

XShell v3

Cross Development Environment

Revision 3.2x

December 1991

XShell v3 Cross Development Environment

©MicroProcessor Engineering Ltd, 1991
133 Hill Lane
Southampton
SO1 5AF

Phone: (+44) (0)23 8063 1441
Fax: (+44) (0)23 8033 9691

Table of Contents

The XShell v3 Development Environment

1. User Interface and Tools	1
1. Keyboard	1
2. Editors	1
3. Device programmer support	1
4. DOS functions	2
5. Monitor support	2
6. Target logging	2
7. Serial line support	2
8. Information functions	2
9. User-defined functions	2
10. Help	2
2. Screen Modes	3
3. Using dialogue boxes	4
4. Edit ASCII File	5
5. Blow EPROM	6
6. Cross Compiler	7
7. DOS Shell	8
8. Emulate EPROM	9
9. Forth Screen Editor	10
10. Generate Intel Hex	11
11. Program Information	12
12. Keyboard configuration	13
1. Configure function keys	14
2. Set up commands	15
3. Set Screen Colours	17
4. Save Configuration	18
13. List Files	19
14. Monitor Modes	20
15. Output log.	21
16. Serial line settings	22
17. Quit	23
18. Time	24
19. Terminal Emulator	25
20. ASCII Values	26
21. View Image	27

22. User definable functions	28
23. Help	29
24. Forth Glossary	31
25. Special Terminal Mode Functions	32
1. Text Downloader	32
2. Uploader	32
3. Forth File Server	33
26. The Command Line	34
27. Summary of functions	35

Using the supplied editors

1. The TED text editor	37
1. File Selection	37
2. Moving through the file.	38
3. Searching for Text	39
4. Configuring TED	40
5. Pagebreaks	42
6. Blocking Text	42
7. Moving Lines	43
8. Leaving TED	44
9. Miscellaneous	44
2. Forth macros for BRIEF	45
1. Installation	45
2. Customisation	46
3. Usage	47
3. Introducing FRED.	50
1. Entering the editor	50
2. The editor display	50
3. Keys used by the editor	52
4. Commands	52
5. Control Codes	53
6. Cursor Movement Commands	53
7. Word Handling Commands.	54
8. Line Handling Commands	54
9. Character Commands	55
10. String Search Functions	56
11. Screen Commands	56
12. Barrel Commands	57
13. Miscellaneous Commands.	58
14. Using the barrel system	59

Appendix A - On-line glossary tools

1. Generating an on-line glossary	61
2. Using the Indexer	61
3. Glossary File Format	62
4. Index File Format	63
1. Notation	64

The XShell v3 Development Environment

1. User Interface and Tools

This chapter details the use of the User Interface, how to call the functions of the development environment, and how to use them.

Keyboard

All functions are accessible by pressing an appropriate <Alt> key. Some functions (usually the most often used ones) are also available on function keys. It is possible to re-arrange the function keys, and to change the functions assigned to them.

Editors

The system may be set up to call both a Forth screen editor and a text editor. Both types of files may be compiled into a target application, and any mixture may be used. The environment can be configured to call your favourite text editor, such as *BRIEF* or any other.

Device programmer support

Functions are provided to call both an EPROM programmer and an EPROM emulator. During development of a ROM-based system, an emulator improves the turn-round time, but for the final version an EPROM must be blown. A function is also provided to convert a binary image file into an Intel Hex format file. This file may subsequently be transferred to an EPROM programmer or emulator, in place of its equivalent binary file.

DOS functions

A DOS function option is provided. This may be used to either run a DOS shell, or just run a single DOS command. A directory function is also provided, further to the DOS function.

Monitor support

When the system signs on, it is usually running in 25-line mode. If you have an EGA or VGA video card, you may toggle this to 43- or 50- line mode respectively. If you do not have either of these video cards, an error message will appear. It is possible to make the system default to 43/50-line mode. This will be described later.

Target logging

A log is provided, in order to save to disc all communications with the target system. Either the output from the target, or the output from the PC may be logged.

Serial line support

All main serial port configurations are supported, on ports 1-4. The settings may include baud rate, number of data and stop bits and type of parity and handshaking. These settings are valid for both the Umbilical link and the serial terminal emulator mode.

Information functions

Other windows are provided, containing information such as a table of ASCII values, a real-time clock and the version number of the system.

User-defined functions

Two user defined functions are provided. These may be configured with both a menu title and a DOS command line to execute. These functions may perform any legitimate DOS function or application, and may be assigned to function keys, just like all the built-in functions.

Help

Help is available at all times, on <F1>. This help is context-sensitive.

2.Screen Modes

There are several screen modes in the environment. These are Target Mode, Cross-Compiler, Umbilical Mode and Host Mode. The first of these is a terminal emulator, with which you may communicate with a target system on a serial line. The second mode is the Cross-Compiler, which is the portion of the development system which produces the target binary. The third mode is Umbilical Mode, through which you may interactively develop a single-chip target system. The screen for one of these modes is shown below. Note that the bottom line of the screen shows the serial line settings, including baud rate, parity, number of data and stop bits, type of handshaking and comms port in use. These settings relate to both the Target Mode and the Umbilical Mode. The bottom line of the screen also shows the name of the current (working) directory, and the names of the current source files, both screen and text formats. The final mode is Host Mode, which provides a small Forth environment on the PC. This may be used to compile and test small pieces of code before cross-compiling them. This Forth behaves in much the same way as the Target and Umbilical Forth systems.

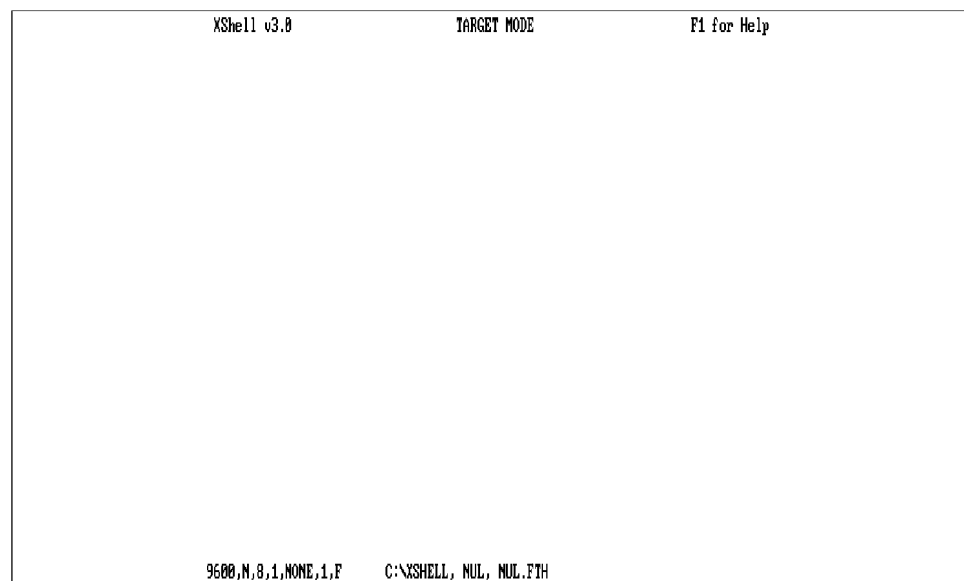


Fig. 1 - A Screen Mode

3. Using dialogue boxes

Many options request information as they are run. This may include such things as filenames, command strings and addresses. When entering these items, a default value will often be provided. This will be shown in yellow (or highlighted on an MDA). To accept and use the default, simply press <Enter> (sometimes labelled <Return>). To type a completely new response, start typing. The default will be removed. To edit the default, use the cursor keys, along with <Home> and <End>, and insert new text as appropriate. To delete characters, use <Delete> or <Backspace>. If you want to insert characters at the beginning of the default, press <Insert> first. When you are happy with the response, press <Enter>. If you want to abort the whole operation, press <Escape> to return to where you were before selecting the option.

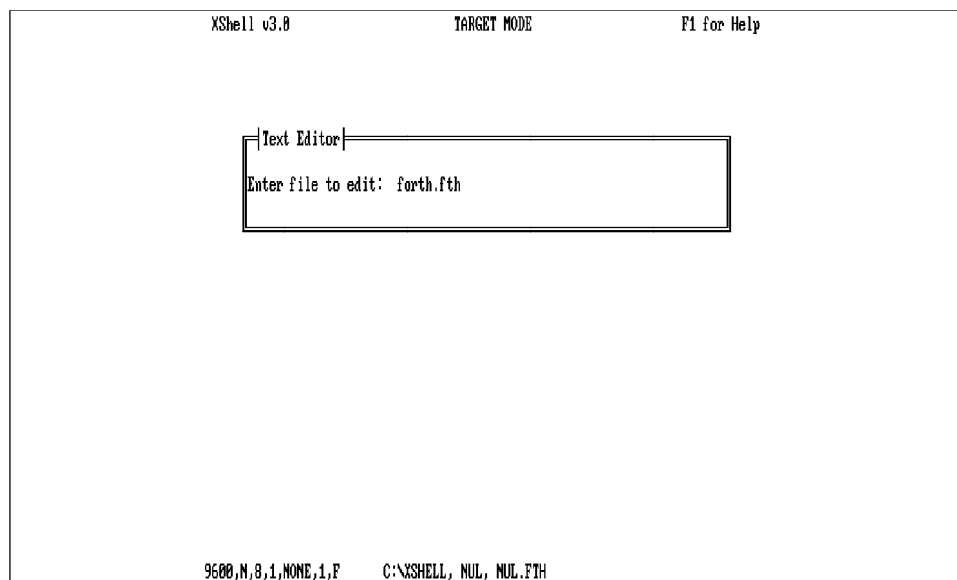


Fig. 2 - A Typical Dialogue Box

4.Edit ASCII File

Key — <Alt> A

This option runs a DOS command to call a text editor.

The DOS command should first be set up from the keyboard configuration option (<Alt> K). However, if it has not already been set up, the first time this option is selected, it will prompt for the DOS command. It is possible to specify whether or not to pass a source-file name to the editor as it is called. Some editors prefer a filename to be provided when they are called. Others, such as *BRIEF*, allow the previous session to be restored when they are called, and therefore do not need a filename to be passed to them.

For further information on how to configure the environment to call your editor, see a later section in this chapter.

If and when prompted for the name of a file in the first dialogue box, there is a file selector which may be used to select a filename, rather than just typing it in. To use the file selector, press either the <Up> or <Down> keys. You may then cursor around the pick list, pressing <Enter> to select a filename.

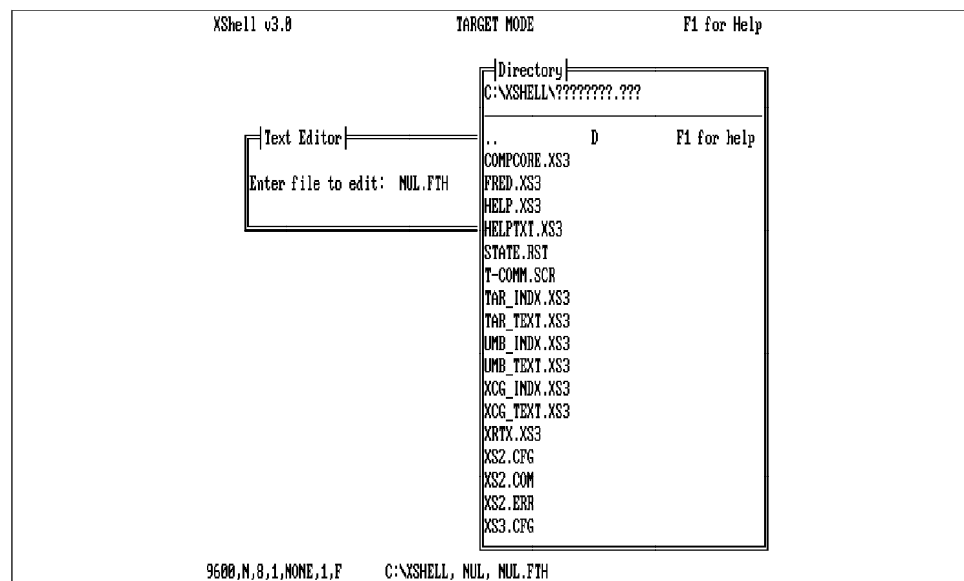


Fig. 3 - Selecting a File to Edit

5. Blow EPROM

Key — <Alt> B

This option runs an EPROM programmer.

It is possible to specify the name of a PC-based EPROM programmer, or the name of a comms package with which to drive a serial-based EPROM programmer. This option will run that programmer software. The name of the programmer software should first be set up with option <Alt> K, but if this has not been done, this option will prompt for it the first time it is run.

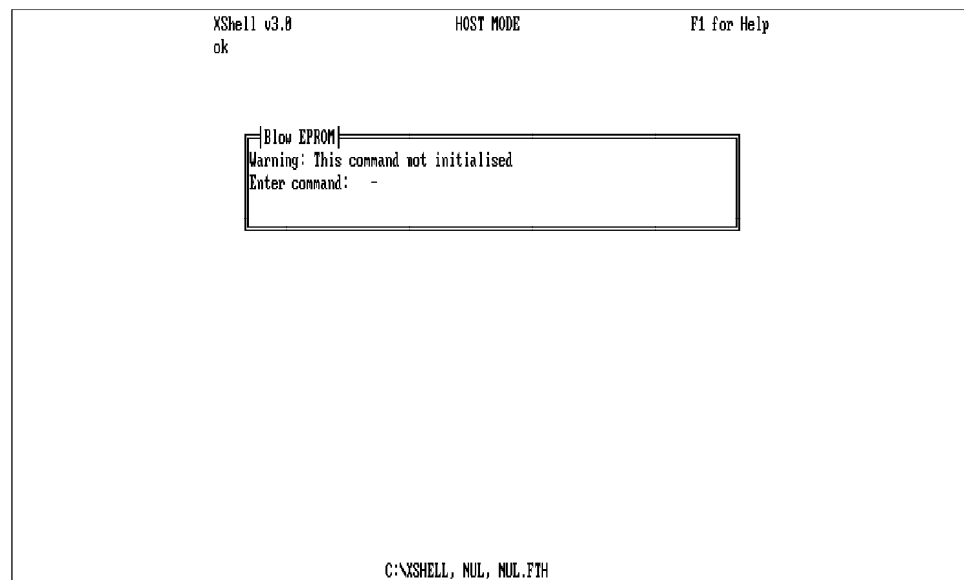


Fig. 4 - The 'Blow EPROM' Command not Initialised

6. Cross Compiler

Key — <Alt> C

This option runs the cross compiler, producing target code. For details on compiler directives, target code and use of the compiler, see a later chapter in this manual.

If the name of the compiler has not been set up beforehand, the system will issue a warning, and advise you to use the <Alt> K set-up option to configure the compiler, for both its name and the command with which to run it.

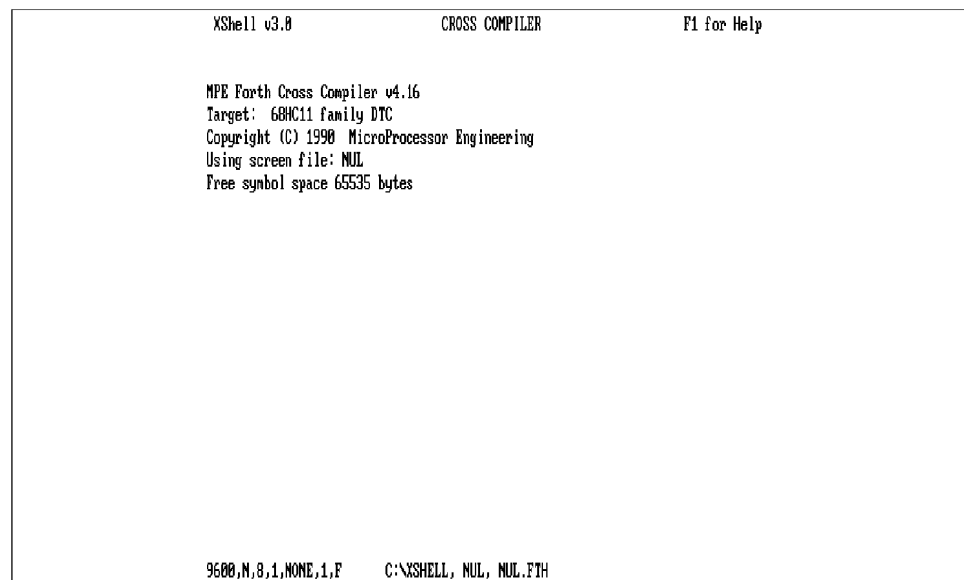


Fig. 5 - A Cross-Compiler Sign-on

7.DOS Shell

Key — <Alt> D

This option allows you to run a DOS command or command shell.

This option opens a dialogue box, requesting the DOS command to be performed. It is now possible to type any valid DOS command. The system will execute that command, and ask you to press a key before returning to the development environment.

If you want to run a command shell, simply press <Return> in place of typing a command. The screen will clear, and the DOS prompt will indicate that the system is resident. When you have finished in the DOS shell, type

EXIT

to return to the development environment.

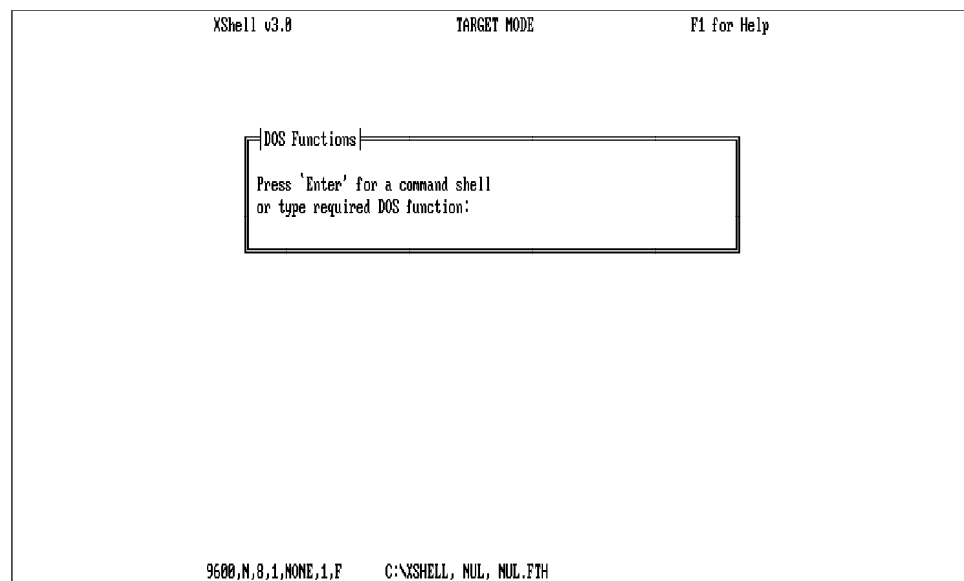


Fig. 6 - The DOS Command dialogue box

8.Emulate EPROM

Key — <Alt> E

This option runs an EPROM emulator.

When this option is provided with the name of either a PC-based EPROM emulator, or the name of a comms package with which to drive a serial-based EPROM emulator, the named package will run allowing you to use an EPROM emulator. This option is very similar to the 'Blow EPROM' option, in that if the command is not set up, the first time it is run it will prompt for the command.

This option is provided alongside the 'Blow EPROM' option so that an emulator may be used in development of an application, but a programmer may be used to produce the final system without having to reconfigure the development environment.

9.Forth Screen Editor

Key — <Alt> F

This option runs the FRED Forth screen editor.

If there is no screen file open, the system will prompt for the name of a file, creating it if necessary. The commands and functions of FRED are detailed in another chapter of this manual. When prompted for the name of a file in the first dialogue box, there is a file selector which may be used to select a filename, rather than just typing it in. To use the file selector, press either the <Up> or <Down> keys. You may then cursor around the pick list, pressing <Enter> to select a filename.

If and when prompted for the name of a file in the dialogue box, there is a file selector which may be used to select a filename, rather than just typing it in. To use the file selector, press either the <Up> or <Down> keys. You may then cursor around the pick list, pressing <Enter> to select a filename.

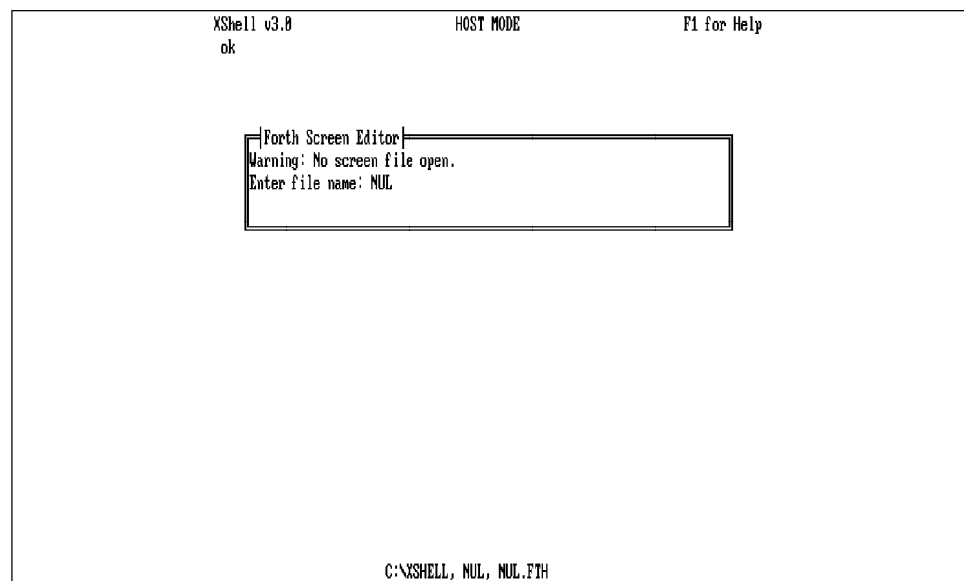


Fig. 7 - Entering FRED with no File open

10. Generate Intel Hex

Key — <Alt> G

This option will generate an Intel Hex format file, based on a binary or other source file.

A dialogue box opens, asking for the name of the source file. This is intended to be a binary file as generated by the cross-compiler section of the development system, but any file can be converted. It will then ask for the name of the output file to produce. A default will be offered. Note the use of the '.HEX' extension to denote Intel Hex. If you require the output to be sent directly to the serial port, then specify a destination filename of 'AUX'. The converted Hex file will then be sent directly to the serial port as specified with the ALT-P settings option. Having entered the filenames, you will be prompted for the start address of the Intel Hex. A default value of '0000' will be offered. After entering these names and value, the conversion will take place, and the dialogue box will close.

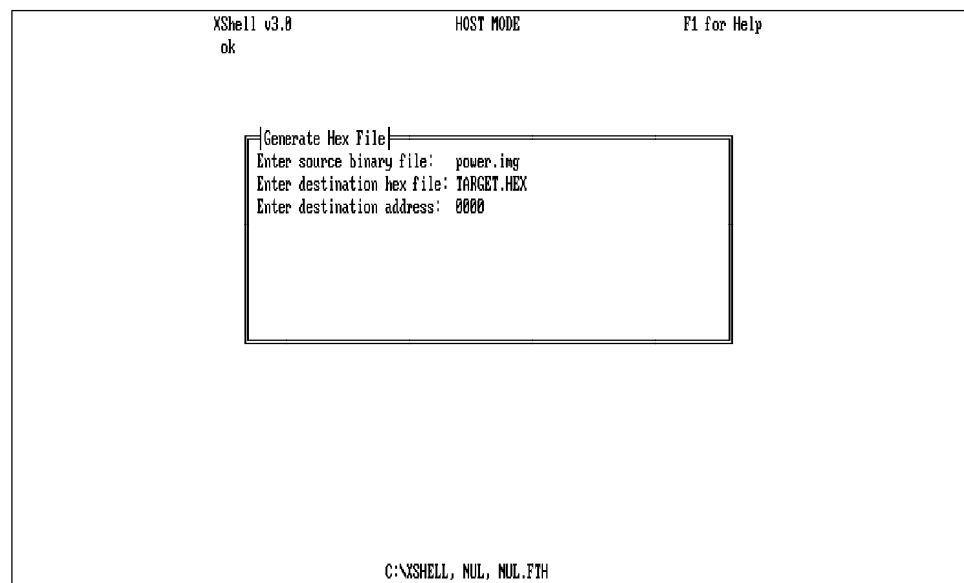


Fig. 8 - The Intel Hex Converter

11. Program Information

Key — <Alt> I

This option displays the name and version number of the product. If contacting your distributor for technical support, you will be asked to quote the information in this window.

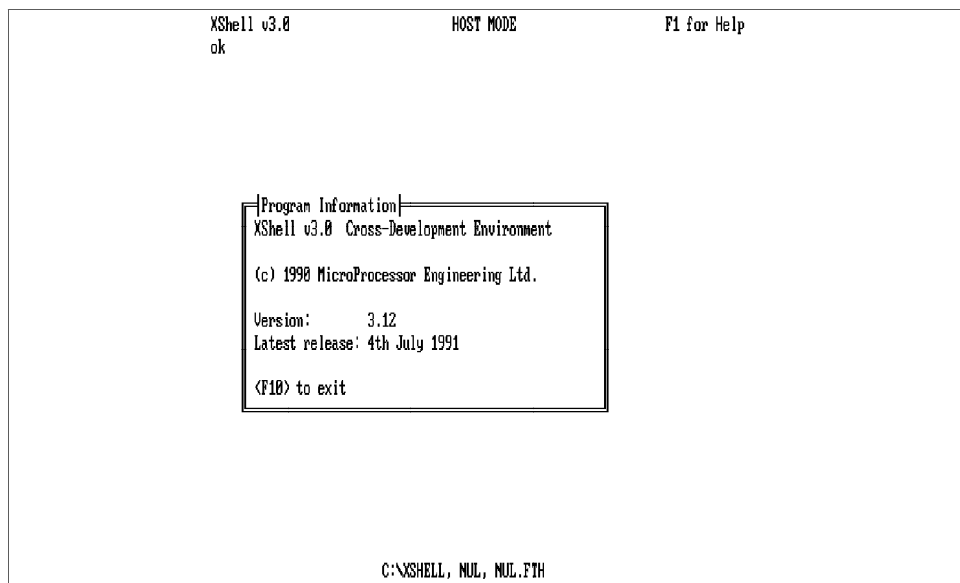


Fig. 9 - The Program Information Window

12.Keyboard configuration

Key — <Alt> K

This option allows you to configure the development environment, and save the configuration information.

The keyboard configuration window opens to reveal three options. The first of these allows you to specify which functions are assigned to which function keys. The second option allows you to define the commands to be performed by the various functions, and sundry other configuration options. The third option allows you to set up the screen colours and attributes. The fourth option allows you to set the serial line configuration of the system. Finally, option five allows you to save the current configuration to disc, ready for next time.

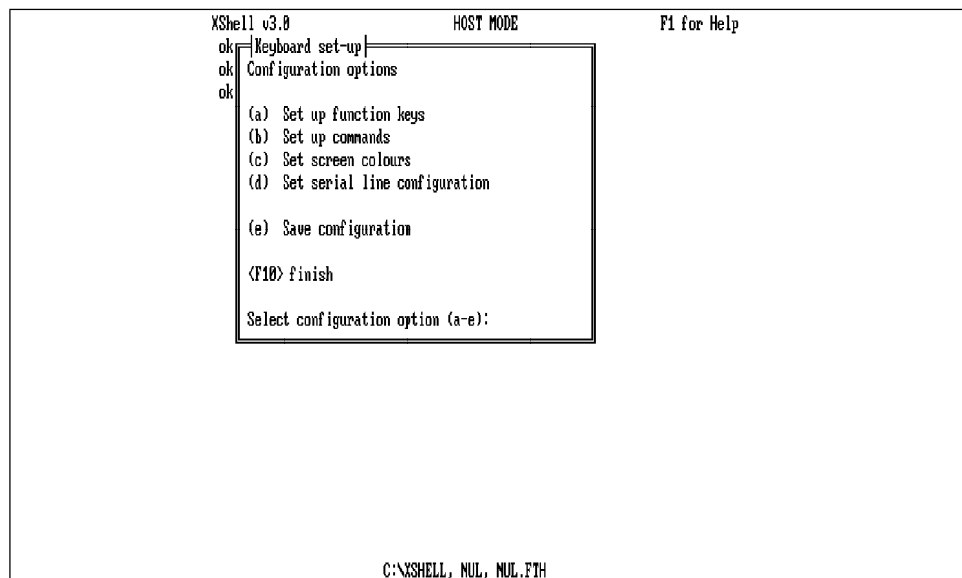


Fig. 10 - The Main Configuration Options

Configure function keys

Key — <Alt> K A

This option provides a list of <Alt> key functions, and a list of function keys with their current assignments. With the exception of <F1> (the 'help' key) it is possible to re-assign any function to any function key.

To change the assignment of a function key, first press the function key you wish to change. This will highlight it in the list of function keys. Now press the letter corresponding to the function you require the function key to perform. For example, to make <F2> run the text editor on <Alt> A, first press <F2>, then press 'A'. You will now see that the message alongside the function key has changed to that of the new function. This will be reflected in the main help screen (<F1> from the main screen).

If you want to remove a function from a function key, instead of pressing an <Alt> key letter, press '-'. This will de-assign the function key, causing it to beep when it is pressed.

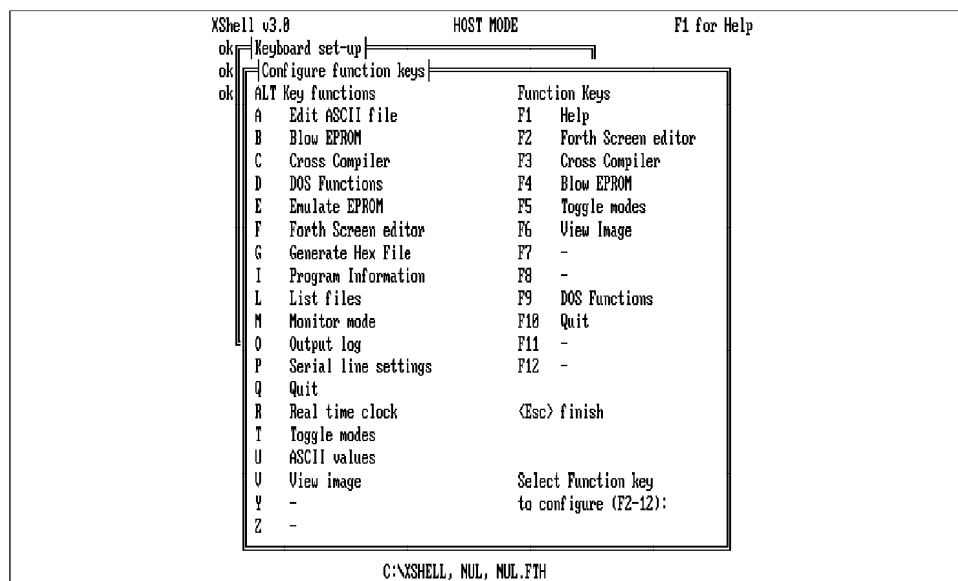


Fig. 11 - The Function Key Configuration Menu

Set up commands

Key — <Alt> K B

This option allows you to specify the DOS commands to be performed when particular system functions are executed. It also further configures the environment for screen modes (EGA/VGA) and for the 'Edit ASCII file' option.

In this window, a list of options is presented. Some of these are toggles, some allow you to enter a command line. First, the toggles.

Option 'A' allows you to specify the name of the ASCII file editor. If using an external editor, such as BRIEF, type the path and name of the editor program. If using the internal editor, TED, specify the editor name as 'TED.XS3'.

Option 'B' allows you to specify whether or not a filename is to be passed to the text editor from the <Alt> A function. Some editors need a filename, others prefer not to take one, but to restore the previous session. As you press 'B' the message by it will say 'Yes' or 'No'. Select the right one for your editor or preference.

Option 'L' allows you to specify whether the development environment beeps and asks to confirm actions. It is a four-way toggle, the showing either 'Quiet' or 'Loud' and either 'Verbose' or 'Terse'. Press the key until the required combination appears.

Option 'M' allows you to specify the default window and screen size. It is possible to run the development environment in either 25 or 43/50 line mode (EGA/VGA respectively) by default. If the system is being run in 43/50 line mode, it is possible to run DOS applications/functions in either the full screen (temporarily reverting to 25 line mode to do so), or to run them in just the top 25 lines of the larger screen. Note that this presumes the application/function knows nothing about 43/50 line mode, and will not intelligently use 43/50 lines if in that screen mode. To select these settings, option 'M' is a four-way toggle. Press it until the appropriate messages appear in combination.

Option 'D' specifies the name of the cross-compiler to use within the development environment. This may be either a '.COM' program, or a compiler module with a '.XS3' extension. The appropriate form will be installed with the system when it is installed from the issue discs.

Option 'E' specifies the Forth command string to pass to the compiler when it is executed. This will usually consist of the name of a control file (either '.SCR' or '.FTH') and the number of the screen/page to load from. Typical commands will be of the form

USING ROMCTL.SCR 2 LOAD

or

2 ALONE FROM-FILE ROMCTL.FTH

Options 'H' and 'J' allow you to specify the messages displayed alongside the User Defined options <Alt> Y and <Alt> Z. These messages will appear in the

help screen as the name of the function. There are other options to specify the actual commands to be performed.

The other options not listed above all allow you to specify a command to pass to DOS to execute as the basis of the function to which they relate. These are all valid DOS commands, and may include pathways, filenames and command-line switches. Examples are given below.

To run an executable program in the current directory:

XRTX.COM

ED.EXE

To run an executable program in another directory:

C:\TOOLS\LIST.COM

..\WS\WS.EXE

To run a DOS batch file:

C:\COMMAND.COM /C B.BAT

C:\COMMAND.COM /C ..\COMPILER\XC.BAT

To run a DOS internal command:

C:\COMMAND.COM /C TYPE COMPILER.LOG

C:\COMMAND.COM /C DIR

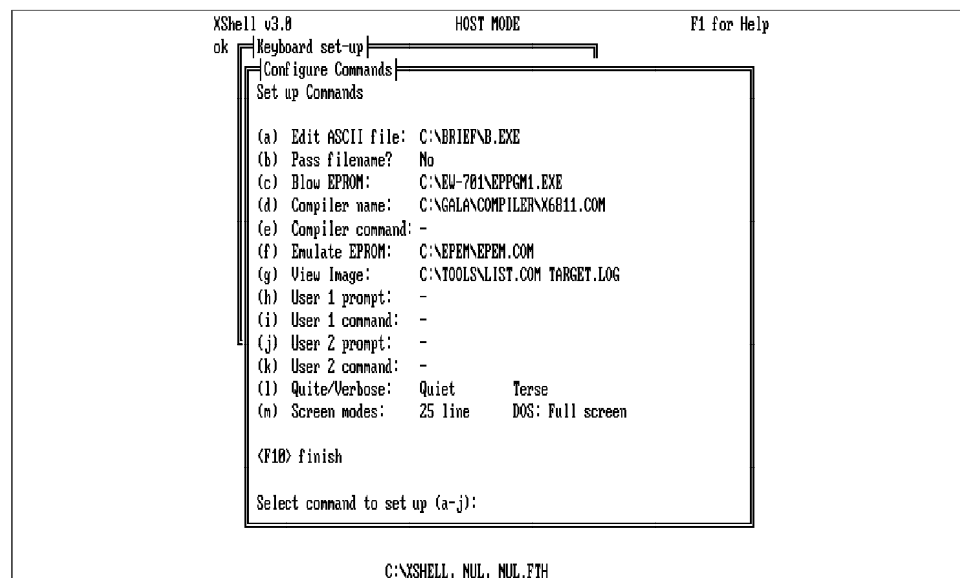


Fig. 12 - The Command String Configuration Menu

Set Screen Colours

Key — <Alt> K C

This option allows you to set up the colours and attributes for the various sections of the screens.

The window opens, showing the name of an area of the screen, and a replica of the full screen. The replica is drawn in the current set of colours. When the name of an area of the screen is shown near the top of the window, the <+> and <-> keys on the number pad may be used to move through the choice of colours. There are 255 for each area of the screen. The selections will depend on the type of monitor and video card you have connected. When the colour or attribute shown is the one you want, press <Tab> or <Shift><Tab> to move on to the next or previous area of the screen. The <+> and <-> keys may now be used to change that colour.

When you are happy with all the attributes and colours, press <Enter> to set them. The colours will take effect fully after the 'Keyboard configuration window has been closed'.

If you do not want to set the new colours, press <Escape>. This will quit back to the main configuration menu without changing the colours.

If you have a colour monitor plugged in, the system will use one set of default colours, if you have a monochrome monitor, a different set will be used.

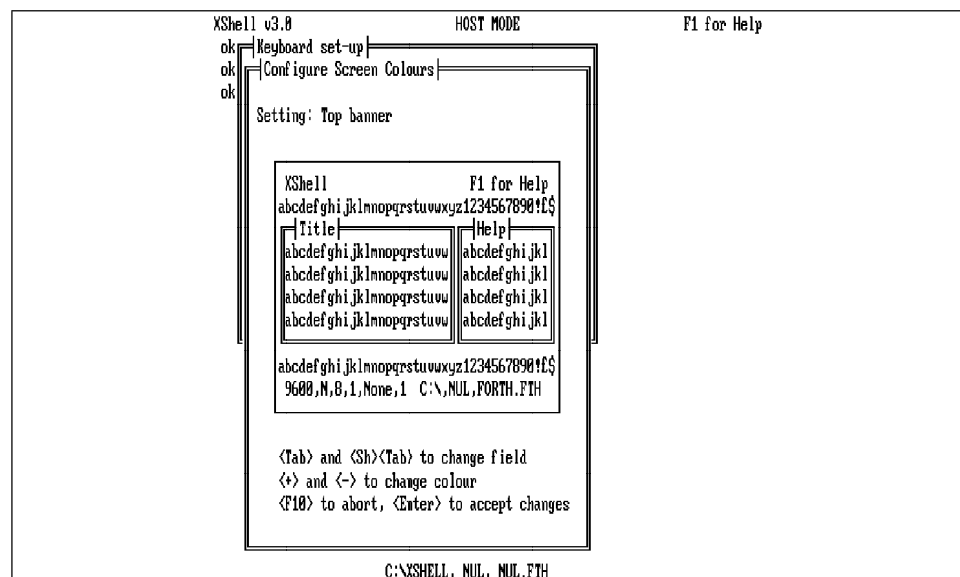


Fig. 13 - The Screen Colours Window

Save Configuration

Key — <Alt> K E

This option saves the current configuration to disc, in the current directory. When it has done so, it beeps to say it has finished.

In the configuration file are the following data:

- All serial line settings
- All command strings
- All function key assignments
- FRED time and date stamp and tab settings

These configurations may also be saved from within the serial line settings window and the Forth Screen Editor.

13.List Files

Key — <Alt> L

This option prompts for a file specification (including optional pathway) and will open the file pick list to show this file spec. If the list exceeds the length of the window, the cursor keys may be used to browse the list. To change directory, select the ‘..’ (parent) entry or an entry which is a subdirectory and press <Enter>.

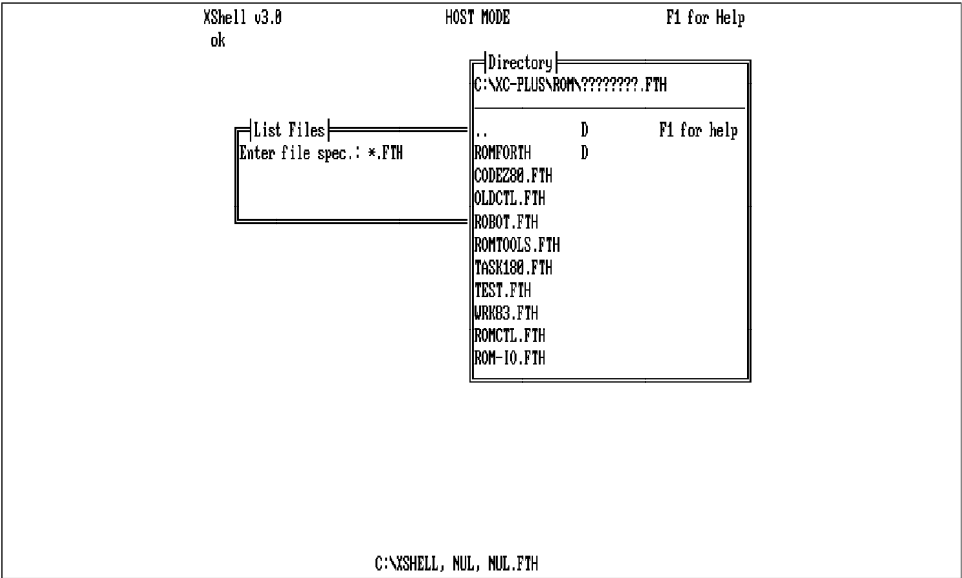


Fig. 14 - The 'List Files' Pick List

14. Monitor Modes

Key — <Alt> M

This option toggles from 25 line mode to 43/50 line mode (if an EGA/-VGA video adaptor is fitted). If your video card will not support 43/50 line mode, this fact will be reported to you in an error window. When the mode is toggled the screen will clear.

15. Output log

Key — <Alt> O

This option controls a log file of target input/output.

The dialogue box contains three options. The first may be used to select the name of the log file to which to write the target communications log. The file pick list may be used whilst selecting a file name.

The second and third options toggle the state of the log file. One option enables/disables a log of target output, the other a log of input to the target. As an option is selected, the state of the log is shown at the top of the window, and the prompts for the options change to reflect the current state of the log. When a file has been selected, and enabled or disabled, press <F10> to quit back to the main screen. If the log has just been enabled, and the log file already exists, you will be prompted to either overwrite the old file with the new one, or cancel the operation. If you want the log output to be directed to the printer, select a log file name of

PRN

Note that many targets echo input. If this is the case, when logging output from the target, the input will also be captured.

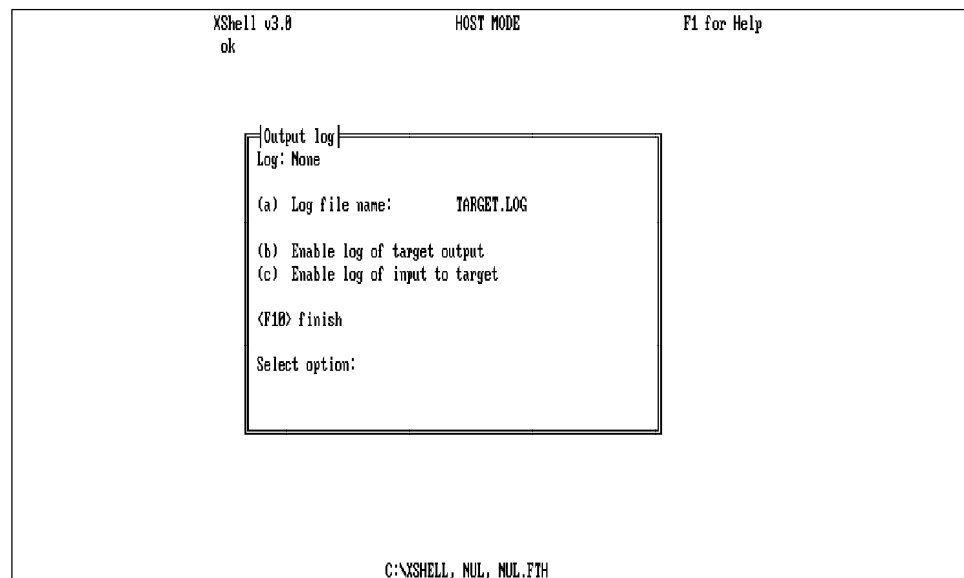


Fig. 15 - The Target Log Control Window

16. Serial line settings

Key — <Alt> P

This option allows you to set up the serial port on the PC for both terminal mode and umbilical mode.

The window opens showing, at the top, the current line settings. There a number of options, covering baud rate, number of data and stop bits, parity, handshaking and port number. Each option is associated with a key. Pressing the appropriate key makes the required setting, and the new setting is reflected in the top of the window. There is also a 'Save configuration' option, which will save both the new serial line settings and all the other configuration information. When you are happy with the new serial line settings, press <F10> to return to the main screen. One option allows you to specify that in target mode, the PC may be used as a file-server for the target Forth system. This is indicated by either 'Dumb terminal' or 'File server'. If the system is configured to act as a file server, an 'F' will be appended to the serial line setting as shown on the bottom line of the screen. A list of functions provided by the file server is provided later in this chapter.

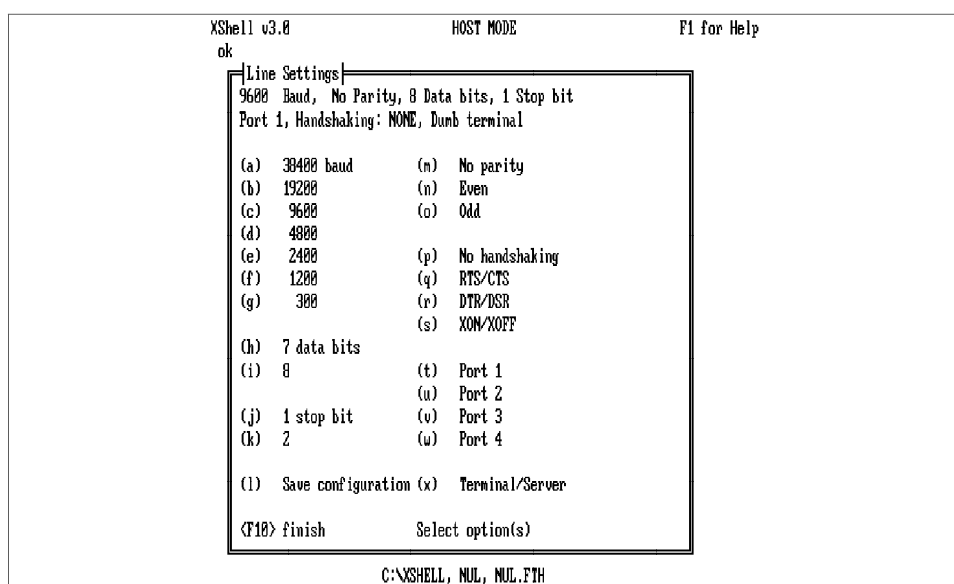


Fig. 16 - The Line Settings Window

17.Quit

Key — <Alt> Q

This option quits the development environment, and returns to DOS.

Before it quits to DOS, a window pops up asking you to confirm the selection. Pressing 'y' or 'Y' will quit to DOS, any other key will return to the development system. Note: this confirmation window may be turned off by setting the system to 'Terse' with <Alt> K B option L. When 'terse', <Alt> Q quits immediately to DOS.

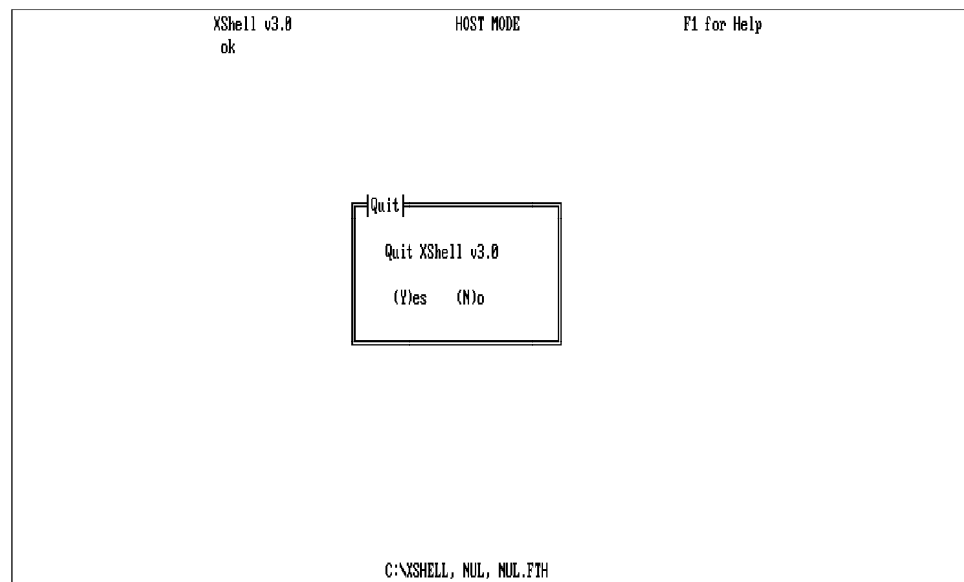


Fig. 17 - The Quit Confirmation Box

18.Time

Key — <Alt> R

This option reveals a real-time clock and calendar.

To return to the main screen, press <F10>.

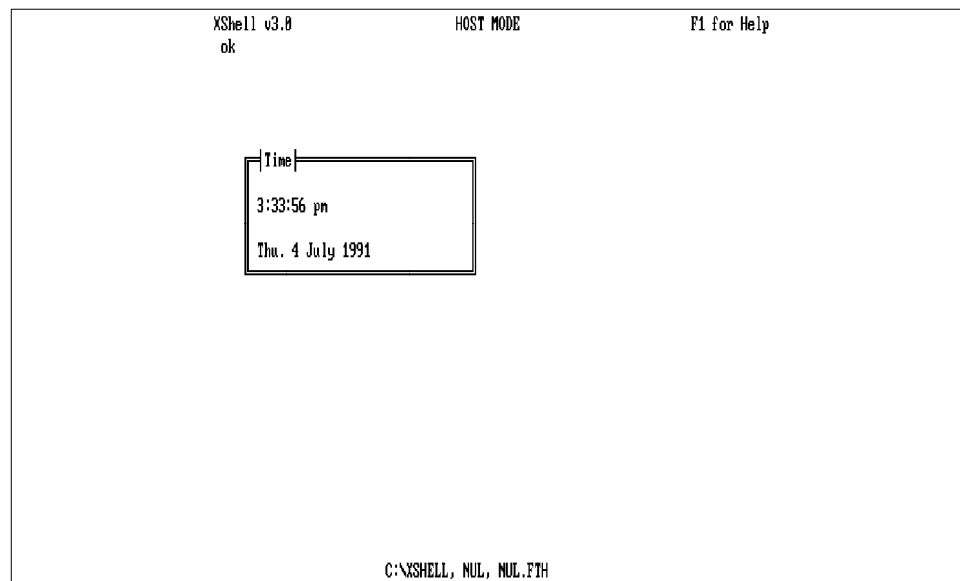


Fig. 18 - The Clock Window

19. Terminal Emulator

Key — <Alt> T

This option switches to and from Terminal Mode and Host Mode.

Whichever mode is running, pressing <Alt> T will toggle to the other. The settings for the terminal emulator are controlled with <Alt> P. The screen will restore to that shown last time terminal mode was used. If it is the first time in a session, the screen will clear. When toggling back out of terminal mode, again with <Alt> T, the screen from the previous mode will be restored. Thus, this toggle may be treated as that between two windows.

20.ASCII Values

Key — <Alt> U

This option displays a table of ASCII values from 00 to FF (hex).

The window shows hex and decimal values and the printed character, with control key value and mnemonic for those below 20 (hex). As the window only shows a portion of the table, to scroll along the list, use the <Up> and <Down> keys. <Page Up>, <Page Dn>, <Home> and <E-nd> may be used to move quickly around the table.

The table may be moved around the screen. This may be done by holding down the <Shift> key, and then pressing the appropriate cursor key.

Pressing <F10> will quit from the window back to the main screen.

D	H	Ch
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?
64	40	@

Fig. 19 - The ASCII Values Table

21.View Image

Key — <Alt> V

This option calls a file browser to dump the binary output of the cross compiler.

This option runs a DOS command, much like other options, but to call a file browser. This may be any one of your choice. The command must be set up with <Alt> K, and may include the name of the file to browse.

22. User definable functions

Keys — <Alt> Y and <Alt> Z

These options may be set up to perform any DOS function you care to define. The command may be any valid DOS command, application or function. However, unlike the DOS shell function, these may be set up and saved in the configuration file, and may be given descriptive names for the menu and help text. They may be assigned to function keys, just like all the other functions. If a user defined function is not set up the first time it is called, you will be prompted to enter both the command string and the menu text. However, these options may be set up more fully from the <Alt> K menus.

23.Help

Key — <F1>

Help is always available, from the <F1> function key.

This function key may not have other functions assigned to it. From the main screen, <F1> shows a list of keys, both <Alt> and <Fn>, and their functions. Within the help window, other function keys may be used to show other help information.

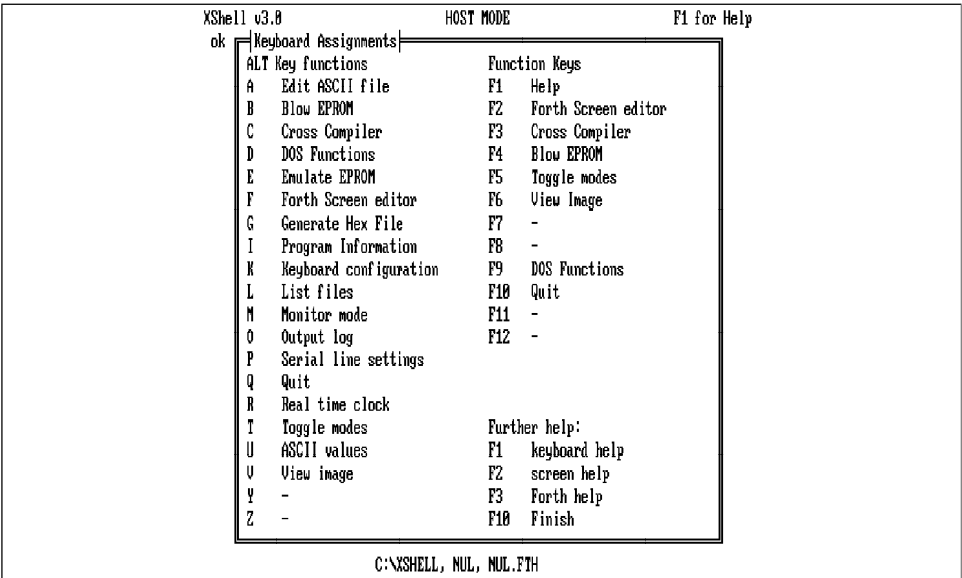


Fig. 20 - The Main Help Window

In other parts of the system, such as within dialogue boxes or data entry options, the 'Help' key will provide more context-sensitive help. Unless otherwise indicated, help is not available from within help boxes.

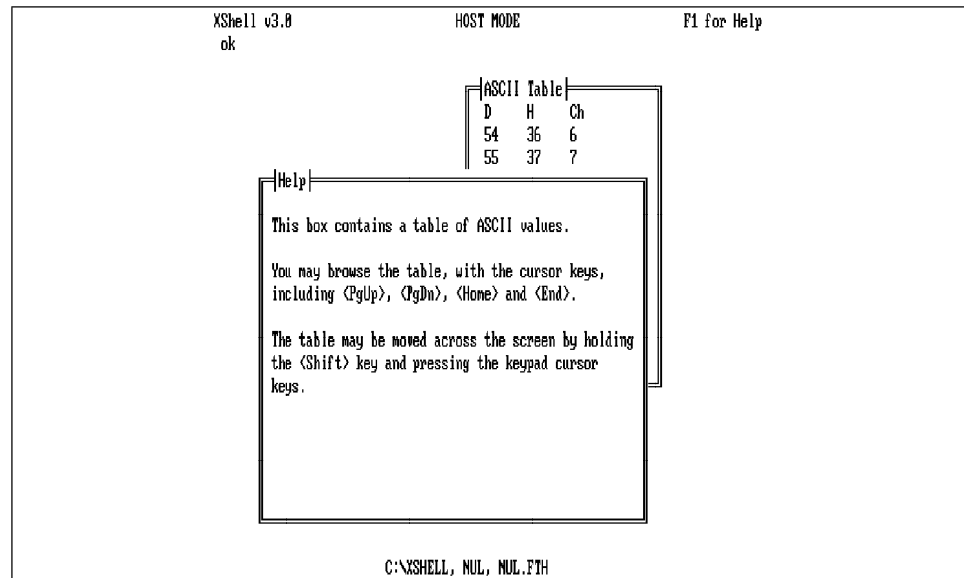


Fig. 21 - A Context-Sensitive Help Window

24.Forth Glossary

A Forth glossary is available on-line from any screen mode. It is obtained by first pressing <F1> to show the list of <Alt> keys and Function keys. The Forth glossary is available as an option on <F3> inside this window.

The Glossary window opens, offering a selection of three different glossaries. One relates specifically to the Umbilical target, one to the external ROM/RAM target, and the third to the cross-compiler directives.

Press A,B or C to select the required glossary. You will now be in a dialogue box in which you may type the name of the Forth word for which help is required. If the word is to be found in the glossary, the main glossary window will show the word. If the word cannot be found, this fact will be reported. When done with the glossary, press <Escape> to return to the main Help window. Help is available in the glossaries, indicating which cursor keys to use and how to initiate a search through the glossary.

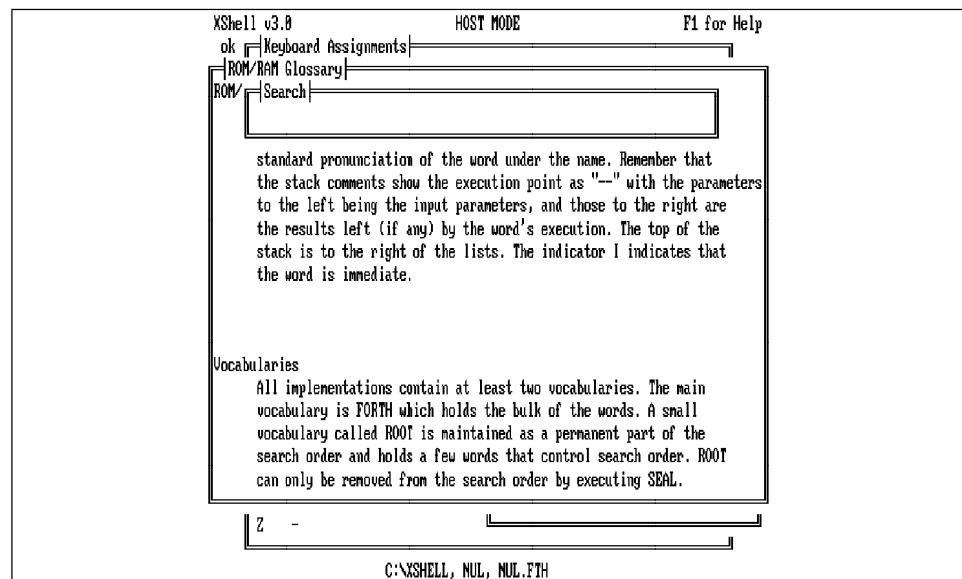


Fig. 22 - Example Forth Glossary

25. Special Terminal Mode Functions

Some functions just for use with an interactive target are available inside Target Mode. These functions are a text-file downloader and a file capture. These are designed for use with RomForth targets.

A text file downloaded is available on the <PageUp> key. To start the download, type

UP-LOAD

This will then prompt with instructions. Press <PageUp>. Type the name of the text file to download. The line counter will indicate progress. At the end of the file, the download window will close, and you may resume work on the target system. If the target generates an error while the file is being downloaded, the downloader will time out. Before you are asked what file to upload, you will be asked whether the target is a ROM PowerForth or other target. This selection defines whether to use the XON/XOFF handshaking protocol used on the ROM PowerForth target. Press 'R' or 'O' to suit the target machine.

A file uploader is also provided on the <PageDown> key. This may be used to capture output from the target, either Hex or Binary.

To start the capture, type the appropriate command on the target, but without completing it (ie do not yet press <Cr> or other start key). Now press the <PageDown> key, and type the name of the file to capture. After specifying the filename, press keys to complete the command to the target. A counter will indicate the amount of data captured from the target, and this is written to disc. When the target stops sending data, the uploader will time out, and finish capturing. You may then continue with the target.

Forth File Server

It is possible to use XShell v3 as a file server to a serial-based target system. This facility must be enabled from the serial line settings window (<Alt> P), and the target end of the code must be compiled into the target ROM. This must be included from either T-COMM.SCR or the RomForth source files via the cross-compiler. When file serving is enabled on the PC, the bottom line of the screen includes a 'F' at the end of the serial settings indicator.

The PC may then be used to provide file services for the following target Forth words:

USING	\ <file> — ; set scr file name
LOAD	\ n — ; load scr no.
LIST	\ n — ; list scr. no.
USE	\ <file> — ; set fth file name
FROM-FILE	\ n1 n2 — ; load pages numbered

26.The Command Line

The development environment may be run with the simple command line
XS3 <Cr>

but it is also possible to provide command line options and switches.

The following options are available:

/T	\ runs terminal mode
/M	\ runs 43/50 line mode
/S:<name>	\ sets current screen file to <name>
/F:<name>	\ sets current text file to <name>
/C"<command>"	\ runs compiler with <command>

The file name options override the default file names. The video mode option overrides the default as saved in the configuration file. The /T and /C options override the normal default mode.

Note: A command must be supplied to the /C option, even a null one, ie /C"".

An example: To run the compiler immediately, and set a default text file, and run in 43/50 line mode:

XS3 /F:appl.fth /C"all from-file romctl" /M <Cr>

27. Summary of functions

<u>Alt Key</u>	<u>Function name</u>
A	Edit ASCII File
B	Blow EPROM
C	Cross Compiler
D	DOS Shell
E	Emulate EPROM
F	Forth Screen Editor
G	Generate Intel Hex
I	Program Information
K	Keyboard Configuration
L	List Files
M	Monitor Modes
O	Output Log
P	Serial Line Settings
Q	Quit
R	Time
T	Toggle Modes
U	ASCII Values
V	View Image
Y	User definable option 1
Z	User definable option 2

Blank Page

Using the supplied editors

1.The TED text editor

TED is a simple text editor which behaves as a smaller version of *BRIEF*. It is designed for use within the interactive XShell environment. Many of its keystrokes are the same as those of *BRIEF*, and any user familiar with *BRIEF* will find TED a comfortable environment for editing smaller text files. When the files become too large or a mammoth editing session is necessary, switch to *BRIEF*.

File Selection

file - Alt-E

To edit an existing file, type Alt-E. The editor will prompt for the file's name. A file name can be entered or the picklist can be used, by pressing up-arrow or down-arrow. The picklist will appear and a file can be selected from the list using the up and down arrows followed by <Enter>. If the picklist is not used the editor will attempt to load the file of the entered name. If editor cannot find the file, and no file extension was specified, it will attempt to load the same file with the extension .FTH. If this file does not exist a new file will be created.

Moving through the file

The cursor keys on the numeric keypad are used to move through the file. The keys have the actions:

Up-arrow	Moves the cursor up one line. If the top line is reached, the text will scroll.
Down-arrow	Moves the cursor down one line. If the bottom line is reached, the text will scroll.
Left-arrow	Moves the cursor left one column. If the left edge of the screen is reached, the text will scroll.
Right-arrow	Moves the cursor right one column. If the right edge of the screen is reached, the text will scroll.
Page-Up	Displays the previous screenfull of text. If the start of the text is reached, the cursor is placed on the first character of the text.
Page-Down	Displays the next screenfull of text. If the end of the text is reached, the cursor is placed on the last character of the file.
Ctrl+Page-down	Moves the cursor to the next page break. If there are no more pagebreaks, the cursor is placed at the end of the text.
Ctrl+Page-up	Moves the cursor to the last page break. If there are no more pagebreaks, the cursor is placed at the start of the text.
Alt-G	Asks the user to specify a page number to jump to. The cursor is left at the page-break character which starts this new page.
Home	Moves the cursor to the start of the line.
End	Moves the cursor to the end of the line.
Home Home	Moves cursor to the start of the top line.
End End	Moves cursor to the start of the bottom line.
Ctrl+Home	Moves the cursor to the start of the text.
Home Home Home	Moves the cursor to the start of the text.
Ctrl+End	Moves the cursor to the end of the text.

End End End

Moves the cursor to the end of the text.

When the cursor is moved where no text exists the cursor's size will increase. This will happen when the cursor is:

- within a tab character
- off the end of a line of text
- past the end of the file

Searching for Text

Searching for a text string - F5

The editor can search for a string of characters in the text. The search starts at the cursor and ends at the end of the text. If the string of characters are found, the cursor is moved to the first character of the string. On pressing F5, the editor prompts for a string and displays the previous search string. By pressing <Return> the editor will search for that string. By pressing any other character, the previous string will clear and let a new string be entered. The search begins on pressing <Return> and can be aborted by pressing Esc.

repeat search - F4/Shift-F5

The editor will search for the next occurrence of the last search string by pressing F4 or Shift-F5.

search and replace - F6

The editor will search for a string of characters in the text and, if found, replace it by another string of characters.

To use search and replace, press F6. The editor will prompt for the search string, and when followed by <Return>, will prompt for the replace string. If Esc is pressed at any time the Search and Replace will be cancelled. On finding a matching string the string will be highlighted and the cursor positioned at the start of the string. The editor will prompt for the action to be taken:

- Pressing Y will replace the highlighted text and the next occurrence will be searched for.
- Pressing N will search for the next occurrence in the text
- Pressing G will replace all the remaining occurrences of the string in the text.

- By pressing the Esc key at any time the replace will be aborted and the string will be 'unhighlighted'.

Configuring TED

The TED editor uses a configuration file called TED.CFG. This file contains information on how the editor will look and act. When TED loads, it looks in the current directory for a configuration file. If it finds one, it sets the configuration from this file. If it does not find one, it looks in the directory where XShell came from. If it finds one here, it uses it, otherwise it sets the pre-programmed defaults. Having attempted to load a configuration file, TED tries to find a status file, TED.STA. This file contains the name of the last file edited, and the cursor position within it when the editor was exited from last time. It also contains the most recent search and replace texts. The configuration file specifies where the status file should come from. If it does not find the status file in either the place identified in the configuration file or the XShell directory, TED will ask for the name of a file to edit. However, if XShell is configured such that it always passes a file name through to the text editor, the file name is taken from XShell, and the status file is effectively bypassed.

The configuration menu

The configuration menu is located on key F2. This is a menu of configuration options. Cursor through it to select the option(s) to change. Press <Enter> to select an option, then either select the setting for the option from a sub-menu, or type the required response on the bottom message line. From a sub-menu, press <Escape> to return to the main configuration menu. Press <Escape> to return to the editor screen. The configuration will be optionally saved when the editor finishes.

Configuring the colours

The colours used by TED may be specified. These colours include the main screen - foreground and background - the message area, the header (with the name of the file) and the borders. Selecting any of these options will bring up a sub-menu of colour options. Select one, and press <Enter>. The new colour scheme will come into effect when returning to the main editor screen.

Configuring the stamp

When inserting a pagebreak (see 'inserting a pagebreak') there is an option to insert text on the following line. This text can be any string of characters, including spaces, and the text can optionally be followed by the date.

To make the editor insert text after a pagebreak, the command 'Stamp On' option must be set. The actual text is set by the stamp text option. For example,

\ this is my stamp text <date>

will, on inserting a pagebreak, insert the text '\ this is my stamp text' followed by the date on the line after the pagebreak.

Configuring tabs

When the tab key is pressed the editor will move the cursor to the next tab column. This column is set by the tab setting option. Select this option, and type the number of columns to place between tab stops.

For example, if tab spacing was set to every 8, the cursor would tab to the next multiple of eight (i.e. 1, 17, 25 ...). The tab can be filled with spaces if the option:

Pad tab with spaces

is in the configuration file.

Configuring the Picklist

When changing to a different file the picklist can be used (see 'Editing a file'). The extension of the files displayed can be specified. For example, if in the configuration file the extension is set to

*.FTH

the picklist would only display files that had an extension of .FTH.

Specifying the location of the configuration file

One of the configuration options allows you to specify where the configuration is loaded from and saved to - either the current directory or the XShell load directory. If you need different configurations for different projects, then specify to save the configuration in the current directory.

Specifying the location of the status file

One of the configuration options allows you to specify where the status is loaded from and saved to - either the current directory or the XShell load directory. You will normally need a different editor status for each project. Therefore specify that the status is loaded from the current directory rather than the XShell load directory.

Pagebreaks

Inserting a pagebreak

A pagebreak is used to segment a source file into manageable chunks and related words can be defined in one page. Text can also be automatically inserted on the line after the pagebreak (see 'Configuring the stamp'). To insert a pagebreak either type

- Ctrl+Enter, or
- Ctrl+L

This will insert a pagebreak on the current line. If the cursor is not at the start of the line of text, the line will be split by the pagebreak.

Deleting a pagebreak

To delete a pagebreak,

- Move to the first column of the pagebreak and press the delete key, or
- Move to the end of the previous line and press the delete key.

Blocking Text

Text can be blocked and then deleted, copied to scrap or deleted to scrap. Text which is in scrap can be copied back anywhere in the original text.

block text - Alt-L

To block a line of text, type Alt-L. Then, by using the cursor keys, the block can be extended up and down. By typing Alt-L again, the text will be unblocked.

Deleting a block - Del

A block of text can be deleted by blocking a portion of the text and pressing the delete key.

Deleting a block to scrap - F9

A block of text can be deleted to scrap by blocking a portion of text (see 'How to block text') and then pressing the F9 key. The block of text will disappear and a copy will be kept in the scrap. Any text originally in the scrap will be over-written.

Copying a block to scrap - F8

A block of text can be deleted to scrap by blocking a portion of text (see 'How to block

text') and pressing the F8 key. The highlighting will disappear and a copy of the text will be placed in the scrap. Any text originally in the scrap will be over-written.

Inserting a block from scrap - F10

To insert the contents of the scrap into the text:

i) Position the cursor on the line where the text is required

ii) Press the F10 key

The scrap will be inserted before the line the cursor is on and the scrap's contents are not modified.

Moving Lines

The editor can delete and copy a line to a 'line scrap' with a single keystroke. The line can then be inserted into the text.

Deleting a line - F7

The line which the cursor is on can be deleted by pressing F7. To delete a block see 'Deleting a block'.

Copying to the line scrap - F8

The line which the cursor is on can be copied to line scrap by pressing F8. The scrap referred to in 'Blocking Text' is not affected.

Deleting to the line scrap - F9

The line which the cursor is on can be deleted to line scrap by pressing F8. The scrap referred to in 'Blocking Text' is not affected.

Inserting from the line scrap - F10

The contents of the line scrap can be inserted by pressing F10.

Leaving TED

Saving your text - Alt-W

To save your text press Alt-W.

Leaving the editor - Alt-X/Esc

To leave the editor press Esc or Alt-X. If the text has not been modified since the last save, the editor will simply quit. If the text has been modified then the editor will prompt whether you wish to save the file. Typing 'S' will save the text and quit. Typing 'A' will abandon the changes and just quit the editor.

Miscellaneous

Getting the Time - F3

To display the time press F3. A window will appear with the current time... When the coffee break is over, press any key to continue with the editor.

2.Forth macros for BRIEF

The *BRIEF* program editor includes a powerful macro facility, which can be used to provide language specific support.

The support extends *BRIEF* to support paging within text files, and to provide documentation support for Forth. The documentation support will scan a text file, extract the names of all the colon definitions, and generate an ASCII alphabetically sorted file of these words with stack comments. This file can then be used as the basis of the user's own glossary.

Installation

BRIEF must first be installed using the *SETUP* program. This will ask for your initials, and will build a default macros file called XXX.M where XXX is the set of the initials given to *SETUP*. It is this file which must be edited to provide Forth support.

The file FORTH.M contains a set of macros to give *BRIEF* full page and Forth language support.

This file should be included (Alt-R) into your initials file (XXX.M). This file is in the \BRIEF\MACROS directory.

Leave your original

```
_INIT
```

macro as it is. Then rename

```
(MACRO XXX  
)
```

to

```
(MACRO <your initials>  
)
```

where <your initials> is the XXX part of the XXX.M file-name. Comment out the original initials macro. Note, a semicolon at the beginning of a line will comment it out, in the same way as a '`'`' in Forth.

The source file XXX.M now contains all the required macros for Forth support, but they must first be compiled before they can be used.

When the *BRIEF SETUP* program installed the system, it made a few changes to your AUTOEXEC.BAT file. These include a line setting BFLAGS, and appending the path C:\BRIEF; to your PATH= line. After installing *BRIEF*, reboot your PC and navigate to the C:\BRIEF\MACROS directory. This is where your initials file is kept. Once here, and once the initials file is edited to include the Forth support from the file FORTH.M, your initials file must be re-compiled. This may be done by typing

CM XXX

where XXX is, again, the name of your initials file. If all goes correctly, the compiler should report no errors. Next time you run *BRIEF*, the new macros file will be automatically loaded, and you will have Forth language support.

If errors occur when compiling your initials file, check the additions against the original FORTH.M. Make any necessary corrections, and recompile. If errors still occur, check the '*BRIEF* Macro Language Guide' for help.

The macro XXX is performed when *BRIEF* loads, thus automatically providing the features added by the file FORTH.M (now in your own macros file).

Customisation

The macro

```
(MACRO XXX  
)
```

contains a string variable INITIALS, which is used to insert your initials in the top line of a page. The initials are currently set for some default, but will need to be edited to suit your own name. This may be done by editing the line in the XXX macro which contains the phrase

```
(= initials "X.X.X")
```

The initials macro also sets up the keyboard for Forth editing, and for smart indenting. These may be changed by modifying this macro.

If any of the given functions do not perform as desired, they may be changed to suit your own editing style.

Usage

The functions added are:

Page-Up <PgUp>

This moves the cursor to the top of the current page, or the top of the previous page. The normal *BRIEF* PageUp function is moved to <Ctrl-PageUp>. The new function maintains a page counter in the left-hand end of the bottom line of the screen.

Page-Down <PgDn>

This moves the cursor to the top of the next page. The normal *BRIEF* PageDown function is moved to <Ctrl-PageDown>. The new function maintains a page counter in the left-hand end of the bottom line of the screen.

Goto-Page <Alt-P>

This function requests a page number, and then scrolls through the file until the top of the required page. The cursor is left on the top line of the required page. (Note that page numbering starts at Page 1, unlike Forth screens which start at Screen 0.)

Which-Page? <Ctrl-P>

This function displays the number of the current page. This is printed at the left-hand end of the bottom line of the screen. The cursor is moved to the top line of the current page. See the above note on page numbering. This function is provided because the <Left>, <Right>, <Up> and <Down> keys do not keep track of the page number. Also when *BRIEF* restores from a previous session, the page number is not saved.

Page-Break <Ctrl-Enter>

This function inserts a page-break (0Ch) into the file. It also places a line reading

\ ———End of Page———

on the last line of the current page. On the first line of the next page, it places

\ — dd/mm/yy — XXX —

leaving the cursor in the right place to add the name of the next word on the top line of the page. The initials XXX are gathered from a string variable INITIALS in the initials macro. The page-break macro is also responsible for inserting the date into the top line. These may be commented out or edited quite freely.

Smart-Indent <Ctrl-I>

This key is a toggle on/off to enable or disable Forth-smart indenting. When this key is pressed, the computer beeps, and a message is printed saying whether smart

indenting is enabled or disabled. When indenting is enabled, an additional two spaces are inserted at the beginning of a line, after the start of a control structure such as DO, IF, BEGIN. The indenting is removed after ends of control structures - LOOP, ENDIF, UNTIL, etc. Colon and Semicolon also control indenting. Note: the smart indenting is only designed to work on code laid out to the BT layout standard (obtainable from MPE). Other layout methods are not guaranteed to operate it correctly. However, the indent code may be modified freely.

DOS Function <F11>

This key prompts the user for a DOS command. Any valid dos command may be typed. The user will then be prompted for the name of the output file into which to redirect the output from the command. When the command has been executed, a window will open in the lower half of the current editor window, displaying the contents of the output file, ie. the results of running the command. If you wish to run a second command in this fashion, the default output file name must be edited, or else *BRIEF* will not be able to display both result files.

Edit File Ctrl-E

This version of 'Edit File' will accept a Forth '.FTH' filename without the extension. It is therefore possible to edit just

ROMCTL

rather than

ROMCTL.FTH

If the file cannot be found with either no extension or a '.FTH' extension, a new file will be opened with the name as typed. This macro directly replaces the normal *BRIEF* 'Edit_file' macro.

Quick Contents Ctrl-Q

This is a version of the Forth QC word (quick contents), which displays the first line of each page. A new window is opened in the lower half of the screen. Into this window are placed the first lines of all pages. The output filename for this window is NUL, but if you want to retain the contents list, rename the output file (Alt-O) to something else.

Glossary Ctrl-G

This function generates a glossary of all Forth definitions (colon, code, constant and variable) in the current file. For code and colon definitions, the stack effect as displayed on the line containing the word name is also included. The function first prompts for the name of a temporary file in which to work. A default name is provided. It then prompts for the name of the final glossary file. Again, a default is provided. Please note that if a glossary file has previously been opened in the current directory, with the same name *BRIEF* cannot guarantee to display the new file after it is generated. The function then scans the source file, generating an unsorted glossary. Having done this, the DOS function *SORT* is called, to sort the file into the file as named second by the user. This file is then displayed in the glossary window.

New-Stamp Ctrl-S

This function changes the date stamp on the top line of the current page to reflect the current DOS date. Please note that this macro is designed to work in conjunction with the top-line generating macro PAGE-BREAK. If you redefine or edit that macro, this one may not work correctly.

3.Introducing FRED

The screen file editor supplied with XShell is **FRED**, an editor specifically designed to edit Forth screen files. If you prefer to edit using a conventional text editor, a later section of this chapter describes and discusses the techniques available to you.

FRED has several features new to Forth editors, such as full screen editing, and a line stack/barrel, and is completely reconfigurable by the user (see below).

Entering the editor

To start the editor use the words **EDIT** or **FRED**.

EDIT

EDIT will load the editor from disc and enter the editor, displaying screen 0, or the last screen previously edited. If the editor module has already been loaded, the editor will restart without having to reloading it.

The editor can be entered displaying a specific screen number by giving the screen number to the word **FRED**.

25 FRED

As before, the editor will be loaded if not already present, and screen 25 will be displayed. In XShell, it is also possible to run the screen file editor by pressing <Alt> F, which is often mapped onto function key <F2>.

The editor display

As the editor signs on you will see the screen to be edited surrounded by a frame. To the left of the frame are the line numbers (1..16).

To the right is a box. This box is used to display current status information. The current screen number is displayed at the top of the box. Below this are the status flags or toggles. The toggles are identified by letters, with an asterisk below for an active toggle, or a dash for an inactive one.

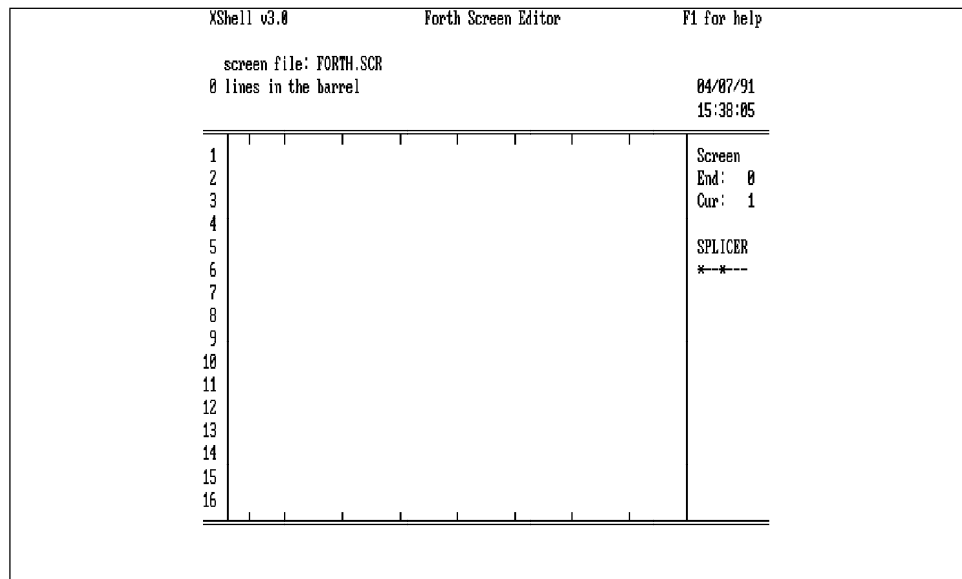


Fig. 23 - The FRED screen

The “I” toggle indicates whether new characters are to be inserted, or are to overwrite the existing ones. In insert mode an asterisk is displayed, otherwise a dash is displayed.

The “E” toggle indicates by an asterisk that an escape key sequence is in progress.

The “S” toggle indicates by an asterisk, that the stamp text and optional time and date information will be added to the right hand side of line 1 if the contents of the screen have changed.

The “P” toggle indicates by an asterisk that lines pushed off the bottom of the screen by line-insert/split are to be placed in the barrel.

Below the frame is the command line, below that the status (error) line, and below that is the search string display line. When you cannot remember the key sequences to control the editor, look at the bottom of the box. A line there tells you which key(s) to press for a help listing, the default being ESCAPE-H.

The help screen will display all the available key sequences which control the editor.

If a command requires additional information (e.g. screens to copy), this information is requested on the line below the screen frame. As the information is gathered a summary is provided in the box. When numbers are required, the command can be aborted by entering a carriage return only.

Keys used by the editor

The editor is fully configurable by the user, and reconfiguration details are provided later in this chapter. The control key functions described here refer to the configuration provided with the system. When using control keys, press the control key, hold it down, press the indicated key, and then release both. This is shown as ^X and referred to as control-X. The ALT keys are used the same way and are shown as <alt> X and referred to as alt-X. When using an escape key sequence, press and release the escape key, and then press and release the indicated key. This is shown as <esc> X, and referred to as escape-X.

When a number is requested, enter a decimal number (any number of digits) followed by a carriage return. Use the backspace key to delete digits. If only a carriage return is entered, the command will be aborted. To leave the editor, either press the escape key twice, or the END key once, and you will be returned to the Forth command line interpreter.

Commands

All editing is done within a 16 row by 64 column box drawn on the screen. The four cursor keys will move the cursor within this box only. In Overwrite mode typing any printable character will overwrite the existing character at the cursor position and move the cursor on one position. In insert mode, typing a printable character shifts all the characters after the cursor one place to the right, and then inserts the new character. If a character is pushed off the end of the line, the bell is sounded.

Screens are stored and loaded by the words **BLOCK** and **FLUSH** and so use the current screen file.

FRED recognises control keys (ASCII codes in the range 0..31) as specifying an editing command. One of these keys, the escape key (ASCII code 27) is used to lead in to a second set of commands. The second character of an escape sequence may be pressed in upper or lower case, with or without the control key, and it will still be recognised.

Control Codes

The control and escape sequences described here are the default ones provided with XShell.

Cursor Movement Commands

^L LEFT

Move cursor left. If the cursor is at the beginning of a new line, it wraps around to the end of the previous line above.

^R RIGHT

Move cursor right. If cursor is at the end of the line it wraps around to the start of the next line.

^U UP

Move cursor up. If cursor is on the top line, it wraps around to the bottom line.

^D DOWN

Move cursor down. If cursor is on the bottom line, it wraps around to the top line. Enter Move cursor to beginning of next line. If the cursor is on the bottom line, it wraps around to the top line.

TAB or ^I TAB-RIGHT

Move cursor right to the next TAB position. From the last tab position of the line, the cursor is moved onto the next line.

shift-TAB or ^O TAB-LEFT

Move cursor left to the next shift TAB tab position. From the start of the line the cursor is moved to the last position of the previous line.

HOME or ^T HOME

Top. The cursor is homed to the “home” top left hand corner of the screen.

Word Handling Commands

^F STEP-FORWARD

Move cursor forward one word. The cursor moves to the next space/non-space boundary.

^<- or ^B

STEP-BACK

Move cursor back one word. The cursor moves back to the previous non-space/space boundary.

<esc> F

DEL/W-FORWARD

Delete word to right of the cursor. If the cursor is at the start of a word, all the characters of the word are deleted. If the cursor is on a blank space, all the blanks up to the next word are deleted, provided that a word exists on the same line. The cursor remains in the same position.

<esc> B

DEL/W-BACK

Delete word to the left of the cursor. Like the previous command, repeated use of this command alternately deletes spaces and text. After each use the cursor moves to the starting position of the deleted text or spaces.

Line Handling Commands

^X

ERASE-LINE

Erase the line containing the cursor, and replace it with a blank line.

^K

DELETE-LINE

Delete the line containing the cursor, and move all the lines below it up one line, leaving a blank line at the bottom.

^S

INSERT-LINE

Insert a blank line at the cursor, moving the lower lines down. The bottom line is lost.

<esc> S

SPLIT-LINE

Split line at cursor. A new line is inserted. All the text to the right of the cursor is moved down to the new line. The bottom line is lost.

^V

TOGGLE-PUSH

Enables the facility to preserve lines, pushed off line 16 by INSERT-LINE or SPLIT-LINE, to be inserted into the barrel.

<esc> O

JOIN-LINE

Join lines at cursor. The line below (starting at the first non-space character) is copied to the cursor line. The line below the cursor is then deleted, the lines below moving up by one line. Try it - this command is useful when a definition becomes visually disorganised.

^Y **DEL-EOL**

Delete to end of line. All characters, from the one under the cursor, to the end of the line, are cleared to spaces.

Character Commands

^A **INS-SPACE**

Insert a single space at the cursor. The rest of the line is moved to the right and the last character lost.

“del” or ^G **DEL-UNDER**

Delete the character under the cursor. The rest of the line moves one position to the left.

“<-” or ^H **DEL-LEFT**

Delete the character to the left of the cursor. The rest of the line moves one position to the left.

“ins” or ^Q **TOGGLE-INSERT**

Insert mode toggle. The status of this is indicated by the “I” flag in the status box, an asterisk indicating that the insert mode is active.

String Search Functions

<esc> L **SEARCH**

Find string. The find string is asked for. To use the same string as on the previous search, just press <enter> in response to the prompt. The search may be interrupted by pressing any key.

<esc> R

REPLACE

Find and replace string. User is prompted for the find and replace strings. The replace string need not be same length as the find string, the other characters on the line will be moved as required. Characters moved off the end of the line are lost. To use the existing strings for subsequent operations, just push <enter> in response to the prompts. The find/replace may be interrupted by pressing any key.

<alt> L

REPEAT-SEARCH

Repeat search. The find string is looked for, using the string last specified to a find or find/replace command.

<alt> R

REPEAT-REPLACE

Repeat search and replace. The find and replace strings are as last specified.

Screen Commands

^E

ERASE-SCREEN

Clear the entire screen and leave the cursor at the top left corner.

“PgUp” or ^N **NEXT-SCREEN**

Step to the next screen. At the end of the file, a new blank screen will be presented. It will only be written back if changed.

“PgDn” or ^P

PREV-SCREEN

Step to the previous screen in the file.

<esc> C

COPY

Copy a screen. FRED asks you for the source and destination screens, and then copies them. After the copy the DESTINATION screen is displayed.

<esc> M

MCOPY

Multiple screen copy. This command copies a group of screens. FRED asks you for the first and last source screens, and for the first destination screen. The order of the copy will be adjusted to cope with overlapping ranges.

<esc> N

CHANGE-SCREEN

Edit another screen. FRED asks you for the number of the next screen you want to edit.

<esc> Y

DEL-EOS

Delete to end of screen. All the characters from the cursor position to the end of the screen are cleared to spaces.

^Z LOAD-SCREEN

Restore the screen to its state before it was last displayed. This is a sort of “oops” key for when things go badly wrong.

Barrel Commands

<esc> D ROLL-DOWN

Roll barrel down one line.

<esc> E CLEAR-BARREL

Erase barrel. All the lines in the barrel are deleted.

<esc> I >SCRATCH

Insert current screen line into the barrel buffer. The current screen line is left unchanged, but the cursor moves down one line. The line is inserted at the current position in the barrel, and is displayed above the editing screen. The number of lines in the barrel is updated.

<esc> J SCRATCH>

Copy current barrel buffer line to the current screen line, whose previous contents are lost. The screen cursor and current barrel line move up one line. The overall contents of the barrel remain unchanged.

<esc> U ROLL-UP

Roll barrel up one line.

<esc> X S-DEL-LINE

Delete current barrel buffer line. The rest of the barrel moves up one line.

<esc> A INS-S>

Insert the current barrel buffer line in to the current screen line. The current line and those below are scrolled down, line 16 is lost. The current barrel line is moved up one line. The overall contents of the barrel remain unchanged.

<esc> G >S-DEL

Insert current screen line into the barrel buffer. The current screen line is deleted, the lines below the cursor are scrolled up and the cursor moves down one

line. The line is inserted at the current position in the barrel, and the barrel is rolled up. The number of lines in the barrel is updated.

<Fn3> SCR>SCRATCH

Insert the entire screen into the barrel.

<Fn4> SCRATCH>SCR

Copy 16 lines from the current position in the barrel to the screen. The current contents of the screen is lost.

<Fn5> >S-LINE#

The user is requested to enter a line number to reposition the current line number of the barrel.

Miscellaneous Commands

<Fn1> HELP

Display menu of control codes.

<esc> T SET-TABS

Set tab spacing. FRED displays the current tab spacing, and asks for the new value. Just entering a carriage return will re-use the old value.

<esc> V SCR-INDEX

Index the top line of each screen. FRED asks for the required range, and entering just a carriage return will specify the start or end of the file.

Shift <Fn2> CHANGE-FILE

Change the current file. The user is prompted to enter the required file name. If the file cannot be found a request is made, whether or not to create. The user is then requested as to which screen number is required for continued editing. If a "not found file" is not to be created the current file is reopened at the current screen number. If no screen number is entered when requested the previous current screen number will be used in the new file.

<Fn6> TPLAT>SCR

Copy the template screen to the current screen. Characters from the template screen will only be copied if there is no character at the same position in the current screen.

<Fn7> SET-TPLAT-SCR

Nominate the screen to be used as the template screen.

<Fn10> TOGGLE-STAMP

Enable the facility to stamp text and optionally time and date to line 1 when a changed current screen is written back to disk.

<alt><Fn10> STAMP-TEXT

The user is requested to enter the text string to be used as the stamp text. The date and time may also be added to the text string when the screen is stamped.

“end” or <esc><esc> SET-LEAVING

Leave editor and return to XShell.

Using the barrel system

The line stack and barrel is a very useful feature of this editor. It serves two main functions.

The first is to collect groups of lines which are needed for re-use in other screens. Such lines are copied into the barrel with the “to barrel” command and the screen cursor is moved down ready for copying the next line to the barrel. When all the lines are collected, you can then move to the screen where the lines are needed, and copy them from the barrel to the screen with the “from barrel” command. The screen cursor moves up ready to receive the next line.

The second main function of the barrel is as a store, or library, of useful little Forth phrases. In this function the barrel is used as a notepad. You can browse through the barrel with the “barrel up” and “barrel down” commands. Notice how the top section of the screen shows you how many lines are in the barrel, and which line is being displayed. Individual lines can be deleted using the “del. barrel” command, or the whole barrel can be purged using the “empty barrel” command.

The number of lines which can be stored in the barrel is limited only by the amount of free dictionary space in the editor module. FRED uses this space for screen and temporary buffers, and for the barrel. When you fill this area you will receive a “barrel full” message. When FRED exits it makes a note of the value of HERE. If FRED is re-entered with the same value of HERE, i.e. no new definitions have been created, the old barrel contents will be used, otherwise the barrel is cleared.

Appendix A - On-line glossary tools

1. Generating an on-line glossary

The development environment includes an on-line for the Forth of the cross-compiler and the Forth of the target systems. These glossaries, as shipped, only cover the general cross-compiler directives, and the words in the MPE extended Forth-83 kernel. However, it is possible to add application words and new compiler extensions to the on-line glossaries. This appendix outlines the method to follow, the tools to use and the format of the files so produced.

2. Using the Indexer

The indexer is supplied in two forms. It is supplied in the form of a PowerForth with the words precompiled, and also as a forth text file which can be compiled into a Forth system. A batch file is also supplied, which can virtually automate the process of generating indices from glossary files. In order to create an index from the file GLOSSARY.HLP simply type:

```
INDEX GLOSSARY.HLP
```

This will automatically create an index file called GLOSSARY.IDX

Alternatively the process can be performed manually by first entering the PowerForth by typing

```
IND
```

in order to enter the Forth and then once the **ok** prompt has been given

```
INDEX-GLOSS GLOSSARY.HLP
```

to actually generate the index file.

If it is wished to incorporate some or all of the indexer into another system, a text file version of the source is supplied. In order to compile this simply enter a Forth system, and type

ALL FROM-FILE INDEX.FTH

This will compile a number of words into the Forth system. The word which actually performs the indexing is called index-gloss and this takes the name of the file to index as an in-line parameter.

3. Glossary File Format

In order for a correct index file to be generated with the supplied index generation utility, it is necessary that the glossary file be in a certain format. This format is, in fact, quite flexible and care only needs to be taken with titles - i.e. the text that is actually going to be inserted into the index. The source text is plain-text with only tabs, carriage returns, ASCII text and bold/unbold characters (see below). The layout of the file is important, and is designed to match, on screen, the layout of the glossary chapters of this, and other manuals.

The utility automatically assumes that any text which is surrounded by *Word-Perfect* bold and unbold characters (9Dh and 9Ch respectively), and which is also preceded by two newlines (CF/LF pairs) is to be inserted into the index. Therefore it should be ensured that each word name in the glossary file is separated by at least two newlines, and that the name of the word **only** is emboldened.

e.g.

AND n1 n2 — n3
“and”

Leaves the bitwise logical of n1 and n2 as n3.



4.Index File Format

An index file consists of a number of information blocks. There are two types of information blocks. These are index blocks and length blocks.

Every entry in the glossary file should have an associated index block of the following form:

Block Offset	Description
0	Count Byte (value of n)
1 .. n	Text of index entry
n+1 .. n+4	Double length offset into glossary in the form HL, HH, LL, LH

The first byte of the block is a count byte containing the number of characters in the text of the index entry. This is followed by the rest of the entry itself. Finally the offset into the glossary file at which the entry may be found is encoded as a double number. This double number consists of two words in big-endian format. However the individual bytes within the words are in little-endian form. Since the help utility relies upon these offsets being correct, a new index should be generated whenever the corresponding glossary file is changed.

Each of these blocks will be contiguous in the index file and should not be delimited.

At the end of the file the length of the glossary file to which the index refers should be encoded in a length block of the following format:

Block Offset	Description
0 .. 3	Total length of glossary file encoded as a double length number in HL, HH, LL, LH format

As with the offset into the glossary file that is encoded as part of an index block, this length is stored as a double number of the big-endian word/little-endian byte format.

Thus an index file will consist of a variable number of contiguous index blocks followed by a single length block.

Notation

n - A single byte number

HH - Most significant byte of most significant word

HL - Least significant byte of most significant word

LH - Most significant byte of least significant word

LL - Least significant byte of least significant word