

JRT SOFTWARE
42 TURNERS AVENUE, COROMANDEL VALLEY
SOUTH AUSTRALIA, AUSTRALIA 5051

◆ COPYRIGHT

Copyright (C) 1984 1985 1986 1987 JIM TUCKER and IAN CURL. All rights reserved world wide. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language in any form or by any means without the prior written permission of JIM TUCKER or IAN CURL.

◆ TRADEMARKS

FLASHPRINT and FLASHKEY and FLASHBASIC and FLASHPATH and FLASHGEN and FLASHCODE and FLASHNET are trademarks and copyright JIM TUCKER.

◆ DISCLAIMER

The suppliers and authors make no representations or warranties with respect to the contents hereof and specifically disclaim any implied warranties of merchantability or fitness for any particular purpose and reserve the right to change this publication and the product it describes without obligation to notify any person or organization of such revision or changes.

◆ COPY PROTECTION

Programs supplied by JRT SOFTWARE are not copy protected and you may make as many copies as you wish. But only for your PERSONAL use on a single computer.

◆ CARTOONS BY STEPHEN STANLEY

THIS USER GUIDE

CONTENTS

THIS USER GUIDE is meant to be read. It describes FLASHPRINT as we would describe it to you personally if we could visit you at home or place of business. Here is a brief description of each section:

- SECTION 1: Making a quick start. For those anxious people who can't wait to read the book. An introduction to FLASHPRINT and what it does. We describe the CP/M and DOS versions and the files on your disk.
- SECTION 2: How to load FLASHPRINT.
- SECTION 3: Entering commands to your printer. There are several types of commands which make FLASHPRINT very powerful.
- SECTION 4: How to customize FLASHPRINT for your printer. You may begin with the files supplied. But if you have a complex printer or special requirements you can make FLASHPRINT print anything your printer can handle.
- SECTION 5: Using CONVERT. Here's how to turn a normal word processing table of printer commands into a short and swift special file which FLASHPRINT uses to send code to the printer.
- SECTION 6: How to send printer control codes from your system prompt using FLASHSET.
- SECTION 7: Some hints for Wordstar users.
- SECTION 8: How it works. This is mainly for wizards who are curious about these things. It is not required reading.
- SECTION 9: For CP/M only. How to use FLASHKEY with the CP/M version and Wordstar.
- SECTION 10: For DOS only. How to remove FLASHPRINT and reclaim the memory it occupies.
- SECTION 11: Designing your own characters with FLASHGEN. We describe both the CP/M and DOS versions.
- APPENDIX A: A chart to convert hex and decimal.
- APPENDIX B: ASCII character chart.
- APPENDIX C: A short tutorial on your printer manual.
- APPENDIX D: The DVORAK keyboard (for CP/M FLASHKEY only).

The file FPNEWS.DOC contains the latest information which may not be included in this user guide.

SECTION 1: INTRODUCTION

THE QUICK START

No details, no explanations, no customizing, no guarantees. But for those who want to try it straight away...

CP/M USERS: Format a disk containing CP/M. Copy your normal version of Wordstar on to the disk, along with the overlay files, then three files from your FLASHPRINT disk -- FPn.COM (where n is your version of Wordstar), FLASH.DEF, and FPDEMO.TXT. Boot FPn. You will find yourself (if you did it right) running Wordstar. Now print the file FPDEMO.TXT.

DOS USERS: Copy FP.COM, FLASH.DEF and FPDEMO.TXT on to a disk containing DOS. Boot FP. This will return you to the DOS prompt. Then print (using PRINT or CONTROL-P then TYPE) the file FPDEMO.TXT. FP will stay in memory until you reboot.

WELCOME TO FLASHPRINT

OUR PRINTER is clever. It can print pica, elite, proportional, compressed, wide, superscript, subscript, bold, enhanced, italics and underlined. It can change line spacing and margins, set vertical and horizontal tabs and print graphics.

And although it was made in Japan it can print characters in Swedish, French and even English. And being a civilized sort of thing it can even go fishing ~~to the lake~~.

The problem is our word processor and other programs, aren't so smart and they don't let us print these things easily. Some programs don't let us print them at all.

So welcome to FLASHPRINT. FLASHPRINT lets you control your printer and print the magic things the printer salesman promised. If your printer can print it so can FLASHPRINT. This user guide was printed on a dot matrix printer which costs less than \$600. But you can just as easily use FLASHPRINT to control a \$5000 laser printer or a \$50,000 professional typesetter.

FLASHPRINT allows you to select any type style or any printer function by keying one or two characters in your document or program. Example:

under **lined** elite *wide* and compressed
normal **bold** *italics* ^{SUPERSCRIPT} and QUALITY MODE

And let us boast that this user guide was keyed and printed using only Wordstar, which is not the most printer-friendly program. And although Wordstar does not do proper justification our printer has inbuilt commands which do that. We can control it thanks to the huge variety of control codes we can send it using FLASHPRINT.

TWO VERSIONS OF FLASHPRINT

Now before we pick up the nuts and bolts, let's tell you that there are actually two versions of FLASHPRINT. One for CP/M 8-bit machines running Wordstar and another for IBM and compatibles.

The CP/M version was created to solve a particular problem: The lack of printer control codes available to Wordstar users. It has been upgraded several times but it is specific to Wordstar. You have this version if you have an 8-bit CP/M computer. And you must have Wordstar (either version 2.2, 3.0 or 3.3).

The DOS version of FLASHPRINT is for the IBM PC. It is memory resident and allows printer control from any program. Any word processor, any spread sheet, or any programming language.

This user guide covers both the CP/M and DOS versions. Both versions use the same command structure and printer tables so if you know how to use one version you will know how to use the other. Files containing FLASHPRINT printer tables or commands transferred from CP/M to DOS and vice versa will be compatible.

FLASHPRINT IS POWERFUL, BUT...

FLASHPRINT is very powerful and we have tried to make it simple for the novice. But that's a conflict. The more power we build into it the more you have to read. No user guide is perfect and not everything can be learned from books. If that were possible the world would not have schools, only libraries.

You probably spent many hours learning how to use your computer from a book. We cannot come into your home or office and tell you all you need to know about FLASHPRINT. Wish we could. Instead, we suggest you spend an hour or two reading this guide thoroughly then another hour or two experimenting. If you are a computer illiterate (and we all were once) joining a computer club is an excellent idea.

Skim through this guide to get the feel of FLASHPRINT. Learn what it's all about. Don't worry if you don't understand everything first time round. After skimming the guide go back to the beginning and read it thoroughly. Scribble notes in the margin. Use a marker pen to highlight things you think are important.

This user guide assumes you are familiar with your computer's operating system. If we did not do that we would have had to include several books within this book.

And before you continue... Don't forget to make a copy of your FLASHPRINT disk. It is not copy protected. We trust you to make copies only for your personal use.

Now we will talk about the files on your FLASHPRINT disk. It probably contains more files than you expected. What are they? The description is in three parts: Files on DOS disks only, files on CP/M disks only and files which are on both versions.

FILES ON DOS DISKS

FP.COM is FLASHPRINT for IBM-PCs or compatibles using MS- or PC-DOS - it's the file you specify at the system prompt (or include in your AUTOEXEC.BAT file).

FPAPCIII.COM is the same thing for NEC APC-III users. If you do not have an NEC APC-III you can delete this file. But if you do have an NEC APC-III you must use this instead of FP.COM. NEC users should delete FP.COM and rename FPAPCIII.COM to FP.COM. This user guide will assume you have done that.

MARK.COM and RELEASE.COM are two special programs which allow DOS users to release FP from memory and reclaim the memory. They are described in Section 10.

FLASHGEN.EXE is the character creator for DOS. It allows you to design you own characters and print them using FLASHPRINT. See Section 11.

FILES ON CP/M DISKS

FP22.COM, FP30.COM and FP33.COM are three versions of FLASHPRINT for different version of Wordstar. The digits refer to the version of Wordstar you are using. FP22 is for Wordstar V2.26, FP30 is for V3.0 and FP33 is for V3.3. No other versions are supported. Make sure you have a legitimate unmodified version of Wordstar else FLASHPRINT may not work. FLASHPRINT will most likely not work if the printer output routine has been "patched".

You may rename FP???.COM anything you like, so long as it is a COM file. From now on we will assume you have renamed the version you want to use FP.COM.

FLASHGEN.COM is the character creator for CP/M and Wordstar users. It enables you to create you own characters and print them from Wordstar with FLASHPRINT. See Section 11.

FLASHGEN.BAS is the BASIC source code for FLASHGEN. You will require Microsoft or a compatible BASIC if you wish to run this. If you understand BASIC you can modify the program for your requirements.

Also included in the TBL and DEF files are demonstrations of FLASHKEY for CP/M users. (FLASHKEY is a separate program for DOS users).

FILES ON ALL DISKS

YOU HAVE the following files no matter which version of FLASHPRINT you are using.

*.TBL are several text files containing the code required by printers. You may change or create these files using any word processor. Generally the filename describes the printer. Normally you will need only one of these files. You may customize the file for your printer or create your own TBL file.

*.DEF are files created from *.TBL files using the CONVERT program supplied with FLASHPRINT. Section 5 explains how to convert a TBL file into a DEF file. Note that DEF files are not in ASCII and cannot be printed or changed using a word processor. Table and definition files are described in Sections 4 and 5.

FLASH.DEF is the default printer definition file which loads if you do not specify a filename when you load FP.COM. FLASH.TBL is the "source" file. It is based on Epson code and should be sufficient to get you started. However, it contains only a few commands. FLASHPRINT allows hundreds and you might like to change this table or create your own.

CONVERT.COM is a program. It converts a text printer table TBL file into a definition DEF file. FLASHPRINT allows you to create a table of commands for your printer using plain English (well, as near as we can get to English in this computer world) and CONVERT turns the TBL file into a compressed file with the extension .DEF (saves space and makes it faster) which FLASHPRINT uses. See Section 5.

BIGCAP.TBL and BIGCAP.DEF are code which allow you to print double-height capital letters on an Epson compatible printer. How to use these files is described in Section 3.

FPDEMO.TXT is a text file containing a few simple FLASHPRINT commands. You may like to print this when FLASHPRINT is loaded to test compatibility with your printer.

SUPER.TBL and SUPER.DEF are table and definition files which we used to print this user guide. SUPER.TBL has lots of bells and thingamebobs for our printer. If you really want to know how to drive FLASHPRINT print SUPER.TBL (without FLASHPRINT loaded as it contains @ characters). The SUPER.TBL file is heavily commented and will give you plenty of info on driving your printer.

FLASHSET.COM is a program which allows you to initialize your printer from the CP/M or DOS command. It is particularly useful for DOS batch files. See Section 6.

Finally, FPNEWS.DOC is the equivalent of a "readme" file and contains the latest information about FLASHPRINT. Please read or print it.

SECTION 2: LOADING FLASHPRINT

GETTING STARTED

TO RUN FLASHPRINT you must have FP.COM (and we have assumed CP/M and NEC users have renamed FLASHPRINT to FP.COM) in the active drive.

You also need a printer definition file. These are the files on your FLASHPRINT disk ending with the extension .DEF. You need only one of them. Choose the one with the name which matches your printer. Most likely the default FLASH.DEF will work on your printer if it is Epson compatible.

If you are using CP/M you must also have the three Wordstar files (WS.COM WSMGS.OVR and WSOVLY1.OVR) on your FLASHPRINT disk. Or you may copy FP.COM and a DEF file on to your Wordstar disk. That is probably easier as it will probably already contain CP/M.

FLASHPRINT always needs at least one DEF file containing printer commands. By default the DEF file is named FLASH.DEF. When you boot FLASHPRINT it will look for a printer definition file named FLASH.DEF and load it into memory. The definition file contains the code sent to your printer.

At the system prompt simply enter the program name FP and hit the enter or return key. FLASHPRINT will sign on and tell you it has loaded the default definition file FLASH.DEF. The CP/M version will also load and run Wordstar. The DOS version will return you to the system prompt and remain in memory.

But what if the FLASH.DEF file is not what you want? Simple. When you load FLASHPRINT you may specify the name of one or two DEF files you wish to use instead. Here's how:

```
A>FP FILE1 FILE2
```

Here are the rules: One or two file names (but no more) may be specified. They are assumed to end with .DEF and the extent should not be specified. You may append a drive name such as B:FILENAME but DOS users cannot append a path.

Here's an example of loading a special definition file. Suppose you want to load a definition file to drive a daisy wheel printer and the DEF file is called DAISY.DEF.

```
A>FP DAISY
```



Now suppose you have special coding in a file called SPECIAL.DEF but want to use the printer codes contained in FLASH.DEF. Here's how it looks:

```
A>FP FLASH SPECIAL (or) A>FP SPECIAL FLASH
```

It doesn't matter which order you load the two DEF files but if they both contain the same commands the first file takes precedence.

If a definition file is not found FLASHPRINT will report the error and return you to the operating system without loading.

If the supplied default file FLASH.DEF does not suit your purposes you may delete it (or rename it) and rename or create any other definition file with the filename FLASH.DEF. This allows you to have a customized version without having to specify a filename each time you load FLASHPRINT.

Now that you have learned how to load FLASHPRINT you may do it immediately. Disks will turn and messages will appear on the screen - CP/M users will find they are running Wordstar and DOS users will find themselves back at the system prompt. It doesn't matter that you do not know how to use it. It will be there ready when you have finished reading the next section.

After loading FLASHPRINT you might like to print the demonstration file FPDEMO.TXT. CP/M users will of course use Wordstar and DOS users can use just about anything that prints. For instance copying the file to the PRN device will work. DOS users can even display the file on the screen and print FLASHPRINT commands via the PRINT SCREEN key.

ATTENTION DOS USERS

Normally there will be no conflict between FLASHPRINT and other resident programs. However, in general you should load FLASHPRINT after other programs which intercept printing, such as a printer spooler.

For instance, we found that if we load our print spooler AFTER loading FLASHPRINT then FLASHPRINT becomes inactive. You should note that FLASHPRINT behaves itself regarding printer interrupts but other programs may not be so elegant. If there are problems try loading files in a different order. Generally you may always load FLASHPRINT after any other program. See also the section on how to remove FLASHPRINT from memory.

SECTION 3: FLASHPRINT CODES

HOW TO PRINT WITH FLASHPRINT

This section tells you how to use FLASHPRINT commands. What each command does is entirely up to you when you create your own table and definition file. How to create tables and customize FLASHPRINT for your requirements and your printer is explained later. But before you do that we think it's important that you know how to use FLASHPRINT.

THE SIMPLE COMMAND

The @ character is unique to FLASHPRINT. When FLASHPRINT finds the @ character it is not printed. Nor is the character which follows - Instead, FLASHPRINT uses the character following @ for its own purposes. The character following can be any character. Caps and lower case are different. We call @ and the character which follows the FLASHPRINT command.

If you simply want to change type, set margins, do underlining or change colors on a printer with a color ribbon this is all you need.

The character @ followed by any other character simply sends a string of bytes (printing characters or control codes) to your printer. The @ does not print. Nor does the character after it. Instead the command sends the code contained in your DEF file to the printer.

Let's look at an example. The default FLASH.DEF file supplied uses @U to begin underlining (note that the U is in caps) and @u (lower case) to turn it off. We may key the following:

Underlined @Uthis is underlined@u and this is not underlined. This prints as follows when FLASHPRINT is loaded.

Underlined this is underlined and this is not underlined.

What happened? The character @ told FLASHPRINT that the character "U" is a command. FLASHPRINT read the command and scanned FLASH.DEF, found a matching entry, and instead of sending "U" to the printer it sent the code for underlining. It did the same sort of thing for @u to turn off underlining.

Incidentally, we usually use a capital character to turn on something and the same lower case character to turn it off, but you can create your own table and use any characters.

Well, that's not particularly flash, you may say. Wordstar can underline. But you might have noticed that the spaces between words are also underlined by the printer. Wordstar's underline does not underline spaces between words. So far, not a giant leap for mankind but certainly a small step.

Those who have tried to configure Wordstar for printers will know that Wordstar printer commands are limited to a few dozen bytes. FLASHPRINT @ commands can send very long strings of code to your printer (thousands of bytes). You could, for example, use a single command to turn on underlining, bold, italics and wide. Also, commands can contain text. You can print your name and address with a simple command. We'll tell you how in the section on customizing FLASHPRINT.

Almost any character may follow the @ character to make a simple command. We use @B for bold, @I for italic, @C for compressed and so on. But you may customize FLASHPRINT with the commands you prefer.

Finally, we must tell you that you must not use the following commands (they are used internally for other commands) @@ @# @< @! @ (@\ @| @/ @! -- keep reading and you will know why.

You could stop here. The simple command is the easiest to use.

THE TRANSLATION COMMAND

We thought it might be handy if you could substitute one character for another. The translation command does a simple substitution. In other words, when FLASHPRINT finds X in a document it will print Y -You do not have to precede the character with the @ character.

Here's how it works. When EVERY character is sent to the printer FLASHPRINT tests it to see if a substitution is to be made. Your customized DEF table is scanned for a translation entry. If one is found the substitution is made. If not the character is printed normally. (In the simple command if no command is found in the table nothing is printed and the command is ignored).

Your DEF table is scanned only when translation is switched on. You switch translation on with @[and switch it off with only a] (note no @ preceding the right square bracket). In other words, every character following @[scans the DEF table to see if something else has to be substituted. Once you wish to stop the scanning and want the characters themselves to print (no substitution) key a right square bracket in your document.

You can use the translation command in several ways. Here is a single translation: @[=. If an entry for = is found in the DEF table the appropriate code will be sent to the printer else = will print. Note this is not the same as @= but a different command.

Here is an example of translation in a document. Assume we have set up the characters * % ^ in our definition file to print something else.

[< This switches translation on. This is a document. If the characters * % ^ are in our definition table then they will not be printed. Instead the code in our definition table (a single character, a string of characters, printer control codes or any combination of all three) will be sent to the printer instead.] Now we have switched off

translation with a right square bracket and * % ^ will be printed normally. Any translation entries in the table will be ignored.

With translation turned on characters which you do not normally use can print something else. For instance, + can print a hollow square and * can print a blob.

Remember, when translation is switched on with @[EVERY character in your Wordstar document is intercepted by FLASHPRINT and a substitution is made if there is an entry in your definition table. A single right square bracket] stops it.

The translation command is unique. It is the only command which prints the actual character if no entry is found in your definition file. All other commands ignore the command if no entry is found.

THE BYTE COMMAND

The byte command is similar to the translation command. The difference is that if no entry is found in the DEF table for the command nothing happens. (In translation if no entry is found the character found is printed). The byte command is turned on by @(and turned off by) -- note no @ is required before the right round bracket.

THE HEX COMMAND

FLASHPRINT lets you send hex values to the printer from a file. You can send any code, even if it is not in your DEF table(s).

Your printer manual explains values required to turn various printing modes on and off. These are usually given as hex numbers. We cannot give a tutorial on hex. All you need know is that a hex value consists of a two-character pair. The characters may be any valid digit (0 to 9) and any character from A to F is a number 10 to 15. You may use hex whenever you want to send the printer a sequence of codes not contained in your definition DEF file.

The hex facility uses the characters < and > as delimiters. The command to turn on hex is @< and hex is turned off by >. Note @ is not required before the > to turn off hex.

Here is an example of a FLASHPRINT text containing hex values: This is text @<0D 0A>and this is more text.

This example sends a carriage return (0DH) and line feed (0AH) to your printer. Although it is a continuous line in your document it will print as follows:

```
This is text
and this is more text.
```

You can use this in a Wordstar header or footer for multiple lines - (You may, of course, create a simple command to send a return).

Here are some rules: Hex alpha characters MUST be keyed in caps and each hex number must be two digits (a leading zero may be required) - The two-digit hex numbers must be separated by a space. There is no check that the hex number is valid so @<1J X?> will send garbage. A carriage return or line feed is ignored within a hex string. This allows you to key a string over several lines.

Being able to send hex numbers directly from a document is very useful when printer commands require a variable number, such as setting the margins for justified printing.

We think the hex facility fills a need but you would not use it all the time. If you have a sequence of code you use often it is best to include it in your definition table and print it using the simple command which requires only two keystrokes.

THE ESCAPE COMMAND

We have described four commands. But there are other commands built into FLASHPRINT that you should know about.

Most printers need an escape (ASCII 27 decimal or 1B hex) preceding a sequence of control codes. Your printer manual may describe the escape character as simply ESC. FLASHPRINT sends ESCAPE to the printer whenever a # character is printed. For example, if ESCAPE+6 (in BASIC: LPRINT CHR\$(27)+"6";) turns on italics on your printer (it does on ours) you can simply include #6 in your program or document.

Here's another example. The NEC Pinwriter uses ESC+!(value) where value is a number between 0 and 255 to choose from a wide range of print styles. The following would turn on print style 20 decimal: #!@<14>. Use the decimal-hex table at the back of this book to make the conversion.

And what if we really want to print a @ or a # -- doesn't FLASHPRINT interpret these as commands? Yes it does.

But we fixed that, too. To actually print a # all you have to do is key @#. And simply key @@ (two of them) to print a single @ - FLASHPRINT recognizes these as special simple commands. This is the reason you MUST NOT include @ or # as simple commands in your definition table. They will be ignored and the actual characters will print.

PRINTING BIG CAPS

The double-height capital letters for Epson printers are contained in a file called BIGCAPS.TBL and BIGCAPS.DEF. They are meant to be used in word processing applications although some users may find them useful in other applications. Note that these only work on a true Epson compatibles.

The big capital letters are not meant to replace more sophisticated programs which allow you to print several different type styles. But for section headings (such as in this user guide) or for printing labels they are very useful.

As the printer can only print one line at a time each line of big caps requires two actual printing lines. You must key the line twice in your document. The first line is preceded by the command @/ and terminated by the single / character. The bottom line is preceded by @\ and terminated by the \ character.

The @/ and @\ commands are really just byte commands which look up a different type of entry in your printer file. They can be used for other purposes.

Note that the graphic characters need to be joined with reduced line spacing. We use 8/72-inch which is @Z in our SUPER table. We resume normal six lines per inch spacing after printing the bottom line of big caps with @z. You must change the line spacing else the top and bottom lines will not touch.

Here's an example of how big caps appear on the screen:

```
@/THIS IS BIG CAPITALS/@Z  
@\THIS IS BIG CAPITALS\@z
```

If you use big capital letters with Wordstar or other word processing programs you will have to make adjustment for the slight difference in line spacing. Wordstar users should print using form feeds to ensure the next page is correctly aligned.

THE SPARE COMMAND

We keep adding things, so in FLASHPRINT we have included a spare command which, although not in any printer tables supplied, can be used by you for any purpose.

It is @(your commands) -- it operates the same as the byte command and no @ is needed to turn it off. In other words, besides @() you have @{} as an additional byte command level.

And there is one more spare commands you might like to know about. It is @| terminated by a single | character. Note this is not a colon but the one that sometimes looks like it. We have reserved this for the middle line of big caps which are three lines high. But meanwhile you may use it as another spare byte command.

We would like to emphasize that you can use any command for any purpose. For instance, by creating a table the commands to print the rows of big caps can be used for something else such as printing different characters which you design.

A SUMMARY OF COMMANDS

Don't let the different type of commands confuse you. Begin with the simple command and work your way from there. @ something simply sends something else to the printer.

In this summary we have preceded the command with the entry type in your TBL files described in the next section. Read how to customize FLASHPRINT for their meaning. And in this summary the word "substitute" means a printing character, a control code or a string containing any mixture of both.

C - THE SIMPLE COMMAND may be all you ever need. @X (where X is any character except those reserved) sends a substitute to your printer.

T - THE TRANSLATION COMMAND where @(contains translated characters) is sort of special. It sends a substitute if the characters within the brackets exist in the printer table, else the characters themselves are sent.

B - THE BYTE COMMAND where @(characters) sends a substitute to the printer if the command exists in the table, else nothing is sent.

H - THE TOP OF BIG CAPS where @/BIG CAPS/ sends the bit image character for the top line of big cap chars. It is simply an alternate byte command and may be used for other purposes.

M - THE MIDDLE OF BIG CAPS where @|BIG CAPS| is not used and may be used as an alternate byte command.

L - THE BOTTOM OF BIG CAPS is @\BIG CAPS\ where this sends the bit image mode for the bottom of big caps. It is a byte command and may be used for other purposes.

S - THE SPARE COMMAND is @(anything goes) and is a byte command for any use.

THE HEX COMMAND @<hex values> can be used to send control codes to the printer not contained in your DEF table. If you don't know about hex you can ignore it.

MISCELLANEOUS COMMANDS not in any table where # sends ESC to the printer, @@ really prints @ and @# really prints #.

So what command prints what? It's up to you. Whatever you put in your printer table. @U can tell the printer to begin underlining. But it could tell your printer to print your name in wide bold pica, your address in elite italics and your phone number in Greek. You can customize FLASHPRINT for any command and for any printer.

SECTION 4: JUST FOR YOU

HOW TO MAKE YOUR PRINTER SING

Now here's the nitty gritty. How to customize FLASHPRINT for your printer and your requirements.

Printers require various control codes and ASCII characters to turn on and cancel various print modes. Exactly what codes are required are described in your printer manual.

Now there are good manuals and bad manuals. In our experience most of them are awful! If you simply CANNOT understand your printer manual TALK TO THE PEOPLE WHO SUPPLIED YOUR PRINTER.

The code needed for your FLASHPRINT commands is keyed by you into a table file and ends up (we'll show you how) in a definition file. When you boot FLASH it loads the definition file into your computer's memory. The definition file contains the special code required by your printer to do its special things, such as print bold, elite, wide and print special graphic characters.

Wizards may ask us: Why convert the file? The answer is this: A DEF file will be much shorter than a TBL file and therefore work much faster. A DEF file is actually an assembled TBL file. For instance 3-digit ASCII numbers are packed into a single byte, comments are stripped, and a unique table is created. This means there is no loss of printing speed when you run FLASHPRINT.

A definition file is full of "magic" numbers (programmers call it a binary file). If you had to create this file in binary you would curse the complexity of FLASHPRINT. So we made it easy.

You simply key your printer commands into what we call a "source" file using a word processor such as Wordstar. A separate program named CONVERT reads your source file and does all the things needed to create a DEF file.

Creating a new definition file for your printer and requirements is easy. There are only two steps: Modify or create a source file and create a new definition file by running CONVERT.



Before reading further we suggest you print some TBL files we have supplied with FLASHPRINT. Refer to your print out while reading the following instructions. And we'll say that again. Please print the TBL files and refer to them while reading the next section. It is important for the understanding of this section.

Now let's go. You will have read that there are several types of commands in FLASHPRINT: The simple command, the translation command, the byte command and so on.

Examination of the source TBL file(s) will show you that each entry begins with a letter which describes the type of command. We call it an entry type. Here are the entry types:

- C - A simple command entry @x (where x can be any character)
- B - A byte entry: character(s) between @(and)
- T - A translation entry: character(s) between @{ and }
- H - High (top) line of big caps: char(s) between @/ and /
- M - Middle line of big caps: character(s) between @| and |
- L - Low (bottom) line of big caps: char(s) between @\ and \
- S - Spare for character(s) between @[and]

So an entry must begin with one of these letters (caps or lower case) at the start of line. The character describes the type of entry we are making. For instance, if this is a simple command for @B for bold the type of command is simple so the letter is C (for command).

Following the entry type is a space followed by the character you want for the command. In this case we suggested B for bold. The letter B (caps) can be entered in any of the following formats:

ASCII - A character preceded by a single quote. Example: 'B enters the letter B (ASCII value 66 decimal) in the table.

DECIMAL - Any decimal number from 0 to 255. The number may optionally be appended with d or D (for example the letter B has an ASCII value of 66 decimal so we could key 66 instead of 'B or we could key 66D but the D is not needed).

HEX - Any hex number from 0 to FF. The hex number must be appended with H h X or x. A leading zero is not required (FFH is okay and so is FX for decimal 15). Numbers A to F may be in caps or lower case.

CONVERT also allows OCTAL and BINARY values. If you know about these things you may use them. Most people will not. Still, here for wizards, are the rules.

OCTAL - Any number from 0 to 377 appended with O o Q or q.

BINARY - Any number from 0 to 11111111 appended with B or b.

In summary, your source file can contain values in several different forms. The letter capital B can be:

'B	< ASCII character
66 66d 66D	< value in decimal
42h 42H 42x 42X	< value in hex
102o 102O 102q 102Q	< value in octal (wizards)
01000010b 01000010B	< value in binary (wizards)

Control codes can also be entered by preceding the character with a circumflex ^. For example Control-M (return or ASCII 13 decimal) can be entered as 13 or ^M.

Entry types may be in any order as the complete table is scanned but for easy of reading we suggest you group your entries.

So far we have explained that each entry consists of a type and the character (ASCII hex decimal and so on) which will activate the entry. The entry type and character which will activate the entry are normally in the first two columns of your table.

What follows the entry type and command character is simply the code you want the command to send to your printer. The code can be entered in any of the formats described -- ASCII, decimal, hex, octal or binary or as a control character or string. More about strings soon.

Here's an example. Assume you want the command @S to tell your printer to go into condensed superscript. Look up your printer manual for the required coding. Suppose it says ESC+S+0 turns on superscript and 15 decimal turns on compressed printing. In BASIC you would write this:

```
10 LPRINT CHR$(27)+"S"+CHR$(0);
20 LPRINT CHR$(15);
```

With FLASHPRINT you simply create a table entry and send the same thing by keying @S in your document.

```
C 'S 27 'S 0 15      * @S = compressed superscript
```

```
^ ^ ^ ^ ^ ^      ^
1 2 3 4 5 6      7
```

- 1 - This is a simple Command (C is NOT preceded by a quote)
- 2 - The ASCII command character to follow the @
- 3 - Sends escape (27 decimal) to the printer
- 4 - Sends a capital S to the printer
- 5 - Sends a value of zero (null)
- 6 - Sends 15 decimal (ASCII SI) for compressed
- 7 - This is comment and is ignored by CONVERT.

When you key @S into a document and print it using FLASHPRINT the values in columns 3 to 6 will be sent to the printer.

COMMENTING YOUR TABLE FILES

Comments are an important part of any computer language. They allow you to make notes for your use, or others, so that even years later you can read exactly what you have done (or what you intended to do) and why you did it.

Comments can be written on a piece of paper and filed and lost. We believe the best place to put a comment is to include it with the computer code.

Comments or remarks can be included in your source file by keying * or ; (semi-colon). The asterisk is for DBASE people and the semi-colon is for assembler wizards. Either may be used. Any characters which follow on the same line will be ignored. (Similar to REM in BASIC). We encourage comments and they may be included freely as they add practically nothing to the CONVERT time and are not included in the final definition file.

Blank lines are ignored and do not need commenting so we include them to make our table files easier to read.

An entry is terminated by <RET> (new line) or comment. However, the backslash character \ may be keyed to continue the entry on to the next line. Characters following \ on the same line are treated as comment. There are several examples of this in the TBL files.

This simply allows you to create single entries hundreds of bytes long but keep them within the printing or display margins on your printer or screen.

INCLUDING TEXT IN YOUR TABLES

You'll notice in the previous examples that ASCII characters were preceded by a single quote. Now we'll show you an easy way to include text in your TBL file.

Instead of keying 'a'b'c you may key <abc>.

You simply start the text with a < character and end it with a > character. Here's an example for a function which prints my name.

```
C 'J <JIM TUCKER> 13 10 * @J prints my name/return/lf.
```

Note that the values 13 10 are carriage return and line feed. Most printers require the line feed as well as the return. A few do not.

Here is an example which prints your name and address when @N is printed:

```
C 'N\           This comment after \ character is ignored
<Mr John Citizen
55 Walnut Street
Anytown CA USA 12345
>
```

There are a few points to note about this entry. All characters keyed between < and > including the return and line feed at the end of line are saved in the DEF file just as you keyed them. When you are entering a string you do not need the \ continued line character. Also note that the end of text > character is on the next line. This ensures that the return and line feed after the zip code 12345 is also saved in the DEF file.

Here's how to actually include the > character in a string entry.

```
C 'X <End of text is the > '> < character.>
```

This will print: End of text is the > character.

To see how TBL files are created we strongly suggest you print the tables we have supplied. They can become a model for your personalized table. You can borrow an entry from here and one from there. The table files are full of comments. We suggest you build a table file a few commands at a time.

Once you have created a table you must convert it to a DEF file for FLASHPRINT. That's easy because your computer does all the work. It's described next.



SECTION 5: MAKING A DEF FILE

DEF FILES SAVE TIME AND SPACE

Conversion is simple. The table file(s) we have described are simply the instructions for FLASHPRINT. But FLASHPRINT does not read a TBL file. A TBL file may contain notes, comments, values as hex, ASCII, decimal or strings. If FLASHPRINT had to work this out on the fly it would be slow and consume tons of disk space and memory. So instead...

CONVERT.COM turns a TBL file into a DEF file which FLASHPRINT can process very fast. The process is simple. At the system prompt you simply convert your TBL file by keying `CONVERT FILENAME`

In this example FILENAME is the name of your TBL file. The TBL file can be named anything but must be of type TBL. PRINTER.TBL is legal but PRINTER.PTR is not.

If no errors are detected the source code will be saved as PRINTER.DEF on the current disk.

A different name for the DEF file and drives may be specified. For example you may `CONVERT B:SOURCE A:DEST`. In this example the file SOURCE.TBL on drive B will create DEST.DEF and write it to A drive. DOS users may NOT specify a directory.

CONVERT is written in assembler and is VERY swift. No source file should take more than 60 seconds to convert into a DEF file. The DOS version with a hard does disk does it in second or two.

If DEF file of the same name already exists CONVERT will ask you if you want to overwrite it. Pressing any key except Y will abort the program.

CONVERT ERROR DETECTION

During conversion CONVERT detects errors (such as incorrect hex numbers, short lines, missing comment delimiters and so on) and reports them on your screen. As the error messages are in English, and not reported as magic numbers, they will not be described here. You will know what they mean when you see them.

CONVERT displays the offending entries on the screen and pauses so you can see them. This prevents multiple errors from scrolling off the screen at a rapid rate. Press any key to proceed or press Control-C to abort.

To make errors easy to find CONVERT reports the line number and character number. For instance, 00011/00567 means CONVERT found an error on line 11 at the 567th character in your table. You may use this to easily locate the error when editing.

You MUST have an error-free assembly before a DEF file will be created or changed by CONVERT. If your source file contains errors nothing on the disk will have been changed. But we cannot know if CONVERT detects all errors so please CONVERT only on a backup disk.

CONVERT is forgiving about layout in the source file but for ease of reading we suggest the following: The line type character should be hard left in column 1 followed by a space then the command character (usually preceded by a single quote). Tab to column 8 to begin keying the actual printer coding. Our examples show the preferred layout.

Although CONVERT detects most errors you might find a couple of error messages ambiguous. For instance, take the following entry...

COMMAND ENTRIES FOLLOW

We meant to start the line with a * for comment. An error. But CONVERT finds the line starts with C which is okay. It thinks this line is a simple command entry. However, it then looks for either a space, a number or an ASCII character preceded by a quote. But there is no space or quote so CONVERT looks for a number. The letter "o" is not a number so it reports a numeric error. Better it reported "comment line not preceded by *" but it doesn't know that.

Still, as CONVERT displays the line number and line containing the error we do not consider this a serious deficiency. It is better than many error reports such as "E47" detected.

- Comments:
- If the value of n exceeds the right margin.
 - Setting the left margin position clear printing.
 - In the proportional pitch.
 - Once the left margin is set.



SECTION 6: FLASHSET

SENDING CODE FROM THE SYSTEM

Both CP/M and DOS disks include a program called FLASHSET.COM. This simple program allows you to initialize a printer from the CP/M or DOS prompt. To use it key:

```
FLASHSET <STRING>
```

The program name FLASHSET is followed by one or more spaces then follows the code you want to send to the printer. The code can consist of normal ASCII characters. For example: FLASHSET MY NAME IS JIM will send the string "MY NAME IS JIM" to the printer.

Well, that's not very useful. But FLASHSET allows you to send control codes to the printer from the command line. Codes must be expressed as decimal numbers and must be enclosed in square brackets. For example, some Epson-type printers require ESC+C+n to set the length of the form where n is the number of lines.

To set your printer to 70 lines (A4) size the FLASHSET command is:

```
FLASHSET [27]C[70]
```

Note that [27] is the decimal value for escape, then follows the letter C then n which is 70 (the number of lines on the page).

Several printer commands can be sent in a single string. The maximum length of the string is 100 characters including the square brackets.

We have done little error checking in this program and leave it to the operator to ensure the string is correct.

Common errors may include expressing an invalid decimal number (the maximum is 255). There should be no spaces in the string else they will be sent to the printer. For instance "[27] A" will send three codes to the printer: ESC, SPACE, A and may not perform as you expect.

FLASHSET is particularly useful for DOS users as it can be included in a batch file. Including FLASHSET in your AUTOEXEC.BAT allows you to easily initialize the printer when you switch on your machine. Or you might like to create batch files with names such as COMPRESS.BAT, BOLD.BAT, A4.BAT and so on.

DOS users may also include FLASHPRINT commands using @ in a FLASHSET string. For example you could use FLASHSET @B@W@U to set bold wide underlined. You must of course load FLASHPRINT into memory before executing a FLASHSET string containing FLASHPRINT commands.

You may include FLASHSET on most of your disks as it is written in assembler and is very compact (512 bytes).

SECTION 7: WORDSTAR HINTS

MAKING WORDSTAR WORK

Wordstar prints spaces for the left margin. If the printer is in underline mode the margin will be underlined. You must turn it off at the end of line. However, many printers allow you to set the margin internally. In this case set Wordstar's margin to zero using .PO0 and set a printer margin. This will prevent the margin from being underlined. Using the printer margin also means you may change the width of characters and keep a constant width in the left margin (the print head always returns to the set position). If your printer has no left margin control you must turn off underlining and reset the normal character width at the end of each line.

FLASHPRINT commands are counted as printed characters when Wordstar does word wrap or justification. The solution is simple. Enter your text and justify it if necessary (using ^B) then insert your FLASHPRINT commands prior to printing. The same goes for Spellstar. Use it first then insert your FLASHPRINT commands.

Your printer will begin printing a new document in the same mode set when you finished printing the previous document. You must be careful to switch all FLASHPRINT coding off at the end of printing. Or do what we do, turn the printer off and on again before printing. A printer "cold boot."

Some printers have an escape code for this which you can include at the start of each document. We include @! in some our tables to send the printer initialization code and include C '! 27 40 in our table.

A warning. Our printer requires ESC 40H to reset it. But 40H is the @ character! When we included this in Wordstar's printer initialization string the @ was intercepted by FLASHPRINT and strange things happened. You can include FLASHPRINT code in a printer initialization string. That way you send a huge string to the printer by using only two bytes. In fact we see no reason why the four Wordstar user controls ^Q ^W ^E ^R cannot contain FLASHPRINT commands.

You can include FLASHPRINT commands in Wordstar headers and footings - In fact it's a good idea to do this if you want to begin printing a document other than at Page 1. For example if you include @B to print bold at the start of text then start printing at Page 2 the command will not be sent to the printer. Instead put it in the head as .HE@B then the bold command will be sent to the printer no matter where you begin printing. Headers (and footers) need not contain text.

The escape # character is considered to be a page number by Wordstar when included in a heading or footing. Instead, use the hex command for ESCAPE which is @<1B> or precede the # character with a backslash "\" (see your WS manual which explains this).

FLASHPRINT allows Wordstar users to easily print multiple line headers and footers. Simply key hex carriage return, line feed in the header or footer: @<0D 0A>. Or make it a simple command. But you must print documents of more than one page using FORM FEEDS or reduce the page length because Wordstar does not know you can include a CR/LF in the middle of a header or footer. Silly thing.

A Wordstar header or footer must not be more than about 200 characters on the screen. But by including the characters in a FLASHPRINT command they can be hundreds of bytes in length. Wordstar knows only about the characters you keyed such as @something, not the number of characters in your table.

The latest hints are contained in the disk file FPNEWS.DOC. If you have any hints to add to the list we would be very pleased to hear from you.

SECTION 8: HOW IT WORKS

TECHNICAL TALK, FOR WIZARDS ONLY

This is not required reading. Most users can skip this. But for those who like to know about these things here's how FLASHPRINT works.

First the CP/M version: FP.COM loads at 100H where all CP/M programs load. If 5CH is empty then FLASH.DEF is assumed and appended to FP. If 5CH is busy the file is appended to FP and if 6CH is busy that file is appended too. Then FP is moved in memory above the space normally occupied by Wordstar. The location depends on the version. Wordstar is then read into 100H.

Then Wordstar is patched by FP so that printer output is directed to FP and the start of data (Wordstar PBGMEM) is relocated to protect FP. Finally a jump to 100H executes Wordstar. This is all rather neat because although Wordstar is patched it is done in memory. The disk version remains virgin pure.

In the DOS version FP.COM loads wherever (you never know where) and reads the DEF file(s) (5CH and 6CH) if specified then modifies the printer I/O interrupt. It exits back to DOS leaving itself resident - The techniques to do this are well documented. Any program which uses interrupt 17H for printer I/O (and that's just about everything) will be trapped by FP. The NEC APC-III version uses interrupt 1BH.

FLASHPRINT is written in assembler and occupies very little space - Computers are much faster than printers and FLASHPRINT will not slow down your printer unless it can print more than 5000 characters per second!

You might find your printer appears to be slow if you are sending complex bit image graphic characters. This is not because your computer has gone into go slow mode. It's because your printer is busy working out what to do with all the zeros and ones being sent to it at a rapid rate. Some printers do this much faster than others.

CONVERT is also written in assembler and can chew a 30K table file down to a 6K DEF file in less than 15 seconds on our DOS machine. In both version the entire table file is read into memory before conversion. This speeds things up no end. However, the length of the TBL is limited to about 50K which should be long enough for all applications.

Both versions of FP load at 100H and begin with a jump (3 bytes). What follows is a table of the FLASHPRINT commands beginning with the @ character (at 103H). If you know how to use DDT or DEBUG you may change any of the characters in the table to something else as they are always referenced rather than absolute in the code. If you change @ to something else you may key two something-elses to really print something else.

SECTION 9: FLASHKEY

HOW TO CONFIGURE FLASHKEY

NOTE: This section relates only to the CP/M version of FLASHPRINT. It describes how you can create function keys for use with Wordstar. A separate FLASHKEY program is available for DOS which allows function keys to be created for use with almost any program. How to use it is described in a separate comprehensive FLASHKEY user guide.

In fact, the DOS version is much smarter because the IBM keyboard has many more keys than the standard CP/M keyboard and each key stroke is returned as a key number and can be made to do almost anything. But if you are running CP/M start reading.

You will not find a program called FLASHKEY on your disk. It is part of FLASHPRINT. FLASHKEY is loaded with FLASHPRINT and becomes part of Wordstar.

In this section we will tell you how to use FLASHKEY and how to customize it. We strongly suggest that you become familiar with FLASHPRINT commands before reading this section. We would like you to become familiar with one aspect of FLASHPRINT at a time. This is an enhancement, which though simple, requires some careful thought.

FLASHKEY functions are contained in your DEF printer tables(s) in the same manner as FLASHPRINT commands. However, FLASHKEY functions are performed when you hit a key while in Wordstar - even within Wordstar's main menu.

FLASHKEY functions are invoked by keying the ^ (circumflex or caret) character. The ^ does not appear on the screen. Instead, when the next character is found in your table the appropriate code is sent to Wordstar. If no entry is found nothing happens. To actually key ^ you have to hit it twice.

It's simple. Keying ^x (the caret followed by x where x can be any character) will send a string of characters (or control codes) contained in your DEF table to Wordstar. These may perform Wordstar functions as well as put text on the screen. Or a combination of both.

Functions can be extremely powerful. For example, you can use a function to print your name and address at the top of an envelope or any complex sequence of Wordstar commands can be built into a function.

FLASHKEY TRANSLATION

And here is another feature of FLASHKEY. Any key may be assigned to another character or string of characters. This lets you reconfigure your keyboard to a non-QWERTY layout. But not only can a single key send another character to Wordstar - it can send hundreds.

The translation facility also lets you assign unusual characters such as | + * and ~ to send a different character to Wordstar or perform a complex sequence of keystrokes. For example, you can use ~ to reformat a complete document by keying just ~ if you include the ~ character as an entry in your table.

You can also change Wordstar commands. Control-F normally moves the cursor right one word. But you can change it to reformat a document instead. Or save the cursor position, write the file to disk and continue editing.

FLASHKEY is powerful. Using the translation facility you can even configure a Dvorak keyboard for Wordstar.

CUSTOMIZING FLASHKEY

This is easy once you know how to create a FLASHPRINT table. A **FLASHKEY** entry begins with the letter K in your printer table if you want the following character to send something to your Wordstar SCREEN after you have hit the ^ character.

For instance, suppose you have designed a letterhead using printer graphics and saved it on disk as LETHED.DOC. To read it at the start of each letter using normal Wordstar you must key ^KR then key in the name of the file which is LETHED.DOC then hit the return key. A bunch of keystrokes. To do the same thing with FLASHKEY you hit the ^ character then the character you assigned in the table to do this.

FLASHKEY entries can be anywhere in your TBL files but each entry must start with the entry type K (for keyboard) in caps or lower case. Here is an entry which reads the letterhead. We have used the letter L to perform the function (but you may chose any letter you wish).

```
K 'L ^K'R <LETHED.DOC> ^M * ^L reads letterhead
```

Note that exactly the same format is used for FLASHKEY entries as for FLASHPRINT entries. You may use control codes for Wordstar preceded by a ^ character.

The other type of FLASHKEY entry is the translation. Each entry starts with the letter X (think of Xlate) and simply tells Wordstar that when you hit one key you meant something else. The "something else" can be another key or sequence of keys.

```
X '+ <plus> * Hitting + key will display "plus"
```

In this instance whenever you hit the + key you will see the word "plus" on the screen instead.

Obviously there are a lot of uses for this sort of thing. You could put the following entry into your TBL file:

```
X ^F ^Q'R^Q'Q^B * ^F reformats a document
```

In this entry when you press Control-F while running FLASHKEY (normally cursor word right) it will send the cursor to the start of file and format the entire document. You have lost the word right key but gained one which we think is more useful.

FLASHKEY grabs the keystrokes before Wordstar looks at them so you can change a Wordstar command to another command or a string.

Now we have to point out that the speed of FLASHKEY can vary with computers. Some are fast and some (such as Microbee) are a bit slow - It really depends on your BIOS (and if you don't know what that means you do not have to know).

Also, if your FLASHKEY entry produces a Wordstar prompt some versions of Wordstar require you to actually hit a key before anything will be read. If you find Wordstar hangs (such as after the ^KR prompt to read a file) you will have to hit a key. Hitting ESC is best because it does nothing but let WS know you have indeed hit a key. Wordstar will then continue to read the FLASHKEY entry. (This "feature" is only in Wordstar 3.3 - earlier versions read entries without the keystroke).

Finally, a suggestion: As FLASHPRINT allows you to load two DEF files you might like to keep FLASHPRINT and FLASHKEY files separate. Create one DEF file for your printer and another for your keyboard. But it is not necessary. You can include FLASHKEY commands anywhere in your FLASHPRINT table.

DOS users might like to know that the separate DOS version of FLASHKEY allows all function keys (F1 to F10) at four levels to be configured as well as all normal ASCII keys and ALT- ASCII keys. In total more than 200 function keys. The 40 function keys are very useful with Wordstar. For example F1, SHIFT-F1, CTRL-F1 and ALT-F1 can be set to perform similar but different functions. Write to us for details.

SECTION 10: FP AND MEMORY

HOW TO RELEASE FLASHPRINT

This section is for DOS users only.

FLASHPRINT for DOS is a memory resident program which simply means it remains in your computer's memory and can be used while you are running other programs. Other memory resident programs include FLASHKEY, printer spoolers, Sidekick and Turbo Lightning. DOS itself is also memory resident. So is the DOS PRINT command.

Although several memory resident programs allow you to deactivate them by pressing certain keys they still remain in memory consuming valuable space which may be required for other programs. Two programs supplied with FLASHPRINT, MARK and RELEASE, solve that. Not only for FLASHPRINT but for other memory resident programs as well.

MARK is also a memory resident program. You simply load MARK before (I repeat BEFORE) you load any other memory programs. We include MARK in our AUTOEXEC.BAT file.

In technical talk, MARK.COM saves the current interrupt vectors, some of which will be changed later by memory resident programs. FLASHPRINT for instance changes interrupt 17H which is printer I/O.

To remove subsequent memory resident programs run RELEASE from the DOS prompt or a batch file. RELEASE looks at the last MARK loaded, restores the vectors, and deletes the MARK and the memory resident programs which followed. You may load MARK several times. It occupies about 1.5K of memory each time it is loaded. (Three MARKS occupying less than 5K of memory are better than having to reboot).

RELEASE will delete everything back to the last MARK loaded. For instance let's load some resident programs (you could include these in a batch file):

```
MARK          <set marker 1
SPOOL LPT1://128 <a 128K print spooler
MARK          <set marker 2
LIGHT         <turbo lightning
MARK          <set marker 3
FP            <and flashprint with flash.def
```

Now let's assume you want to print without FLASHPRINT. Simply run RELEASE. To get rid of Turbo Lightning run RELEASE again. And to free up the memory occupied by the spooler run RELEASE a third time.

You could of course set only marker 1 then load the spooler, Turbo Lightning and FLASHPRINT without markers. A subsequent running of RELEASE will remove all three as well as marker 1. All you have to remember is RELEASE removes all programs loaded after the last MARK as well as MARK itself.

You might like to create a word processing batch file so that everything can be released when you exit back to DOS. We use the following RUNWS.BAT file:

```
REM RUNWS.BAT      <name of batch file
MARK               <set a marker in case not set already
SPOOL LPT1:/360   <load printer spooler
LIGHT              <load lightning
FP SUPER          <load flashprint with super.def
WS                 <run wordstar
RELEASE           <release back to mark after ws exit
MARK              <reset a marker
```

In this batch file the memory taken up by the spooler, Turbo Lightning and FLASHPRINT (about 480K total) is released automatically when Wordstar returns to DOS. We can now run any program without resetting our computer to release memory occupied by a memory resident program.

When you run MARK and RELEASE they will tell you they are working. If RELEASE cannot find a previous MARK it displays a message. Don't forget that RELEASE also deletes the previous MARK and it should be loaded again in case it's required.

Most users will find both programs invaluable for all applications which use memory resident programs.



SECTION 11: FLASHGEN

CREATING YOUR OWN CHARACTERS

Before we begin note there are two versions of FLASHGEN. Both were written in BASIC and compiled. The DOS version is specific to DOS facilities while the CP/M version is less powerful but more all-purpose and should run on most machines. This section describes both.

Both are compiled BASIC programs and will run directly from the operating system without BASIC.

FLASHPRINT allows Wordstar to print characters which you have designed simply by keying something else, even a single character. This makes FLASHPRINT great value for those who need to print their own scientific symbols, for example, or characters in another language from a Wordstar document.

Printing characters which you have designed is as easy as keying @X where X is the FLASHPRINT command you have assigned to the character - Or you may assign your character to the translation table and simply key X. Then, whenever X (any character you like) is found Wordstar will not print X but instead will print YOUR character.

We found designing characters and working out the coding required by the printer to be a frustrating, time-consuming and boring. We began with a piece of graph paper, filled in the dots, then laboriously added up the dots to calculate the magic number required for each row of dots to form the bit image character on our printers.

We entered this into our .DEF file. Used CONVERT. Booted FLASH then printed the character. Often the character did not print as we had imagined and so we started again. A lot of testing was needed before we achieved the correct coding.

So, to make the job much easier, we wrote FLASHGEN and made our computer work out all those magic numbers. We cut the development time for a single character down from 30 minutes to about five minutes.

FLASHGEN allows you to draw the character on the screen and dump the required code to your printer. This code can then be entered into your printer source file.

In this section we assume you are familiar with FLASHPRINT and how to customize it.

BIT IMAGES AND YOUR PRINTER

We suppose most FLASHPRINT users have had little or no experience using the printer's bit image mode. After all, until now you most likely could only send the required code to the printer via a BASIC program, not Wordstar or word processor on DOS machines. So here is a little background on what your printer may require to print bit image characters.

Bit image mode is a sequence of coding which tells the printer to not print characters but fire certain pins on the printing head according to the code sent. Your normal characters are made of dots. In bit image mode you tell your printer: "Don't print characters -- print the following dots."

We have tested FLASHGEN on two printers. A CItch and Epson clone. Your printer is most likely compatible with one of these.

Your manual will describe the coding required to begin printing a bit image character. The required code may be ESC+S (our CItch) or ESC+K (our Epson).

By sending these two bytes you have told the printer you want to print a bit image character. You must now tell it how many columns of dots form the character you are about to send. In other words, the width of the character. (All characters will be eight dots from top to bottom).

Our CItch requires four ASCII digits. Note these are ASCII keyboard characters 1 2 3 4 and so on. You MUST have four digits so leading zeros are required. For example the table entry for a character 24 dots wide is '0'0'2'4 -- note the leading zeros and note each number is a character and is therefore preceded by a single quote.

In a FLASHPRINT table you key the string -- for example <0024> or else '0'0'2'4 which is the same thing.

Our Epson is different. It requires two values (not characters). To get these divide the number of columns of dots by 256. The first value is the remainder. The second value is the whole number ignoring the remainder. For example a character 300 dots wide yields the following two values 44 (the remainder) and 1 (the whole number). You enter this into your table as simply 44 1. Normally your character will be less than 256 dots wide and in this case the first value will be simply the width the second will be zero.

Here are two entries for an 8-row wide character in the simple command table using the X character to print the bit-image character.

```
C 'X 27 's '0'0'0'8 n n n n n n n n * 8-dot wide CItch char
C 'X 27 'K 8 0 n n n n n n n n * 8-dot wide Epson char
```

Where n are values the printer requires for each column of dots. Well, how do you determine the values for n? You use FLASHGEN.

The following two parts describe FLASHGEN for CP/M and FLASHGEN for DOS. Read the part appropriate to your version.

FLASHGEN FOR CP/M

This version is slightly more restrictive than DOS due to the different flavors of CP/M machines and some restrictions in Microsoft BASIC. Also, many CP/M machines operate on terminals which will not display graphics. However, many users have successfully designed quite complex alphabets using it.

To run FLASHGEN simply type FLASHGEN at the system prompt.

After loading, FLASHGEN asks if you want brief instructions then asks you for your type of printer. Then it asks for the width of the character. All characters are eight dots high and the maximum width is 64 dots. Enter the width and press return. If no width is specified (you just press RETURN) the width will be eight dots.

A grid of dots is then displayed. You may then move the cursor around the grid using the normal cursor keys. (WARNING!! The Wordstar cursor keys do not work).

Initially the grid is blank, indicated by full points. You may create a printing dot in any position on the grid by keying an X or x and erase a printing dots by keying a dot or space.

You may abandon the character at any time by pressing A (abandon) - When the character has been completed you may dump it to the printer by pressing D (dump).

The screen pattern of dots and Xs will be printed. Then the actual character you designed will be printed. Finally the printer's bit image escape sequence will be printed followed by the values needed to print the character.

Look at the character. If it's not exactly what you had in mind press C (for continue) and you may keep moving the cursor around the grid and make changes. Else press any other key to quit this character and begin designing a new one.

Remember, FLASHGEN prints the following: (1) An image of the dots and crosses that were on the screen when you dumped it. (2) The character itself (for proofing purposes. If you don't like it press C to continue and make changes). (3) All of the coding required to print the character on the printer you specified (Epson or CItch).

If the CHARACTER does not print properly with FLASHGEN in either Epson or CItch mode, well, your printer requires some unique coding. So far printers we have tested work in one of these two modes but there is probably some unique ones out there.

A word about a "feature" of BASIC. If the dot image is CHR\$(9) then BASIC may send spaces to the printer because it assumes you are sending a TAB. This is a "feature" of CP/M BASIC and you may ignore the

funny output - FLASHPRINT will get it right if you use the numbers printed.

Examination of the code will show the difference between CitoH and Epson. Besides different bit image mode commands the printers "count" the dot values in different directions. On the C.itoH the top dot in each row is the least significant bit. On Epson the bottom dot is the LSB.

Finally, as FLASHGEN was written in Microsoft BASIC we have supplied the source code as FLASHGEN.BAS so wizards out there can make changes for their specific implementation.

FLASHGEN FOR DOS USERS

This part relates only to the DOS version of FLASHGEN. FLASHGEN is a QUICKBASIC compiled program which allows you to:

- (1) Create bit image characters on your screen.
- (2) Print the code required by your printer.
- (3) Save the code directly to a table file on disk.

To run FLASHGEN simply key FLASHGEN at the system prompt. It then gives you some prompts which requires some answers.

FLASHGEN needs to know what sort of printer you have. If your printer is Epson compatible (and most printers are these days) press E for Epson. However, if you have a CitoH or perhaps Toshiba, press C for CitoH. If you are not sure try Epson and if that doesn't work reboot FLASHGEN and try CitoH. Our IBM printer is CitoH compatible.

If you chose the Epson option FLASHGEN will ask you for the character following ESC to print bit image mode. This is normally the letter K. Your printer manual should explain this in detail.

ESC+K (which is the default if you simply hit RETURN in response to the question) prints bit image characters in standard character density of 60 dots per inch. However, other characters following ESC allow different densities. Here are some options:

- ESC+L < Double density graphics (120 dots per inch)
- ESC+Y < Double speed double density (120 dots per inch)
- ESC+Z < Quadruple density (240 dots per inch)

Refer to your printer manual. If you wish to change graphics density from normal (K) then the appropriate letter should be keyed at the prompt. If you don't care simply hit RETURN and the default K will be used by FLASHGEN.

FLASHGEN will then ask you for the character width. If you hit RETURN you will get the default width which is 8 dots. You may enter any value up to a maximum of 80. If you enter more than 80 FLASHGEN will default to 80 (the maximum width of the screen).

Note that if you change the horizontal dot density of the characters for Epson you might also wish to increase the width.

Once you have answered the prompts FLASHGEN will display a grid of dots eight dots deep and the selected width. Initially the character is blank and dot positions are shown by the full point character.

To set a dot position to print hit the letter X (caps or lower case) - The display will change from a dot to a solid shape and the cursor (in this case the + character) will move to the next position on the line.

You may move the cursor around the grid by pressing the appropriate cursor arrow keys. The RETURN key will move the + cursor to the first position on the next line. The HOME key will move the cursor to the top left corner. The cursor will not move outside the grid.

If you have keyed X to print a dot in the selected position you may change it to a blank by positioning the cursor on the solid shape and hitting the space bar or full point. The solid will change to a dot (indicating a blank) and the cursor will move to the next position on the line.

If you wish to erase the whole grid back to blanks press the letter A (caps or lower case) to abandon this character. The grid will blank and the + cursor will appear at the top left for you to begin again.

Once you have filled in the dots you will want to see what the character looks like on your printer. Press the letter P (caps or lower case) to print. The cursor position is not important for this to work.

The image of the screen will be sent to the printer, followed by the character, then the code required to print the character.

Once printed you have the following options:

- C = Continue with this character.
- A = Abandon this character.
- S = Save the FLASHPRINT character code on disk.

If you choose to save the character you will be asked for a file name for the character. You may specify a new file or an existing file. If the file already exists the code for this character will be ADDED to the end of the file. This is useful for creating a sequence of characters and building a character set.

You may enter up to eight characters for the file name. The extension as .TBL is added automatically and specifying an extension will cause an invalid file name message.

Once you have specified a file name you will be asked for the type of entry. This will normally be C T B and so on. In other words the first character to appear in a FLASHPRINT entry. You may key any letter(s) here as there is no check for validity and you may change the disk file later using your word processor before running CONVERT.

You will then be asked for the character to activate the entry. This is normally the second parameter in your FLASHPRINT entry. Enter it

exactly the same as you would key it in a table entry. Precede characters with a single quote, control codes with the ^ character or enter a decimal or hex number.

After you have specified a file name, the entry type and the character to activate the entry, FLASHGEN will write the printer code to disk.

If you have entered the parameters correctly FLASHGEN will make a file which you can CONVERT immediately. (But you might like to check it out with your word processor first). Or you can use your word processor to make changes to the entry types or characters which activate the entry. (Don't change the numbers sent to the printer unless you really know how bit image characters are printed).

Once the FLASHGEN character is saved on disk you have the option of changing the character (but not on disk), abandoning the character and starting a new character, or exiting to the system.

In fact you can exit from FLASHGEN whenever keyboard input is required by pressing Control-C.

A table file created by FLASHGEN can be added to your regular printer table file using your word processor or DOS.

SUMMARY

FLASHPRINT is unique. It allows you to print anything from a Wordstar file including characters which you have designed. FLASHGEN makes designing the characters easier.

APPENDIX A: HEX-DECIMAL

HEX TO DECIMAL CONVERSION CHART

DECIMAL TO HEX CONVERSION: Find the decimal number in the table. The character in the LEFT column is the FIRST hex digit, the character in the top row is the second hex digit.

HEX TO DECIMAL CONVERSION: Find FIRST hex digit in LEFT column and read across top row to the column containing the second hex digit. The co-ordinate contains the decimal number.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

APPENDIX B: ASCII CHARACTERS

ASCII VALUES

HOW TO USE THESE TABLES: To find the HEX value for a character locate it in the table. The value in the first column is the first hex digit and the value on the top row in the same column is the second digit - To find the DECIMAL number add the value in the second column on the same row to the value in the second row in the same column.

TABLE-1 ASCII CONTROL CODE BY NAME

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DEC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1 16	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US

TABLE-2 ASCII CONTROL CHARACTERS

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DEC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 0	^@	^A	^B	^C	^D	^E	^F	^G	^H	^I	^J	^K	^L	^M	^N	^O
1 16	^P	^Q	^R	^S	^T	^U	^V	^W	^X	^Y	^Z	^[^\	^]	^^	^_

TABLE-3 ASCII PRINTING CHARACTERS

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DEC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2 32	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3 48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4 64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5 80	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6 96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7 112	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

TABLE-1 shows the ASCII names for control codes. In most cases you need not know the names. It is here for completeness. TABLE-2 shows control codes which cannot be keyed. They must be entered in your FLASHPRINT table preceded by the caret as shown here or as a decimal or hex number. To enter them directly into a document you must use the hex value. In TABLE-3 SP is a space. DEL is delete and does not print.

APPENDIX C: YOUR PRINTER

A GUIDE TO THE GOBBLEDEGOOK

Here we will try to help you make sense of your printer manual. We have more than 20 printer manuals ranging from understandable to incomprehensible. Before we begin please note that this is not a complete guide to printers. However, we hope this brief guide helps.

Let's begin. When you print your computer sends characters to your printer. But as you probably know, computers only know about numbers - So each character is assigned a number. The number is stored in the computer and translated to a character pattern when it is displayed on the screen. The same thing happens on a dot matrix printer. The computer sends the printer a number and the printer translates this into a pattern of dots.

Fortunately, there is a standard relationship between the numbers. A number means a character. The relationship is known as ASCII pronounced "askey." Characters are assigned numbers between 32 and 127.

ASCII characters are between 32 and 127 (32 decimal is a space, 65 decimal is the capital letter A etc) but computers store their numbers in bytes and can store any number between 0 (zero) and 255. So what about the other numbers which are not characters and which cannot be keyed? Some of these are used by your printer to switch on modes or print graphic characters and so on.

If these numbers are not characters how do you send them to the printer? From your FLASHPRINT table or by keying them as hex values into your document.

The hard bit is to discover what your printer requires. You will most likely find this information in a section of your printer manual headed with something like "Printer Control Codes." Here is a typical entry from a C.Itoh manual for underlining.

Command	Decimal Data	Hex. Decimal Data	Function
ESC X	27 88	1B 58	Start Underline
ESC Y	27 89	1B 59	Stop Underline

Example: Printing underlined characters.

```
DATA: 1 2 3 [ESC] X 4 5 6 [ESC] Y 7 8 9
PRINT: 123456789
```

The C.Itoh has two commands which turn underline on/off. Column one lists the commands in ASCII. ESC means ESCAPE and X is the ASCII character which turns underlining on. So if we send ESC followed by the letter X to the printer it will switch into underline mode. If we send the printer ESC then the letter Y it will stop underlining. Now ESCAPE is not a keyable character. You may have an ESCAPE key on

your keyboard but it displays nothing on the screen. To send an ESCAPE to your printer you must include it in your FLASHPRINT definition file or key # in your document.

When FLASHPRINT finds a # character in a file being printed it sends an ESC to the printer so to turn on underline we can simply key #X in our document. And we can key #Y to turn off underlining. Note there must be no space between the two characters.

But what if you want to put this into a table? The second column shows the ASCII values and these can be used. ESC has a value of 27 and a capital X has a value of 88. So the code required is simply 27 88.

The third column in our C.Itch example gives the same values in hexadecimal. You may enter @<1B 58> in text or 1B 58H in your table.

Two character instructions are common in printers. The first character is usually ESC (27) followed by either an ASCII character or the value of the character. If the value is not a keyable character (anything with a decimal ASCII value of less than 32 or greater than 127) you must enter it as a hex number.

VARIABLE VALUES

Some printer commands require values which are variable. For instance setting the left margin. The value tells the printer where the margin is. There are two types of variables - ASCII numbers (0 to 9) and absolute values which you cannot key. Here's a C.Itch example:

Command	Decimal Format	Hex. Decimal Format
ESC L N ₂ N ₁ N ₀	27 76 n ₂ n ₁ n ₀	1B 4C (n ₂) (n ₁) (n ₀)

N₂ - N₀ are Decimal numbers 0 - 9.

n₂ - n₀ are Decimal equivalent ASCII numbers (48 - 57).

(N₂) - (N₀) are Hexadecimal (ASCII) numbers (30 - 39).

Although the manual does not make it clear, this simply means that to set a left margin the printer requires ESC followed by the letter L followed by three ASCII characters (not values) between 0 and 9 equal to the position of the left margin. Note three digits are required and so one or two leading zeros is needed.

To set the left margin to 12 we can simple key #L012 in our document To clear the margin (set it back to column one) we can key #L001. Or we can of course put the entry into a table. However, note that what follows #L are ASCII characters, not values. The table code to set the margin to 12 would be 27 <L012>. Do not use 27 'L 0 1 2. This sends values and will not work.

C.Itch and look-alike printers usually use ASCII characters for

values. However, Epson type printers use values. Here is a typical Epson code to set the left margin:

LEFT MARGIN:

Sets position of left margin.

Name: ESC+I+n

Code: 27,108,n DEC 1B,6C,n HEX

Input Format: LPRINT CHR\$(27)+"I"+CHR\$(n);

Epson printers need ESC followed by the letter I followed by the left margin value. A left margin of 12 requires a value which is not a keyable ASCII character. So our entry in the FLASHPRINT table is 27 'I 12. To actually key this into text we would have to enter 12 as a hex value (see decimal/hex conversion table in Appendix A). The result is #I@<0C>.

GRAPHICS

There are several different ways printers access built in graphic characters. One common method is to turn on graphics using an ESCAPE sequence. The printer then does simple translation. This character = that character. If you have an IBM compatible printer it usually will require a character number greater than 127.

You will find in your printer a manual a table of characters and the value required to print them. As values greater than 127 have no ASCII character equivalent you must put the value in a table or key them directly into text as hex values. Good printer graphic tables show both the decimal and hex value of the characters else you may use the decimal/hex conversion table.

CONTROL CODES

The values from 0-31 are called control codes. The value is named after the capital ASCII minus 64. For example Control-B has an ASCII value of 66 decimal. Subtract 64 from it and you have a value of 2 - Control-B has an ASCII value of 2. This is the value programs such as Wordstar read when you press the control key and the letter B.

Some control values are used for certain functions. For instance, Control-M is the same as RETURN or ENTER. Control-I is the same as pressing the TAB key on most computers. You may key control codes as a FLASHPRINT entry in your table by either preceding the character with a ^ circumflex or by keying the actual value. For instance, ^M is the same as 13 or 0DH.

Finally, if you find your printer manual inadequate, and some are hopeless, complain to the person who sold you the printer.

APPENDIX D: DVORAK

A DIFFERENT KEYBOARD

The computer keyboard layout you are using is called QWERTY. It is named after the six letters at the top left.

It is very inefficient. For instance, when you are keying English text your left hand does most of the work. And there is only one vowel on the "home row" where most typists rest their fingers.

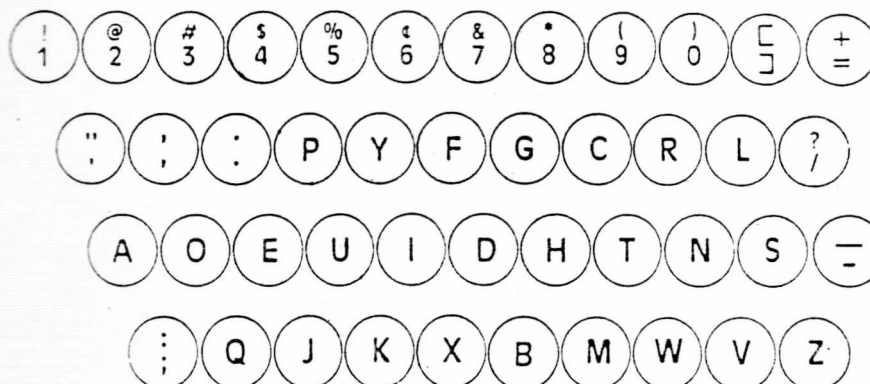
QWERTY was designed in 1872 so that the most used characters are as far apart as possible. This prevents typewriter keys from jamming as characters strike the platten. Not a computer problem.

In 1932, after testing 250 keyboards, August Dvorak, of Seattle USA, showed a more efficient keyboard. On a Dvorak keyboard 70 per cent of the keys hit are on the home row, 22 p.c. on the row above, and only 8 p.c. on the row below. All vowels are on the home row and the right hand works harder. Dvorak's layout is now known as the Dvorak Simplified Keyboard.

During World War II Dvorak trained a group of 44 women to use his keyboard. After a month they were turning out 74 per cent more work and were 68 per cent more accurate. Using Dvorak's keyboard typing speeds of more than 100 words per minute are common.

FLASHKEY (for CP/M users only) allows you to configure a Dvorak Simplified Keyboard on your computer using Wordstar. DVORAK.TBL is the FLASHKEY table which has been converted to DVORAK.DEF.

Here is Dvorak's layout. However, note that Dvorak designed his keyboard for typewriters and the position of some keys will be different on your computer.



DVORAK SIMPLIFIED
KEYBOARD