

```

global=<ESC>G
help=<ESC>H
hyphen=<ESC>Y
index=<ESC>I
search=^S
spelling=<ESC>S
un-paginate=^UT
#
# General user key control
#
keycap=<ESC>UX
keygo=<ESC>X
#
# Print codes
#
acute=<ESC>'
cedilla=<ESC>,
circmflx=<ESC>/
diacesis=<ESC>"
grave=<ESC><CTRL'>
tilde=<ESC><CTRL~>
uprtcode=<ESC>UP
#
# Misc
#
endpage=<ESC>?
hardspce=<ESC><SPACE>
insert=<F1>
mult-del=<ESC><F2>
mult-ins=<ESC><F1>
marker=^K
quit=<ESC>Q
read=<ESC>R
recall=^R
redraw=^Y
scr-prt=<ESC>.
sof-cncl=^Q
tab=<TAB>
spectab=<ESC><TAB>
un-delete=^U<F2>
write=<ESC>w
status=<ESC>E
un-status=^UE

```

Notes:

- o ^ and \ at the end of the string do not need to have \ in front of them -- SCRIPSIT 16 "knows" that there is no character following them to be control or special.
- o Some commands are reserved for future versions of SCRIPSIT 16.

The easiest way to make a new config file is to copy one that already exists and modify the copy. After you do this, you must let SCRIPSIT 16 know that you have created a new file. For example, if you have created a new config file for a DEC VT100 (calling the file `vt100`), you would make the file the way you want it, and, after exiting SCRIPSIT 16, type:

```
scripsit -i vt100
```

SCRIPSIT 16 will examine the file and look for errors in it, and tell you if it finds any. If there are any errors, correct them and run `scripsit -i` again, until it gives no error messages. Note that you only need to do this when adding a config file that didn't exist in that directory before. If you just change a file, SCRIPSIT 16 will detect that when you run it. But it doesn't hurt to run `scripsit -i` on the file anyway, to check for errors.

SCRIPSIT 16 is supplied with printer capability ("printcap") files for Radio Shack printers. Additional printers can be used with SCRIPSIT 16 if you create printcap files for them.

Printcap files can be in any of three places. If you have set an environment variable SCRIPSIT (see chapter 1), the directory named there is searched first. If no printer file is found there, SCRIPSIT 16 looks in a subdirectory of your home directory. If not there, the last place searched is /usr/lib/scripsit/printers.

Each printer is represented by a subdirectory. (For simplicity, we will only deal with printers in /usr/lib/scripsit/printers, but the structure is the same for the other locations.) In this directory is a file called description and a file for each font that the printer can handle.

The description file consists of lines that look like this:

```
name=value
```

interspersed with comment lines. Comment lines have # in the first position and SCRIPSIT 16 completely ignores the rest of the line.

The valid names are defined below. The value, unless otherwise specified, is a string of characters that will be sent to the printer to accomplish the desired action. This can be specified in any of several ways:

- o Any ASCII character represents itself (for example, A).
- o Any character with ^ in front of it is a control character (for example, ^A represents control-A). This has meaning only for the letters of the alphabet. Note that ^a and ^A mean the same thing.
- o A backslash (\) in front of a character gives it a special meaning:

```
\t      tab (same as ^I)
\r      carriage return (same as ^M)
\f      form feed (same as ^L)
\b      backspace (same as ^H)
\E      escape (note: the E must be uppercase!)
\^      ^
\\      \ itself
\nnn    nnn represents an octal value. Any character can
        be specified this way. For example, \101 is the
        same as the character A. A space character must
        be represented as \040.
```

Following are two printer description files. The first is the file for the Daisy Wheel Printer II (dwii):

```
#
# Main entry for the DW-II
```

```
move=<ESC>M
paginate=<ESC>T
pause=^UP
print=<ESC>P
shift=<ESC>]
un-center=^UN
un-hyphen=^UY
un-justify=^UJ
un-freeze=^UZ
un-lock=^U]
#
# Attributes
#
boldfce=<ESC>+
comment=<ESC>[
dbl-unl=<ESC>=
fontchg=<ESC>[
strkthru=<ESC>-
subscr=<ESC>\>
super=<ESC>\<
underln=<ESC>_
un-boldfce=^U+
un-comment=^U[
un-dbl-unl=^U=
un-strkthru=^U-
un-subscr=^U\>
un-super=^U\<
un-underln=^U_
#
# Menu Control
#
execute=<ESC><ESC>
#
# Control
#
abort=^UQ
cancel=<BREAK>
#
# Mode settings
#
video=^V
ins-mod=^N
overstrk=^U^N
un-video=^U^V
#
# Utilities
#
utilities=^X
convert=<ESC>V
```

```

11-keygo=<ESC>!
12-keygo=<ESC>@
13-keygo=<ESC>#
14-keygo=<ESC>$
15-keygo=<ESC>%
16-keygo=<ESC>^
17-keygo=<ESC>&
18-keygo=<ESC>*
19-keygo=<ESC>(
#
# user key control
#
keyexit=^Z
#
# Motion Keys
#
backspce=<BKSP>
enter=<ENTER>
home=^O
#
down=<DOWN>
right=<RIGHT>
left=<LEFT>
up=<UP>
#
# Text Quantities
#
hdr-ftr=<ESC>K
column=^F
document=^T
line=^L
page=^G
paragrph=^P
sentence=^E
word=^W
#
# Define block commands
#
define=^D
center=<ESC>N
copycat=<ESC>;
copy=<ESC>O
delete=<F2>
format=<ESC>F
freeze=<ESC>Z
justify=<ESC>J
spacing=<ESC>L
lock=<ESC>}
movecat=<ESC>:

```

```

#
# Underline is done the hard way
UT=Manual
# Same with double underline
DT=Manual
# Boldface is manual
BT=Manual
# Strike through is manual
ST=Manual
# Increment spacing
I1=^E\001
I2=^E\002
I3=^E\003
I4=^E\004
I5=^E\005
I6=^E\006
# Superscript (Half up)
HU=^E\036
# Subscript (Half Down)
HD=^E\034
# Backspace kind
BK=Fixed
# Distance is 5
BW=5
#
# Double underline
#
DC=^337
DW=J
# Acute
AC=^247
AW=+
# Diaeresis
YC=^276
YW=+
# Default Font
DF=standard.10
# Offset from left margin
OF=5
# We can only have 6 lines/inch
L6=
# Use dwii driver
DR=bp.dwii

```

The second is for the DMP-2100 dot matrix printer (dmp2100):

```

#
# Main entry for the DMP2100
#

```

```

# Initialization String = Graphics mode off, Select DP mode, elongate off
IS=\036\023\E\017
# Underline is automatically done by the printer
UT=Automatic
# Underline start/stop codes
US=\017
UE=\016
# No double underlining
DT=None
# Boldface is automatic
BT=Automatic
# Bold start/end codes
BS=\E\037
BE=\E\040
# Strike through is manual
ST=Manual
# Increment spacing
I1=\E\001
I2=\E\002
I3=\E\003
I4=\E\004
I5=\E\005
I6=\E\006
# Superscript (Half up)
HU=\024\E\036\023
# Subscript (Half Down)
HD=\024\E\034\023
# Backspace is variable
BK=Variable
# Acute description
AC=\240
AW=+
# Diaeresis description
YC=\276
YW=+
# Form Feed
FF=^L
# Form Feed Set page length
FS=\E4
# Default Font
DF=standard.10
# Select lines per inch
L6=\E6
L8=\E8

```

- o In general, any keystroke given as part of a multiple-key sequence is case insensitive. (It doesn't matter if <ESC> is followed by M or m in the MOVE command, for example.) This applies only to keystrokes specified by a letter (example: move=<ESC>M). If you want the value to be case sensitive (exact match, including SHIFT key status), you must specify the keystroke in the \nnn octal format.

Following is the command section for the Model 16 Console. It is grouped by command type. Note that the actual order in which the commands are defined does not matter.

```

#
#
Key
#
# user keys
#
0-keycap=<ESC>UD
1-keycap=<ESC>U1
2-keycap=<ESC>U2
3-keycap=<ESC>U3
4-keycap=<ESC>U4
5-keycap=<ESC>U5
6-keycap=<ESC>U6
7-keycap=<ESC>U7
8-keycap=<ESC>U8
9-keycap=<ESC>U9
10-keycap=<ESC>U)
11-keycap=<ESC>U!
12-keycap=<ESC>U@
13-keycap=<ESC>U#
14-keycap=<ESC>U$
15-keycap=<ESC>U%
16-keycap=<ESC>U^
17-keycap=<ESC>U&
18-keycap=<ESC>U*
19-keycap=<ESC>U(
#
0-keygo=<ESC>0
1-keygo=<ESC>1
2-keygo=<ESC>2
3-keygo=<ESC>3
4-keygo=<ESC>4
5-keygo=<ESC>5
6-keygo=<ESC>6
7-keygo=<ESC>7
8-keygo=<ESC>8
9-keygo=<ESC>9
10-keygo=<ESC>)

```

```

End Document=1/\ERG}\ERg
#
Print Code=1/\ERGs\ERg
Tilde=1/\ERGV\ERg
Grave=1/\ERGb\ERg
Circumflex=1/\ERG~\ERg
Cedilla=1/\ERGx\ERg
Diacesis=1/\ERGu\ERg
Acute=1/\ERGa\ERg

```

Note that there are more lines in this section than in the normal video section. This is due to the fact that certain things don't display at all in normal video mode, such as the end-of-line character or boldface indicators.

The final section is the command section. It begins with the word Key on a line by itself (because it describes the keyboard actions for a particular command). Each of the possible SCRIPSIT 16 commands is then listed with the sequence of keystrokes necessary to execute that command. There are certain restrictions on this section that must be met, or the program won't function properly:

- o The cancel command must be defined, and it must be a single keystroke. In the standard configuration files, it is defined as BREAK or control-C. It can only be defined once.
- o All commands must begin with a non-printing character (generally either a control key or ESCAPE). The printable characters always represent themselves, and cannot be changed in the config file.
- o Two commands cannot have the same key sequence. It is possible, however, for the same command to be defined more than once, with different key sequences. (The cancel command, however, must be defined only once.) Example:

```

execute=<ESC><ESC>
execute=<ESC><ENTER>

```

would allow the execute command to be entered with either of the two sequences given.

- o One command's key sequence cannot be the same as the first part of another command's key sequence. For example, the following two command lines would cause an error:

```

execute=<ESC>
left=<ESC>A

```

since when ESC was pressed, SCRIPSIT 16 would not know whether it was the execute command or if it should wait for the rest of the left command.

The following gives the explanation of each string:

Initialization and Termination

- IS Initialization string. This will be sent to the printer at the beginning of the document.
- TS Termination string. This will be sent at the end of the document.
- BP Begin page. This string will be sent at the beginning of each page.
- EP End page. This string will be sent at the end of each page.

Attribute Handling

- UT Underline type. This can be Automatic, Manual, or None. If UT=Automatic is specified, then SCRIPSIT 16 will underline using the US and UE strings defined below. If UT=Manual is defined, SCRIPSIT 16 will underline by printing the underline character and backing up to print the character over it. If UT=None is specified, SCRIPSIT 16 will not try to underline on this printer.
- US Underline start. This string is sent to start underlining if UT=Automatic.
- UE Underline end. Used to end underlining if Automatic is specified.
- DT Double underline type. Same as UT but for double underlining.
- DS Double underline start. Used to begin automatic double underlining.
- DE Double underline end. Used to end automatic double underlining.
- DC Double underline character. Since double-underline is not a standard ASCII character, different printers will use different values for this character. This defines the character or sequence of characters that will be used.
- DW Double underline width. This is a character that is the same width as the double-underline character. SCRIPSIT 16 needs this value to know exactly how much space on the line is taken up by double underlining.
- BT Bold type. If BT=Automatic, then the codes BS and BE will be used to boldface characters. If BT=Manual, then SCRIPSIT 16 will print the character over itself five times. If BT=None, then no boldfacing will be done.
- BS Bold start. This is the string used to start automatic boldface.
- BE Bold end. This is the string used to end automatic boldface.

ST Strike through type. As with the others, this can be **Automatic**, **Manual**, or **None**. If **Manual**, SCRIPSIT 16 will strike through by typing a dash character, backing up, and printing the character over it.

SS Strike through start. This is the string to use for automatic strike through.

SE Strike through end. This is the string to end automatic strike through.

Diacritic Handling

YC Diaresis character. This is the string used to print the diaresis character (¨). If this string is not defined, the diaresis cannot be printed on this printer.

YW Diaresis width. This is a character that is the same width as the diaresis character in a proportional character set.

AC Acute character. This character is used to print the acute character (´). If it is not defined, the acute cannot be printed on this printer.

AW Acute width. This is a character that is the same width as the acute character in a proportional character set.

Horizontal Spacing

BK Backspace kind. This can be **Fixed**, **Variable**, or **None**.

If **BK=Fixed**, then a backspace is a fixed character size. If the font is a fixed size, then the backspace is assumed to be the same size as the other characters. If the font is proportional, the size of the backspace is defined by the value of **BW** (below).

If **BK=Variable**, SCRIPSIT 16 will print the backspace code and follow it with the distance to backspace.

If **BK=None**, SCRIPSIT 16 will not try to backspace on this printer.

Note: The actual character used to backspace is always 08 hexadecimal.

BW Backspace width. If **BK=Fixed**, this is the width (in decimal) of a backspace in a proportional font. (The Daisy Wheel II, for example, always backs up five units in proportional spacing when a backspace is received.)

I1 Increment 1. String to send for one incremental space. An incremental space is a unit in the current font (for example, this is 1/60th of an inch on the Daisy Wheel II). For printers that can't do incremental spacing, this string should send out a space, and **I2-I6** should be left undefined.

I2 Increment 2. String to send for two incremental spaces.

(**Note:** **\ERG** is the command sequence to put the Model 16 Console into graphics mode, and **\ERg** is the sequence to put it back into normal mode.)

The names in front of the equals sign must be listed exactly as given, or an error will result when SCRIPSIT 16 tries to use the file.

The following codes **MUST** be defined in normal video mode, or SCRIPSIT 16 will not function properly. (They can appear as a blank, as in the example above.):

Insert Dot
Hard Space
Tab
Special Tab
Return

The next section is the **Full Video** section. It begins with the word **Full** on a line by itself. This gives the same codes as the normal video section (with additions), but this time it tells how they are to be displayed in full video mode. Following is the full video section for the Model 16 Console:

```
#
# Define the full video codes
Full
#
Insert Dot=1/\ERGo\ERg
Hard Space=1/\ERGP\ERg
# Line endings
End Line=1/\ERG`\ERg
Return=1/\ERGW\ERg
# Page Stuff
Header=1/\ERGe\ERg
Footer=1/\ERGC\ERg
Forced Page=1/\ERgk\ERg
Soft Page=1/\ERgN\ERg
# Tabs
Tab=1/\
Special Tab=1/\`
# Start-Stop and attributes
Start=1/\ERgj\ERg
Stop=1/\ERgh\ERg
Underline=1/_
Double Underline=1/=
Strike Through=1/-
Bold=1/+
Superscript=1/\<
Subscript=1/\>
Comment=1/\ERg_\ERg
# Document codes
Pause Printer=1/\ERg!\ERg
```

```

CTRL6=~
CTRL'='
ESC=\E
TAB=^I
BKSP=^H
BREAK=^C
F1=^A
F2=^B
ENTER=^M
LEFT=<ESC>D
RIGHT=<ESC>C
LP=<ESC>A
DOWN=<ESC>B
SPACE=\040

```

The values to the right of the equal sign are explained above. Notice that the names to the left of the equal sign are all in uppercase. This helps distinguish keynames from actual values in the following sections but is not a requirement.

The second section is the normal video section. It is started by a line containing the word Normal by itself. Following this line is a list of lines that give the name of each internal control code followed by how it is to display in normal video mode. If a code is not in this list, it will not be displayed at all in normal video. Following is the normal video section for the Model 16 Console:

```

#
# Define the normal video codes
#
Normal
#
Insert Dot=1/\40
Hard Space=1/\40
# Tabs
Tab=1/\40
Special Tab=1/\40
Return=1/\40
# Page Stuff
Header=1/\ERGe\ERg
Footer=1/\ERGc\ERg
Forced Page=1/\ERGk\ERg
#
Print Code=1/\ERGs\ERg
Tilde=1/\ERGu\ERg
Grave=1/\ERGb\ERg
Circumflex=1/\ERG~\ERg
Cedilla=1/\ERGx\ERg
Diacesis=1/\ERGu\ERg
Acute=1/\ERGa\ERg

```

- I3 Increment 3. String for three incremental spaces.
- I4 Increment 4. String for four incremental spaces.
- I5 Increment 5. String for five incremental spaces.
- I6 Increment 6. String for six incremental spaces.

Vertical Spacing

- HU Half up. This is the string used to space the printer up one half line, for superscripts.
- HD Half down. This is the string used to space the printer down one half line, for subscripts.
- FF Form feed. This string is used to send the printer to the top of the next page. If this is not defined, then SCRIPSIT 16 will move the printer to the top of the page with line feeds.
- FS Form set. This is the string (if any) required to set the form length for automatic form feed.
- L6 Line 6. This string is used to set six lines per inch printing. This is the normal "typewriter" vertical spacing, and most printers don't need to have anything set for this value (see the dwii example, above), but it should be defined even if it has no value.
- L8 Line 8. This string is used to set eight lines per inch printing. This allows you to print more per page, but the lines will be closer together.

Miscellaneous

- DR Driver. This is the name of the printer driver that SCRIPSIT 16 will use with this printer. There are three supplied with the SCRIPSIT 16: bp.dwii, for the Daisy Wheel II; bp.dwp410, for the Daisy Wheel Printer 410; and bp.standard, for all other printers. (bp.standard is used if this string is not defined.) You can also supply your own driver; see chapter 4.
- DF Default font. This should be set to the font your printer comes up in when you turn it on. It is used if you are printing in draft mode and SCRIPSIT 16 cannot find a certain font you have requested, or if you have never requested a font. (This font only affects printing, not formatting -- if SCRIPSIT 16 doesn't know what font you are using, it will have formatted your document in 10 pitch.)

This is also the font that the printer will be left in after printing a SCRIPSIT 16 document.

OF Offset. This is the offset added to the left margin for each line printed on this printer, in units (as for incremental spacing). Sometimes it is necessary to move an entire document to the right. (An example is with the Daisy Wheel II Sheet Feeder, which requires you to begin the printout in the center of the platen rather than the far left edge.)

In addition to the description file, a font select file is needed for each font supported by the printer. These files are in the same directory as the description file. Each font select file is three lines long.

The first line is a string that is used to select the font when SCRIPSIT 16 needs to print in it.

The second line tells how to deselect that font. This is needed only for fonts that are "subfonts" of other fonts (for example, an elongated font, which is twice the width of a proportional font).

The third line is the number of increments per inch in the font, multiplied by 10. For example, the dmp2100 in correspondence-quality 10-pitch has 180 dots per inch. Multiplying this by 10 gives 1800, and this is the number we use.

Following is the font select file for correspondence-quality 10-pitch for the DMP-2100. The file is called correspond.10 and is in the directory /usr/lib/scripsit/printers/dmp2100:

```
\E\22
1800
```

One more example: The font dmp2100.el needs both a select and a deselect string.

```
\E\21\E\16
\E\17
1800
```

- o A backslash \ in front of a character gives it a special meaning:

\t	tab (same as ^I)
\r	carriage return (same as ^M)
\f	form feed (same as ^L)
\b	backspace (same as ^H)
\E	escape (note: the E must be uppercase!)
\^	^
\<	<
\>	>
\\	\ itself
\nnn	nnn represents an octal value. Any character can be specified this way. For example, \101 is the same as the character A.

- o A keyname enclosed in angle brackets < > represents the value of that name as defined in the keynames section. For example, if F1 is defined as ^A, and ESC is defined as \E, then the following two definitions are identical:

```
mult-ins=<ESC><F1>
mult-ins=\E^A
```

except that the first is easier to read.

The above describes key codes. Video display codes look the same, for the most part, but instead of representing values returned by keys, they represent values to be output to the screen. In addition, a video code needs to have an explicit length, since not all codes sent to the screen actually display; some just set a mode or change the type of display. The format for a video value is:

length/value

where length is usually 1, the / is required, and value is as defined above for key codes. (See the examples for more information.)

Lines with # as the first character are completely ignored by SCRIPSIT 16 and may be freely inserted as comments.

The first section begins with a line containing the word **Keynames** all by itself. It describes the names of certain special keys that the terminal might have. For example, the Model 16 keyboard has keys labeled F1, F2, TAB, and some others. In order to be able to use these names in the following sections, you must tell SCRIPSIT 16 what keys are available and what they send. Following is the Keynames section of the Model 16 configuration file:

```
#
# Define key names
Keynames
#
```


(Whatever terminal type you use must have an entry in a file called "/etc/termcap" that tells TRS-XENIX certain characteristics of it. This is normally set when you install TRS-XENIX and shouldn't need to be changed unless you add completely new terminal types to it. See the TRS-XENIX documentation for details of what a "termcap" file looks like.)

Once the terminal type is known, SCRIPSIT 16 will look for a file with that name in one of three directories. If you have set an environment variable called SCRIPSIT, all of SCRIPSIT's special files will be searched for first in the directory specified there. For the terminal configuration files, the subdirectory name is "config", so it will look first for a file called \$\$SCRIPSIT/config/\$TERM. (A dollar sign '\$' in front of a word represents the contents of that environment variable: if you have set SCRIPSIT to "/usr/joe", for example, and are using an DT-1 in adds-25 emulation mode, the file it looked for first would be called "/usr/joe/config/adds25".)

The second place it will look is \$HOME/scripsit/config/\$TERM. HOME is set automatically when you log in; it should never need to be changed.

The third, and last, place it will look is with the configuration files installed when you installed SCRIPSIT 16: usr/lib/scripsit/config/\$TERM. If it can't find a file by that name here, it will give up and display an error message.

Format of the "config" file

This section tells you how to set up your own configuration file for SCRIPSIT 16. Note that it's probably easier to copy an existing one and modify it than to type it in from scratch!

The config file contains four sections. Each is preceded by a distinctive keyword to tell SCRIPSIT 16 which section follows. The general format is lines that match the pattern

```
name=value
```

where name depends on the section and value represents either a key code or a video display code. These can be specified in a variety of ways:

- o Any ASCII character represents itself (for example, A).
- o Any character with ^ in front of it is a control character (for example, ^A represents control-A). This has meaning only for the letters of the alphabet. Note that ^a and ^A mean the same thing.

SCRIPSIT 16 can use every font on the printers it supports. If you add a new printer type that has its own proportional font, however, you will need to add a new font file.

Font files can be in one of three places, just like the printcap files or config files (see chapters 1 and 2). The subdirectory name for font files is fonts. There is a separate file for each font that SCRIPSIT 16 recognizes.

Fixed-pitch fonts are named either **standard** or **correspond**, followed by a dot, followed by the number of characters per inch. (Examples: **standard.10**, **correspond.12**). For 16.7 pitch, however, the number 16 is used (example: **standard.16** is really 16.7 pitch).

The font file for a fixed font contains two lines. The first line is an F, and the second line is the number of characters per inch times 10 (so that for a 10-pitch font, the number would be 100).

Since each printer has its own proportional font, proportional fonts are named by the name of the printer, followed by a dot, followed by **ps**. For example, the proportional font for the Daisy Wheel II printer is **dwii.ps**, while for the DMP-2100 it is **dmp2100.ps**.

The font file for a proportional font has one line for each character, plus two. The first line has V. The second line has the number of increments per inch, times 10. For example, the **dmp2100** prints 180 dots per inch in proportional spacing mode, and so the number that would appear here is 1800.

Following this is one line for each character in the font containing the size of that character in increments. The order of characters is the standard ASCII sequence (space, !, ", etc., up to ~).

An elongated font is one that is twice as long as a corresponding proportional font. For example, **dmp2100.el** is the same as **dmp2100.ps**, but each character is exactly twice as wide.

For an elongated font, the first line of the font file says **Double**. The second line gives the name of the doubled font.

Here are some examples. The first one is the font file for **standard.10**:

```
F
100
```

The second is the font file for **dmp2100.ps**:

```
V
1800
14
08
11
14
16
```

19
19
08
08
08
13
14
08
08
08
14
14
14
14
14
14
14
14
14
14
14
08
08
14
16
14
14
19
18
18
18
18
18
16
18
18
11
14
18
16
23
18
18
16
18
18
16
18
18

SCRIPSIT® 16 is supplied with terminal description files for the Model 16 console ("trs16", "trs16+") and the DT-1 Terminal in ADDS-25 emulation mode ("adds25"). However, SCRIPSIT 16 can easily be used with a new terminal type if you supply a terminal description ("config") file for it.

Not all terminals are created equal, however, and SCRIPSIT 16 needs certain features in order to function properly. The following features are REQUIRED, and SCRIPSIT 16 won't work properly (or at all) if these features aren't present:

- o Direct cursor addressing. This means that the cursor can be positioned anywhere on the video screen with one control sequence.
- o A standout mode (usually reverse video). This means that characters on the video screen can be "highlighted" in some fashion.
- o Automatic wrap. This means that when the character at the end of one line is written, the cursor will automatically move to the beginning of the next line.
- o A screen that is at least 24 lines by 80 characters. SCRIPSIT 16 will not use any more than that but must have at least that much.

In addition, the following features are highly desirable for word-processing use:

- o Arrow keys that are easily accessible. (Most word-processing commands involve cursor motion.)
- o Backspace and left arrow keys that send different codes. If they send out the same code, the backspace command will have to be moved to another key. (Normally, backspacing is destructive, while the left-arrow key causes non-destructive cursor motion.)
- o Graphics characters, or some other "alternate character set" that can be used to display internal control codes.

SCRIPSIT 16 finds out what kind of terminal you have by looking at the environment variable TERM. This is set automatically when you log in to the type of terminal attached to that line. If you change terminals for any reason, you must let TRS-XENIX know by typing the following command at the shell prompt "\$":

```
TERM=<type>; export TERM
```

or, if you are using csh, at the shell prompt "%":

```
setenv TERM <type>
```

In both cases, replace <type> by the type of terminal you are attaching (for example, TERM=vt100 if you are attaching a DEC VT100 terminal).

18
23
18
18
16
08
14
08
14
16
08
14
14
14
14
14
14
14
11
14
14
08
08
14
08
19
14
14
14
14
11
12
11
14
14
19
14
14
11
08
08
08
16
00

The third example is the font file dmp2100.el:

Double
dmp2100.ps

SCRIPSIT 16 comes with three printer drivers that will handle all the printers it supports: bp.dwii for the Daisy Wheel II, bp.dwp410 for the DWP-410 printer, and bp.standard for all others.

The function of the printer driver is to perform any needed character translation. When the driver is loaded, it will have its standard input, output and error connected back to SCRIPSIT 16. It should pass any character through to the printer, with the exception of a hexadecimal FF (377 octal), which is a possible "signal" condition. If this is followed by an octal 001, the driver must write a character to the standard error output -- this means that SCRIPSIT 16 is waiting to regain synchronization with the driver. Any other character following this code is to be sent to the printer.

The printer driver must be located either in the current directory or in the directory /usr/lib/scrpsit/bin. It will not be found anywhere else.

The following is a usable, if not very sophisticated, printer driver.

```
#define STDIN      0
#define STDOUT    1
#define STDERR    2
#define RUBOUT    0xff
#define EOF       (-1)

/*
 * sample printer driver -- doesn't do anything special
 */
main()
{
    register int c, rubflag = 0;
    char out;

    while ((c = bp_read()) != EOF) {
        out = c;          /* convert to character */
        if (rubflag) {
            rubflag = 0;
            if (out == '\001') {
                write(STDERR, &out, 1);
                continue;
            }
        } else if (c == RUBOUT) {
            rubflag = 1;
            continue;
        }
        /*
         * Do any special translation here
         */
        write(STDOUT, &out, 1);
    }
}
```

Technical Information for Customizing SCRIPSIT® 16

1/ Adding a New Terminal Type to SCRIPSIT 16	1
2/ Adding a New Printer to SCRIPSIT 16	12
3/ Adding Additional Fonts to SCRIPSIT 16	19
4/ Adding a User Printer Driver to SCRIPSIT 16	22

```
/*
 * bp_read -- return the next character from the parent. Use buffer
 * to speed things up.
 */
bp_read()
{
    static char buf[256];
    static char *p;
    static int count = 0;

    if (count == 0) { /* get new buffer full */
        count = read(STDIN, buf, sizeof(buf));
        if (count <= 0)
            return (EOF);
        p = buf;
    }
    --count;
    return (*p++ & 0xff);
}
```

Technical Information
for
Customizing SCRIPSIT® 16