



FORGE

For On-line Real-time
General Editing

USER'S GUIDE

POINT  **AUTOMATED SOFTWARE PRODUCTS**

2569 McCabe Way Irvine, California 92714 (714) 754-4114 / TWX (910) 595-1113
3668 "The Barnyard" / Camel, California 93923

PUBLICATION NO. ASP-400-4144

REVISION B

I. FORGE

FORGE is a text editor written in Business Basic under IRIS. It is used to create and modify Basic source code and other text files. It can be used by any number of ports concurrently.

FORGE is a useful tool for fast and efficient applications programming. It combines full-screen editing with cursor tracking and extensive line modification capabilities. FORGE also can insert portions of one file into another and save blocks of code by creating alternate files.

A unique feature of FORGE, extremely beneficial to the inexperienced user, is the inclusion of comprehensive help modules which allow the user quick access to information about the FORGE edit commands and error messages. The help modules can be invoked at any time during an edit, and will return the user to the point where the edit was suspended.

II. USING FORGE

To invoke the editor from SCOPE, simply type "FORGE". If FORGE is not on logical unit zero or the user's assigned logical unit, type "nn/FORGE", where nn=the logical unit on which FORGE is resident. The terminal will display a screen as shown below.

```
PORT NUMBER: 14      FOR ON-LINE REAL-TIME      FORGE 2.1 05/20/81
                     GENERAL EDITING

(0) EXIT FROM FORGE
(1) EDIT AN EXISTING TEXT FILE
(2) CREATE A NEW TEXT FILE

COMMENT: ENTER THE NUMBER OF THE FUNCTION YOU WISH TO EXECUTE
COMMAND:
MESSAGE:
```

The top of the screen shows port number, program name, version, revision and release date. A menu occupies the middle of the screen. Three lines at the bottom of the screen are for system/user interaction. The COMMENT line provides system prompts to the user. The COMMAND line is for user command input and the MESSAGE line displays error messages from the system.

Option "0" (or 'escape') will exit FORGE and return the user to SCOPE. Option "1" lets the user retrieve lines from an existing text file, display them on the screen, and perform edit functions. Option "2" allows the user to create and edit a new text file.

Selection of Option "1" will cause FORGE to request an existing file name and its associated logical unit. Option "2" causes FORGE to ask for the new file's name and its associated logical unit. For both options, the logical unit entry may be omitted by pressing 'return', causing IRIS to default to the user's assigned logical unit.

Option "1" will cause FORGE to clear the screen and display the first line of the specified file. Option "2" will automatically create a "last accessed" line as the first record of the new file. (This record is updated with the date and time of last access every time the file is modified using FORGE.)

III. THE HELP FACILITY

While using FORGE to edit a file, the help facility may be invoked at any time simply by executing the command ",H". The edit is suspended and a summary help screen is displayed. Information contained in this display includes a short description of every command and error message within FORGE.

At this point, any item on the screen may be selected for reference. Entering either the command itself or the number of an error message will call up a screen describing the subject in greater detail. The data available from a command detail includes general syntax structure as well as illustrative examples. Further information on the syntax notation conventions within FORGE may be accessed by entering a question mark ("?").

To return to editing the original file, press the 'escape' key. The edit will resume at the point where it was suspended, the screen will be re-displayed as before, and the line pointer will be positioned at the bottom line. Editing may be continued as though no interruption had taken place.

IV. CONCEPTS

FILES

When FORGE is invoked, it opens various work files in addition to the actual file being edited. Each user has exclusive access to a unique set of files, which is insured by incorporating the port number into the file names, (represented by "nnn" below). The FORGE work files are:

File Name	Description
EDITWKAnnn	Edit work file "A" -- Used as the input scratch file for every even-numbered pass through the file, and as the output for every odd-numbered pass through the file.
EDITWKBnnn	Edit work file "B" -- Used as the input scratch file for every odd-numbered pass through the file (except pass one) and as the output for every even-numbered pass through the file (except when the ",E" command is executed).
EDITSC0nnn	Edit scratch file -- Used to maintain various parameters and status information internal to FORGE.
EDITSV0nnn	Edit save file -- The default file name which is used on group save and insert file commands (protection 0).

The first three files listed above are deleted by FORGE upon completion of an edit. The edit save file, however, may be accessed later.

CURSOR TRACKING

With cursor tracking, FORGE can accurately recognise and interpret cursor movement keys, thereby allowing the user to move directly to and modify any area of the screen. Implementation of cursor tracking involves enabling the \$TERM module with cursor tracking appropriate to the CRT. Full-screen editing capabilities then become available. For terminals without directional keys, the following control codes may be used:

<u>Keyboard character</u>	<u>Control code</u>
Escape	CTRL D
Left Arrow	CTRL H
Right Arrow	CTRL I
Down Arrow	CTRL J
Up Arrow	CTRL K

FORGE invites new text by displaying the cursor at the beginning of the bottom line. FORGE will roll up each new line of text as it is entered and return the cursor to its original position.

To request any edit function other than the acceptance of new text, the user must supply a command at the beginning of the bottom line. In FORGE, commands are distinguished from ordinary text by a very simple rule:

All FORGE commands are preceded by a comma and must start at the beginning of the bottom line.

Full-screen editing capabilities are a key feature of FORGE. The cursor movement keys may be used to position the cursor anywhere on the screen, and text may then be directly modified. The potential therefore exists to perform edit functions without ever using the line modification commands, simply by moving directly to and modifying text as it appears on the screen.

FORGE was designed as an editor of Basic source code. Although FORGE is capable of many word processing functions, these functions are not necessarily relevant to its purpose.

The following walk-through of various FORGE functions should be helpful to the first-time user. Read the comments that appear in the "DESCRIPTION" column. Then enter the information exactly as it appears in the "INPUT" column. Note the result of each entry and proceed to the next step. Remember to observe correct spacing between all words. (CR) indicates that the 'return' or 'enter' key should be pressed. (ESC) indicates that the 'escape' key (CTRL D) should be pressed.

DESCRIPTION	INPUT
1. Enter FORGE.	FORGE (CR)
2. You wish to create a new text file.	2 (CR)
3. Specify the name of a file.	<filename> (CR)
4. Specify your assigned logical unit.	(CR)
5. You will now have a "last accessed" line showing on the bottom of your screen. FORGE is ready to accept new lines of text data. To build a file containing information which may then be modified, type in each of the lines on the right.	10 REM (CR) 10 REM "RN" PROGRAM (CR) 10 DIM Z\$(2) (CR) 20 PRINT "TAP ENTER "; (CR) 30 INPUT ""Z\$ (CR) 40 PRINT (CR) 50 PRINT "STARTING" (CR) 60 REM (CR) 60 REM MAIN LOGIC (CR) 60 REM (CR) 60 FOR P=0 TO 9 (CR) 70 PRINT P (CR) 80 NEXT P (CR) 90 REM (CR) 90 REM FINISH UP (CR) 90 REM (CR) 90 PRINT (CR) 100 PRINT "FINISHED" (CR) 110 CHAIN "" (CR)

DESCRIPTION	INPUT
6. Verify that you are at the end of file, and bring the pointer (arrow on the left) to the bottom of the screen.	(CR)
7. Use the cursor movement keys, or the equivalent control codes, to move the cursor to the word "UP" in line 90. Change the word "UP" to "PROGRAM".	[See section IV for cursor control codes, if needed] PROGRAM (CR)
8. Move the pointer up one line.	(ESC)
9. Move the pointer up three more lines. After pressing the 'escape' key for the first time, wait for the pointer to move up before pressing it again.	(ESC) (ESC) (ESC)
10. Re-display the screen, ensuring that the display is accurate.	,X (CR)
11. Move the pointer down one line.	, (CR)
12. Move the pointer down four more lines. Note that after the pointer reaches the bottom of the screen, it wraps back to the top.	, (CR) , (CR) , (CR) , (CR)
13. Position the pointer to line four on the screen.	,4 (CR)
14. Copy line number 10 (the pointed line) to the bottom of the screen. Note that the pointer automatically moves to the next line.	,C (CR)
15. Copy the remaining lines, down to and including line number 70, to the bottom of the screen.	,C (CR) ,C (CR) ,C (CR) ,C (CR) ,C (CR) ,C (CR) ,C (CR) ,C (CR) ,C (CR) ,C (CR) ,C (CR) ,C (CR)
16. You now wish to go back to the start of the file, or "wrap" around. The first line will again be shown at the bottom of your screen.	,W (CR)
17. Bring up the next three lines of file onto the screen.	(CR) (CR) (CR)
18. Bring in a whole "page" of the program. This places the line currently at the bottom of the screen at the top.	,P (CR)
19. Retrieve another page. Note the "EOF" message at the bottom right of your screen, indicating the end of file has been reached. You are now looking at the last 23 lines of the file.	,P (CR)
20. Wrap to the start of the file again.	,W (CR)

DESCRIPTION

INPUT

- 21. Find the line beginning with "40" (line number 40). ,F 40 (CR)
- 22. Roll up ten lines from the file. ,R 10 (CR)
- 23. Roll up to the end of the file. Note: This form of roll may be interrupted at any point with the 'escape' key, but a ",X" should then be performed to ensure the accuracy of the screen. ,R (CR)
- 24. Move the pointer to line number 50. ,18 (CR)
- 25. Insert a line after line number 50. ,I (CR)
60 REM MORE COMMENTS (CR)
- 26. Move the pointer to the first line number 10 on the screen. [Use any of the previously described techniques]
- 27. Delete five lines (line numbers 10 through 20). ,D5 (CR)
- 28. Enter a new line after line number 110. 120 STOP (CR)
- 29. Return the cursor to the command area. (CR)
- 30. Position the pointer to line number 50. ,17 (CR)
- 31. Modify the pointed line. Replace the word "STARTING" with "START". ,M STARTING<START (CR)
- 32. Modify the pointed line. Insert the word "PROGRAM" after "START". ,M START> PROGRAM (CR)
- 33. Look at the help module to see a summary of all FORGE commands. ,H (CR)
- 34. Select more information about the modify command. ,M (CR)
- 35. Read the syntax conventions for describing commands in the help modules. ? (CR)
- 36. Return to the modify command detail. (CR)
- 37. Look at part 2 of the detail for the modify command. (CR)
- 38. Return to the screen that you were editing. Note that the pointer has moved to the last line. (ESC)
- 39. End the edit, keeping all of the changes made to the file. ,E (CR)
- 40. You do not have any more editing. N (CR)

V. COMMAND SUMMARY

<u>FUNCTION OF COMMAND</u>	<u>SYNTAX</u>
Position pointer	,{nn}
Abort edit	,A
Copy pointed line	,C
Delete line(s)	,D(A,B,nn)
End edit	,E(D,X,#nnn)
Find label	,F'blank'<string>
Group save	,G[A,B][E,'blank']["<filename>",""]
Help	,H
Insert line(s)	,I{nn}
Insert file	,I["<filename>",""]{#nnnnn,/<string>}
Locate string	,L'blank'<string>
Modify line(s)	,M(nnnnn)'blank'<string1>[<,/>]<string2>
Show record number	,N
Page forward in file	,P
Quit file insert	,Q
Roll through file	,R(nnnnn)
Skip to record number	,S nnnnn
Wrap to start of file	,W
Extract and refresh	,X
Position cursor	,@{nn}
Comment pointed line	, <string>
Set comment delimiter	, =<character>
Print screen	,\${n}
Move pointer up	'escape'

COMMAND SYNTAX CONVENTIONS:

,H -Items not bracketed by adjacent matching delimiters (i.e., {}, [], <>, or single quotes) should be entered exactly as shown.

nnn -The small letter n represents a positive numeric value. The number of n's indicates the maximum size of that field.

[A,B] -Items in brackets, separated by commas, indicate that one of the bracketed items must be included in the command.

'BLANK' -Items surrounded by single quotes indicate the single keyboard characters described, in this case, a blank space.

(A,B,C) -One of the items, separated by commas and appearing in braces, may optionally be included in the command.

<STRING> -These delimiters indicate that the item is to be replaced by the type of data indicated, in this case, any alphanumeric string.

VI. ERROR MESSAGES

<u>Number</u>	<u>Description</u>
01	Cursor was moved down past bottom line
02	General syntax or punctuation error
03	'Escape' was pressed when not allowed
04	Numeric parameter out of valid range
05	In ",M" command, <string1> not found
06	In ",S" command, driver busy or missing
07	In ",Q" command, not in file insert mode
08	For insert file, already in file insert mode
09	<filename> same as one of edit work files
10	<filename> can not be accessed properly