dictionary disc to check the text file against. As SPELL sorts the words in the text file into alphabetical order before commencing to check and as the dictionary contains no words, the end result will be that all words will be unrecognised and therefore sent to the file in alphabetical order.

j). C - Catalogue files

This is a duplication of the same command in the main menu for convenience of use.

SPELL 4-6

MAIL MERGING

1. WHAT IS MAIL MERGING?

In its simplest form, mail merging is the process of incorporating text from one file into the text of another file. One of the most common forms of mail merging is incorporating names and addresses from a data file into a standard letter, in order to create a "personalised" letter for each person. There are, however, many variations and the next few chapters will describe these, starting with simple examples and gradually introducing the more advanced features of mail merging with PROTEXT.

The mail merging facilities of PROTEXT are very advanced and will permit the carrying out of virtually any sort of mail merge. Selective, conditional and alternative text merges are all possible, as are arithmetic calculations and each will be covered in detail in the following chapters.

Despite being a very advanced program, it has been designed in such a way that straightforward mail merging is very simple to understand and use, whilst at the same time facilities are incorporated which will permit the most extensive of mail merges and calculations.

Because mail merging can be a complex subject, this part of the manual is written on the basis that the user has no knowledge of the subject and instead of listing all the facilities at once, introduces them gradually. In this way it is easy to carry out simple mail merges and with experience, to progress to more complex examples. At the back of the manual is a technical appendix which details all the stored commands in alphabetical order.

Throughout the chapters on mail merging, many examples are given, with the recommendation that they be typed in, tried and saved. On occasions earlier examples will be used again and expanded to demonstrate the use of new commands. Many of the examples can be used as 'template' files and modified at a later date to suit the needs of users.

One of the features most commonly found in connection with mail merging is the use of data files and the next chapter covers data files in detail.

MERGE 1-2

2. DATA FILES

Data files are a very important aspect of mail merging, so before looking at mail merging, it is important to know exactly what they are and how they are created.

a). What is a data file?

A data file is a collection of items of information which may be anything from names, addresses and telephone numbers, to paragraphs of text, or collections of numbers.

Data files can be created in a number of ways. The simplest example would be one created by PROTEXT, in which case the data would be entered using PROTEXT in the normal way and then saved to a file on disc.

Most good databases and many spreadsheets are capable of creating data files suitable for use by PROTEXT, but in a few cases, programs may create files which are largely, but not entirely, suitable. The utility program, CONVERT.COM provides a number of options which may be used to easily convert such files into a suitable format. Full details are given on how to use these options in the chapter on 'Utility programs'.

A third source of data is 'output' from suitable Basic programs, written to process and store data.

Any Ascii file may be used by PROTEXT as a data file, providing the data is stored in a suitable format.

b). Data file format

A data file will usually contain a number of records, each of which will contain one or more fields of information.

For example:-

A data file might contain a list of names and addresses, in which case the name is considered as one field and each line of the address as a further field. All the fields that constitute one complete name and address are considered to be one record. All the records constitute one data file.

c). How PROTEXT reads data

When PROTEXT reads from a data file, it reads characters from the beginning of the file, until it finds either a comma or a carriage return character, either of which it will take to indicate the end of the first field. It will then take the next character it finds to be the start of the next field and repeat the process until the appropriate number of fields has been read. Any leading spaces in a field will be ignored.

The only exception to this is when the first non space character of a field is found to be either a single or a double quotation mark; in which case PROTEXT will continue to read the characters until it finds matching quotation marks, which it will then take to indicate the end of that field. Any quotes of the alternate sort, or commas, contained within the matching quotes will not be treated as end of field markers in this case, but as part of the text. In addition, any leading or trailing spaces inside the quotes will be retained as part of the text. It can therefore be seen that on occasions it may be desirable to 'wrap' fields in quotation marks to ensure that fields do not get broken up by commas and quotation marks in the text.

Normally each record will be separated from the next by a blank line and this will be used by PROTEXT to indicate the end of a record.

Occasions can arise where one or more of the records may contain fields which have no information, for example when one of the fields is for telephone numbers. Unless the absent field is the last field in the record, it is vital that the field is not left out of the record and empty fields can be represented either by an empty line, or alternatively, the dollar symbol '$' may be used to represent an empty field. When reading data, PROTEXT understands a single dollar symbol to indicate an empty field and when it has been read, converts it to an empty string. One of the advantages of using the dollar symbol is that it makes it easier to view the data file and see where each record starts and ends, but there are other reasons why it may be preferable, and these will be covered in later chapters.

Note: Whilst it is preferable to have a blank line between each record, some programs produce 'Comma delimited' export files. These files have all the fields of a record on one line, with each field wrapped in quotation marks and separated by a comma and sometimes without a blank line between each record. As long as all the records contain the same number of fields, even if some of them are empty, these data files are still quite usable without any conversion. In the event that they contain unequal numbers of fields, they may still be used after conversion with the CONVERT utility (See Utilities) to insert a blank line between each record

d). Examples of valid data file records.

The following are all examples of valid formats for data files suitable for use with PROTEXT and could all be used in the same data file, although it is preferable to use only one format in any data file, if only for reasons of ease of creation and alteration.

```
J Smith
$
The Cottage
Little Deighton
Nr York
West Yorkshire

D Grey

15 Ambleside Road
Moortown
Harrogate
Yorks

"M Black"
"0539 412345"
'"The Leys"'
"Cartmel Fell"
"Ulverston"
"Cumbria

"P Brown","","16 High Street","Sandylands","Kendal","Cumbria"

R White, 9923 46143, 24 North Ave, Hampton, Keswick, Cumbria

J Green, $, 64 Park Road, Stockport, Cheshire, $
```

Note: In the examples where the fields are wrapped in quotation marks, a 'null' field may be represented either by two consecutive quotation marks or a dollar symbol wrapped in quotes.

e). Exporting data from other programs.

Note: This section is included here as it concerns data files, but it is suggested that it is only studied in detail after the chapters on mail merging have been read and understood.

The most common source of data, other than data files created with PROTEXT, is in the form of output from a database program. There are a number of databases available for the Amstrad PCW and CPC computers and most of them provide 'export' facilities. The degree of flexibility of output varies from program to program, as also does the terminology used in these programs.

The object of this section is not to give specific details on how to export data from each individual program, but to indicate the general principles involved and the terminology used. It will also be noted that some programs make it easier to export data than others and if a database is being purchased for use with PROTEXT, consideration should be given to this feature as well as to the normal database features.

Masterfile 8000 is the latest version of the well known Masterfile database programs and is written specifically for the PCW computers. It is ideally suited for use with PROTEXT mail merging as it has an unusually wide range of export options, ensuring that suitable data files can be created with little effort. It is used as an example of the type of options available, all or some of which MAY be available with other programs.

The 'Export an ASCII file' option in Masterfile is used from the Disc Function menu and the fields which are to be exported are then selected. These may be specified in any order and the order chosen does not matter to PROTEXT, with one exception, which is described later in this section.

Having selected the export fields, a number of options are offered, one after the other. The first of these is 'Data identifiers', which means that the field references may be exported as well. This is not relevant to PROTEXT, so 'N' should be selected. The second option is 'Surname Shuffle' which is a term unique to Masterfile and 'Y' would normally be selected.

Similarly, the 'Line breaks' option is unusual to Masterfile and refers to the fact that an address may be stored in a single field yet be displayed as separate lines. Selecting 'Y' will result in each line of the field being exported as a separate field, which is what PROTEXT requires.

It should be noted that for reasons which are explained in the relevant sections on mail merging, it is important that a field which contains these 'line breaks' should be specified as the last field to be exported if there is any possibility that the number of lines might not be constant in every record.

'Comma delimited' is a term frequently used in database export facilities and if this is selected, all fields will be wrapped in quotation marks and each field will be separated from the next field by a comma.

'Wrapped in quotes' is a similar option, but instead of fields being separated by commas, each field will be wrapped in quotes on a separate line.

The final Masterfile option is 'Record separators'. If 'Y' is selected, then each record will be separated by a blank line in the data export file. The only exception to this is when the comma-delimited option is selected, in which case an extra comma will be positioned at the end of each record, instead of a blank line between records.

The most commonly used selection with Masterfile 8000 would be:

Data Identifiers	N
Surname shuffle	Y
Line breaks	Y
Comma delimited	N
Wrapped in quotes	Y
Record separators	Y

Some or all of these options will be available in most database programs, though the methods used to achieve the same results differ widely. Some programs require a special 'report format' to be created, whilst others require 'processes' to be written.

The option to wrap fields in quotes, or 'comma delimit' is provided by some programs, but many are only capable of sending 'raw' data to a data file. In these cases, it will frequently be necessary to use the PROTEXT 'CONVERT' program on the resultant data file, in order to wrap fields in quotes or insert blank lines between records. Wrapping fields in quotes is desirable in case there are any commas or quotation marks within the fields, which would otherwise be treated by PROTEXT as indicating the limit of a field.

Due to the wide variations in facilities, it is not possible to give specific details for all programs, but with the above information and a trial export of data it should be possible to make virtually all export files suitable for PROTEXT. If all else fails, it will often be possible to load the file into PROTEXT and modify it as necessary by use of the REPLACE option.

A special note:

Many people will have discovered the frustration of trying to transfer data from an existing database program to a new database, where the format of the export files is incompatible with the new database. PROTEXT may be used to reconstruct the data file in the required format by creating a merging file to read the data from the existing file and then using the PRINTF command to send the output to a new file in the required form. A simple example of this process is given in the Appendix listing examples of mail merge files, together with an example of converting a date into a different format.

MERGE 2-6

3. SIMPLE MAIL MERGING

Note: For the remainder of the chapters on mail merging, stored commands will be described in a simplified fashion. A fully detailed appendix, giving technical details of each command, will be found at the back of the manual.

At many stages throughout this section of the manual it will be suggested that text and data files are saved. There are two reasons for this. Firstly, because many of the examples will be used at a later stage and expanded as new commands are introduced and secondly, many of the examples will readily convert into 'template' files, which can easily be modified to create mail merge files for a variety of purposes.

The concept of keeping 'template' files has been mentioned in the chapter on stored commands in the word processing section of the manual and is equally valid, if not more so, with mail merging documents. For many people the example files created in these chapters will be the only 'templates' they need, requiring only a few changes to suit different purposes. In this way, it is possible to quickly create new documents without the need to start from the beginning each time.

a). Getting started

The principle on which PROTEXT carries out mail merging is very simple. At the start of a document, PROTEXT opens a specified data file and reads the contents of each field, allocating them to variables. When all the specified variables have been filled, PROTEXT continues printing the document, inserting the contents of each variable into the text, wherever a variable reference is found. By the time the end of a document is reached, all the variables for a field must have been read. The process is then repeated, with further fields being read from the data file and the next copy being printed.

Variables are inserted into the text, at the appropriate position, by placing an ampersand ('&') immediately before and immediately after the variable name. For example, to insert a variable called 'name' into the text, it is simply entered as '&name&'. There is no restriction on the number of times each variable may be used in the text.

If variables are positioned in the text using tab markers, then the contents of the variable will be justified about the tab location when printed, in the normal way.

Note: Users should note that the contents of any variable positioned in the text using the decimal tab will automatically be positioned accordingly in the course of printing, enabling automatic justification of any tables of figures contained in a data file.

One of the most common forms of mail merging is where a number of letters are to be sent to individuals and their name and address is required on each letter.

b). Creating a Data file

At this stage, it would be a good idea to create a small data file, with only five or six records, for use with the examples which follow in the remainder of the chapter. The easiest way to create a small data file is simply to type it in, using PROTEXT. Enter the name on the first line and press RETURN then type in the address on separate lines, using RETURN at the end of each line. For the purposes of the example, make the records a mixture of three and four line addresses and for the time being, do not put any punctuation marks or quotes in the fields. After entering the last address line of each record, press RETURN a second time, so that there is a blank line between each record, and do the same at the end of the file. Save the file with the name ’DATFILE1’.

c). Reading the data

Initially, there are only three stored commands that may be required, two of which will be used in the majority of cases of mail merging. These are:-

>DF file {file}	Define data file.
>RU var {var}	Read variables unconditionally.
>RV var {var}	Read variables.

Note: The use of curly brackets indicates that optionally, one or more extra items may be specified. ’file’ may be any valid filename and ’var’ may be any word starting with a letter from A-Z or a-z and may contain letters, numbers, ’.’, ’_’ and ’?’.

DF Define data file or files.

Example: >DF file1 file2 ... filen

Description: The files should contain data which is going to be used for mail merging. Usually there will be only one file, but several are permissible. When each file is completed, the next one will be opened for reading. This command is used to tell PROTEXT the name of the file to be used, and so must occur before reading data from the file.

Note: If more than one DF command is required in a document, a CF command must first be used to close one file before opening the next (See Advanced mail merging).

Note: The drive letter may be specified, preceding the file name. eg. B:file1 A:file2.

RU Read variables unconditionally from the data file.

Example: >RU var1 var2 ... varn

RV Read variables from the data file.

Example: >RV var1 var2 ... varn

The number of variables named in these commands must be the same as the maximum number of fields in a data record, usually with one added for the blank line between records. They need not all be in one 'RU' or 'RV' command, but by the time a complete document has been printed once, all the fields for a record must have been read.

'RU' and 'RV' are very important commands, being the means of extracting data from a data file for use during mail merging. Whilst both commands are used in a similar fashion, with the same syntax, there are important differences between the way that the two commands work, the choice of which command to use being largely determined by the format of the data file. A thorough understanding of both these commands is important and the differences in their uses will be discussed at various stages in the manual.

Clear any text from the computer, using PROTEXT's 'CLEAR' command and enter the following example program. Save this as 'EXAMPLE1', as it will be used in later examples and modified as further commands are introduced.

Example - simple mail merge to read names and addresses from a data file and insert them into a letter.

```
>CO   EXAMPLE1
>PL 24                              ;for example purposes only
>CP OFF                             ;single sheet
>DF datfile1                        ;define data file
>RV name addr1 addr2 addr3 addr4 dummy        ;rem read variables

&name&
&addr1&
&addr2&
&addr3&
&addr4&

Dear &name&,
     Thank you for your letter about the insurance for &addr1& &addr2&
&addr3& &addr4&. We think that you will find the rates quoted.......
```

Before trying out the example, it is worth studying it to see what has been done. The first line is a comment line and is used as a reminder of what the document is. The 'PL 24' command in the second line and the 'CP OFF' in the next line are present only for the purposes of this example, in order that when it is printed to the screen, each record will be printed and then stop, so that it can be studied.

'DF datfile1' defines the name of the data file, so that PROTEXT can open it for reading. The following line, which in this first example makes use of the 'RV' command, is there to read the various fields of the record and allocate them to the variables. It is good practice to choose variable names that are related to the data that the variables will contain.

Special notice should be taken of the variable 'dummy', which will be used in the majority of mail merges and is used to read the empty line between each record.

'RV' has been used in this example for a reason. One of the special characteristics of 'RV' is that when it reads from the data file, it stops reading any more data as soon as it finds an empty field and automatically gives a null value to any remaining variables on that 'RV' command line which have not yet been allocated values.

With the data file that has been created, this is important, as some of the records have less fields of data than others. If the record was one that had only three address lines and all the variables were to be read and allocated, then the variable called 'addr4' would contain the empty line between records and 'dummy' would contain the name field from the next record, so that when the next record was read, the first field would be allocated to the variable 'name', which in this case would be the first address line. When using 'RV', as soon as the blank line is read, no further fields are read and any remaining variables are given a null value.

The remainder of the example is the text that will be printed and, as described earlier, wherever the data is required, the appropriate variable is entered in the text, wrapped in its markers.

To try out the example and save wasting paper, use the PRINTS (PS) command to direct the output to the screen instead of the printer. PS is particularly useful with mail merging, as it allows a check that everything is happening as expected, before printing it out properly.

One thing that will be noticed is that even though some of the addresses may have only three lines, PROTEXT closes up the blank line and does not leave unsightly gaps. The space between the last line of the address and 'Dear..' will always remain as one line.

If, for any reason, it is desired to retain the gap, PROTEXT has an alternative set of variable markers, '!' to replace the '&' and these can be used instead. Example '!addr4!'

Even more noticeable is the way that the remainder of the paragraph of text containing the address, is automatically reformatted during printing, ensuring continuity of the text.

The two commands used so far are the only ones required for the simplest of mail merges.

d). Comparing RV and RU

In order to see the different result obtained by using the 'RU' command, change the 'RV' command to 'RU' in 'EXAMPLE1' and repeat the PRINTS. When using 'RU', the names and addresses will soon get confused and mixed up.

Load the data file back into PROTEXT and go through the records, inserting a blank line at the end of any record that is missing the fourth address line, making each record effectively the same length. Save this amended data file with the name 'DATFILE2'.

Reload the text file 'EXAMPLE1' and change the name of the data file specified by 'DF' to 'datfile2'. Carry out another print, using 'RV' to read the data and everything will get confused again, but when replaced with 'RU', everything will work correctly.

From the above, it should be seen that the main use of 'RU' is with data files which contain records with the same number of fields in each record. Many database programs construct data files of this sort, which are suitable for use with PROTEXT. Except for the simplest of data files, containing only names and addresses, it is normally preferable to make sure that all records are of the same length, even if it means that either blank lines or lines containing the '$' marker, are used to signify an empty field. Some programs permit you to specify default text, to be used in the event of an empty field being found, in which case '$' would normally be specified.

'RV' can also be used to read data files which contain a constant number of fields, but if any of the fields in a record are null, then 'RV' will interpret these as the end of a record and give a null value to the remainder of the variables, which is not the required result. To see this in action, reload 'DATFILE2' and insert some telephone numbers after the name and before the address. Leave one or two of them as blank lines and save as 'DATFILE3'.

Reload 'EXAMPLE1' and alter the data file name to 'datfile3'. Change the 'RV' line to 'RU' and insert a variable called 'tel'. The line should now look like:-

```
>RU name tel addr1 addr2 addr3 addr4 dummy      ;rem read variables
```

Save this file with the name 'EXAMPLE2' and carry out another PRINTS. Everything will work correctly, as 'RU' reads all the fields, including any empty ones.

Note: When amending earlier examples, to create new ones, do not forget to alter the title in the first comment line to suit the new example number.

Note: It is necessary to read the telephone field and allocate it to a variable, even though it is not actually required for the document. It is essential to read all the fields of a data record, even though only certain of them may be required, otherwise PROTEXT will get out of step with the data records.

Now change the 'RU' to 'RV' and repeat the print. This time things will get out of order again, as 'RV' will cause PROTEXT to stop reading fields as soon as it comes across an empty field.

There are two ways to get round this problem with 'RV', the simplest one being to use the '$' marker, as described in the chapter on Data files, in place of an empty line wherever there is an empty field in a record, but this is not always possible with data files created by other programs. The second method requires the splitting of the 'RV' command on to a number of lines and is covered in detail in the chapter on advanced mail merging.

As a general rule, 'RU' is used to read the contents of data files having records which contain a constant number of fields, whatever the format of the file, whilst 'RV' is used for simple data files where the last field in a record may, or may not, be present and for a number of specialised uses, discussed later in the manual.

e). Creating data by other means

So far, the only way that it has been possible to insert text into a document has been by reading it from a data file, but there are two commands which enable data to be provided in other ways.

```
>AV (message) var (number) {(message) var (number)}     Ask variable
>SV var = string of text {var = string of text}          Set variable
```

Note: The use of normal '( )' brackets indicates that the parameter is optional, whilst curly brackets, as before, indicate that one or more extra items may be specified. 'message' is a piece of text to be displayed on the screen and 'number' is a number between 1 and 255.

AV Ask for variable from keyboard.

Examples:

```
>AV "Name",name,20,"Address line 1",addr1
>AV name,addr1,addr2,addr3,addr4
```

Description: The messages are displayed on the screen as prompts. If no message is specified for any particular variable then the variable name is displayed on the screen followed by a question mark. Text may then be typed in, and this is assigned to the variable. If 'number' is specified this sets the maximum length of text that may be entered. The maximum length permitted is 255 characters, and this is the default value.

SV Set variable to a value.

Examples:

```
>SV firstname="Peter"
>SV name=firstname+" Smith"
>SV counter=1
```

Description: This command automatically sets a variable to a given value, which can then be used in the text in the same way as other variables. The 'string of text' may be either a string of characters, which must be enclosed in quotes, or it may be another variable, in which case the variable name is used, without quotes, or it may be a combination of both, in which case they are joined with a '+' sign, as in the second example. Valid decimal numbers may also be used, either with or without quotes.

There are many other things that can be done with the 'SV' command, but they are beyond the scope of this introductory chapter and are discussed in detail in the later chapter on variables.

The easiest way to see the effect of these two commands is to modify 'EXAMPLE2', used earlier in the chapter. The following two lines should be inserted after the line which starts '>RU'.

```
>AV "Enter Date " date
>SV year="1986"
```

and the following line after 'Dear &name&'

```
>CE  Renewal date - &date& &year&
```

Save the modified file as 'EXAMPLE3' and use PRINTS again, to see the results.

The main commands concerned with simple mail merging have now been covered but there is one final command which should be mentioned in this chapter, which, whilst not strictly being mail merging in the normal sense, is a simple and yet very powerful command.

f). Inserting separate text files

'IN' is a very useful command which permits the merging of commonly used pieces of text into a document. For example, the user's address could be kept in a file on disc called 'address' and instead of loading it into memory each time a letter is written, it would only be necessary to put '>IN address' at the start of each letter. As another example, if the same paragraphs of text are frequently used in business letters, these could be saved on disc with appropriate names and called as necessary.

IN Insert file into text.

Example: >IN filename

Description: This command may be used at any point in the text of a document and the named file is read from disc and the contents printed. When the end of the file is reached, printing of the original file continues at the line after the IN command. The file is not loaded into memory, merely printed. Any number of 'IN' commands may be issued.

Note: Drive letters may be specified preceding the filename, in the normal fashion.

Another use of 'IN' is to permit the continuous printing of a number of files as one document. A text file consisting of the following entries will, when printed, result in 'file1' being printed, followed by 'file2' and finally 'file3'.

```
>IN file1
>IN file2
>IN file3
```

One of the advantages of this is that page numbering, Headers, Footers, or margins set at the start of the document will follow through correctly, unless altered by any stored commands in the files being printed. In this way, it would be possible to print out a whole book, from beginning to end, in one run, with headings and page numbers running correctly through.

Further uses of the 'IN' command are detailed in the chapter on 'Advanced mail merging'.

g). Address labels

Finally, before moving on to more advanced mail merging, this very common and straightforward use of PROTEXT should be mentioned.

```
>CO Address labels
>---------------------------------------|-----------------------------------R
>PL 9          ; Number of lines to repeat label (6 per inch)
>ZM            ; Set all margins to zero
>DF datfile1   ; data file to read
>CO Next six lines read contents of one or two records
>RV name,addr1,addr2,addr3,addr4,dummy
>IE
>SV nexname="",  nexaddr1="", nexaddr2="", nexaddr3="", nexaddr4=""
>EL
>RV nexname,nexaddr1,nexaddr2,nexaddr3,nexaddr4,nexdummy
>EI
!name!                                  !nexname!
!addr1!                                 !nexaddr1!
!addr2!                                 !nexaddr2!
!addr3!                                 !nexaddr3!
!addr4!                                 !nexaddr4!
>PA
```

There are several points that should be noted with the above example.

a). 'PL' should be set to a value equal to the number of lines that are required to move from the top of the first label row to the top of the next label row. The normal setting of a printer is six rows to an inch. In the above example, the setting is therefore for address labels with a one and a half inch repeat.
b). 'ZM' must be used to set all margins to zero.
c). 'RV' is used, as 'datfile1' contains variable length addresses.
d). Full explanation of IE, EL and EI is given in the following two chapters.
e). Note also that there should be no blank lines after the >PA stored command, which is used to force a new page and the labels to be advanced to the start of the next label.
f). The alternate set of variable markers, '!', have been used to ensure that even if a field is empty, the empty line will be printed, ensuring that the printer always moves on the correct amount to the next row of labels.
g). TAB should be used to position the right-hand column of variables so that the addresses are lined up correctly.

4. CONDITIONAL PRINTING AND MAIL MERGING

Conditional printing and conditional mail merging are really two different things, but, as they are frequently used together, both will be covered at the same time.

A conditional mail merge is one where text will be either printed or not printed, depending on whether certain conditions are met. It may affect whether all or only part of the text is printed, or even whether one piece or an alternative piece will be printed.

A little thought will give some idea of the scope of these commands and once the principles are understood, some time spent experimenting will soon bring to light numerous possible uses. There are certainly far too many uses to cover all of them in this manual.

There are four commands which are directly concerned with straightforward conditional printing, some of which can be used on their own, but most of which are used in conjunction with each other.

>IF condition	If condition is true then print block.
>EL	Else print other block if IF condition false.
>EI	End of an IF block.
>SK condition	Skip copy if condition true.

Before considering these commands, it is important to have a thorough understanding of what is meant by 'condition'. A condition is where a string (piece of text), a number, or a variable, is compared with another string, number, or variable. The result will be either 'true' or 'false'. For the purpose of these commands, there are eight possible comparisons that can be made:-

=	equal to
< >	not equal to
<	less than
< =	less than or equal to
>	greater than
> =	greater than or equal to
IN	is contained in (e.g. "jo" IN "Mr Jones" is true)
NOTIN	is not contained in

Decimal numbers are compared numerically. A numeric argument is one which the program recognises as a valid decimal number, such as '37' or '5.4'. If a valid number is found, it will be compared numerically, otherwise it will be compared character for character. 'IN' and 'NOTIN' will automatically carry out a character comparison, but the other comparators will attempt a numeric comparison first, before defaulting to a character comparison if the comparison is numerically invalid. For the purposes of a character comparison, no account is taken of whether a letter is upper or lower case and both are considered equal.

The first and simplest of these commands is:-

IF Print if condition true.

Example:

```
>IF "Married" NOTIN status
>IF firstname IN fullname
>IF age>40  ; (i.e. true if age is over 40)
```

Description: The condition is tested and if true the block of text which follows is printed (and any stored commands acted upon). The block of text extends until either an 'EL' command, an 'EI' command, or the end of the text. If the condition is false then everything within the block is ignored, including stored commands.

Note: IF blocks may be nested to a depth of 7.

EI Endif. End of IF block.

Description: This command marks the end of an 'IF' block. If the 'IF' block was nested within another, then printing reverts to the remainder of the previous 'IF' block, otherwise unconditional printing resumes.

Note: Every 'IF' command must have a matching 'EI' command.

Load 'EXAMPLE3' and add the following lines to the end of the text, save as 'EXAMPLE4' and again use PRINTS to see the effect:-

```
>IF tel>""
        If you would be kind enough to telephone us....
>EI
```

A check is made to see whether the contents of 'tel' are greater than nothing and any of the records which contain a telephone number will have the extra line of text added to the document, but if the telephone field is empty, then it will be omitted.

EL Else.

Description: The block of text following this command is printed if the condition in the previous IF command was false, but is ignored if that condition was true. The block extends until an EI command or the end of the text.

Note: IF blocks may be nested within EL blocks.

'EL' is a very useful command, as it permits the inclusion of an alternative piece of text, to be used in the event that an 'IF' condition is found to be false. ie. "IF this condition is true then print this block, ELSE print the next block".

Add the following lines to 'EXAMPLE4', immediately before the '>EI' command, save as 'EXAMPLE5' and again use PRINTS.

```
>EL
        If you would be kind enough to write to us ....
```

This time either one or the other message is printed, depending on whether a telephone number is present in the field or not.

'Nesting' has been mentioned in the previous commands and this is the process where one IF block is situated inside another. IF blocks can be nested to a depth of seven. It is not possible to have IF blocks overlapping each other and if this is attempted, either the results obtained will not be what was expected, or else an error message will be given by PROTEXT, depending on the precise situation. The example below will give some indication of nesting blocks.

For example.

```
>IF tel>""                  ; (IF 1) if condition 1
>IF name="Smith"                  ; (IF 2) then if condition 2
   Your name is Smith and you have a telephone...

>EL                               ; (EL 2) else if not condition 2
>IF name="Jones"                        ; (IF 3) then if condition 3
   Your name is Jones and you have a telephone...

>EL                                     ; (EL 3) else if not condition 3
   Your name is not Smith or Jones, but you have a telephone...

>EI                                     ; (EI 3) end condition 3
>EI                               ; (EI 2) end condition 2
>EL                         ; (EL 1) else if not condition 1
   You do not have a telephone...

>EI                         ; (EI 1) end condition 1
```

From the above example, it can be seen that condition 2 is nested completely inside condition 1 (between the IF and the EI commands). In turn, condition 3 is nested entirely within the EL part of condition 2.

It may be found convenient when creating documents of this sort to do so in a particular order and to annotate each command line with comments. In the example, each level of nesting has been marked with the command to which it is associated and the comments have been tabulated across the document to make it clear which commands are related. It may also be found convenient to insert all of the related IF, EL and EI commands at the same time and then to insert the text within them as appropriate.

SK Skip printing if condition true.

Examples:

```
>SK "Mr" IN name       ; If name contains 'Mr', not printed
>SK add4="London"      ; If add4 is "London", not printed.
```

Description: The condition is checked and if true the remainder of the text is skipped. This allows the selective use of data from a file.

Note: The same effect could be achieved by the use of 'IF', but 'SK' is preferable where it is required to omit printing the remainder of the text for that record, as the effect is immediate, whereas 'IF' will continue to check through the document for further commands, until the end of the document is reached.

The easiest way to see how this command works is to insert the following line into 'EXAMPLE5' immediately after the '>RU' line and save it as 'EXAMPLE6', before printing.

```
>SK addr4 > ""
```

This line tests to see whether the variable 'addr4' contains anything or not and if it does, then the condition is true and the remainder of the document will be skipped. In this case, as the test is made before anything is printed, the complete record will be skipped. The effect of this is that only records with three address lines will be printed. Another example would have been to use '>SK tel=""', which would have resulted in only records containing a telephone number being printed.

The commands covering the simplest forms of conditional mail merging have now been covered, but there are several other commands which extend the possibilities further and these are covered in the chapter on advanced mail merging.

To make the most of PROTEXT it is recommended that some time is spent experimenting with the various commands covered so far, before continuing with the chapter on advanced mail merging.

MERGE 4-6

5. ADVANCED MAIL MERGING

Many new commands were described in the earlier chapters and in this chapter further new commands will be introduced, as well as extensions to a number of commands already covered in their simpler forms.

a). More new commands

>CF	Close data file.
>IE	If data file exhausted then print block.
>ID var	If variable defined then print block.
>IU var	If variable undefined then print block.
>RP	Repeat.
>UN condition	Until the condition is true.
>WC	Write file close.
>WF filename (A)	Open file for writing.
>WF ON/OFF	Turn writing to file on and off.
>WM text	Write message to file.

(i) Closing data files

In earlier chapters the 'DF' command was used to open data files for reading. Normally a data file will be closed automatically when the last of the data has been read, but there are a number of occasions when it may be preferable to close a data file prematurely.

One of the restrictions mentioned previously was the fact that it is only possible to have one data file open for reading at a time. For many purposes this will be no problem, but it can be useful to be able to open another data file, having extracted the required information from an earlier file. 'CF' is provided expressly for the purpose of closing one data file, in order that another data file can be opened in a subsequent command.

CF Close data file.

Example: >CF

Description: CF closes a previously opened data file, permitting a further file to be opened for reading.

Note: A CF command must be issued between two DF commands, otherwise the second DF command will be ignored.

Note: A data file closed by the use of CF can be reopened at a later stage with a further DF command, but the contents will be read from the start of the data file each time.

(ii) Special conditional commands

The use of the 'IF' stored command has already been explained in the chapter on conditional printing and mail merging, but in addition to 'IF' there are three further commands which can be used instead of 'IF' for specialised purposes. These are:

IE If exhausted.

Description: The data file is checked and if it contains no further data, then the block marked by 'IE' is printed. It is usually positioned at the end of a document so that when the current print is completed, the data file will be checked to see if there is any further data. This command is particularly useful for displaying a summary when a mail merge run is completed.

ID If defined.

Example: >ID var1

Description: ID is a special form of the IF command, which returns 'true' if the variable has been defined, and 'false' if not. If true, the text following will be printed, or any stored commands acted upon, until the matching 'EI' is found. This command is very useful when used in conjunction with AV.

IU If undefined.

Example: >IU variable

Description: IU is the opposite of ID, in that if the variable has NOT been defined, then the condition is true.

A variable has been defined if it has already been given a value earlier in the text printing, by any of the commands AV, RU, RV, or SV. This allows, for example, an AV command to be executed only once and the resultant value to be used for the remainder of the mail merging.

The following example gives a demonstration of the use of both the IE and IU commands. Reload 'EXAMPLE4' and make the following alterations:-

Insert this line before '>AV "Enter date" date':

```
'>IU date'
```

Insert these lines following '>SV year="1986"':

```
'>SV total=0'
'>EI'
'>SV total=total+1'
```

Insert these lines at the end of the file:

```
'>IE'
'>SV total=total[w1]'
'>DM You have printed &total& copies.'
'>EI'
```

Save the text as 'EXAMPLE7' and using PS it will now be found that at the start of printing, the date will be requested. This has the effect of defining the variable 'date', so that in subsequent copies, as 'date' has now been defined, the date will not be requested and the existing value will be used for all further copies. Finally, when all copies have been printed, a message will be displayed on the screen stating how many copies were printed.

'ID' is similar, but the text which follows it will only be printed if the variable has already been defined.

(iii) Repetitive looping

PROTEXT also provides the means to carry out repetitive actions in a document. These are carried out using the following commands.

RP Repeat.

Example: >RP

Description: 'RP' marks the start of a block of text that will be repeated until a condition specified in 'UN' is found to be true. Normal printing will then continue.

UN — Until a condition is true.

Example: >UN "N" < name

Description: 'UN' marks the end of a Repeat - Until loop. When PROTEXT finds this command, it checks to see if the specified condition is true and if not, jumps back to the earlier 'RP' command. This sequence will repeat until the condition is true, at which point the loop will be exited and normal printing of the rest of the document will continue.

Note: A Repeat - Until loop will always be carried out at least once.
Note: When RP UN is used to carry out printing, it should be remembered that any natural page breaks may be altered and it may often be desirable to use a forced page break ('>PA n') at some stage after completion of the loop.

These commands can be used in a number of ways, either to repeat a block a set number of times, or until a certain condition is met.

Example to create a data file:

```
>ZM
>RP
>AV "Enter name " name
>IF name>""
>AV addr1 addr2 addr3 addr4
>SV dummy=""
!name!
!addr1!
!addr2!
!addr3!
!addr4!
!dummy!
>EI
>UN name=""
>ST
```

In the above example, the name is requested and checked to see that it contains text. If so, then the address is requested and the text printed, before looping back to repeat the process. If 'name' contains nothing, then the check in 'UN' will be found true, the loop will be left and printing will stop. Using PRINTF, to print to a disc file, will create a file suitable for use as a data file.

Note: The alternate markers, '!', have been used to ensure that the same number of lines are sent to the data file each time, even if a field is found to be empty.

Example to request information a given number of times.

```
>---------------------!----------!-----------!--------!----------R
Description of goods  Unit cost  Quantity    Discount
>SV counter=0
>RP
>AV descrip cost number disc
>SV counter = counter+1
>--------------------------.----------.-----------.--------------R
&descrip&            &cost&   &number&      &disc&
>UN counter=10
```

In this example a variable called 'counter' is set to a value of zero and when the text is printed, requests are made for a description of the goods, the quantity and the discount. The value of the variable is increased by one and the loop repeated until the value of 'counter' reaches ten, when the loop will be left and the remainder of the text printed.

(iv) Writing to a disc file during printing

PROTEXT provides the facility to write part or all of a document to a disc file during the course of printing. There are three commands which provide this facility.

WF Open write file to disc.

Examples:

> WF filename
> WF filename A

Description: A file of the specified name will be opened for writing. The '>WF ON/OFF' commands determine whether the text is written to the file or not, when printing takes place.
If the 'A' suffix is used, any existing file of that name will have the data appended to it.

WF ON/OFF Turn on and off writing to the file.

Examples:

```
>WF ON
>WF OFF
```

Description: When printing takes place, if writing is turned on, all text is sent to the file, including stored commands, printer control codes. rulers etc. at the same time as the document is printed normally. 'WF' may be turned on and off as many times as required during the course of a document.

WC Write file close

Description: Closes the file being written to by the above commands. If no WC command is given the file will be closed when printing finishes.

Note: '>WC' must be used to close a write file before an '>EX' command is used, if the >EX command involves any disc access.

This is an extremely powerful function as it can be used to create individual documents, or parts of documents, during the course of a mail merge printout. When it is used, the file that is created is a PROTEXT document and any stored commands, rulers or printer control codes will be sent to the file as well as the text that is being printed. The final result is a document which can be printed out as a normal file. The only difference from the original document is that where variables had been used in the original document, the new file will contain the data instead.

This command is particularly useful when used in conjunction with stored commands which request input from the keyboard ('>AV') and can be used to provide a copy of what was typed in at the keyboard, together with the remainder of the document. One specific use would be where PROTEXT was being used to create invoices from information input at the keyboard. If the invoice is printed with 'WF' turned on, then a copy of the invoice will be provided on disc.

Care should be taken when using this command to ensure that the appropriate ON and OFF commands are positioned in the correct places in the document. If the whole document is printed with 'WF' turned on, then the result will be a file containing all the stored commands, except the 'WF' commands, as well. If the WF commands are positioned only around the text and are turned off when stored commands are being printed, then the result will be a facsimile of what has been sent to the printer, including printer control codes. It can be seen that it is possible to use these commands in a number of ways.

The following examples will give some indication of how the commands are used:-

```
>WF "agename.lst"
>SV counter=0
>RP
>SV counter= counter+1
>AV "Enter name" name "Enter age" age
        This is a list of ten people and their ages
>WF ON
        This person is called &name& and is &age& years old.
>WF OFF
>UN counter=10
        Remainder of text .......
```

This example will result in a file called 'agename.lst' being saved to disc, which will contain nothing more than consecutive lines stating 'This person is called nnnnnn and is yy years old', where 'nnnnn' is each person's name and 'yy' is their age. The heading and the remainder of the document will not be sent to the disc file. The document which is printed will contain the heading, the name and age lines for ten people and the remainder of the text.

Compare this with the following example:-

```
>WF "copy.let"
>WF ON
>TM 5
>SM 8
>------|---------------------------------------------------------R
Dear Sir,
        Thank you for your letter.....

>WF OFF
```

This example, when printed, will create a file called 'copy.let', which will contain not only the text, but all the stored commands and rulers which are within the ON and OFF statements.

b). Displaying messages during printing

So far, there has been no means of knowing which record is next to be printed, until it is actually being printed. There are some occasions when it would be convenient to decide at printing time whether a record was to be printed, or even what was to be printed. PROTEXT provides a number of commands to display messages on the screen. These commands are 'CS', 'DM', 'ST' and 'WT', which have already been described in the chapter on stored commands. What was not mentioned was the fact that variables could be used in the messages, thus providing a method of displaying their contents.

```
Examples: >DM     The next record is &name&.
          >CS     Do you want to print the record for &name&?
          >WT     Print &name&? STOP to stop or any key to print.
          >ST     There are no more records to print called &name&.
          >WM     This letter is to &name&
```

In all cases the variable must have been defined before the message line and following the message line or lines, suitable stored commands should be inserted to take the appropriate actions.

Example: Modify 'EXAMPLE2' to the form shown below and save as 'EXAMPLE8

```
>CO EXAMPLE8
>PL 24
>DF datfile3
>RU name tel addr1 addr2 addr3 addr4 dummy
>CS  Do you want to print the record for &name&?
>AV " "print
>SK print< >"Y"
&name&
etc

>DM  Copy to &name& has now been printed
>WT  Press any key to continue
```

The above example goes through the complete data file, prompting with the name from each record and asking whether it is to be printed or not and displays messages after printing, to inform who it was for and to request the user to 'Press any key to continue'.

c). More on the RV stored command

In the earlier chapter on simple mail merging, many of the differences between the way that 'RU' and 'RV' function were described, as well as the fact that 'RV' was particularly suitable for use with the simplest of data files. Mention was also made of the fact that it could be used to read more complicated data files and in fact had definite advantages for certain purposes.

It will be recalled that one of the special features of 'RV' was the fact that as soon as it came across an empty field it ceased to read any further data fields and allocated a null value to any remaining variables on that command line. This was also one of the things that caused problems when trying to read records which contained empty fields, as it automatically gave a null value to the remaining variables, even though they were present in the record. This also caused the reading of the records to get out of step.

One solution was to make use of the special PROTEXT indicator '$', to indicate to PROTEXT that the field was present, but with a null value. In this way 'RV' would continue to read the remainder of the variables.

There is another solution to the problem, which can be used where the records contain empty fields, without the '$' marker, of the sort that may be created by some programs. The answer is to split the reading of the variables into a number of lines, in such a way that any field which may be empty is the last field on that command line. Several lines can be used if there is a possibility that a number of fields might be empty.

In the following example it is assumed that the data file contains records in which the second, sixth and seventh fields might be empty in some records and that each record consists of eight fields plus a blank line.

```
>DF datafile
>RV name tel
>RV addr1 addr2 addr3 addr4
>RV occupation
>RV salary dummy
```

When RV reads 'tel', if it finds it to be empty it will give it a null value, but as there are no more variables on that command line, PROTEXT will move on to the next line and continue to read the next few variables. As soon as a null value is discovered, the remainder of the line will again be given null values and so on to the end.

d). More on the IN stored command

In the chapter on simple mail merging a description was given of the use of the 'IN' command to permit the consecutive printing of a number of files, whilst still maintaining continuity of page numbers, headers, footers etc. Mention was also made of the fact that 'IN' could be used to cause other files to be printed at any stage in the printing of a document, permitting the merging of commonly used blocks of text for example.

A third use, is the ability to use 'IN' at the end of a document to call another document. This can be repeated as many times as required. In general, a more satisfactory way of doing this, however, would be to use the previously mentioned list of '>IN filename' commands.

One further use of 'IN' which can be very useful is to use it in a recursive fashion, to call itself. This will permit the same document to be repeated as many times as required. The method of doing this is very straightforward and simply involves using the '>IN filename' command at the end of the file and then saving the file with the name used in the 'IN' command. In this way, when the end of the file is reached, it automatically calls itself. An example of this is shown in the appendix containing sample mail merge documents.

6. MAKING THE MOST OF VARIABLES

There are a number of things that can be done with variables which have not yet been covered and this chapter introduces methods of manipulating the contents of variables in various ways, as well as introducing a number of other uses for variables.

a). Joining variables and text

When 'SV' was introduced, mention was briefly made of the fact that strings of text or variables could be joined together to make one variable. This can be particularly useful on many occasions. In 'EXAMPLE3', 'SV' was used simply to give a variable a value and then two variables were positioned next to each other in the text, so that when printed, they appeared as one.
eg. &date& &year&. It could have been done more simply by joining the two at an earlier stage and then using just the one variable in the text.

For example:

```
>SV year=" 1986"
>AV "Enter Renewal Date " date
>SV date=date+year
```

or more simply:

```
>AV "Enter Renewal Date " date
>SV date=date+" 1986"
```

Note: If variables which are to be joined together contain valid numeric values, then the result will be the numeric total of the two variables, which may not be the required result.
For example.

```
>SV num1="24", num2="10", num3=num1+num2
```

will result in 'num3' containing "34.00", not '2410'. If the result is intended to take the form of the characters being joined together, this can easily be achieved by using:-

```
>SV num1="24", num2="10", num3=" "+num1+num2, num3=num3[2:]
```

which, because the first character being a space, is a non numeric argument has the desired effect. The last statement, 'num3=num3[2:]' has the effect of removing the unwanted space at the start. See the following two sections for details of splitting variables and numeric calculations.

There are numerous uses to which this ability to join variables and strings of text can be put, but one in particular is worth mentioning.

No punctuation was used in the addresses of the earlier examples, with the exception of a full stop at the end of the first sentence, for a very good reason. Frequently the effect of "personalised" circulars is spoilt by unsightly gaps in the text and misplaced punctuation marks. PROTEXT takes care of the gaps automatically, but selecting the correct punctuation is a bit more difficult.

Commas and full stops could have been placed in the text, after each of the variables, so that they were printed. For many simple mail merges this would have worked satisfactorily as long as all the variables contained text. If some of the variables contained nothing, the result would bȩ punctuation marks all on their own, or perhaps, two together. When printing 'EXAMPLE2', it may have been noticed that sometimes the full stop at the end of the first sentence was next to the address and sometimes it was separated because the last address line was missing.

Another way would be to put the punctuation in the data file as part of the text for each field, with the fields wrapped in quotes. In many cases this would work satisfactorily, but it has a number of disadvantages, not only because it becomes necessary to enclose fields in quotes in the data file, but because the situation can arise where the punctuation mark is not the one required in the text.

The solution is simple, once the principle is understood. Using 'EXAMPLE2' from the earlier chapter, if the record was one where the address consisted of only three lines, then the third (and last) line of the address would require a full stop, but if the address consisted of four lines, then the third address line would require a comma and the fourth a full stop. The easiest way to understand this is to look at an example of what is required. Load 'EXAMPLE2' and modify it before saving the modified version as 'EXAMPLE9':-

```
>CO   EXAMPLE9
>PL 24
>CP OFF
>DF datfile3
>RU name tel addr1 addr2 addr3 addr4 dummy
>IF addr4>""
>SV addr3=addr3+"," addr4=addr4+"."
>EL
>SV addr3=addr3+"."
>EI

&name&,
&addr1&,
&addr2&,
&addr3&
&addr4&

Dear &name&,
        Thank you for your letter about the insurance for &addr1&,
&addr2&, &addr3& &addr4& We think that you will find the rates quoted
.......
```

In this example, it is known that 'name', 'addr1' and 'addr2' will always contain something, so it is sufficient to simply insert the punctuation in the text for these address variables. The 'IF' command checks to see whether 'addr4' is an empty variable. If not, it adds a comma to 'addr3' and a full stop to 'addr4', otherwise 'EL' puts a full stop on the end of 'addr3' instead. In this way, when printing takes place, the appropriate punctuation is printed as part of the variable. If there had been a possibility that an address could have consisted of only two lines, then an 'IF' command would also have checked to see whether 'addr3' had been a null string.

b). Splitting variables

As well as joining variables and text together, PROTEXT provides the facility to split the contents of variables down into smaller parts, either by character or by word. This is achieved by specifying which characters or words are required in the following way:-

var[a:b]	from character ’a’ to character ’b’ inclusive.
var[a:]	from character ’a’ to the end.
var[a]	character ’a’ only.
var[Wa:b]	from word ’a’ to word ’b’ inclusive.
var[Wa:]	from word ’a’ to the end.
var[Wa]	word ’a’ only.
var[w-1]	the last word only.

If only numbers are specified, then PROTEXT will split the contents on a character basis, but if the first number is prefixed with a letter ’W’, then the variable will be split by words. The last option is a special case and will select the last word only.

Note: In this context, ’word’ means any group of letters or numbers delimited by any of the following - a space, full stop or comma. Thus for a variable containing ’12.45’, var[w1] will give ’12’ and var[w-1] will give the result ’45’

Examples:

If a variable called ’name’ contains ’Robert Stephen Smith’, then the following will be true:-

name[3:6]	is ’bert’
name[10:]	is ’ephen Smith’
name[3]	is ’b’
name[W1:2]	is ’Robert Stephen’
name[W2:]	is ’Stephen Smith’
name[W2]	is ’Stephen’
name[W-1]	is ’Smith’

Using 'Robert Stephen Smith' as an example again, the following values for the new variable will result:-

```
>SV newname=name[W1]+" "+name[W-1]   gives 'Robert Smith'
>SV newname=name[W2:]                gives 'Stephen Smith'
>SV newname=name[1:3]+" "+name[W-1]  gives 'Rob Smith'
>SV forenames=name[W1:2]             gives 'Robert Stephen'
>SV newname="Mr "+name[W-1]          gives 'Mr Smith'
```

It is also possible to use this form of splitting when checking conditions, using 'IF', 'SK' and 'UN'.

Examples:

```
>IF "Smith" IN name[W-1]        block is printed.
>SK name[W-1]>"M"               remainder is not printed.
>UN name[W1]="Robert"           repeat loop ends.
```

c). Using variables in stored commands

The use of variables in the stored commands concerned with displaying messages has been covered in the previous chapter, but it is also possible to use them in the majority of other stored commands, but in a much more sophisticated fashion.

The contents of a variable can be used in a stored command to define conditions, values and even other variable names in other stored commands. The easiest way to describe the use of this facility is by listing a few of the possibilities, but a full description of all the options is beyond the scope of this manual and it is suggested that some time spent experimenting with stored commands and variables will bring to light numerous possible uses.

Examples:

```
>AV "Name of file " filename
>IN &filename&

>AV "Start at page " startat "End at page " endat
>SA &startat&
>EA &endat&

>AV "No. of copies " copyno
>NC &copyno&

>AV "Enter Data file number" number ; if 3 was entered
>DF "datfile"&number& ;will open file called 'datfile3'
```

d). Numeric arithmetic

Numeric arithmetic is also possible with PROTEXT and calculations can be carried out using the ’+’, ’-’, ’*’ and ’/’ operators. Compound calculations are permitted and the calculations are carried out in the order that they are encountered rather than by operator priority. Brackets may NOT be used to force calculations. Invalid numeric contents of variables will result in an error message and printing halting.

Calculations can be carried out on the contents of variables, as long as they are of numeric type and the results stored in further variables.

All calculations are accurate to two decimal places and the results are substituted into the text showing two decimal places.

Note: The only time a number will be displayed without the two decimal places is if the number is an integer and was assigned directly to the variable. For example, if ’cost’ is defined by ’>SV cost=9’, then ’&cost&’ will display ’9’. To display ’9.00’ add the command ’>SV cost=cost+0’. As the variable value has now been calculated, it will return a result to two decimal places.

Alternatively, if a number is required without the decimal places being shown, adding the line ’>SV number=number[W1]’ will return only the integer part of the number to the variable ’number’.

Example:

```
>AV "Unit cost? " price,"Quantity? " quantity,"Discount? " discount
>IF discount=""
>SV discount=0
>EI
>SV vat=15
>SV disccalc=100-discount/100 vatcalc=100+vat/100
>SV total=price*quantity*disccalc
>SV totincvat=total*vatcalc
>SV pound=totincvat[W1]
>SV pence=totincvat[W-1]
```

The above example illustrates a number of possible ways to manipulate and calculate numeric variables, including an example of obtaining the equivalent of INT(num) and FRAC(num). In this case the variable ’pound’ will contain the integer and the variable ’pence’ the fraction.

Numeric arguments can also be used for making comparisons and in this case ’>’, ’<’, ’< >’, ’> =’, ’< =’, and ’=’ can all be used.

Example:

> IF age > 40; true if age is greater than 40

Variables can also be used as counters and markers in stored commands, to cause actions to happen after a certain number of times.

Example:

```
>SV counter=0
>RP
>AV description cost quantity
>SV total=cost*quantity  counter=counter+1
>-----------------.---------.-------------.----------.----R
     &description&    &cost&    &quantity&    &total&
>UN counter=10
>CO  remainder of text follows
```

In the above example, the counter is set to zero at the beginning and a description, details of cost and the quantity are requested. 'SV' then calculates the total cost, assigns it to a variable called 'total' and increments the value of 'counter'. The next line prints the variables and 'UN' checks to see whether the value of 'counter' has reached ten. If not, then the loop is repeated, otherwise the loop will be exited and the remainder of the text printed.

MERGE 6-8

UTILITY PROGRAMS

1. INTRODUCTION

PROTEXT is supplied with a number of separate utility programs to carry out certain special tasks. These are separate programs which have been specially written so that they can be used in a number of ways.

These programs would normally be used from within PROTEXT, from command mode and will, to all intents and purposes, act in exactly the same way as any other command.

Whereas calling a separate program will normally result in the loss of any data in the program already in use, these programs can be called and on completion will automatically return to PROTEXT with any text in memory at the time the utility was called, still intact (See note below).

Note: CPC6128 users with only one disc drive must save any files to disc before using any of these programs and on completion it will be necessary to load the file back into memory in the normal way. If the file is not saved before entering the utility program, it is probable that part or all of the document will be lost.

These programs may also be used on their own, directly from CP/M command mode, simply by using their name and on completion they will return to CP/M command mode.

There are four programs supplied, as follows:-

CONFIG.COM — A program to re-configure many aspects of PROTEXT to suit individual preferences, such as insert or overwrite mode by default and default page length.

SETPRINT.COM — A program to set up a 'printer driver' which tells PROTEXT how to access the printer(s), giving details of control codes, type of printer etc.

CONVERT.COM — A program to convert text and data files created by other programs into forms more suited to their use in PROTEXT.

DCOPY.COM — A program to copy the entire contents of a disc on to another disc. ie. to make backups of important discs.

The above four programs are described in detail in the following chapters.

UTILITY 1-2

2. CONFIG - PROTEXT CONFIGURATION PROGRAM

PROTEXT is provided already configured to suit the requirements of the majority of users, but the CONFIG utility program is provided to enable many of the features of PROTEXT to be reconfigured to suit individual preferences. The default settings of many of the editing features, such as right justification, insert or overwrite mode may be altered so that when PROTEXT is loaded they are automatically pre-set to the required settings. Any key on the keyboard may be reconfigured to serve a different purpose and different printer drivers may be loaded by default, in addition to the facility to change the default printing options to suit the type of stationery to be used.

Note: Most users will find the default settings to be entirely suitable for their purposes, but it is important that CONFIG is used as soon as possible, in order to make a small number of adjustments that may be required, according to the printer and the number of disc drives being used.

Once CONFIG has been used to change the parameters, a configuration file called PROTEXT.CFG will be created. This will automatically re-configure PROTEXT whenever it is loaded, as long as the PROTEXT.CFG file is present on the disc. In the event that such a file is not found, PROTEXT will simply be set up with the default values supplied.

If a configuration file is created and is later found not to be entirely suitable, then all that is required is to use CONFIG again to adjust the values and a new PROTEXT.CFG file will be created.

CONFIG may be used either from CP/M command mode, or from within PROTEXT by typing 'CONFIG' from command mode, at which stage the screen will clear and an opening menu will appear, listing the following options:

1 - Set editing options
2 - Set spelling checker options
3 - Set printing options
4 - Set general PROTEXT options
5 - Set keys for PCW8256/8512
6 - Set keys for CPC6128
7 - Set printer driver options
8 - Set name for AUTOEXEC file
9 - Save configuration
0 - Quit configuration program

The option required should be selected by pressing the appropriate number key and the screen will clear and be replaced by a further set of options.

Editing the options

Once the selection has been made from the main menu with the number key, all the options within the new menu are selected by one of two methods, depending on the selection made. The 'Set keys' options are slightly more complex and are described separately.

Details of the current values of each option are shown and in the event that there is no existing PROTEXT.CFG file, then the values shown will be the default ones provided by PROTEXT. The up and down cursor keys are used to select which of the options are to be changed

The STOP key (ESC on the CPC6128) can be used at any time, to go back to a previous menu. If a beep is heard, then this means that an illegal value has been entered, or else the key that has been pressed is not relevant. In much the same way that an illegal value may not be entered, it is not possible to move from an entry until a legal value has been provided.

Some options have an easily defined set of possible values (such as yes/no, or internal/serial/parallel printer), in which case using the right cursor key will move forwards through the options and using the left cursor key will move back through them. Once the required value has been selected, the up and down cursor keys may be used to move on to another option.

There are also a number of options which have no set range of parameters, such as those requiring the name of a file. In this case the option to be altered is selected, as before, with the up and down cursor keys and then the value required is typed in.

The Set keys options

The two options to 'Set keys' are slightly different and require a key number entering, to specify which key is to be edited. Each key on the keyboard has an individual 'key number' and these may be found from the diagram in the computer's User Manual. Once the key number has been entered and RETURN pressed, a table will appear.

This table is split in three parts vertically, with the left columns showing the key number, the middle showing the key values as ASCII characters and the right side as hexadecimal codes.

The PCW 8256/8512 key table has five values for each key, representing that key when pressed on its own, with EXTRA, with SHIFT, with ALT, and with ALT and SHIFT simultaneously.

The CPC6128 table has three values - the key on its own, with SHIFT, and with CONTROL.

The cursor will be positioned on the ASCII character of the first column for the selected key. The left and right cursor keys may be used to move backwards and forwards across the ASCII characters. Pressing the required key on the keyboard will change the setting in the table to the new value. In addition, it is possible to move up and down through the key numbers by using the up and down cursor keys.

Normally this is the easiest way to change which characters are allocated to a key, but there are a considerable number of characters and codes which cannot be keyed in directly from the keyboard, as they do not have their own key. Pressing the TAB key will switch the cursor into the 'Hex code' side of the table and the hexadecimal number representing the character may then be typed in. The computer's User Manuals give full details of the hex values of characters and an Appendix at the back of this manual lists the codes for phrases and function keys, as well as the codes for the command keys.

When editing is complete, the STOP key may be pressed to back out to the previous menu.

The remainder of this chapter gives details of the various options available from the main menu and their functions.

a). Set editing options

This option enables the user to configure a number of features for use within edit mode, so that when PROTEXT is first loaded the specified features will be in operation. The configurable options are listed below, with brief notes describing those options for which the use may not be self evident. The options should be selected and altered by the methods described above.

Default drive for text
Default group for text
Insert/overwrite mode
Right justify on
Word-wrap on
Box mode on
Help lines on
Background printing on
Prompt for background printing (1)
Decimal tabs commas/full stops (2)

Tabs and returns displayed
Spaces displayed
Control codes displayed
Document/Program mode
Doc/Prog mode for imported files (3)
Undelete buffer size (4)
Cursor flash rate (5)
Temporary text drive (6)
Sideways scroll increment

1. Prompt for background printing - If selected, PROTEXT will request whether printing is to be carried out in background mode when one of the printing options is selected.
2. Decimal tabs may be configured to align with either full stops or commas.
3. If programs, data files etc., are to be regularly imported from other software, it is preferable to select Prog mode as the default. By default, when a file which is not specifically a Protext document is loaded, Document mode is selected.
4. Undelete buffer size - This specifies the amount of memory which must be set aside to store deleted text. The larger the buffer, the less memory will be available for the text.
5. Cursor flash rate - The larger the number, the slower the flash rate.
6. Temporary text drive - This option defines the disc drive on which PROTEXT will store those parts of a text file which are not in memory.

b). Set spelling checker options

This is a straightforward option and the settings will be those that SPELL will contain on entry directly from CP/M. When used from within PROTEXT, the settings are automatically selected to suit the system in use.

Default drive for text
Default group for text

c). Set printing options

These options are concerned with how PROTEXT should lay out the documents it prints. The default settings provided are suitable for use with A4 single sheet paper, or 11 inch continuous stationery. These settings may be overridden at any time by the use of stored commands in the document. The prime use for this option is to change the default settings if different stationery is used for most of the time.

Page length
Line spacing
Top margin
Header margin
Footer margin
Bottom margin
Side margin

Continuous printing on
New page after print
Page number at top
Page number at bottom
Form feeds enabled

Note: The settings for top, header, footer and bottom margins are not allowed to total more than the setting for page length.

d). Set general options

These options are concerned with setting the key repeat rates and the screen colours (normal or inverse on the PCW). The expansion string buffer size governs the size of the buffer which holds phrases as well as any function key definitions. Note that the bigger the buffer, the less memory will be available for text. The use of the RELAY key to re-format the current paragraph may be disabled if users find its location too close to the cursor keys.

Expansion token buffer size
Key repeat rate
Key startup delay
Disable relay-key off/on

PCW 8256/8512 colour
CPC 6128 border colour
CPC 6128 foreground colour
CPC 6128 background colour

e). Set keys for PCW8256/8512

This option will rarely be needed, particularly by English speaking users, and is primarily included for the benefit of anyone who might prefer to re-configure certain keys on the keyboard to perform different functions. The most likely need for this would be to arrange for special symbols or foreign characters to be located on more suitable keys.

f). Set keys for CPC6128

This option serves the same purpose as e), and is for use by CPC6128 users.

f). Set printer driver options

This option allows two default printer drivers to be specified for PROTEXT to load (one for PCW8256/8512, one for CPC6128) automatically. The maximum amount of memory to allocate for control codes and for character translations may also be specified (See SETPRINT). Note that if more space is reserved than necessary, it will reduce the amount of memory available for text.

g). Set name for AUTOEXEC file

When PROTEXT is first loaded, it will look for an EXEC file to execute. By default, this is a file called EXFILE. This option permits another file with a different name to be specified instead of EXFILE.

h). Save configuration

Once all the required options have been selected and values specified, the configuration details can be saved to a file by using this option. The configuration file will be saved with the name PROTEXT.CFG.

Note: Only one configuration file may be present on a disc, as it must have the name PROTEXT.CFG if it is to be automatically configured on entry to PROTEXT.

Note: Once a configuration file (PROTEXT.CFG) has been created, it should be copied onto the other side of the Start of Day disc, so that it is available when PROTEXT is loaded.

i). Quit configuration program

Using this option will return control to PROTEXT command mode (or CP/M if CONFIG was loaded from CP/M). If any changes have been made to the configuration, but the Save configuration option has not been used, a request will be made for confirmation that the changes that have been made are to be discarded.

3. SETPRINT - CREATE A PRINTER DRIVER

A 'printer driver' contains information for PROTEXT about the printer being used and the way it is to be used. This information is contained in a file with the extension '.PTR'. PROTEXT comes with three printer drivers already on the disc.

PCW.PTR	- specifies the control codes for the PCW internal printer.
EPSON.PTR	- specifies control codes for a standard Epson compatible printer.
JUKI6100.PTR	- specifies the codes for a Juki 6100 daisy wheel printer and many other daisy wheel printers.

PROTEXT also includes the details of a 'simple printer', i.e. a printer which recognises no control codes except the backspace code and can use this to provide underlining and bold facilities.

Unless CONFIG has been used to specify a different default printer driver, PROTEXT, when used on a PCW8256/8512, will use PCW.PTR to print to the internal printer, or if it fails to find it, will treat the internal printer as a 'simple printer'. Similarly, if no default printer driver is specified for the CPC6128, PROTEXT will assume a 'simple' parallel printer.

If an Epson compatible printer is to be used, then CONFIG should be used to specify 'EPSON.PTR' as the default printer driver. Similarly, if a Juki 6100 printer is being used, JUKI6100.PTR should be specified.

If the printer being used is not one covered by the supplied printer drivers, it is recommended that SETPRINT is used as soon as an understanding of the requirements has been gained and after studying the printer control code section of the printer manual, in particular.

SETPRINT is specially written so that it may be used from within PROTEXT without loss of text which may be in memory at the time. Once complete, it will automatically return to PROTEXT command mode (see note below).

Note: CPC6128 users with only one disc drive must save any files to disc before using the SETPRINT program and on completion it will be necessary to load the file back into memory in the normal way. If the file is not saved before entering SETPRINT, it is probable that part or all of the document will be lost.

The file called SETPRINT.COM must be available on one of the drives at the time SETPRINT is called, or the error message 'SETPRINT.COM not found' will be given. Typing SETPRINT from PROTEXT command mode will result in a slight pause whilst PROTEXT searches the drives for the file and does some checking.

The screen will clear and an opening menu will appear, listing the following options:

1 - Set printer options
2 - Set serial options
3 - Set printer control codes
4 - Set character translations
5 - Set character widths
6 - Load printer driver
7 - Save printer driver
0 - Quit SETPRINT

The option required should be selected by pressing the appropriate number key and the screen will clear and be replaced by a further set of options. The up and down cursor keys are then used to select which of these options are to be changed.

If SETPRINT is entered from PROTEXT, then the currently loaded printer driver will automatically be loaded into SETPRINT and the current values of each option will be displayed. If there is no printer driver loaded, or SETPRINT is entered from CP/M, it will be necessary to use option 6 to load the required printer driver.

Editing the options

The STOP key (ESC on the CPC6128) can be used at any time, to go back to a previous menu. If a beep is heard, then this means that an illegal value has been entered, or else the key that has been pressed is not relevant. In much the same way that an illegal value may not be entered, it is not possible to move from an entry until a legal value has been provided.

Some options have an easily defined set of possible values (such as yes/no, or internal/serial/parallel printer), in which case using the right cursor key will move forwards through the options and using the left cursor key will move back through them. Once the required value has been selected, the up and down cursor keys may be used to move on to another parameter.

There are also a number of options which have no set range of parameters, such as those requiring the name of a file. In this case the parameter to be altered is selected, as before, with the up and down cursor keys and then the value required is typed in.

Some options require a series of codes to be entered and these can be entered either as text or as numbers. The top of the screen will display either the message, 'text' or 'numeric'. In text mode typing any normal character will insert that character as a code to be sent; some characters cannot be entered in this way, such as ESC (decimal 27). The TAB key is used to toggle between text and numeric input, so in order to input codes of this sort, numeric mode should be toggled on.

Once in numeric mode there are also two options, decimal and hexadecimal (the top of the screen will display 'decimal' or 'hex'). ALT-TAB will switch between decimal mode and hexadecimal mode. When entering numbers, they should be separated either by commas or spaces.

In all modes, the cursor keys will move to different parts of the code in the normal way. ALT with left and right cursor keys will move to the beginning and end of the line of codes; ALT-E will delete from the cursor position to the end of the line.

Editing 'Control codes', 'Character translations' and 'Character widths'

In the case of the control codes and the character translations, a table will appear showing which options already have values assigned to them. The cursor keys should be used to move through the table and the codes assigned to the particular option the cursor is on will be displayed in the lower part of the screen.

The character translation and character width tables can only display half of the characters at a time and the other half may be selected or reselected, at any time, by pressing the TAB key.

Pressing RETURN when the cursor is positioned on the required character will select that option and the cursor will move down to the displayed codes to enable them to be edited. Editing and entry of the codes is carried out as described above. Pressing STOP will complete entry of the codes and return to the main table.

When editing is complete, pressing STOP a further time will return to the main menu.

a). Set printer options

These are general options and are used by PROTEXT to determine in what format and where the document is to be sent when printing takes place. It is important when creating a printer driver, or modifying an existing one, that as many of these questions as possible should be answered. It may not be possible to answer all of them, as some of the features may not be available on the printer being used.

The options available are:-

Printer type
Control code interpretation
Proportional printing on
Microspacing on
Character width
Microspacing code
Abort code
Continuous paper code
Single sheet code
Paper sensing on code
Paper sensing off code
Half line feed code

By default, if no printer driver is loaded, the only settings made by PROTEXT will be 'unknown' printer type and 'simple' control code interpretation. No control codes and translations are defined.

(i) Printer type.

The setting of this option determines which of the printer ports will be used when the printer driver is loaded and the setting selected should be the relevant one for the printer to which the printer driver refers. The options available are 'parallel', 'serial', 'internal' and 'unknown'. The 'internal' option refers to the standard PCW printer and is not relevant for the CPC6128. If 'unknown' is selected, then no change will be made to the current setting when the printer driver is loaded.

Note: Only the 'internal' setting will be relevant on the PCW8256/8512 unless a serial/parallel interface has been fitted to the computer. On the CPC6128, the only relevant setting will be 'parallel' unless a serial interface has been added to the computer.

(ii) Control code interpretation.

There are two possible settings, 'simple' and 'send'. The default setting is 'simple', in which case only the simple printing facilities described at the start of the chapter will be available. If 'send' is selected then the control codes specified in the remainder of this chapter will be sent to the printer as required.

(iii) Proportional printing on.

By default the setting will be 'N', but if 'Y' is selected then proportional printing will be in operation when the printer driver is loaded.

(iv) Microspacing on.

The default setting is 'N', but may be changed to 'Y' if microspacing is required as standard. Full details of this option are in the section on microspacing in chapter Word 2.

(v) Character width.

This specifies the number of microspacing codes required to make up one character and is related to the microspacing code option below. Full details of how to work out the required codes are given in the section on microspacing in chapter Word 2.

(vi) Microspacing code.

This is the sequence of codes required to print a microspace. Full details of this option are in the section on microspacing in chapter Word 2.

(vii) Abort code.

If the printer supports this code, then when printing is aborted, this code will be sent to the printer to abort printing instantly.

(viii) Continuous paper code.
(ix) Single sheet code.

Continuous paper and single sheet codes are supported by some printers and one has the effect of turning the other off. In the absence of these codes, the appropriate DIP switch on the printer should be set to continuous and PROTEXT will determine whether printing should stop or not.

(x) Paper sensing on code.
(xi) Paper sensing off code.

Not all printers support these codes, but if they are available then background printing can be automated to a greater degree when single sheet stationery is being used.

(xii) Half line feed code.

This is the code required to advance the paper by a half line feed and is supported by most printers. If half line feeds are to be used in documents, then these codes must be specified.

b). Set serial printer options

This option only requires setting if the printer being used has a serial interface and is used to set the values of the following parameters:-

Baud rate (9600)
Number of data bits (8)
Parity (NONE)
Number of stop bits (1)
Mode Selector (0)

The above options are all self explanatory, except for the last one, Mode selector'. This option is used to set whether handshaking is 'on' or 'off'. The default setting is 0 (zero), which is 'off' and the setting for 'on' is 255. If the serial printer being used requires handshaking, then the value should be changed to 255.

The default values, shown in brackets above, will suit most serial printers, but, if necessary, they may be changed to suit those available on the printer. It is important that the settings for PROTEXT and those for the printer are the same. These options are all of the 'limited set' type and the left and right cursor keys should be used to rotate through the options.

c). Set printer control codes

The codes which are entered in this option are the ones required by the printer to carry out the different styles of printing and other specified functions. The printer manual should be consulted to ascertain the codes required.

The control codes which may be defined are @ and letters a to z. If both an 'on' code and an 'off' code are set then PROTEXT will send the 'on' code the first time the printer control code is encountered and subsequently alternate between 'on' and 'off'. It should be noted that most codes have an 'on' and 'off' value, but that Pica/normal (10 characters per inch) and Elite (12 characters per inch) usually only have an 'on' value as, when one is selected, it will automatically turn the other off. One or the other will always be in use, whereas the other codes, such as enlarged or condensed are used in addition to them, to provide further typestyles.

Note: The supplied printer drivers provide the codes for most of the normally used printing options, but any of the unused control code letters may be defined to add further options if they are supported by the printer.

d). Set character translations

This can be used to redefine one character either as another character or as a sequence of codes. This is useful for printing characters that do not have a standard ASCII code, or those that a printer does not normally support. The printer manual should be studied for details of how to create new symbols.

Users of daisy wheel printers may well find this option to be particularly useful, as a number of them have the less frequently used characters allocated values different from the more common standards.

Note: The PCW.PTR printer driver already has a number of characters translated, or redefined and all characters between ASCII 32 and 127 and also those characters listed in the PCW user manual, between 160 to 254, may be printed without any alterations being required. If the PCW.PTR driver is modified slightly to adjust the few codes which differ from the 'Epson' standard, it will be possible to use this printer driver with an Epson compatible printer, to print all of these characters as well.

Note: The code 255 decimal, the 'equivalent' sign should not be used or allocated to any character.

Any characters which have been redefined will be displayed in the table in inverse video.

A common problem is printing the '£' sign. If the printer is not printing this (it often comes out as a '#'), the character number for '£' should be ascertained from the printer manual and the '£' character redefined to be printed as this character number.

Note: If the CPC6128 is being used and the character number is greater than 127 (hexadecimal &7F) it will not be possible to print the character by this method, as the CPC6128 only has a '7 bit' printer port, which means that only codes lower than 128 may be sent to the printer. There are two possible methods of overcoming this limitation. Some printers have a special code, sometimes called 'Set MSB' or 'Set MSB to 1' and if so, these codes should be sent, then the code for the character and finally the codes to 'reset top bit' (or 'Reset MSB' or 'Set MSB to 0'). Alternatively, if the character is one that is contained in the printer's foreign character sets, it is possible to send the codes required to change the 'nationality', followed by the character and finally the codes to return the 'nationality' to its original setting.

e). Set character widths

If full proportional printing is required, this section is used to define the width of each character. When full proportional printing is in use, PROTEXT refers to the widths in this table and calculates how many characters it can fit on a line. If there is sufficient space on a line, it will bring further words up from the following line to fill out the line.

Note: It must be appreciated that PROTEXT has to do a lot of work calculating the exact size of each line and this option is not recommended for use with dot matrix printers, as the text output is fairly slow. The option is intended primarily for use with daisy wheel printers.

The JUKI6100.PTR and PCW.PTR printer drivers come with these codes already set to the appropriate values, but EPSON.PTR is supplied with these codes set to the default value of 10, though there is no reason why they may not be redefined, remembering that the rate of printing will be considerably reduced.

The table normally shows the widths in 1/120s of an inch, but the printer manual should be studied for details, both of the widths of each individual character and of the codes required to move the carriage by 1/120 of an inch. The section in this manual on 'Proportional printing and Microspacing' should be studied for further details.

f). Load printer driver

Existing printer drivers may be loaded for modifications by use of this option. The default filename extension for printer drivers is '.PTR'.

g). Save printer driver

This option permits the modified or newly created printer driver to be saved for future use. The name by which the printer driver is to be saved, should be entered. The default filename extension for a printer driver is '.PTR' and it is recommended that this is used.

Note: Printer drivers should be saved onto both sides of the 'Start of Day' disc, so that they are available when PROTEXT is loaded and also when SETPRINT is loaded.

h). Quit SETPRINT

If any changes have been made to the driver and the save option has not been used, SETPRINT will request confirmation that any changes are to be discarded before returning to PROTEXT command mode or CP/M.

4. CONVERT - FILE CONVERSION PROGRAM

CONVERT is a separate program supplied on the master disc and is primarily provided for the purpose of converting text and data files created by other programs into a form more suitable for use by PROTEXT. It may also be used to alter certain features of files created with PROTEXT.

Note: ASCII files created with the 'simple' save from Locoscript are unusual in that they do not contain any carriage returns at the end of each line. As a result, it is unnecessary to use CONVERT on these files and they can be loaded into PROTEXT in the normal way. Once loaded, each paragraph will appear as a single long line of text. Suitable rulers should be installed and the document formatted with the FORMAT command, before saving as a normal PROTEXT document. Only if the document contains any 'non-English' characters, should CONVERT be used to process the document with the 'Select convert' option and the 'convert non- English characters' option selected. This is necessary because PROTEXT stores non-English characters in a different form from Locoscript. Note that the file must be loaded into PROTEXT, formatted and saved, before it is processed with CONVERT.

In order to make use of CONVERT, the file called 'CONVERT.COM' must be present on one of the drives. It does not matter which drive, as PROTEXT will search for it on each drive in turn.

The program is specially written so that it may be called by typing 'CONVERT' when in PROTEXT command mode, in which case, on completion of conversion of the file or files, a return will be made to PROTEXT and any text which was in memory at the time will still be intact (See note below).

Note: CPC6128 users with only one disc drive must save any files to disc before using the CONVERT program and on completion it will be necessary to load the file back into memory in the normal way. If the file is not saved before entering CONVERT, it is probable that part or all of the document will be lost.

CONVERT.COM may also be used from CP/M command mode by typing 'CONVERT', in which case it will return to CP/M command mode once conversion of files is complete. It would normally be used from within PROTEXT however.

After a brief hesitation whilst PROTEXT carries out certain functions and checks for the presence of CONVERT.COM, an opening menu will be displayed, listing the following options:

ASCII TO Protext
Wordstar to Protext
Wordstar to ASCII
Select convert options
Quit

a). ASCII to PROTEXT

When this option is selected, the name of the file to be converted is requested, followed by a request for a name for the new file that will be created. The filenames may be prefixed with drive letters in the normal way (eg. B:filename) so that a file may be read from one drive and the modified file written to another drive. As soon as the second filename has been given, CONVERT will commence processing the document. This is fully automatic and a count of the number of lines that have been converted will be displayed on the screen, to give an indication of progress. Once the complete file has been converted a prompt will appear, requesting whether any more files are to be converted or not.

If 'Y' is selected, a return will be made to the opening menu for further selection, otherwise a return will be made to PROTEXT command mode.

The effect of the conversion is to create a file containing soft returns and spaces (which are used by PROTEXT in the formatting process), which once loaded into PROTEXT and, with the addition of suitable rulers, can then be treated as a normal PROTEXT Document file.

The operations carried out in the course of conversion are:

(i) Any hard returns which are followed by a character that is not a space or another hard return will be changed into a PROTEXT soft return.
(ii) Where a hard return has been changed to a soft return by the above process, any spaces preceding the soft return on that line, that are followed by another space will be changed into soft spaces.

Note: Spaces at the start of a line are not converted to soft spaces, otherwise justification of tabulated sections of text would be lost, as would any inset left marginis at the start of a paragraph. If required, these spaces can easily be replaced with tabs using PROTEXT's 'REPLACE' command. For example, for a section of text containing a number of lines with an inset of five spaces, REPLACE would be used and when prompted for 'String to find', the space bar would be pressed five times and when prompted for 'String to replace', the TAB key would be pressed once. Answering 'Y' to the prompt to 'Replace y/n' at each occurrence would then replace the spaces with tabs. An appropriately placed tab or left margin marker should be inserted into the ruler line to restore the format.

Note: Some ASCII files may contain characters, freq'ıently printer control codes, which are outside the strictly normal range of the ASCII character set. These may be removed or changed during the course of conversion by making use of the 'Select convert options', option from the main menu rather than this option. Alternatively they may be edited out, either by the use of REPLACE, or by normal editing methods, from within PROTEXT.

b). Wordstar to PROTEXT

This option performs a similar function to the 'ASCII to Protext' option already described, but on Wordstar type files. These are files created by a number of programs and if loaded into PROTEXT without conversion, the effect will be a document in which the last character of each word will be indecipherable. Using this option will rectify the problem and in the process make the file suitable for normal editing by conversion of the appropriate returns and spaces, as described above.

Note: There are also some programs which produce files which are almost completely indecipherable when loaded into PROTEXT. Frequently these can also be converted satisfactorily by the use of this option.

Note: If the document contains any control codes, they can also be converted at the same time by making use of the 'Select convert options' option from the main menu, rather than this option.

c). Wordstar to ASCII

This option will convert a Wordstar type of file into a straightforward ASCII file, complete with hard carriage returns at the end of each line. As with both the above options, the name of the file to be converted is requested and also a name for the converted file.

The main use for this option is for the conversion of files to be used as mail merging data files into a suitable format.

d). Select convert options

This is the most flexible and powerful of the options available, but does require some thought and care to ensure that all the correct options are selected.

Facilities are provided to modify files that are going to be used as data files, as well as for the conversion of specified characters into other characters.

On selecting this option a number of further options are offered for selection, one after the other, each of which require the answer 'y/n' and depending on whether 'Y' is used to select the option, a further subset of options may be offered. Once selection is complete, a request is made for the name of the file to convert and a name to save the converted file with. Conversion will then take place automatically, with a running total of the number of lines which have been converted displayed on the screen. Once complete, either more files may be converted by selecting 'Y' in answer to the request, or a return to PROTEXT may be made by choosing 'N'.

The full list of options offered are:

Convert to PROTEXT document format
Strip bit 7
Split file
Translate any characters
Remove any characters
Convert non-English characters
Wrap each line in quotes
Insert blank line after every line

(i) Convert to PROTEXT document format

If 'Y' is selected the converted file will be saved as a PROTEXT document, complete with soft returns and spaces, as described above, otherwise the file will be saved as an ASCII format file, complete with hard returns and hard spaces.

(ii) Strip bit 7

If a character has bit 7 set, this means that 128 has been added to the ASCII value of the character, as with the last character of a word in a Wordstar file.

Selecting 'Y' to this option will remove bit 7 from any character in the document which has this bit set. Selecting 'N' will leave them intact.

Note: If selected, this will have the effect of removing bit 7 from all characters, including any printer control characters which may have bit 7 set. If the document is known to have printer control codes and these are required, the option to 'Translate any characters' should be used to convert the codes to the equivalent PROTEXT control codes at the same time. Alternatively, if they are not required at all, the option to 'Remove any characters' should be selected.

(iii) Split file

Whilst PROTEXT is capable of editing documents of 'unlimited' file size, many people prefer to work with documents of a more convenient size. Selecting this option enables large documents to be split into a number of smaller files. Once 'Y' is selected, the 'Maximum file size in bytes' will be requested and the size required, in bytes (characters) should be entered.

When CONVERT saves the new files, it will save them with the specified name and will add an incrementing number as the filename extension, eg. the new files will be called 'FILENAME.1', 'FILENAME.2' etc.

Note: CONVERT will always attempt to split the file at the end of a line, rather than at exactly the number of bytes specified.

Note: Files may still be printed as one file, by making use of the '>IN filename' command, described in the chapters on stored commands and mail merging, to link them together. Alternatively they may be merged into one large file again, when all editing has been completed, although it would normally be advantageous to leave them as a number of more manageable files.

(iv) Translate any characters

If this option is selected a request for the 'first code to translate' is made and the code of the first character to be changed should be entered. This is followed by a request for the 'last code to translate', which should also be entered. In the event that only one code is to be changed, the same number should be entered in answer to both questions. Often, however, it will be possible to specify a range of characters.

Finally the 'new code or +/- offset' will be requested. If only one code was specified, then the new code value should be entered, but if a range was selected, then the offset should be given, prefixed by '+' or '-', depending on whether the new codes are smaller or greater than the original ones.

For example: If characters with values between 193 and 218 (Upper case A-Z with bit 7 set) are to be altered to lower case letters a-z with bit 7 set (PROTEXT printer control codes have these values and will appear in PROTEXT as inverse lower case characters), then the offset should be given as '+32'.

Once the offset or new character value has been selected, the option to specify 'more translations' is offered until such time as all translations have been specified and 'N' selected in answer to this question.

Note: A convenient way to find out the value of any characters is to use TYPE from CP/M and compare the character with the table provided in the CP/M part of the Amstrad manual.

(v) Remove any characters

This option will remove any specified characters from the file and is a convenient way of removing any unwanted control codes which may no longer be required in the document. Either one, or a range of characters may be removed by entering the codes in a similar way to that described in the previous option and as before, more than one range of characters may be specified.

(vi) Convert non-English characters

This option can be used to ensure that any 'Non-English' characters are preserved. Some of the codes used by other programs for these characters are used by PROTEXT as control codes and PROTEXT stores non-English characters in a special format.

(vii) Wrap each line in quotes

The main use for this option is in the conversion of unsuitable data files into a form which PROTEXT can process. Sometimes data created by other programs may contain either commas ',' or quotation marks (either single or double) in the middle of a data field, both of which PROTEXT understands to indicate the start or end of a data field. The solution is to wrap each field in quotation marks. Many databases offer the option to either 'comma delimit' or 'wrap in quotes' when specifying export options and if so, then these options would be used. Some do not provide this facility, in which case this option would be specified to solve the problem.

When fields which contain commas and quotes are not 'wrapped' in quotation marks, the result is that the fields will become split up and confused when mail merging takes place.

This option wraps each line of data in a set of double quotes and in the process it also converts any existing double quotes within the field into single quotes, otherwise they would be understood to mean the end of the field. Any quotes used in a data field must be of an alternate sort to those which are used to wrap the field.

(viii) Insert blank line after every line

This option is used for the purpose of modifying data files created by other programs which are unsuitable for use with PROTEXT.

Normally PROTEXT requires a data file to have each record separated by a blank line and many databases permit the insertion of dummy fields into data to be exported, in which case a dummy blank field would be specified for the last field in each record. Some, however do not provide this facility and with these programs this option would be used to create the necessary blank line.

5. DCOPY - DISC COPYING PROGRAM

DCOPY is a utility program provided for the purpose of copying the contents of any CF2 single density disc onto another disc for the purpose of making a back up copy. It is good practice to keep back ups of any important files as it is always possible for accidents to happen, or discs to fail. It is also good practice to make backups regularly and systematically. DCOPY provides a complete copy of the original disc, including any .BAK files.

Warning: DCOPY will copy the entire contents of one disc onto another. If a blank formatted disc is accidentally copied onto a disc which contains files, the result will be two blank discs, so care should be taken to ensure that the 'write protect' tab on the original disc is set in the open position to ensure that this cannot occur.

Note: The COPY option of DISCKIT should be used on the PCW to make back ups of any CF2DD format discs of the type used by the second drive (Drive B) on the PCW8512 and expanded PCW8256. The DCOPY program is only intended to be used to copy single sided CF2 discs.

DCOPY MUST be used to make backup copies of the dictionary disc as neither the DISCKIT program provided with CP/M, nor the COPY command in PROTEXT are capable of copying the special format used on the Dictionary disc.

DCOPY is specially written so that it may be used from within PROTEXT without loss of text which may be in memory at the time. Once complete, it will automatically return to PROTEXT command mode (see note below).

Note: CPC6128 users with only one disc drive must save any files to disc before using the DCOPY program and on completion it will be necessary to load the file back into memory in the normal way. If the file is not saved before entering DCOPY, it is probable that part or all of the document will be lost.

The file called DCOPY.COM must be available on one of the drives at the time DCOPY is called, or the error message 'DCOPY.COM not found' will be given. Typing DCOPY from PROTEXT command mode will result in a slight pause whilst PROTEXT searches the drives for the file and does some checking.

Instructions are then given about which drive should contain the original disc and also which drive to put the disc to be copied into. Users of machines with only one drive will be given instructions about which disc to insert at the appropriate times.

Before any copying is undertaken, the write protect tab should be opened on the disc to be copied, to make sure that it cannot be accidentally overwritten. When ready, press 'S' to start, or press 'C' to cancel. Copying will then take place automatically. On completion, the opportunity to copy another disc is offered. Selecting 'N' at this point will return to PROTEXT command mode.

Note: This program takes an exact copy of the entire contents of a disc. Any files on the disc onto which the original is to be copied will be erased, so it is important to ensure that there is nothing of importance on the backup disc before commencing copying. If it is required to copy only a number of files onto another disc which already contains files, the COPY or ERACOPY commands should be used in PROTEXT, to copy the relevant files across.

SPECIAL OPTIONS

1. SPECIAL CHARACTERS

PROTEXT is capable of being used with most non-English languages and fully supports the use of accents and characters such as c-cedilla.

Characters containing accents may be typed in during the course of editing and will appear correctly on screen. They may also be stored in the SPELL dictionary and will be checked and may be corrected in the normal way.

There are seven main accents which are required to cover the usual range of European languages and these may be obtained in the following way.

The base character should be entered first and then immediately followed by EXTRA and the number key which contains the required accent (the accents and their keys are listed below). The accent will then be positioned over the character. Accents may be used with any character, which permits the use of PROTEXT with a number of languages which normally are not catered for. Welsh and many of the Eastern European languages are covered as a result.

Note: CPC6128 users should note that CTRL-1 to CTRL-7 are used to obtain accents, instead of the EXTRA key and a number key. The special characters are obtained by pressing CTRL-0, followed by the appropriate letter key, or by pressing one of the function keys with either SHIFT or CONTROL (see below).

If an accent is required by itself, press space followed by the accent key. Should any of the accent characters be required frequently it is possible to redefine the keys to give just the accent.

Accents supported

PCW key	CPC key		Accent
EXTRA-2	CTRL-3	¨	Umlaut
EXTRA-5	CTRL-5	°	Ring
EXTRA-6	CTRL-7	´	Acute accent
EXTRA-7	CTRL-6	^	Circumflex
EXTRA-8	CTRL-1	`	Grave accent
EXTRA-0	CTRL-4	ˇ	Inverted circumflex
EXTRA-hyphen	CTRL-2	~	Tilde

Note: The keys used on the PCW are the same ones used under CP/M, with the exception that EXTRA-0, the 'Inverted circumflex', which is used by a number of Eastern European languages, is an additional accent.

In addition to these accents, which may be used on any character, a number of phrases are initially defined as special 'non-English' characters and in the case of the PCW, these are the same keys as are used in CP/M. A list of the keys to be pressed is given below, a full list of phrases is in the Appendix which lists the 'Expansion string tokens'.

Note: The phrases may be redefined and care should be taken when selecting phrases, if any of these characters are required.

It is possible to display on the screen any of the characters between 160 (A0 hex) and 254 (FE hex), as described in the PCW Users Manual, but as they do not have their own keys on the keyboard normally, the characters must be allocated to KEYs or PHRASEs. The appropriate code should be allocated to the chosen key using the 'escape' characters described in the chapter on Phrases.

Note: Character 255 (FF hex), the 'equivalent' sign, must NOT be defined in this way, as it has a special use in PROTEXT.

The printer driver PCW.PTR has the characters redefined so that these characters may also be printed on the PCW printer. With a few modifications to this driver, it is also possible to use this driver with an 'Epson' compatible printer.

Note: It is not possible to use these characters on the CPC6128, as the printer port on this computer only has a '7 bit' port, which means that the highest code value that it is possible to send to the printer is 127.

It should be noted that it may be necessary to re-define some printer codes to enable these characters to be printed and with some printers it will not be possible. The chapter on the SETPRINT utility should be consulted for advice on how to proceed, if this should be necessary.

A complete summary of the characters that are available by default on the function keys and phrase keys is given in an appendix at the back of the manual. In addition, the following keys are defined to give arrows:-

PCW key	PCW key	CPC key	Character
ALT-V-←	SHIFT-ALT-←	CTRL-V ←	Left arrow (←)
ALT-V-→	SHIFT-ALT-→	CTRL-V →	Right arrow (→)
ALT-V-↑	SHIFT-ALT-↑	CTRL-V ↑	Up arrow (↑)
ALT-V-↓	SHIFT-ALT-↓	CTRL-V ↓	Down arrow (↓)

APPENDICES

A1. SUMMARY OF PROTEXT COMMANDS

This summary is divided into the following categories and gives concise details of the syntax of all commands used by PROTEXT.

(a) Edit mode commands
(b) Stored commands
(c) Command mode commands
(d) External utility program commands

(a) Edit mode commands

The commands marked '*' are not available in program mode.

Cursor Movement

PCW8512	PCW8512	CPC6128	
ALT-CHAR	←	←	Move left one character
CHAR	→	→	Move right one character
	↑	↑	Move up one character
	↓	↓	Move down one character
ALT-WORD	SHIFT-←	SHIFT-←	Move left one word
WORD	SHIFT-→	SHIFT-→	Move right one word
	SHIFT-↑	SHIFT-↑	Scroll back 1 line
	SHIFT-↓	SHIFT-↓	Scroll forward 1 line
LINE	ALT-←	CTRL-←	Move to start of line
EOL	ALT-→	CTRL-→	Move to end of line
	ALT-↑	CTRL-↑	Scroll back 18 or 25 lines
	ALT-↓	CTRL-↓	Scroll forward 18 or 25 lines
	ALT-Q	CTRL-Q	Scroll back one screen (no overlap)
	ALT-Z	CTRL-Z	Scroll forward one screen (no overlap)
	ALT-[	CTRL-[	Move to start of memory
	ALT-]	CTRL-]	Move to end of memory
ALT-DOC	ALT-[ALT-[	CTRL-[CTRL-[	Move to start of document
DOC	ALT-] ALT-]	CTRL-] CTRL-]	Move to end of document
ALT-PAGE	ALT-(	CTRL-(	* Move back one page
PAGE	ALT-)	CTRL-)	* Move forward one page
ALT-PARA	ALT-<	CTRL-<	Move back one paragraph
PARA	ALT->	CTRL->	Move forward one paragraph
		CTRL-£	Move to top left of screen
	ALT-L	CTRL-L	Go to last position
	ALT-G	CTRL-G	Go to line/page/column

ENTER	RETURN	RETURN	Insert mode - split line and move to left margin of next line Overwrite mode - move to left margin of next line
	SHIFT-RETURN	SHIFT-RETURN	Move to left margin of next line
	ALT-RETURN	CTRL-RETURN	Move to left margin of next line
	TAB	TAB	Insert mode - insert tab character Overwrite mode - move to next tab
	SHIFT-TAB	SHIFT-TAB	Insert mode - move to next tab Overwrite mode - insert tab character

Insertion and deletion

	ALT-I	CTRL-I	Insert line
ALT-CAN		CTRL-CLR	Delete line
	ALT-←DEL		Delete to start of line
ALT-E	ALT-DEL→	CTRL-E	Delete to end of line
	←DEL	DEL	Delete character before cursor
	DEL→	CLR	Delete character at cursor
	SHIFT-←DEL	SHIFT-DEL	Delete word left
	SHIFT-DEL→	SHIFT-CLR	Delete word right
	ALT-TAB	CTRL-TAB	Toggle insert/overwrite mode
	ALT-A	CTRL-A	Transpose (alternate) characters
	ALT-U	CTRL-U	Undo last delete operation
	ALT-*	CTRL-*	Split line at cursor
	ALT-+	CTRL-+	Join lines
SHIFT-ALT-CAN			Clear text

Block commands

SHIFT-⊞	SHIFT-COPY	SHIFT-COPY	Set or clear block markers
CAN	ALT-K	CTRL-K	Clear all block markers
PASTE	ALT-M	CTRL-M	Move block
COPY	ALT-COPY	CTRL-COPY	Copy block
CUT	ALT-CUT	CTRL-DEL	Delete block
	ALT-B	CTRL-B	Box mode on / off
	ALT-O	CTRL-O	Copy block from other document

Formatting and rulers

RELAY	ALT-F	CTRL-F	* Format to end of paragraph (RELAY may be disabled in CONFIG)
	ALT-C	CTRL-C	* Centre line
	ALT-D	CTRL-D	* Copy default ruler (first active ruler in document)
	ALT-R	CTRL-R	* Copy previous ruler but one
	ALT-J	CTRL-J	* Justify on/off
	ALT-W	CTRL-W	* Word wrap on/off

Find and replace, place markers

FIND			Find string
EXCH			Replace string
⊞		COPY	Next find
⊟	ALT-@ @	CTRL-@ @	Previous find
	ALT-@ n	CTRL-@ n	Set/Go to marker (0 to 9)
	ALT-@ [	CTRL-@ [	Go to [block marker
	ALT-@]	CTRL-@]	Go to] block marker
	ALT-@ L	CTRL-@ L	* Go to left margin
	ALT-@ R	CTRL-@ R	* Go to right margin
UNIT	ALT-⊞	CTRL-@ +	Go to next marker
ALT-UNIT	ALT-⊟	CTRL-@ -	Go to previous marker
	ALT-@ ?	CTRL-@ ?	Insert multiple marker

Other commands

	ALT-H	CTRL-H	Help on/scroll forwards
	ALT-N space	CTRL-N space	* Non-break space
	ALT-N hyphen	CTRL-N hyphen	* Non-break hyphen
	ALT-P	CTRL-P	* Page mode on/off
SHIFT-ALT-H	ALT-V B	CTRL-V B	Help scroll backwards
	ALT-V H	CTRL-V H	Edit help off
	ALT-V S	CTRL-V S	Hard spaces visible/hidden
	ALT-V T	CTRL-V T	Tabs and returns visible/hidden
	ALT-V V	CTRL-V V	Control codes visible/hidden
	ALT-V cursors	CTRL-V cursors	Insert arrow symbols
	ALT-S	CTRL-S	Reserved for future expansion
	ALT-T	CTRL-T	Reserved for future expansion

Note: ALT-S and ALT-T currently duplicate ALT-V S and ALT-V T but future versions of PROTEXT will use these for other functions.

	ALT-X	CTRL-X	* Insert printer control code
	ALT-Y	CTRL-Y	Switch between documents in memory
	ALT-/	CTRL-/	Convert to upper case
	ALT-.	CTRL-\	Convert to lower case
	ALT-hyphen	CTRL-hyphen	* Insert soft hyphen
	ALT-space	CTRL-space	Insert a space
	EXTRA-letter	CTRL-O letter	Insert stored phrase
	EXTRA-number	CTRL-number	* Insert accent
PTR			Access internal printer controls
EXTRA-PTR			Screen dump (internal printer)
ALT-ENTER		CAPS LOCK	Caps lock
SHIFT LOCK	CTRL-CAPS LOCK		Shift lock
EXIT	STOP	ESC	Exit from edit mode

Extra characters

SHIFT-ALT-↑	CTRL-V ↑	Up arrow
SHIFT-ALT-↓	CTRL-V ↓	Down arrow
SHIFT-ALT-←	CTRL-V ←	Left arrow
SHIFT-ALT-→	CTRL-V →	Right arrow

PCW8256/8512 extra characters

EXTRA-.	¦
EXTRA-@	\
EXTRA-:	↑

(b) Stored commands

Key:	a	an ASCII character
	f	a filename
	n	an integer between 0 and 255
	v	a variable identifier
	(x)	an optional parameter
	{x}	a parameter that may occur 0 or more times
	expr	a string expression
	text	a string of chars, optionally in quotes
	cond	a conditional string expression

Where more than one parameter is listed on a command line, the parameters may be separated by spaces, commas, or equals signs. This is not shown in the syntax descriptions below.

Stored commands are broken down into a number of categories according to their most common usage. Within each section they are arranged alphabetically. A separate appendix gives full technical details of all the stored commands in strictly alphabetical order. The subheadings are:-

Paper layout commands
Page formatting commands
Messaging commands
Miscellaneous general purpose
Printer control commands
Variable and data input - mail merging
Conditional printing and mail merging

The commands marked ’*’ take immediate effect, the remainder taking effect at the next new page printed.

Paper layout commands

	>BM n	Bottom margin.
*	>EM n	Even side margin.
	>FM n	Footer margin.
	>HM n	Header margin.
	>OM n	Odd side margin.
	>PL n	Page length.
*	>SM n	Side margin.
	>TM n	Top margin.
	>ZM	Zero margins.

Note: all parameters are in lines or columns as appropriate.

Page formatting commands

*	>CE text	Centre line.
	>CP ON/OFF	Continuous/Single sheet printing.
	>EA nn	End at page no.
	>EF text	Define even footer text and turn footers on.
	>EH text	Define even header text and turn headers on.
*	>EP (n)	Even page throw (can be conditional).
	>FF ON/OFF	Form feeds enabled/disabled.
	>FO text	Define footer text and turn footers in.
	>FO ON/OFF	Turn footers on/off.
*	>FP ON/OFF	Formatting whilst printing on/off.
	>HE text	Define header text and turn headers on.
	>HE ON/OFF	Turn headers on/off.
*	>LS n	Line spacing.
	>NC n	Number of copies.
	>NP ON/OFF	Enable/disable new page at end of printing.
	>OF text	Define odd footer text and turn footers on.
	>OH text	Define odd header text and turn headers on.
	>OP (n)	Odd page throw (can be conditional).
*	>PA (n)	Page throw (can be conditional).
	>PE ON/OFF	Print even pages only.
	>PN nn	Page number of next page.
	>PO ON/OFF	Print odd pages only.
*	>PS ON/OFF	Paper sensing on/off.
*	>RJ ON/OFF	Right justifying on/off.
	>SA nn	Start at page no.

Messaging commands

*	>CO text	Comment line.
*	>CS text	Clear screen and display message.
*	>DM text	Display message.
*	>ST text	Stop printing and display text.
*	>WM text	Write message to file - used with WF.
*	>WT text	Wait and display text.

Miscellaneous general purpose

*	>EX text	Execute command.
*	>IN f	Insert file.
*	>RC a {n}	Redefine character.
*	>WC	Write file close.
*	>WF f (A)	Open file for writing (appending).
*	>WF ON/OFF	Writing to file on/off.

Printer control commands

* >CC a {n} (;a{n})	Define printer control code.
* >CW n	Define microspace character width.
* >OC n {n}	Output codes to printer.
* >MC n {n}	Define microspace code sequence.
* >MS ON/OFF	Microspacing on/off.
* >PP ON/OFF	Proportional spacing on/off.
* >PR f	Load printer driver.

Variable and data input - mail merging

* >AV v {v}	Ask for variables.
* >CF	Close data file.
* >DF f {f}	Define data file.
* >RU v {v}	Read variables unconditionally.
* >RV v {v}	Read variables and pad with nulls.
* >SV v=expr	Set variables.

Conditional printing and mail merging

* >EI	End of ID, IE, IF or IU block.
* >EL	Else - print block if previous IF condition false.
* >ID v	Print block if variable defined.
* >IE	Print block if data file exhausted.
* >IF cond	Print block if condition true.
* >IU v	Print block if variable undefined.
* >RP	Repeat - until following UN condition is true.
* >SK cond	Skip printing if condition true.
* >UN cond	Until condition is true - repeat from RP command.

Note: If column one contains '>', and columns two and three each contain one of: space, 'L', '-', or '!' then the line is a ruler line.

(c) Command Mode

The following commands are listed according to a number of categories and alphabetically within the category for ease of location. The categories are:-

Document handling
Printer control
Printing options
Printer control commands - background printing
Sundry printing options
Disc Drive selection and cataloguing
Disc copying and formatting
Disc file manipulation
File protection
Text manipulation and formatting
Phrase, Exec and Symbol commands
Miscellany

Key to parameters:

a	an ASCII character
dr	drive letter
f	a filename (may include drive)
af	ambiguous filename (may contain wildcards)
inf	input filename
outf	output filename
newf	new filename
oldf	old filename
n	an integer between 0 and 255
nn	an integer between 0 and 65535
(x)	an optional parameter

Items marked with '< .. >' are mandatory.

Command	Abbr.	Description	

Document handling

CLEAR		-	Clear text.
DOC		-	Select document editing mode.
PROG		-	Select program editing mode.
LOAD	<f>	L	Load new file.
MERGE	<f>	MER	Merge file with current text.
NAME	<f>	N	Assign name to current file.
PRINTF	<outf> (inf)	PF	Print to disc as Ascii, opt. from disc.
PRINTFB	<outf>	PFB	Print block to disc as Ascii file.
SAVE	<f>	S	Save text.
SAVEB	<f>	SB	Save block.
SPOOL	<f>	SPON	Echo all screen output to a file.
SPOOLOFF		SPOFF	Turn off echo to file.
SWAP		SW	Swap between two documents in memory.
TYPE	<f>	T	Display file contents on screen.
CAN or CLR key			Clear screen.

Printer control

INTERNAL		INT	Select PCW Printer.
PARALLEL		PAR	Select parallel (Centronics) printer.
PRINTER	<f>	PR	Load printer driver.
SERIAL		SER	Select serial printer port.

Printing options

PRINT	(f)	P	Print file, optionally from disc.
PRINTB		PB	Print marked block of text only.
PRINTF	<outf> (inf)	PF	Print to disc as Ascii, opt. from disc.
PRINTFB	<outf>	PFB	Print block to disc as Ascii file.
PRINTP	(f)	PP	Print pages selectively, opt. from disc.
PRINTPQ	(f)	PPQ	Print pages selectively in NLQ, opt. disc.
PRINTPS	(f)	PPS	Print to screen selectively, opt. disc.
PRINTS	(f)	PS	Print text to screen, opt. from disc.
PRINTSB		PSB	Print block to screen.
PRINTQ	(f)	PQ	Print, NLQ mode, optionally from disc.
PRINTQB		PQB	Print block, NLQ.

Printer control commands - background printing

ABANDON		AB	Abandon printing and clear buffer.
CONT		CO	Continue printing contents of buffer.
STOP		ST	Stop printing but leave buffer intact.

Sundry printing options

PRINTON		PRON	Echo all screen output to printer.
PRINTOFF		PROFF	Turn off echo to printer.
TYPEWRITER (n)		TW	Typewriter mode.

Disc Drive selection and cataloguing

A:		A	Select drive A.
B:		B	Select drive B - valid if drive installed.
C:		C	Select drive C - valid if drive installed.
D:		D	Select drive D - valid if drive installed.
M:		M	Select drive M (PCW only).
DRIVE	<dr>	DR	Select drive. - Any drive A-P if installed.
GROUP	<n>	GR	Select group/user number.
USER	<n>	U	Select group/user number.
CAT	(dr)or(grp)or(af)	DIR	Catalogue files. Opt. drive, group, name.

Disc copying and formatting

DCOPY		-	Copy single sided disc. Utility program.
DFORM		-	Format disc. CF2 or CF2DD format.
DFORMD		-	Format disc CPC6128 Data format.

Disc file manipulation

COPY	<oldf> <newf>	-	Copy file, creating backup.
COPY	<af>(<grp>)(<dr>)		Copy file(s). Wildcards permitted.
ERACOPY	<oldf> <newf>	ECOPY	Copy file, erasing old version.
ERACOPY	<af>(<grp>)(<dr>)		Copy file(s), erasing old. Allows wildcards.
ERASE	<af>	ERA	Erases file(s). Wildcards permitted.
RENAME	<newf> <oldf>	REN	Rename file.

File protection

ACCESS	<af>	ACC	Set file(s) to read-write status.
PROTECT	<af>	PROT	set file(s) to read-only status.

Text manipulation and formatting

FIND		F	Find string.
FIX		-	Remove tabs, soft spaces and soft returns.
FIXB		-	As for FIX, but on marked block.
FORMAT		FT	Format text.
FORMATB		FTB	Format block.
REPLACE		R	Find and replace string.
NUMBER		NUM	Number lines of text.
NUMBERB		NUMB	Number lines of text in marked block only.

Phrase, Exec and Symbol commands

EXEC	<f>	X	Execute file of commands.
LPHRASES		LP	List defined phrases.
PHRASE		KEY	Define phrase or function key.
SYMBOL	<n> <n1>...<n8>	SYM	Re-define a screen character.

Miscellany

CALC		CA	Calculator.
COUNT		-	Count words.
COUNTB		-	Count words in block.
GOTO	<a> <nn>	G	Goto Page/Line/Column number.
HELP	(subhead)	H	Display HELP subheadings. (opt. commands)
PAUSE		-	Pause (optionally with message).
QUIT		Q	Quit PROTEXT.

COPY or (PCW only) ⊞ key — Recall last command typed in.

(d) External utility program commands

The following commands call programs which must be available to PROTEXT. In other words the specified '.COM' files must be on one or other of the discs. These programs are specially configured so that when completed, control returns to PROTEXT without loss of the original text (See note below). Any program may be called using the * prefix, and any text in memory will be saved to a temporary file the name of which will be passed as a parameter, but unless specifically written to work with PROTEXT, control will not automatically return to PROTEXT.

Note: CPC 6128 users with only one disc drive and PCW users who are not using drive M as the 'Temporary' drive should ensure that any documents are saved before using these options, as use of these external programs will usually require the changing of discs in order to run the utility program. On return to PROTEXT, the document can be reloaded in the normal way.

CONFIG	-	Protext configuration utility program.
CONVERT	CV	File conversion to PROTEXT format.
DCOPY	-	Copy single sided disc. Utility program.
SETPRINT	SP	Printer driver creation utility.
SM	-	Main spelling checker menu.
SPELL (f)	-	Spell check document, opt. from disc.
* <progname>	-	Call a separate program, automatically passing existing text file as a parameter.

A2. TECHNICAL SUMMARY OF STORED COMMANDS

This chapter contains full syntax descriptions of all the stored commands and default values, where applicable. Note that many of the default values may be changed by the use of the CONFIG and SETPRINT utility programs.

STORED COMMANDS

AV Ask for Variable from keyboard.
Syntax: >AV (<string>) <identifier> (<int>)
  { (<string>) <identifier> (<int>) }
Examples: >AV "Name",name,20,"Address line 1",addr1
>AV name,addr1,addr2,addr3,addr4

BM Bottom Margin.
Syntax: >BM <int>
Default: 3 lines

CC redefine Control Code.
Syntax: >CC <letter> {<code>} (; {<code>})
Example: >CC "h",27,82,6 ; 27,82,3

CE CEntre line.
Syntax: >CE <text>

CF Close File.
Syntax: >CF

CO COmment line.
Syntax: >CO <message>
Examples: >CO This line is just a reminder
> > > This is an alternative form of CO

CP select Continuous Printing or single sheets.
Syntax: >CP ON
>CP OFF
Default: Continuous printing on.

CS Clear Screen.
Syntax: >CS <message>

CW	set Character Width for microspacing.	
Syntax:	>CW <int>	
Examples:	>CW 10	(elite characters)
	>CW 7	(condensed)
	>CW 14	(condensed enlarged)
DF	Define data File or files.	
Syntax:	>DF <filename> {<filename>}	
Examples:	>DF "data"	
	>DF names1,names2,names3	
Notes:	It is not possible to have more than one DF command in use at a time. If more than one DF command is required, a CF command must close the first one before the next is opened.	
DM	Display Message.	
Syntax:	>DM <message>	
EA	End At page number.	
Syntax:	>EA <integer>	
Default:	Last page.	
EF	Even page Footer text and turn footer on.	
Syntax:	>EF <message>	
EH	Even page Header text and turn header on.	
Syntax:	>EH <message>	
EI	EndIf. Matches with ID, IF, or IU.	
Syntax:	>EI	
EL	ELse. Matches with ID, IF, or IU.	
Syntax:	>EL	
Note:	IF blocks may be nested within EL blocks.	
EM	Even side Margin.	
Syntax:	>EM <int>	
Default:	5 characters.	
EP	Even Page throw, optionally conditional.	
Syntax:	>EP (<int>)	
Note:	As PA,except that if new page has an odd number, a complete blank page will be printed. Text will start at the top of an even numbered page.	

EX Syntax:	**EX**ecute command. >EX <string>
FF Syntax: Default:	**F**orm **F**eed codes - enable or disable. >FF ON >FF OFF Form feed disabled.
FM Syntax: Default:	**F**ooter **M**argin. >FM <int> 2 lines (if footers on, otherwise zero).
FO Syntax:	**FO**oter text and turn footers on. >FO <message>
FO Syntax:	turn **FO**oters on or off. >FO ON >FO OFF
FP Syntax: Default:	**F**ormatting whilst **P**rinting - enable or disable. >FP ON >FP OFF Off.
HE Syntax:	**HE**ader text and turn headers on. >HE <message>
HE Syntax:	Turn **HE**aders on or off. >HE ON >HE OFF
HM Syntax: Default:	**H**eader **M**argin. >HM <int> 2 lines (if headers on, otherwise zero).
ID Syntax:	**I**f **D**efined. Matches with EI. >ID <identifier>
IE Syntax: Note:	**I**f **E**xhausted. Matches with EI. >IE Tests data file for no more data and if so, block is printed.
IF Syntax: Note:	**IF** condition true - print. Matches with EI. >IF <condition> IF blocks may be nested to a depth of 7.

IN	INsert file.
Syntax:	>IN <filename>
IU	If Undefined - variable. Matches with EI.
Syntax:	>IU <identifier>
LS	Line Spacing.
Syntax:	>LS <int>
Note:	Half line feeds are also permitted - specify as ’.5’ or ’½’.
Default:	Single line spacing.
MC	Set Microspace character Code sequence.
Syntax:	>MS <code> {<code>}
Example:	>MS 27,"L",1,0,0
MS	MicroSpacing - enable or disable.
Syntax:	>MS ON >MS OFF
Default:	Off.
NC	Number of Copies.
Syntax:	>NC <int>
Note:	The number of copies is ignored if a data file is defined.
Default:	One.
NP	New Page after printing - enable or disable.
Syntax:	>NP ON >NP OFF
Default:	Enabled.
OC	Output Codes to printer.
Syntax:	>OC <code> {<code>}
Examples:	>OC 27,"3",12 >OC 27,J,20
OF	Odd page Footer text and turn footer on.
Syntax:	>OF <message>
OH	Odd page Header text and turn header on.
Syntax:	>OH <message>
OM	Odd side Margin.
Syntax:	>OM <int>
Default:	5 characters.

OP	**Odd Page throw, optionally conditional.**
Syntax:	>OP (<int)>
Note:	As PA, except that if new page has an even number, a complete blank page will be printed. Text will start at the top of an odd numbered page.
PA	**PA**ge throw, optionally conditional.
Syntax:	>PA (<int>)
Note:	If parameter specified, then a new page will be started if the number of blank lines is smaller than or equal to the number specified.
PE	**P**rint **E**ven pages only.
Syntax:	>PE ON
	>PE OFF
PL	**P**age **L**ength.
Syntax:	>PL <int>
Default:	66 lines (suits 11 inch continuous or A4 single sheet).
PN	**P**age **N**umber.
Syntax:	>PN <integer>
Default:	Page 1.
PO	**P**rint **O**dd pages only.
Syntax	>PO ON
	>PO OFF
PP	**P**roportional **P**rinting - enable or disable.
	>PP ON
	>PP OFF
Default:	Off.
PR	load **PR**inter driver.
Syntax:	>PR <filename>
PS	**P**aper **S**ensing - enable or disable.
Syntax:	>PS ON
	>PS OFF
Default:	Paper sensing off.
RC	**R**edefine **C**haracter.
Syntax:	>RC <character> = <code> {<code>}
Examples:	RC "!" = "!" #08 "."
	RC £ 27 ">" 1 27 "="

RJ Right Justification - enable or disable.
Syntax: >RJ ON
>RJ OFF
Default: Corresponding to the edit mode setting at time of printing, unless specified.

RP RePeat. Matches with UN.
Syntax: >RP
Note: Repeat loops may not be nested.

RU Read variable Unconditionally.
Syntax: >RU <identifier> {<identifier>}

RV Read Variable from data file.
Syntax: >RV <identifier> {<identifier>}
Example: >RV name,addr1,addr2,addr3,addr4,addr5,dummy

SA Start At page number.
Syntax: >SA <integer>
Default: Page 1.

SK SKip copy if condition is true.
Syntax: >SK <condition>

SM Side Margin.
Syntax: >SM <int>
Default: 5 characters.

ST STop printing.
Syntax: >ST <message>

SV Set Variable.
Syntax: >SV <identifier> = <expression>
{<identifier> = <expression>}

TM Top Margin.
Syntax: >TM <int>
Default: 3 lines.

UN UNtil. Matches with RP.
Syntax: >UN <condition>

WC Write file Close.
Syntax: >WC

WF **Write File.**
Syntax: >WF <filename> (A)
Note: Opens file for writing. 'A' option appends to existing file.

WF Turn Writing to File on and off.
Syntax: >WF ON
>WF OFF

WM Write Message to file.
Syntax: >WM <message>

WT WaiT and display text.
Syntax: >WT <message>

ZM Zero Margins.
Syntax: >ZM

KEY TO SYNTAX

(...) an item in parentheses is optional.

{...} an item in curly brackets may occur any number of times (possibly zero).

< character > an ASCII character, optionally in quotes.

< string > a sequence of ASCII characters, enclosed in quotes.

< message > a sequence of ASCII characters, optionally in quotes. It may include variable references.

< filename > a legal file identifier under the active filing system.

< integer > an integer between 0 and 65535

< int > an integer between 0 and 255.

< identifier > a variable identifier. Must start with a letter and may contain letters, digits, ".", "_" and "?".

< code > an ASCII code. The codes may be specified as decimal numbers, hex numbers (prefix with & or #) or ASCII characters. If an item is not a number it is sent as an ASCII character. To send the ASCII code of a decimal digit enclose the digit in quotes. Codes may be separated by commas or spaces.

< expression > is < num expr > or < str expr >

< num expr > is < num item > (< num op > < num expr >)

< num op > is +, -, * or /

< num item > is < fixed pt no. > or < identifier > with value a fixed pt no.

< fixed pt no. > is (< sign >) < uint > (. (< uint >)) or . < uint >

< uint > is unsigned integer

< str expr > is < str item > (+ < str expr >)

< str item > is < num item >
or < id > [(W) < int >]
or < id > [(W) < int >:]
or < id > [(W) < int >: < int >]
or < id > [W-1]

< condition > is < str cond > or < num cond >

< str cond > is < str item > < str comp > < str item >

< str comp > is one of: =, <, < =, >, > =, < >, IN, NOTIN

< num cond > is < num expr > < num comp > < num item >
or < num item > < num comp > < num expr >

< num comp > is one of: =, <, < =, >, > =, < >

separators Where more than one parameter is listed on a command line the parameters may be separated by spaces, commas, or equals signs. This is not shown in the syntax descriptions.

Variable references:

To print the value of a variable at any point in the text, type the variable name with an "&" immediately before and after it, e.g. &name&. If the value of the variable is a null string, and there is nothing else on the line, then the line is omitted altogether. If the value of the variable is the null string and there is a space after the variable reference then the space is not printed.

If the blank line or space is required, then type "!" before and after the variable name, e.g. !name!.

APPEN 2-10

A3. DEFAULT PRINTER CONTROL CODES

The following are the printer control codes built in to the PCW.PTR, EPSON.PTR and JUKI6100.PTR printer drivers by default.

All printer codes may be redefined to suit either alternative printers which do not use the standard codes, or for special purposes (See the SETPRINT utility program for full details.). Printer codes may also be redefined within a document by use of stored commands and the sections on printing and stored commands should also be studied.

The JUKI6100.PTR printer driver suits the JUKI6100 daisy wheel printer and is configured to enable the full use of right justified proportional printing. PCW.PTR suits the PCW printer and is also fully configured for justified proportional printing, though this is not really recommended, as printing is very slow with a dot matrix printer. The EPSON.PTR driver is configured for proportional printing, but the 'character width' settings have not been specified and proportional printing will take place in non justified mode if the printer supports proportional printing.

EPSON.PTR and PCW.PTR control codes

name		codes for on	codes for off
@	Reset printer	27, 64	-
b	Bold (emph.)	27, 69	27, 70
c	Condensed	15	18
d	Double-Strike	27, 71	27, 72
e	Elite	27, 77	-
i	Italics	27, 52	27, 53
l	Enlarged	27, 87, 1	27, 87, 0
n	Pica	27, 80	-
p	Proportional	27, 112, 1	27, 112, 0
q	NLQ	27, 40, 27, 120, 1	27, 120, 0
s	Subscript	27, 83, 1	27, 84
t	Superscript	27, 83, 0	27, 84
u	Underline	27, 45, 1	27, 45, 0

Note: There are differences in the other parts of the printer driver which is the reason why there are two different drivers which share the same printer control codes.

JUKI 6100.PTR control codes

name		codes for on	codes for off
@	Reset printer	27, 13, 80	-
b	Bold (emph.)	27, 87	27, 38
c	Condensed	27, 31, 9	27, 83
d	Double-Strike	27, 79	27, 38
e	Elite	27, 31, 11	27, 83
i	Italics	-	-
l	Enlarged	27, 17, 10	27, 17, 74
n	Pica	27, 31, 13	27, 83
p	Proportional	27, 80	27, 81
q	NLQ	-	-
s	Subscript	27, 30, 6, 27, 85	27, 68, 27, 30, 9
t	Superscript	27, 30, 6, 27, 68	27, 85, 27, 30, 9
u	Underline	27, 69	27, 82

Notes on line pitch.

The following list gives an indication of the number of characters printed per inch when the appropriate control code is specified. By default, PROTEXT will print in draft quality with a pitch of 10 characters per inch.

Pica.

n	pica	10	
l	enlarged pica	5	
c	condensed pica	17	
lc	enlarged/condensed	8.5	

Elite.

e	elite	12	
l	enlarged elite	6	
c	condensed elite	20	(Not available on PCW printer)
lc	enlarged/condensed	10	

Proportional.

This figure will vary with different printers and according to the different character widths used for each character, but generally the average may be considered to be equivalent to the elite pitch of 12 characters per inch.

Note: Not all printers will be able to support all the possible combinations of typestyles and the printer manual should be consulted for full details.

A4. EXPANSION STRING TOKENS AND DEFINITIONS

a). Expansions string tokens

TOKEN	LETTER KEY	FUNCTION KEYS				DEFAULT DEFINITION
No.		PCW		CPC		
80	EXTRA-A	SHIFT	f1/f2	SHIFT	f5	ª
81	EXTRA-B	SHIFT	f3/f4	SHIFT	f1	●
82	EXTRA-C			CTRL	f0	©
83	EXTRA-D					†
84	EXTRA-E			CTRL	f6	¦
85	EXTRA-F	ALT	f1/f2			ƒ
86	EXTRA-G	ALT	f3/f4			Ø
87	EXTRA-H	ALT	f5/f6			½
88	EXTRA-I	SHIFT	f5/f6	SHIFT	f4	Æ
89	EXTRA-J	SHIFT	f7/f8	SHIFT	f7	Ç
8A	EXTRA-K	ALT-SHIFT	f1/f2	SHIFT	f8	«
8B	EXTRA-L	ALT-SHIFT	f3/f4	SHIFT	f9	»
8C	EXTRA-M	ALT-SHIFT	f5/f6			
8D	EXTRA-N	ALT-SHIFT	f7/f8			
8E	EXTRA-O			SHIFT	f6	º
8F	EXTRA-P	EXTRA	f1/f2	CTRL	f8	¶
90	EXTRA-Q	EXTRA	f3/f4	CTRL	f5	¿
91	EXTRA-R	EXTRA	f5/f6			®
92	EXTRA-S	EXTRA	f7/f8	SHIFT	f0	ß
93	EXTRA-T					™
94	EXTRA-U					⊦
95	EXTRA-V	NORMAL	f3/f4	CTRL	f1	ø
96	EXTRA-W	NORMAL	f5/f6	CTRL	f4	æ
97	EXTRA-X	NORMAL	f7/f8	CTRL	f7	ç
98	EXTRA-Y			SHIFT	f3	¥
99	EXTRA-Z					
*9A			f1/2	CTRL	f3	254,"CAT",13
*9B		ALT	f7/8	CTRL	f9	254,"ERA *.BAK",13
*9C			FIND	CTRL	f2	254,"FIND",13
*9D			EXCH	SHIFT	f2	254,"REPLACE",13
*9E		EXTRA	ENTER	CTRL	ENTER	254,"EXEC EXFILE",13
*9F		ignore token			unused	

Note: The tokens prefixed with an asterisk (*) are set by PROTEXT and are not re-definable.

APPEN 4-2

A5. KEY TRANSLATIONS

These are the codes which must be used in a phrase definition to string command sequences together.

PCW	CPC	code	PCW	CPC	code
ALT-@	CTRL-@	0	ALT-[	CTRL-[	27
ALT-A	CTRL-A	1	ALT-.	CTRL-\	28
ALT-B	CTRL-B	2	ALT-]	CTRL-]	29
ALT-C	CTRL-C	3	-	CTRL-£	30
ALT-D	CTRL-D	4	-	CTRL-0	31
ALT-E	CTRL-E	5	ALT-<	CTRL-<	218
ALT-F	CTRL-F	6	ALT->	CTRL->	219
ALT-G	CTRL-G	7	ALT-(	CTRL-(	220
ALT-H	CTRL-H	8	ALT-)	CTRL-)	221
ALT-I	CTRL-I	237	ALT-*	CTRL-*	222
ALT-J	CTRL-J	10	ALT-+	CTRL-+	223
ALT-K	CTRL-K	11	ALT-/	CTRL-/	231
ALT-L	CTRL-L	12	ALT-hyphen	CTRL-hyphen	227
ALT-M	CTRL-M	238	ALT-space	CTRL-space	235
ALT-N	CTRL-N	14			
ALT-O	CTRL-O	15	↑	↑	240
ALT-P	CTRL-P	226	↓	↓	241
ALT-Q	CTRL-Q	17	←	←	242
ALT-R	CTRL-R	18	→	→	243
ALT-S	CTRL-S	19	SHIFT-↑	SHIFT-↑	244
ALT-T	CTRL-T	20	SHIFT-↓	SHIFT-↓	245
ALT-U	CTRL-U	21	SHIFT-←	SHIFT-←	246
ALT-V	CTRL-V	22	SHIFT-→	SHIFT-→	247
ALT-W	CTRL-W	23	ALT-↑	CTRL-↑	248
ALT-X	CTRL-X	24	ALT-↓	CTRL-↓	249
ALT-Y	CTRL-Y	25	ALT-←	CTRL-←	250
ALT-Z	CTRL-Z	26	ALT-→	CTRL-→	251
TAB	TAB	9	RETURN	RETURN	13
SHIFT-TAB	SHIFT-TAB	228	SHIFT-RETURN	SHIFT-RETURN	236
ALT-TAB	CTRL-TAB	225	ALT-RETURN	CTRL-RETURN	236
DEL→	CLR	16	ALT-CAN	CTRL-CLR	230
SHIFT-DEL→	SHIFT-CLR	229	ALT-CUT	CTRL-DEL	232
ALT-DEL→		5	ALT-COPY	CTRL-COPY	234
←DEL	DEL	127	⊞	COPY	224
SHIFT-←DEL	-	211	STOP/EXIT	ESC	252
ALT-←DEL	-	212	Enter edit mode		253
SHIFT-COPY	SHIFT-COPY	233	Enter command mode		254

APPEN 5-2

A6. SAMPLE MERGING DOCUMENTS

This chapter consists of a number of sample merging documents intended to give some idea of the variety of possible uses for PROTEXT. Some of them are complete documents in their own right, but many are working extracts from larger documents, intended to demonstrate solutions to a variety of problems. The first comment line in each program describes the use of the document and the next line describes the type of records required in the data file. There is a brief description of how the program works and what it does, after each document.

The last example is worthy of study, not so much for its usefulness, but as an example of methods of programming and the versatility of PROTEXT.

The first example is only for reference and is 'EXAMPLE6' from the earlier chapters, listed in full.

```
>CO   EXAMPLE6
>CO data file requires name, tel, 4 address lines
>PL 24        ;for screen display
>CP OFF       ;purposes only
>DF datfile3  ;open data file
>RU name tel addr1 addr2 addr3 addr4 dummy ;read data fields
>SK addr4>""    ;check for last address line empty.If so, skip
>AV "Enter Date " date   ;request date
>SV year="1986"          ;set the year

&name&
&addr1&
&addr2&
&addr3&
&addr4&

Dear &name&,
>CE  Renewal date - &date& &year&
    Thank  you  for  your  letter  about  the  insurance  for  &addr1&
&addr2&  &addr3&  &addr4&.  We  think  that  you  will  find  the  rates
quoted.......

>IF tel>""            ; if telephone - print
    If you would be kind enough to telephone us......
>EL                   ; otherwise - print
    If you would be kind enough to write to us ......
>EI                   ;end of IF EL block
```

```
>CO  Invoice printer and calculator
>CO  No data file required
>SV vatrate=0.15
Description  Retail Quant   Disc  Total     Net          VAT    TOTAL
>RP
>AV "Description of goods "descrip 12
>AV "Retail price "ret"Quantity "quant"Discount "disc
>IF disc=""
>SV disc=0
>EI
>SV totret=ret*quant off=totret/100*disc net=totret-off vat=net*vatrate
>SV totinc=net+vat gnet=gnet+net gvat=gvat+vat gtotinc=gtotinc+totinc
>----------------.-----.-------.------.--------.----------.---------.-R
&descrip&   &ret& &quant& &disc& &totret& &net&       &vat& &totinc&
>AV "Another entry? Y/N " yeno
>UN yeno[1]<>"Y"
                                             ------------------------------
                                     TOTALS   &gnet&      &gvat& &gtotinc&
                                             ------------------------------
>ST
```

The above template file creates invoices. Based on information requested, it will calculate the various totals, net values and VAT. After each item the option of entering another item is offered. Any answer but 'Y' will stop further entry and the grand totals will be printed. Use the PRINT command.

```
>>>     This example is part of an invoice generating template
>>> no data file is required
>RP      ; repeat enter order until correct
>AV "Protext order (Eprom, disc, cassette): ",orderp
>SV orderp=orderp+" 0 0 0"  ; default quantities of zero
>IF orderp[1]=" "           ; if no figures were entered
>SV orderp=orderp[2:]       ; strip leading space
>EI
>SV ep=orderp[w1] dp=orderp[w2] cp=orderp[w3]
>AV "Re-enter order (y/n)? ",again?
>UN again?[1]<>"Y"
```

This is an extract from another type of invoice generator. In the full template, the prices would be calculated from the quantities and the procedure repeated for other products. The prices would be stored in the template file as variables.

The quantities of each of the 3 formats of PROTEXT are entered as a single string, e.g. '10 20 10', and this string is split into the 3 individual numbers. Each quantity defaults to zero if omitted. The whole is enclosed in a repeat loop to allow the order to be re-entered if a mistake was made.

```
>CO Data file creator
>CO No data file required.
>ZM                         ;set margins to zero
>CS Type END to stop
>AV "Enter name " name      ;ask for name
>IF name="end"              ;see if it equals end
>ST                         ;and if so stop
>EI
>AV "Telephone number " tel "Address 1 " addr1 "Address 2 " addr2
>AV "Address 3 " addr3 "Address 4 " addr4
>SV dummy=""
>IF '"' IN addr1           ;check for double quotes in first address
>SV addr1="'"+addr1+"'"    ;if so, wrap address in single quotes
>EL                         ;or
>SV addr1='"'+addr1+'"'   ;wrap address in double quotes
>EI
"!name!"
"!tel!"
!addr1!
"!addr2!"
"!addr3!"
"!addr4!"
!dummy! .
>IN datamake                ;call file again.
```

The above file will create a data file of names, telephone numbers and addresses. The file must be saved on disc with the same name as used in the '>IN' command at the end. Typing 'end' in answer to 'Enter name' will close file and stop further processing. The 'IF' command checks for house names with double quotes and if found, wraps the data in single quotes. The remainder of the variables are wrapped in quotes at the printing stage. The last line calls the file from disc again. The file is created by using PRINTF with a filename.

```
>CO routine to separate off initials
>CO no data file required
>AV name              ;Request name to separate initials from.
>SV n1=name[w1] n2=name[w2] n3=name[w3] n4=name[w4]  ;split name
>IF n4=name[w-1]      ;check for same as last word in 'name'
>SV inits=n1+n2+n3        ;if so, 'inits' = n1, n2 and n3
>EI
>IF n3=name[w-1]      ;repeat for third word
>SV inits=n1+n2           ;if so, 'inits' = n1 and n2
>EI
>IF n2=name[w-1]      ;repeat for second word
>SV inits=n1              ;if so, 'inits' = n1
>EI
&inits&   &name&
```

This file is an example of manipulating variables, in this case to extract the initials in a person's name. It takes account of likely variations in the number of initials.

```
>CO Example routine to select one of four data files.
>CO 4 data files, called 'address.a-d', 'address.e-k', etc
>AV name
>SV suff="junk s-z l-r e-k a-d"
>RP
>SV suff=suff[w2:]
>UN name[1] >=suff[1]
>SV df="address."+suff[w1]
>DF &df&
```

This file makes use of a RP UN loop to repeatedly shorten a variable until the first letter of 'name' is greater than or equal to the first letter of the variable. 'junk' is simply there to be discarded on the first loop. Finally, joining of variables and text is used, followed by making use of the ability to use a variable in a command, to define the data file. This would be a suitable routine to select a data file to be read, according to the surname, enabling data to be kept in a number of files.

```
>CO using stored commands to set other stored commands.
>CO no data file
>AV "Side margin " sm
>AV "Page length " pl
>AV "Single sheet Y/N" cp
>AV "No. of copies " nc
>IF "Y" IN cp
>CP OFF
>EL
>CP ON
>EI
>SM &sm&
>PL &pl&
>NC &nc&
```

This one simply asks for a number of details about the options required and then uses the variables either as part of the stored command, or to determine the setting of the appropriate stored commands.

```
>CO EXAMPLE to reconstruct a data file for import to another database
>CO data file called original.dat is required
>ZM                       ; zero margins
>PL 7                     ; page length=No. of field lines
>DF original.dat          ; old data file
>RU initials,surname,add1 add2,add3,tel,dummy    ; read existing data
>CO  New file rearranges order, combines intials and surname and adds
>CO  a fourth address field
!surname! !initials!
!tel!
!add1!
!add2!
!add3!
!dummy!
!dummy!
```

The file is created by using the PRINTF command and specifying the new filename. This is a simple example only and any data may be manipulated as necessary. For example, dates may be converted from one format to another.

Note that the optional '!' variable markers are used to ensure that even empty fields are printed to the new file.

```
>>> Example to convert dates to suitable form for conditional comparison.
>>> or different format. No data file required
>>> Date form  01 Mar 86
>AV "Enter Date (dd mmm yy) " date
>IF date[2]=" "
>SV date="0"+date
>EI
>SV mon=date[w2] cal="Junk Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec"
>SV ncal="Junk 01 02 03 04 05 06 07 08 09 10 11 12"
>RP
>SV cal=cal[w2:] ncal=ncal[w2:]
>UN cal[w1]=mon
>SV ndate=date[w3]+ncal[w1]+date[w1]+" " ndate=ndate[1:6]
>SV mdate=date[w1]+":"+ncal[w1]+":"+date[w3]
Original - &date&   New form - &mdate&   Comparison form &ndate&
```

```
>CO                         Prime Number Generator
>DF primes ; never accessed so any filename on disc can be used
>ZM
>IU i
>SV i=3.00
>EL
>SV i=i+2
>EI
>SV j=3.00
>>>
>RP
>SV t=i/j
>SK t[w-1] = 0   ; if remainder zero
>IF j*j > i
>SV iint=i[w1]
&iint&
>SK 1=1
>EL
>SV j=j+2
>EI
>UN 0=1
```

This file calculates prime numbers and should be started with PS. The only point to note is the 'DF' command, which must have the name of a file on disc, even though it is not accessed. It is necessary, as it permits 'SK' to work.

A7. PROTEXT - COMPATIBILITY BETWEEN CP/M AND AMSDOS VERSIONS

Text files created with either version of PROTEXT may be edited using the other version. The AMSDOS version includes both the ROM and CPC disc versions of PROTEXT. The notes below will be of interest to those using both versions of PROTEXT on a regular basis.

1. AMSDOS document files, when loaded into the CP/M version will cause document mode to be selected.

2. AMSDOS program files, when loaded into the CP/M version, will cause document mode to be selected. This can be changed using CONFIG. If both CP/M and AMSDOS versions of PROTEXT are being used then the mode to be selected for 'imported files' should be set to 'program' with the CONFIG utility. With this setting both AMSDOS documents and programs will cause the correct mode to be selected.

3. CP/M documents, when loaded into the AMSDOS version, will cause program mode to be selected. Type DOC or D to select document mode. AMSDOS versions 1.23 and later recognise CP/M PROTEXT documents and select document mode.

4. The CP/M version supports some additional stored commands. If the document is required to be printed using the AMSDOS version of PROTEXT these commands should not be used. Note that the AMSDOS program PROMERGE or PROMERGE PLUS will be required as well as AMSDOS PROTEXT for many of the stored commands.
   The stored commands new to the CP/M version are:

   IE, PE, PO, PP, WF, WC, WM

5. The CP/M version does not use the language stored commands (DA, EN, FR, GE, IT, SP, SW). All characters can be printed without commands to select the printer font, because all characters can be defined with CONFIG. If these stored commands are present in a document they will be ignored by CP/M PROTEXT.

6. The following special characters, supported by CP/M PROTEXT are not recognised by AMSDOS PROTEXT, although they will not be corrupted if a file containing them is loaded into AMSDOS PROTEXT:

   All characters with accents, the special characters available on function keys, the arrow characters, non-break hyphens, multiple markers.

APPEN 7-2

A8. PRINTER AND DISC PROBLEMS?

This appendix covers certain problems that sometimes occur with printers and disc drives, and suggests the most common solutions.

a). The printer

(i) The printer doesn't print anything.

Check the following:

Is the printer power lead plugged in?
Is the printer cable connected to both printer and computer?
Is the cable plugged in the right way up?
Is there paper in the printer?
Has the printer been switched offline?

(ii) A blank line is printed after each line of text.

The printer is automatically feeding the paper an extra line each time. This is a feature of printers that is needed by some software, but must be disabled for PROTEXT.

This feature is usually controlled by a switch ('DIP switch') in or on the printer. The printer manual will give details. The switch should be set to 'no automatic line feeds'.

If this switch is set correctly and double line feeds still occur, then the problem is probably being caused by the way the printer cable is wired up. The best solution is to replace the cable by one wired correctly.

Should both of these solutions fail, there is a software means to prevent the extra line feeds. Use the SETPRINT utility to create a printer driver (or edit the printer driver already in use). Use the 'define character translations' section and define the line feed character, ALT-J (press ALT-J when asked for the character to define) to give zero. This will stop PROTEXT sending line feeds to the printer.

b). The disc drive

(i) The message 'Disc missing or read fail - Retry, Ignore or Cancel?' appears.

The most likely cause is that PROTEXT is trying to access a disc (for example to save a file) and the disc is not in the disc drive. The disc may appear to be in the drive but may not be fully inserted.

Another likely cause is that the disc in the drive has not been formatted. All discs must be formatted before use - this can be done using the PROTEXT command 'DFORM'.

The message can also be given on the PCW computer if a CF2DD disc is put into drive A, or the wrong way round in drive B.

The least likely cause, but one that may occur, is that the disc has become faulty. This occasionally happens, especially with a heavily used disc. If this appears to be the case, 'R' should be pressed to try again. If the error persists the disc should not be used any further - it will however usually be possible to recover most of the files on the disc and save them on another disc. Since there is always a small chance of a disc becoming faulty it is absolutely essential that backup copies are kept of all important files.

(ii) The message 'Disc unsuitable for drive - Retry, Ignore or Cancel?' appears.

This message is given on the PCW computer when an attempt is made to save to a CF2 disc in drive B. CF2 discs must be in drive A in order to save files on them.

A9. SYSTEM ERROR MESSAGES

This section lists error messages relating mainly to disc operation. These are termed system error messages because they do not relate to any particular program or command, and may occur at any time.

Disc missing or read fail - Retry Ignore or Cancel?

(i) The disc being used has been taken out of the drive. Insert the disc and press R.

(ii) The disc is not formatted.
Press C to cancel the operation. The disc must be formatted using DFORM before it can be used.

(iii) PCW only. This error is given if a CF2DD disc is put into drive A, or the wrong way round in drive B.

(iv) The disc may be faulty or corrupted. Press R to retry. If the error persists try re-formatting the disc.

Drive not ready - Retry Ignore or Cancel?

The disc being used has been taken out of the drive. Insert the disc and press R.

Disc error - Retry Ignore or Cancel?
Seek fail - Retry Ignore or Cancel?
Data error - Retry Ignore or Cancel?
No data - Retry Ignore or Cancel?
Missing address mark - Retry Ignore or Cancel?
Media changed - Retry Ignore or Cancel?
Media change occurred - Retry Ignore or Cancel?

If any of these errors occur the disc may be faulty or corrupted. Press R to retry. If the error persists try re-formatting the disc.

Write protected - Retry Ignore or Cancel?

> The write protect tab is pushed in. Remove the disc, slide the tab out, and press R.

Disc unsuitable for drive - Retry Ignore or Cancel?

> PCW only. The disc in drive B is formatted as a CF2 disc, and so may only be written to in drive A.

Bad format - Retry Ignore or Cancel?

> The disc is not formatted, or is of a non-Amstrad format, or the disc may be faulty.

File is read only

> The chosen file cannot be written to because it has been protected. Use ACCESS to unprotect the file.

Directory full

> The maximum number of files allowed on a disc has been reached. There may still be room on the disc, so any unwanted files should be deleted.

Disc full

> The storage capacity of the disc has been reached. Often this can be remedied by erasing backup files. Type ERASE *.BAK or press ALT-f7 (PCW) or CTRL-f9 (CPC6128).

File not found
File does not exist

> There is no file of the chosen name on the disc. Check that the name was typed correctly, that the correct disc is in the drive, and that the correct drive is selected.

Bad filename

The combination of characters chosen as a filename is not allowed. Valid names consists of up to 8 characters, followed optionally by a full stop and an extension of up to 3 characters. Certain characters are not allowed.

Maximum number of files open

There is a limit to the number of files that can be open at the same time. In normal use this limit will never be reached.

Insufficient memory for program

The program has used all the available computer memory. This error will not occur in normal use.

EXEC file read error

A disc error occurred when reading commands from an exec file. This could be caused by removing the disc containing the exec file, or by a faulty disc.

This program will only run under Amstrad CP/M Plus

Arnor CP/M Plus programs will only work on Amstrad computers with CP/M Plus. In particular they will not run on other CP/M systems. This is because special use is made of Amstrad specific features to attain the best performance.

If this message occurs when using an Amstrad computer, which is possible if some other software is installed, all may not be lost. Contact Arnor for help.

APPEN 9-4

A10. SIMPLE LETTER EXAMPLE

The object of this chapter is to provide a guide to the typical procedures that would be followed when writing a document and printing it out.

We shall assume, for the purposes of this example, that PROTEXT has not been used before and that you will not, therefore, have created any 'Template files' to store your most frequently used layouts. The default settings provided by PROTEXT are ideal for creating a letter on single sheet A4 sized paper and 11 inch continuous stationery, so for this example, we shall assume that the letter is to be printed on A5 single sheet stationery.

As we shall use A5 stationery for part of the time, it is a good idea to start by creating a 'template file' to hold the required setting for page length, margins etc. before we continue with writing the letter. Having done this, whenever we want to write a letter on A5 stationery, it will only be necessary to load the correct template file at the start. Template files are ideal for storing the settings for any frequently used document layouts. If we ALWAYS used A5 single sheet paper, it would be more sensible to reconfigure the default settings with CONFIG, so that they were always present whenever PROTEXT was loaded.

Creating a Template File

The object of the template file is to create, through the use of stored commands, the settings we require for page length, all the various top, bottom and side margins, suitable rulers and any other features which we might like to add, such as proportional printing, or double line spacing.

The first thing we need to know is the size of the paper we are going to print on. In the case of A5 paper, it is approximately 5.75" wide by just over 8" tall.

We should start by deciding how many characters across the paper we wish to print. We shall print our letter in the default Pica print style, which gives 10 characters per inch, which means that if we were to print from one edge of the paper to the other, that we would be able to print about 56 characters, but we obviously want a clear margin on either side. The default side margin setting in PROTEXT is 5 characters (that is, 5 spaces will be printed from the left hand edge of the paper before the text is printed), which on a small sheet of paper is a bit large. so we shall reduce the side margin to 4 with the SM command. As we also want a similarly sized margin on the right of the page, that means that we can only print 48 characters on a line (56-4-4), so we will want a ruler that is 48 columns wide. We also want to inset the start of each paragraph by 5 columns, so when we create our ruler, we shall insert a tab marker in column 5.

The next thing to consider is how many lines we can print on a page. The standard setting for line pitch (the number of lines printed per inch) is 6, so we should set the page length to 48 (8 X 6 lines per inch), but one thing we must also take into consideration is that most printers require a certain amount of the paper fed beyond the print head, in order that the bail bar can hold it. Our printer requires about 1", so the effective length of our page must be reduced by 6 lines to allow for this. We shall therefore set the page length to 42 with the PL stored command.

The next items to consider are the top and bottom margins. As we are not bothered about using headers and footers, we can ignore the default settings for header and footer margins, as PROTEXT will ignore them if they are not specifically turned on by defining them. The default settings for top and bottom margins are 3, which are ideal for use with continuous stationery, but we don't really require a top margin as we are going to position the paper at the right position, so we shall set it to zero with the TM command.

We must also consider the fact that it is not normally possible to print on the last few lines of a sheet of paper, as by that time the end will have fed past the roller and will no longer advance as it should. We could have allowed for this by reducing the number of lines per page, but if we did that, then the paper would not be ejected correctly at the end of the page, so we shall leave the bottom margin setting at its default value of 3, which is ideal for our printer. PROTEXT will automatically allow for these three lines when calculating how many lines of text to print and then eject the paper at the end correctly.

The final thing that we must do is set printing to single sheet stationery and the methods used will vary with different printers. Ours is fully 'Epson' compatible, so we can make use of the PS (paper sensing) command to control printing and don't need to use the CP OFF command, which we would otherwise have used to set it for single sheet printing.

That now completes all the necessary settings, but as we are going to use plain paper we can save ourselves some repetitive typing by including our address in our template file.

Our template file will therefore look something like:-

```
>CO A5 letter template.
>---!-------------------------------------------R
>PL 42      ; page length 42
>TM 0       ; top margin zero
>SM 4       ; side margin 4
>PS ON      ; paper sensing on for single sheet
                            Arnor Ltd.
                            118 Whitehorse Road,
                            Croydon.
                            CRO 2JF
```

Depending on the computer and arrangement being used, we should now save this template onto either our text file disc, or, if we are going to copy the templates onto drive M at start up time, onto our start of day disc. We shall save it with the name 'A5LETTER' and in future, every time we want to type an A5 letter, all we have to do is type 'M A5LETTER' before commencing the document. You will notice that we have used the 'M' command to merge the template file in. We could have used the load command instead, but the advantage of using 'M' is that it does not name the document, so there is no chance of us accidentally saving it with the name 'A5LETTER', which could occur if we loaded it in and forgot to change the name of the document.

We can create as many template files as we like, including stored commands for double line spacing, proportional printing, continuous stationery, or whatever we want.

Writing a letter

We shall assume you have loaded PROTEXT with the start of day disc in the normal fashion and PROTEXT is in command mode, waiting for a command. As we are going to write a short letter on single sheet A5 stationery, we will load the appropriate template file, using 'M A5LETTER'. After a second our template will appear in the upper part of the screen and we can press the STOP key to go into edit mode. Once in edit mode, the cursor will be positioned at the start of the document, so pressing ALT-] will take us to the end of the template and we can now start typing.

Once we have created and corrected the text, it is a good idea to have a thorough read through before printing, to make sure that we have made no mistakes, so we shall use ALT-[to go to the start of the document and scan through it using ALT-↓ to make sure. If the document is likely to be more than one page long, it is also a good idea to make sure that the page breaks fall in convenient places in the text and this can be done very easily by using the ALT-(and ALT-) commands, so we shall once again go to the start of the document with ALT-[and then using ALT-) move through the document a page at a time. If we find that a page breaks in an unsuitable place, we have only to move back in the text to a suitable place and insert a '>PA' stored command, to force the page to break where we want.

Once we have checked the page breaks, we shall return to command mode with the STOP key and carry out a spelling check, by typing SPELL from command mode. We insert the dictionary disc as and when told to and make whatever corrections are necessary as the checking proceeds. When completed, we are back in command mode and perhaps we were warned that the document may need re-formatting because we altered the length of a word whilst correcting them, so we type FORMAT.

Before we print the letter out, we should save it onto a disc for future reference, by typing 'S' and pressing RETURN. We can then enter the name we want to give the letter and press RETURN again. We should try to give it a meaningful name, so that when we do a CAT of the disc, it will be obvious what the contents of each file are, even at a later date.

We shall print two copies of the letter, a draft one for our records and a final one in Near Letter Quality. Having made sure that we have put a piece of paper into the printer and that it is positioned correctly, all we need to do is type 'P' (the abbreviation for the PRINT command) and press return and our draft copy will be printed. If the letter is more than one page long, the printer will stop at the end of the first page and allow us to insert a new sheet before pressing the 'on line' button on the printer (If we were using the PCW printer, we would have to press the EXIT key) to print the next page. Once that is done, we can insert another new piece of paper and this time we shall use the 'PQ' command (PRINTQ) to print the second copy in NLQ mode.

That is all there is to it. It has probably taken as long to read this chapter as it would have done to write the letter! If you now want to write another letter, you can either type CLEAR from command mode, or press ALT-SHIFT-CAN from edit mode to remove the old letter from memory.

There are, of course, many other printing commands available, but for most purposes, these will be the ones that you will use most.

GLOSSARY

GLOSSARY

ASCII
A code used to represent text within a computer.

BACKUP FILE
The copy of the previous version of a file that is retained whenever a file is saved.

BLOCK
A contiguous portion of text defined by two markers at the first and last character included.

BOX
A rectangular portion of text defined by two markers at the top left and bottom right corners of the rectangle.

BYTE
The unit of storage of a computer capable of holding a single character.

CAPS LOCK
The state of the computer in which all letters typed are entered as upper case.

CATALOGUE
A list of files stored on a disc.

CHARACTER
Any letter, digit or symbol.

COMMAND
An instruction to the computer to do something.

COMMAND MODE
The mode of operation of a computer program (e.g. PROTEXT, BASIC) that allows commands to be entered.

CONFIGURATION
The operation which customises the word processor for a particular printer or an individual user.

CONTROL CODE
A number that represents an instruction and not a character. In particular an instruction to a printer (e.g. to underline or print italics).

CP/M
A special program called an operating system which controls the use of the disc drives.

CURSOR
The blob on the screen marking the place at which text will be entered.

DATABASE A program to store and display information. An electronic card index.

DATA FILES Files created from the information in a database or created with a word processor, frequently for use in mail merging.

DEFAULT The initial setting of some feature of the word processor, such as insert mode on, page length 66.

DISC A means of magnetic storage of data allowing fast access.

DOCUMENT MODE The usual mode of operation of PROTEXT, in which all the formatting features may be used.

DRIVE Disc drive, the part of the computer into which discs are inserted. On machines which have more than one drive it is possible to select which drive is used to for loading and saving.

EDIT MODE The mode of operation of a word processor that allows text to be entered and changed.

FILE A group of data collected together and stored on disc.

FLOPPY DISC A type of disc commonly used with microcomputers.

FONT One of the character sets available on a printer.

FOOTER One or more lines printed at the bottom of each page, in which titles and page numbers can be printed.

FORM FEED A control code that causes a printer to feed the paper through to the start of the next page.

FORMAT (i) An operation on a portion of text which causes the text to be arranged in some regular way.
(ii) An operation that must be performed on a disc to make it ready for use.

FUNCTION KEYS Special keys on the keyboard, usually configurable to carry out special tasks.

GLOBAL	Everywhere, i.e. throughout the entire text.
GROUP	A collection of files on a disc. Particularly on large capacity discs it can be convenient to separate the files into different groups. The catalogue of files only shows one group at a time.
HEADER	One or more lines printed at the top of each page, in which titles and page numbers can be printed.
K	A unit of storage equal to 1024 bytes.
LINE FEED	A control code that causes a printer to feed the paper through by a single line.
LOAD	Copy data from disc to the memory so that the data may be used.
LOWER CASE	Small letters.
MAIL MERGE	An operation whereby a standard letter or document is printed many times using names and addresses taken from a file.
MARGIN	(i) The normal limit of text to the left and right. (ii) The blank space on the paper surrounding the text.
MARKER	A pointer to a particular location in the text.
MEMORY	The part of a computer where data (programs, text, etc.) is stored, and may be accessed quickly.
MERGE	Load one file into memory without clearing what is already there.
MODE	One of a number of states of operation of a program, e.g. edit mode, command mode, document mode, insert mode.
PAGE THROW	An instruction to start a new page when printing.
PHRASES	Strings or blocks of text which may be inserted into a document with a single key press.

PRINTER DRIVER A file that configures a word processor to print on a particular printer and/or paper.

PROGRAM MODE The alternative to document mode, used for editing programs. The formatting features are not available.

PROTEXT A computer program (word processor) produced by Arnor.

RETURN The code used in the computer to mark the end of a line.

RIGHT-JUSTIFY An operation on a line of text which causes the text to be spread out over the full length of the line.

RULER A line at the top of the screen showing left and right margins and tabs.

SAVE Copy data from the memory of the computer to a disc so that the data may be preserved.

SHIFT LOCK The state of the computer in which all keys pressed give the character normally given if the key is pressed together with SHIFT.

SOFT CHARACTER A character in the text used for formatting.

STATUS LINE The line at the top of the screen when in edit mode which shows various useful information.

STORED COMMAND A command concerned with printing, that is stored in the text but is not printed as part of the text.

TAB A position in a line to which the cursor moves when the TAB key is pressed.

TOGGLE A command which switches some function between on and off.

TOKEN A special character which represents a string of characters. Tokens are the means by which phrases and function keys work.

UPPER CASE Capital letters.

USER AREA — A CP/M term meaning the same as GROUP.

UTILITY — A program or command, distinct from the main program which performs some useful function.

WILDCARD — A special character which will match any character when searching for a string.

WORD PROCESSOR — A computer program that handles the entering, storing, formatting and printing of text.

WORD-WRAP — The operation of moving a word onto the next line when there is not enough room on the current line.

GLOSSARY 1-6

INDEX

INDEX

PROTEXT v2.20 SUPPLEMENTARY DOCUMENTATION

Contents

Protext is fully compatible with the PCW9512 from version 2.20. Note that Protext acts differently depending on the machine on which it is running, and the same version works on PCW9512, PCW8256/8512 and CPC6128.

PROTEXT for the PCW 9512

The instructions given in the manual for creating a start of day disc (page INTRO 3-3) need to be changed in just one detail. The instruction to type 'PROTEXT <MAKEPCW' should be ignored. PCW9512 users should instead type:

```
        PROTEXT  <MAKE9512
```

Printer driver for the PCW9512

A new printer driver PCW9512.PTR has been added which is used by the PCW9512 daisy wheel printer.

PCW9512 Printer ports (see WORD 4-8)

INT - This command will automatically load the printer driver 'PCW9512.PTR' instead of 'PCW.PTR'.

PAR - Selects the built in parallel port on the PCW9512.

PAR2 - This new command has been added to select the parallel port on the optional serial/parallel interface, and is equivalent to PAR on the PCW8256/8512.

SER - Selects the optional serial port.

Keyboard configuration

The PCW9512 keyboard is slightly different to the PCW8256/8512. A configuration file 'PROTEXT.CFG' is provided to account for this variation. This will be copied onto the start of day disc on a PCW9512.

Disc formatting and copying

The PCW9512 uses only the double sided disc format (though it will read the PCW8256 single sided format). The command 'DFORM' will therefore always use the double sided format. The command 'DFORMD' DOES NOT WORK on a PCW9512.

The 'DCOPY' utility does not work on the PCW9512. The DISCKIT program supplied with the computer should be used to copy discs.

SPELL v3.10 Documentation Update

Summary of new features

The dictionary may now be held on any drive, including drive M.
The dictionary drive and group are configurable.
Different size dictionaries can be created.
A list dictionary to file option has been added.
The spelling checker is fully compatible with the PCW 9512.

Dictionary structure

The Protext dictionary is now stored as an ordinary file, and so may be copied by any standard method (such as Protext's 'COPY' command, or the CP/M 'PIP' program). When a dictionary is created it occupies its full size, but will have room for words to be added. The word count option in the utility menu gives an indication of how full the dictionary is as a percentage.

The supplied dictionary is 178K in length. Other possible sizes are 89K, 267K, 356K, 445K, 534K, 623K and 712K.

Selecting the dictionary drive (option V)

This is a new option on the main menu, which allows the drive to be used for the dictionary to be chosen. Pressing V will cause the dictionary drive indication on the status line at the top of the screen to cycle through the available drives.

Listing a dictionary to a file (option L)

Using the list dictionary option it is now possible to create ASCII dictionary files. Selecting L from the utility menu will ask whether the words should be listed to the file 'DICT.ASC'. If Y is pressed then this file will be created on the current text drive, otherwise the words will be listed to the screen.

Remember to use option D in the main menu to select the required drive before listing the dictionary to a file.

If there is insufficient room on any disc for a file containing the whole dictionary (e.g. on a CPC6128), then it is possible to split it between several files. Start the listing and abort by pressing ESC before the disc becomes full (the dictionary sectors are displayed on the screen as an indication of progress). Change the disc and restart the listing at the last word that was in the first file.

Initialising a new dictionary (option I) (see SPELL 4-4)

Unlike previous versions of Protext and Prospell, this option does not format a new disc, because the dictionary may share a disc with other files. The disc must therefore have been formatted previously. Note that if a CF2 format (single sided) disc is being used on a PCW8256/8512 this must be formatted with the Protext command 'DFORMD'. This gives 178K available disc space; other formatting commands only give 173K.

Before initialising a dictionary the dictionary drive displayed on the status line should be checked. If this does not match the drive on which the dictionary is to be created, return to the main menu and use the V option (described above).

The initialise dictionary option will now ask for the name of a dictionary, which may be up to 8 characters long, and the dictionary size. The sizes available will be indicated, depending on the available disc space.

The sizes are:

1	89K	5	445K
2	178K	6	534K
3	267K	7	623K
4	356K	8	712K

A CF2 disc will only hold sizes 1 and 2. Drive M on the PCW8512 or PCW9512 will hold sizes 1 to 4. A CF2DD disc will hold sizes 1 to 7. Size 8 would require a high capacity disc, such as a hard disc.

Note that size 1 is not large enough to hold the supplied dictionary, and would only be used for a supplementary dictionary.

Configuring the Protext dictionary drive (see UTIL 2-4)

The spelling checker options section in the CONFIG program now has two further options. These are the default drive and group for the dictionary. By default these will be set to drive A, group 0.

The dictionary drive may also be selected from the main menu, but using CONFIG is the only way to select the dictionary group.

There are several possible ways of using the dictionary, and different users will find different configurations more suitable.

Dictionary on drive B:

On a PCW8512 or a PCW8256 or PCW9512 with drive B fitted, it may be best to keep the dictionary on a CF2DD format disc in drive B. This allows greater capacity for adding new words, and the convenience of not having to change discs. Text files may also be kept on the same disc.

Dictionary on drive M:

Putting the dictionary on drive M will produce a great increase in the speed of the spelling checker. The start of day disc should be set up to copy the dictionary onto drive M. For example, on a PCW9512 which will have the dictionary file on the start of day disc, add the following line to the file 'STARTUP':

```
ECOPY A:*.DCT M
```

On a PCW8512 it will be necessary to remove the start of day disc and insert the dictionary disc. Add the following lines to the end of 'STARTUP':

```
; Remove the start of day disc and insert the dictionary disc
; Press a key when ready ...
PAUSE
ECOPY A:*.DCT M
```

Copying the dictionary to drive M will of course reduce the space available for files. Paragraph f(ii) on page INTRO3-9 applies. If large files are to be edited it is not recommended that the dictionary be copied onto drive M.

Of course, if words are added to the dictionary on drive M it will need to be copied back to the original disc to save the new words.

Note that a PCW8256 without memory expansion will have insufficient space on drive M for the dictionary.

Conversion of old type dictionaries (option K)

Dictionaries created or supplied with previous versions of Protext (below v2.20, SPELL versions below 3.10) or any version of Prospell below v4.10 will need to be modified before use with the new Protext spelling checker.

An option is provided to update the old type of dictionary to the new format. This is option K in the utility menu. The procedure is extremely quick and will create a file called PROSPELL.DCT. The 'name dictionary' option may then be used to change the name, if required.

How to merge dictionaries

Using the various new commands described above it is now possible for the first time to merge dictionaries together. The method for doing this is:
(i) list the dictionaries into ASCII files. If listing more than one dictionary on the same disc *don't forget to rename DICT.ASC* before using list again.
(ii) create a new large dictionary using the initialise option.
(iii) use the build to dictionary option on each ASCII file in turn. Allow plenty of time for this!