UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

User's Guide to the Data General Computer for the Branch of Exploration Geochemistry and the Branch of Energy Minerals

By

Wendy S. Speckman

Open-File Report 85-132

TABLE OF CONTENTS

General Information 1
Conventions Used in This Text 1
Login and Logout 1
Default Conditions 2
Logon.cli 2
Characteristics 2
Searchlist 2
Uppercase Letters 2
Prefix Character 2
Directory Structure 2
Pathnames 3
Messages 4
Mail 4
Command Line Error Correction 5
Interrupts and Breaks 5
Other Control Key Sequences 6
Filenames 6
Templates 6
Help Facility 7
Commands 8
Command Abbreviations 8
Command Separator (;) 8
Command Switches 8
Expanding Command Lines 9
Specific CLI Commands 9
1. Filestatus 9
2. Delete 10
3. Type 10
4. Qprint 10
5. Copy 10
6. Move 11
7. Create 11
8. Date 12
9. Time 12
10. Space 12
11. Qdisplay 12
12. Qcancel 12
13. Permanence 12
14. Rename 12
15. Execute (X) 12
16. Who 13
17. Ped 13
18. Write 13
19. Directory 13
20. Searchlist 13
21. Acl & Defacl 14
22. Characteristics 15

23. Logfile 15
24. Listfile 15
Programs 17
Macros 17
Pseudo-Macros 18
Batch (Absentee) 18
Text Editors 20
SED Text Editor 21
SPEED Text Editor 25

GENERAL INFORMATION

When quotes (") are used in this text they are used only to set off the phrase or command from the text. They should not be typed.

"<CTRL>" is used in this text to indicate that the control key (labeled "CTRL" or "CTNL" or "CTL") should be pressed. The "<CTRL>" key should be pressed and held down while pressing the other key or keys required. "<ESC>" is used to indicate that the key labeled "ESC" or "ESCAPE" should be pressed. The "<ESC>" key should be pressed and released before pressing any additional keys. The "<ESC>" key on any Tektronix equipment is the "ALT" key.

"<C.R.>" is used in this text to indicate that the key labeled "CARRIAGE RETURN", "NEWLINE", or "RETURN" should be pressed.

"CLI" stands for Command Line Interpreter. The CLI is the initial program executed for each user at logon time.

LOGIN AND LOGOUT

To access the computer: dial 232-5309. When prompted with "enter class", type "datag" and press the carriage return key. After the computer has responded with
**AOS/VS Rev. 4.04.00.00 Press NEW LINE to begin logging on **
enter a carriage return to begin. When the computer prints "Username:" type your username. [Example: JDOE]. In most cases your username will be your first initial and last name (ALL UPPERCASE LETTERS). When the computer prints "Password:" type your password. Initial passwords are "newuser". After the logon procedure is complete, the initial working directory is your username directory.

To exit from the computer type "BYE". A closure message will be printed and your telephone line will be disconnected. Wait until this closure line completes printing before you hang up the telephone and turn off the terminal, otherwise the logout process may not be completed.

To change your password, type your username as usual when logging in, but do not press the carriage return. Instead press "<CTRL> L", respond to the prompt with your old password, press "<CTRL> L" again, and then a prompt will be issued for a new password.

The computer allows multiple login sessions of each user.

DEFAULT CONDITIONS

The "LOGON.CLI" is a file that is executed by the CLI when you logon. It is currently set up to perform the following functions:

(1) Issue an initial "CHARACTERISTICS" command appropriate for hardcopy terminals. The characteristics are modified so that all terminals receive faster output from the computer. If a terminal is being used that cannot perform at this faster speed, type "SLOW_TERM" and the characteristics will be adjusted for that logon session. This will need to be typed at the beginning of each logon session. Persons using Data General terminals should type "D400" after each logon so that screen editing capabilities may be used.

(2) Issue an initial "SEARCHLIST" command to include directories necessary to execute STATPAC programs.

(3) Print any mail you have pending. The mail will remain until you delete it.

The system default is uppercase letters. The computer converts all system-level commands (including filenames) from lowercase letters to uppercase letters. A file named "data" is the same as a file named "DATA" or a file named "Data". "CHAR/ULC" is the command string used to change to lowercase type. There are two MACROS ("LC" and "UC") that can be used to change from uppercase to lowercase and vice versa rather than typing the entire command string.

The computer prints a prefix character ")" to indicate it is ready for a command.

DIRECTORY STRUCTURE

A directory is a file that contains information about and pertaining to files within that directory. Directories are used to organize data files. Data files within directories contain information. Directories may contain other directories, creating an inverted tree structure. The inverted tree begins with the root as the highest level. The symbol ":" is used by the system to represent the root directory.

Under the root directory there are many sub-directories. One of these is UDD, which contains all of the username directories. Another is UML, which contains the source and object code and documentation for applications programs, such as STATPAC.

PATHNAMES

Pathnames are used to refer to a specific data location within the inverted tree hierarchy. Relative pathnames can be used to refer to directories and files, as related to the current working directory. Absolute pathnames are used to address a data location from the root directory.

The absolute pathname to a username directory is:

```
:UDD:JDOE
```

The absolute pathname to the documentation directory in UML is:

```
:UML:STATPAC:DOC
```

The symbol ^ can be used to indicate the next most superior directory. For example: if the current working directory is
:UML:STATPAC:DOC
the following command can be used to change from the current directory to the directory
:UML:STATPAC:OBJECT

```
DIR ^OBJECT
```

The command can also be issued using the absolute pathname:

```
DIR :UML:STATPAC:OBJECT
```

Absolute pathnames are independent of the current working directory.

MESSAGES/MAIL

Messages can only be sent to another user who is currently logged on. Messages are sent to a process_id, not to a user, so they cannot be sent to a user who is not currently on the system. Mail is sent to a username, so it can be sent to any registered user whether or not he is currently logged on.

MESSAGES

To see users who are currently logged on the Data General computer, type "PED" or "?". This will cause a table of processes to be printed. There may be more than one entry per user. If this is the case it does not necessarily mean the user has multiple logon sessions. The process_id table shows the programs users are executing, if any. The process_id number (PID) used to send a message is the one associated with the CLI process connected at "CONx". This PID number is located in the leftmost column of the table.

The command syntax to send a message is:

SEND <PID> message <C.R.>

When the message is received, the system prints "FROM PIDx: message".
Nothing is printed to tell the recipient of the message the username of the sender.

MAIL

To see the list of registered usernames, type the following:

DIR :UDD;F/S;DIR/I <3 commands on one line>

This prints the names of the user directories and then returns you to your initial username directory.

If mail is sent to you while you are logged on, no message is received to indicate that you have new mail.

The mail commands are as follows:

1. To send mail: "MAIL/S USERNAME"
   The system responds with a second ")". Type the message on as many lines as necessary. To terminate a message, type a ")" on a line by itself. The mail is sent to "USERNAME" after the ")" is sent to the computer.

2. To send a file as mail: "MAIL/S USERNAME filename"
   This sends the file named "filename" to user "USERNAME". No prompt is issued to the sender.

3. To read mail: "MAIL/R"
   This prints all mail currently waiting for you
   MAIL/R USERNAME
   Prints all mail sent to you by "USERNAME"

4. To delete mail: "MAIL/D"
   Deletes all mail.
   MAIL/D USERNAME
   Deletes all mail sent to you by "USERNAME"

5. To send your waiting mail to a file: "MAIL/R/L=filename"
   This sends all waiting mail to file named filename

6. To delete mail you have sent: "MAIL/F"
   This deletes all mail you have sent.
   MAIL/F USERNAME
   Deletes all mail you have sent to "USERNAME"

COMMAND LINE ERROR CORRECTION

To delete one (1) character on a typed command line, press the "DEL" key one time. Pressing the "DEL" key five (5) times deletes the previous 5 characters. The "DEL" key does not cause the printhead to backspace, but does print an underscore (_) character for each "DEL" pressed. The "DEL" key on any Tektronix equipment is the "RUBOUT" key.

To delete an entire typed command line use a "<CTRL> U".

INTERRUPTS AND BREAKS

To break from a system command use "<CTRL> C <CTRL> A". This is a "soft" break. If this is used while in a text editor, execution of the current editing command will stop, but you will remain in the text editor. "<CTRL> C <CTRL> A" does not break from a program.

To break out of a program use "<CTRL> C <CTRL> B". This is a "hard" break. It is the only way to break out of a program. If "<CTRL> C <CTRL> B" is used while executing a system command, the process will be logged out.

OTHER CONTROL KEY SEQUENCES

To temporarily suspend printing or execution, use "<CTRL> S".

To resume, use "<CTRL> Q".

It is fairly common for new users to press "<CTRL> S" by accident or for this sequence to be sent as noise. When this happens, the terminal will lock up. It will appear that the computer has gone down or the terminal has quit working. Type "<CTRL> Q" several times and operation should continue.

FILENAMES

Filenames may be up to 31 characters long. Allowable characters for filenames are:

A-Z 0-9 ? (question mark)
_ (underline) . (period) $(dollar sign)

Do not end filenames with any of the special filename characters.

None of the template characters can be used in a filename (- + * # \).

The operating system uses several naming conventions. These are suffixes to part of the filename. Some common ones are listed below.

.F77	indicates a FORTRAN source file
.CLI	indiates a MACRO
.SR	indicates an assembly language source file
.OB	indicates an object (compiled) module
.PR	indicates a program file (executable)

TEMPLATES

Templates are wildcard conventions used with filenames. These are:

+ (plus) matches any string of characters
- (hyphen) matches any string of characters not containing a period
* (asterisk) matches any single character except a period
\ (backslash) omits a string of characters - restricts files matched by template
(pound sign) searches designated directory and all its sub-directories

Examples

In these examples "F" is the "list" command.

To search the current working directory and all its sub-directories for a file named "ABC.DATA"

```
"F #:ABC.DATA"
```

To list all files in the current working directory that end with ".LAST"

```
"F -.LAST" or
"F +.LAST" or
"F +LAST"  will find all files ending with "LAST"
           (no "." necessary)
```

To list all files in the current working directory that have the string "ROCK" anywhere in their name.

```
"F  +ROCK+"
This will find:  ROCKFIRST
                 SECONDROCK
                 THIRDROCKFOURTH
                 FIFTH.ROCK.SIXTH
                 SEVENTHROCK.LAST
```

Anything that can be listed with a given wildcard convention can be deleted with the same convention.

HELP

There is a help facility on the computer. Typing "HELP" will list all of the topics available. Typing "HELP *TOPIC" will list information about that topic. Typing "HELP *COMMAND" will list all commands. Typing "HELP COMMAND_NAME" will list information about that command. When "HELP COMMAND_NAME" is typed, it will print "TYPE HELP/V COMMAND_NAME". The "/V" is the "verbose" switch. Type "HELP/V COMMAND_NAME" and all available information about that command will be printed.

COMMANDS

ABBREVIATIONS

There are no standard abbreviations for system commands. Commands may be shortened to their minimum uniqueness. For example: The command to delete a file is "DELETE". If "D FILENAME" is typed the computer will print this message:

"ERROR: COMMAND ABBREVIATION NOT UNIQUE".

This will also happen if "DE FILENAME" is typed. "DEL FILENAME" will be understood by the system. The shorter versions for the commands are not published because Data General reserves the right to add new commands to the operating system (CLI). New commands might change the minimum uniqueness of an existing command. The computer will of course recognize more than the minimum amount required. "DELE FILENAME" is an acceptable command.

The semi-colon ";" is the command separator character. More than one command can be issued on a command line if the commands are separated by a semi-colon.

SWITCHES

Each command can be modified by command switches. The names of switches can be shortened to their minimum uniqueness. There are command switches and argument switches. Commands whose arguments take switches are not often used, therefore, only command switches will be discussed.

The command switches listed below can be used with any "CLI" command.

/L write "CLI" output to current listfile
<see CLI command #24 LISTFILE>
/L=filename write "CLI" output to file named filename
/Q set squeeze to on for this command
(compresses out all spaces possible)

NOTE: The /B (binary switch) has nothing to do with STATPAC or RASS files. Do not use it.

Each command has a listing of applicable switches. The information of switches for specific commands is available from the "HELP" facility. Some of these switches are very useful and a few of them are mentioned in the list of specific commands that follows.

EXPANDING COMMAND LINES

Parentheses ( ) can be used to expand command lines. The "WRITE" command can be used to see how the "CLI" will interpret the command.

Examples:

"DEL FILE(A B C)" will delete files named "FILEA", "FILEB", and "FILEC"

"(ACL TYPE) filename" will print the access list of the file and type the contents of "filename" on the terminal

"TY/V FILE(1 2 3).F77" will type the contents of files "FILE1.F77, "FILE2.F77", and "FILE3.F77" on the terminal.

SPECIFIC CLI COMMANDS

1. FILESTATUS (F)

Lists names of all files (file segments, directories, and links) in the current working directory.

EXAMPLES:

F/S	(lists the files sorted alphabetically)
F/AS	(lists an assortment of information about each file [i.e. size (in characters), date created and type of file]
F/AS/S	(combination of both of above switches
F/DLM	(lists files and dates the files were last modified)
F/S/AS #:filename+	(lists all files in current directory and all subdirectories that match the template "filename+")
F/TY=DIR	(lists all directory files)

The "FILESTATUS" command lists all types of files and in no apparent order (except when modified by a switch).

2. DELETE (DEL)

Deletes named files from working directory. A template can be used. "DEL +" is allowed and will delete ALL unprotected files (INCLUDING SUB-DIRECTORIES) in the working directory. "DEL/C +" will query the user about each file to be deleted. It is STRONGLY RECOMMENDED that the "/C" switch be used with this command. When the "/C" switch is used, enter "Y" to delete and <C.R.> to retain file.

3. TYPE (TY)

This command prints the contents of an ASCII file on the terminal. The default does not print the title. The printing always begins with line 1. More than one file name may be in the list and templates are allowed.

4. QPRINT (QPR)

This command sends referred files to the line printer. Templates are allowed, but should be used with caution. Common switches for this command are:

```
QPR/DEL        (deletes file after it is printed)
QPR/COPIES=3   (prints  three copies of the file)
```

Users located at remote sites may use procedures named "MCQPR" and "WRQPR". These macros invoke the "QPRINT" command, but the heading of the printout is changed to reflect the destination.

5. COPY (COPY)

Syntax is "COPY newfile oldfile oldfile2 ..."

The copy command copies the contents of the old files into the new file. It will not accept a template as a filename. After a file is copied with the "COPY" command, access to the new file must be set if there are others using this file. If a "COPY" command is issued on a file that you do not have access to, an empty file is created in your area. You must "DELETE" this file before re-issuing the "COPY" command. If several "oldfiles" are given, the contents of all these files are copied into the newfile.

6. MOVE (MOVE)

Syntax is
"MOVE destination:directory:name source-file"

Your current working directory should be the directory in which the source files are located. Templates are allowed for the source file. "MOVE" is usually used to move more than one file. The "MOVE" is not a "COPY/DELETE". The "DELETE" will have to be issued specifically.

EXAMPLE:

MOVE :UML:STATPAC:SOURCE +.F77
would move all files fitting template +.F77 in the current working directory to the directory :UML:STATPAC:SOURCE

7. CREATE (CRE)

The "CREATE" command is used to create files, directories, and links. To create a new subdirectory, the command is: "CRE/DIR dirname". To create a new file the command is "CRE filename". The create command can be used rather than using a text editor to create a file with character data. An example of this procedure follows:

```
CRE/I  FILE45
LINE1
LINE2
LINE3
)
```

The commands listed above are exactly what is typed. These lines do not show the prompts given by the computer. When "CREATE" is invoked with the "/I" switch, the computer prompts with a second ")". To close the file, a third ")" on a line by itself is entered.

The commands listed below show the above procedure again, but also show the "CLI" prompts.

```
)CRE/I  FILE45
))LINE1
))LINE2
))LINE3
)))
)
```

8. DATE

This command gives current date.

9. TIME

This command gives the time.

10. SPACE (SP)

This command will display the number of blocks of storage currently allocated and the number used. The command is applicable only to your username directory. When you receive the message "CONTROL POINT DIRECTORY EXCEEDED" the space allocated to you has been filled.

11. QDISPLAY (QDIS)

This command will display all pending requests in queues for various devices.

12. QCANCEL (QCAN)

The syntax is "QCAN sequence-number". This command will cancel any request you have submitted for a queue device.

13. PERMANENCE (PERM)

The syntax is "PERM filename [ON or OFF]". This command is used to protect files, directories, and links so they cannot be deleted. If "PERM" is "ON" for a file, the file cannot be deleted until the "PERM" attribute is turned "OFF". IT IS RECOMMENDED THAT YOU USE THIS FEATURE ON YOUR SUBDIRECTORIES SO YOU DO NOT DELETE THEM IN ERROR. If the command "PERM/V +" is used, the computer will display this attribute for all files.

14. RENAME (REN)

The syntax is "REN oldname newname"

15. EXECUTE (XEQ or X)

This is the execute command for a program file.

16. WHO

This command displays information regarding your process.

17. PED

This command will display all current processes on the Data General MV/6000 computer. This list shows what programs are being executed, so a user may have more than 1 or 2 entries listed.

18. WRITE

This command is used to print on the terminal what is happening in the execution of a MACRO. It can be used to see how the CLI will expand a command line where parentheses () are used or to print the results of a pseudo-macro operation.

EXAMPLE:

```
WRITE (TY QPR) FILE(A D).LIST
WRITE [!UADD,4,6]
```

19. DIRECTORY (DIR)

The "DIR" command used alone will display the name of the directory in which you are currently positioned (working directory). The "DIR" command with a pathname following will change your working directory to that directory. The "^" can be used in a "DIR" command to indicate moving to next most superior directory. To move to any directory, a relative pathname may be used. "DIR/I" will position you back to your username directory.

20. SEARCHLIST (SEA)

The "SEA" command used alone will display the current searchlist. The "SEA" command with a pathname following will change your searchlist to include that directory ONLY. It will not append that directory to the current searchlist, but will replace the current searchlist. The initial searchlist is set up in the "LOGON.CLI". The system searches the working directory then the searchlist directories from top to bottom. The system uses the searchlist for many "CLI" commands. It does not use it for "DEL", "F", "MOVE", "DUMP" or for MACROS. It does use the searchlist for the "TYPE" command.

21. ACL & DEFACL

The "ACL" command is used to display and set access to files. "ACL pathname" will display the current access list of file "pathname". (File "pathname" could be a directory file, as well as a data file.)

The privileges that can be given with the "ACL" command are as follows:

O	owner access - can change the ACL of file
W	write access - can delete and create files in directory - can modify a file
A	append access - can add files to a directory
R	read access - can print a file
E	execute access - can execute programs

Files with "+,RE" access can be read by everyone. A file you create has USERNAME,OWARE access.

When the "ACL" command is used to modify the access to a file, the access list is REPLACED with the access list you give it. If access is set on a file, then the containing directories must have appropriate access.

Examples:

```
ACL/V +              (lists current access on all files
                      in working directory)
ACL FILENAME JDOE,OWARE,MSMITH,WARE,+,RE
                     (changes access list to include
                      only those named)
```

The "DEFACL" command sets a default access control list for the directory. The "default access" is effective only for files that you create in the directory. It is not effective for files that are copied into the directory or files someone else creates in the directory. When a file is copied from one directory into another directory, the "ACL" on the new file is the same as the "ACL" on the original file. No one has access until specifically given access.

22. CHARACTERISTICS (CHAR)

The "CHAR" command sets and displays the characteristics for your terminal. Initial characteristics are set up for you in the "LOGON.CLI". The "CHAR" command used alone will display the current settings. If you change the characteristics with this command and find you do not like the result, "CHAR/PREVIOUS" will set characteristics to what they were prior to the "CHAR" command. "CHAR/RESET" will reset the switches to the system default. Some common switches are:

```
CHAR/ULC      upper/lower case
CHAR/CPL=80   line length
```

23. LOGFILE

The logfile logs your activity with the "CLI". It keeps an exact log of the "CLI" commands executed. This is helpful for user assistance if you feel the system is behaving strangely. The limitation of the "LOGFILE" command is that no program output is logged.

To issue this command type:

```
LOGFILE filename
```

Do not use the same logfile name every day. The system appends information to this file rather than overwrites it. Do not keep these files longer than one or two days. If you have not experienced problems during a session, there is no need to keep the logfile for that session.

24. LISTFILE

The "LISTFILE" command can be used to direct output from the system to a file. "LISTFILE" is useful only for the output from system commands. Program output cannot be directed to a listfile. This is useful to print listings of directories and to print output from the "HELP" commands on the line printer.

There are two ways of using this command. The first is to set up the listfile using the "LISTFILE" command. This would be used when output from several different commands will be directed to the same listfile. The output from each successive command will be appended to be listfile. The second way is to use the "/L" switch. Use this when the output from only one command is to be directed to the listfile. When the second method is used, the listfile is

opened and closed for that command. If another command is issued in that form with the same listfile name, the original file will be overwritten.

EXAMPLES for directing output to a file

1. 1st Method: Using "LISTFILE" command

Each command where "/L" is used appends that output to the listfile "file1".

```
DIR/I             (positions in  initial working
                   directory)
LISTFILE file1    (creates listfile named "file1")
F/AS/S/L #        (directs the output from the "F" command
                   to "file1")
DIR TDAT          (positions in subdirectory "TDAT")
TY/L file2        (types the contents of "file2" into
                   "file1")
HELP/V/L QPR      (appends the help information on
                   "QPRINT" to "file1")
LISTFILE/K        (closes listfile named "file1")
DIR/I             (positions in  initial working
                   directory)
QPR file1         (sends "file1" to line printer)
```

2. 2nd Method: Using "/L=file" switch

```
HELP/V/L=file3 QPR  (puts the ouput from the
                     "HELP QPRINT" command into file
                     named "file3".)

F/AS/S/L=file4      (puts the output from the
                     listing into file named "file4".)

QPR file(3 4)       (sends "file3" and "file4" to
                     line printer)
```

PROGRAMS

The "EXECUTE" ("XEQ") command is used to execute a program file (shown in directory as program_name.PR). This command can be abbreviated as "X". The syntax is "X program_name". All programs can be invoked this way. However, a "macro" has been created for each applications program so that it is not necessary to type "X program_name". All that is necessary is to type the program name.

The list of programs that are currently operative on the Data General computer is on-line and accessible to anyone. To obtain this list type:

QPR :UML:STATPAC:DOC:PROGRAMS

When most programs are invoked, the program name, a version number, and the last date the program was modified will be printed. Notice this information.

All problems or "bugs" you find should be reported to the Applications Software group. Send the hard copy of of the terminal output and clarifying remarks.

MACROS

"Macros" are usually created to execute a repetitive string of commands. Variables can be input to "macros" through the use of dummy arguments. "Macros" are created with a text editor or with the "CRE/I" command. User-created "macros" should not have the same name as system commands. The filename of a "macro" must end with the suffix ".CLI".

Arguments are sent to a "macro" on the command line when the "macro" is invoked. Their place is held in the macro with the following notation:

%1%	represents the first argument on the command line
%2%	represents the second argument on the command line
%-%	expands to all arguments from first to last
%1-%	same as %-%

There are many possible ways to indicate to the "macro" which arguments, ranges of arguments, or switches should be interpreted. These are detailed in the Data General CLI Manual. "Macro" is a topic in the "HELP" facility.

Pseudo-macros

"Pseudo-macros" can be used for conditional execution or to expand to a value. They are enclosed in brackets ([]) and begin with an exclamation point (!). They can be used in "macros" or at command level. There are many "pseudo-macros". They are explained in detail in the Data General CLI Manual. "Pseudo-macro" is a topic in the "HELP" facility.

To see some examples of "macros" and "pseudo-macros" type the following on your terminal:

```
TYPE/V  (LOGON EAR UC HEAD LOOKST PLOT_CALCOMP).CLI
```

BATCH

Batch processing on the Data General can be used to run a process in background or absentee mode and have the output captured in a file. This file can then be printed either on the lineprinter or your terminal.

The output from the batch processor may not be easy to read because the responses to program queries are always not displayed on the same line as the query itself.

A batch input file is a form of a "macro". The invocation of programs from a "macro" is different than the interactive invocation. The invocation of programs in a batch file is different yet. Interactively, programs are invoked with "X PROGRAM_NAME" or "PROGRAM_NAME". The invocation of the program from a "macro" MUST BE "X/M PROGRAM_NAME". In a batch job the invocation MUST BE "X/I PROGRAM_NAME". To read more about these switches you can type "HELP *I_SWITCH" and "HELP *M_SWITCH". For every "X/I" in a batch input file there MUST BE a corresponding ")" following the last response given for the program.

The batch input file must be created using a text editor or the "CRE/I" command. The batch input file should have the suffix ".ABSIN".

To submit a batch input file named "JOBNAME.ABSIN", type "EAR JOBNAME". A "macro" has been created to enter the batch job into the batch queue. The "macro" looks for a file with the suffix ".ABSIN" and creates a file suffixed ".ABSOUT".

To display the contents of the batch queue, use the command "QDIS". To cancel a batch process, use the command "QCAN seq.no.". "Seq.no." is the sequence number assigned to your batch job when it is entered to the queue. These commands are

the same as for the lineprinter queue. The batch facility is set up at this time to process only one job at a time. Processing begins immediately after the request is made if there are no prior requests.

The file suffixed ".ABSOUT" can be "typed" to the terminal or "qprinted". The file will contain a line near the end that reads "LIST FILE IS EMPTY AND WILL NOT BE PRINTED". This is a function of the batch system and should be ignored.

EXAMPLE

The system prompts and prompts from "CRE/I" are shown.

1. To create and process a batch input file for execution of program "CI"

```
)CRE/I CI.DEN.ABSIN          creates file named "CI.DEN.ABSIN"
))DIR :UDD:JDOE:SUBDIR1      first command in macro is to
                              position to appropriate
                              subdirectory
))X/I CI                     execute program "CI" with
                              "/I" switch
))CARDS.DEN                  response to program "CI" for
                              input file
))STAT.DEN                   response to program "CI" for
                              output file
))))                         ) is necessary as last
                              command in batch job
)))                          this ) terminates the
                              "CRE/I" command

)EAR CI.DEN                  EAR.CLI enters the batch job
                               in the queue
```

TEXT EDITORS

The two text editors on the Data General MV/6000 computer at this time are "SPEED" and "SED".

All editors on the Data General system have "paging". If a large character file is edited, paging will be encountered. A "page" in "SED" is 1023 lines (approximately 53,000 characters in "SPEED"). Most global operations take place only on the current page. Editing printer files (i.e. files that have embedded carriage control) is particularly cumbersome.

SED Text Editor

The "SED" text editor is designed for Data General video terminals. It can be used on hard-copy terminals, but some of the features are sacrificed.

"SED" is a line-oriented text editor. For anyone using "SED" over a modem (not hard-wired to computer) a "MACRO" has been created that invokes "SED" with some switches that make it easier to use. To invoke "SED" with the defaults, type "SED". To invoke "SED" with the switches type "SEDIT". Hereafter, in this text "SED" will be used to mean either "SED" or "SEDIT".

"SED" creates a special binary file with the suffix ".ED", which is used exclusively by "SED". This file contains any parameters that have been established in previous sessions, such as whether line numbers should be displayed. This file also remembers where the cursor was positioned. If this file exists when "SED" is invoked, the cursor will be positioned wherever it was when the previous edit session was ended. When "SED" is terminated, it asks if you want to make the input to that edit session a backup. If you respond with "Y" a file is created having a suffix of .BU. To invoke "SED" type "SED filename". The "SED" prompt is "*".

Some of the basic commands for "SED" follow:

LIST	will print current line
LIST 1 5	will print lines 1 - 5 (does not change cursor position)
LIST ALL	will print entire PAGE
DISPLAY	will print various information about the file
POS 25	will position cursor to line 25
POS LAST	will position cursor to last line on current page
POS PA 2	will position cursor to first line on page 2
DEL 20 30	will delete lines 20 - 30
DEL 10	will delete line 10
UNDO	restores text deleted in most recently issued "DEL" command

```
FIND word              will search to next occurrence of "word"
F                      will search to next occurrence of previous
                         search string
FIND word ALL PAGES    will search all pages, if necessary, to
                         find first occurrence of search string

BACKFIND string        will search from current position backwards

CLEAR LINE             will stop printing of line numbers

SET LINE_NUMBER_DISPLAY will start the printing of line numbers.

ABANDON                will abort from editor with no changes
                         written

BYE                    will save file with changes and
                         exit from editor

SAVE                   writes out a copy of the file without
                         exiting from "SED" to a file suffixed
                         with .SV
```

```
MOVE 50 LAST BEFORE 1      moves lines 50 - last before line 1
MOVE 1 AFTER LAST          moves line 1 to the end of the file
MOVE 1 25 ONTO filename    moves lines 1 - 25 into file named
                             filename (appends if file exists)
MOVE 25 BEFORE 10          moves line 25 before line 10


DUP 25 40 AFTER LAST       copies lines 25 - 40 to the end
                             of the file
DUP 50 LAST ONTO file1     copies lines 50 - last into
                             file named file1.


SUBSTITUTE newstring FOR oldstring   replaces oldstring by
                                       newstring in ALL
                                       occurrences of
                                       oldstring from current
                                       line to last line on page
SUB new F old IN CURRENT             replaces new for old on
                                       current line only
SUB new F old ALL PAGES              replaces new for old in all
                                       occurrences of old in
                                       all lines on all pages

HELP                     prints out information about commands

APPEND <C.R.>            appends to end of the file

INSERT <C.R.>            inserts above current line
INSERT 15                inserts above line 15
```

INSERT FROM filename	inserts above current line data from filename
INSERT 10 from filename	inserts text from filename before line 10
DO	allows execution of a CLI command
DO F	types a listing of all the files in working directory without exiting from "SED"
DO SED	allows editing of a second file without exiting the first edit session
SPLIT	sets a page break -- will not have effect of a form feed to the line printer if "SEDIT" instead of "SED" is used.
JOIN	removes a page break
MOD	for Data General terminals only uses four arrow keys to edit text on current line

Once "APPEND" or "INSERT" have been issued, you are in that mode until "SED" receives the terminating character <ESC> on a line by itself.

The destination for the "MOVE" or "DUP" command has to be within the current page or be an external file.

The "FIND" command will only find the first occurrence. "FIND string ALL PAGES" is not a proper command. Use the command "SUB old FOR old ALL PAGES" and "SED" will print all lines where the "substitution" occurred.

In a "MOVE" command the cursor is positioned after the moved text. In an "INSERT" command the cursor is positioned after the last inserted line.

The "MOVE", "DUP", and "INSERT" commands can be very lengthy command strings. The syntax for them may require or allow an address, a range, a source, or a destination. Key words are used for some of these required items. The address is usually a number. A number alone indicates a line number. If a page number is intended the form is "PA 3". The "range" can be a pair of line numbers, beginning and ending. The words "ALL", "REMAINING", "CURRENT", "TO", or "FOR" can be used to specify a range of text. The "source" and "destination" can be a line number or a filename. The words "TO", "ONTO", "BEFORE", "AFTER" are used with the "destination."

To "JOIN" the cursor must be positioned on the first line

of the second page of the two that will be joined.

For the "FIND" or "SUBSTITUTE" commands the character strings must be in quotes if they contain any spaces, special characters, or lowercase letters.

SPEED Text Editor

The "SPEED" text editor is a character-oriented text editor. Spaces and <C.R.> are valid characters. "SPEED" creates a special file suffixed with ".TM" which is used exclusively by "SPEED".

The "L" and "M" commands change the position of the cursor. The "T" command types a line, but doesn't change the position of the cursor. "SPEED" requires a "1" in front of the "L", "T" and "K" commands. If the "1" is not used with the "T" command, "SPEED" prints a special character (^) denoting the position of the cursor. This makes locating the cursor position easier, but defining column positions more difficult.

The sentinal character in "SPEED" is the "<CTRL D>". In "SPEED" there are no delimiters for the "search" and "insert" commands. "SPEED" assumes there will be a character string after the "S" or "I" command. The flag to indicate the end of the string for both of these commands is the "<ESC>".

Command Action	SPEED Data General
	$$ = <CTRL D> $ = <ESC>
move to top of buffer & print entire buffer	J#T$$
move to line 5 and print that line only	5J1T$$
print current line	L1T$$
position to & print next line	1L1T$$
position to & print previous line	-1L1T$$
insert character string and print current line	Ichar-string$L1T$$

go down 15 lines and print that line	15L1T$$
search for character delete next character and print current line	Schar$1DL1T$$
search and replace	Cstr1$str2$L1T$$
move to right 46 characters, delete 2 chars, insert "23" and print current line	46M2DI23$0L1T$$
kill current line	1K$$
kill 10 lines	10K$$
write out file with changes	FUH$$
abort from edit session	H$$
search to character string, kill remainder of line	Sphrase$1KI<C.R.> $$$
print all occurrences of "phrase" in file	<Sphrase$;0L1T1L>$$
move 20 characters to right, kill rest of line, on each line in the file	VN<20M1KI<C.R.> $>$$
make change on every other line in file (start with line 2)	<1LCstr1$str2$;1L;>$$
insert new line before every occurrence of "str1"	<Cstr1$str2<C.R.> str1$;>$$

OTHER COMMANDS

```
.=$$   display current character number
VN=$$  display number of lines on that page
VL=$$  display current line number
```

ITERATION

An iteration loop in "SPEED" will cycle and loop forever unless the command directs it to stop. The number of times to execute, the number of lines in the file, or a semi-colon (;) in the command string will tell "SPEED" to stop iteration. The semi-colon tells speed to stop iteration when it reaches the end of the page (file). The < and > are the iteration delimiters used.

BUFFERS

Buffers can be used to make copies of particular lines within a file or to move lines within a file. When a buffer is re-used, the original information is overwritten.

To store one line in memeory and then recall it somewhere else:

1. Cursor positioned on the line.
2. To store 1 line in buffer "8" type:
   1BC8$$ (BC = buffer copy)
   OR 1BT8$$ (BT = buffer take - deletes from current position)
3. Position cursor to line where lines from buffer are to be inserted.
4. To recall buffer "8" type:
   I<CTRL B>8$$

Example:

To make a copy of each line in file above original line:

```
VN<1BC8I<CTRL B>8$1L>$$
```

PAGING

1. Need to position cursor from page to page with "R" or "Y" command.

2. Once a page is "YANK"ed, the cursor cannot be positioned on a previous page.

3. All activities (search, type, delete) apply only to current page.

4. "N" is command to search across page boundaries (nonstop search).

5. If "200K" (kill 200 lines) is typed, and 200 lines do not exist,
   there is no error message.

6. If a user tries to "SPEED" a large file, the message "insufficient memory available" is printed. The procedure to use to "SPEED" the file is given below.

```
SPEED
FRname_of_file$$          (opens file)
100WM$$                   (sets window for 100 lines
                             to be read into buffer)
R$$                       (reads 1st group of 100 lines
                             into buffer)
FWoutput_file$$           (creates output file)
R$$                       (writes 1st 100 lines to output file,
                             empties buffer, and reads in next
                             100 lines)
...                       (etc.)
FC$$                      (closes all files)
H$$
YES
```

MERGE FILES IN SPEED

Merge file1 and file2 in file1:

```
SPEED file1           (opens file1)
I<CTRL>Ffile2$$$      (inserts information from file2)
FUH$$                 (writes out merged file into file1
```

MERGE FILES IN SPEED

Merge file1 and file2 to form file3:

```
SPEED
FRFILE1$$          (opens file1)
A$$                (writes contents of file1 into buffer)
FC$$               (closes file1)
FRFILE2$$          (opens file2)
A$$                (writes contents of file2 into buffer)
FC$$               (closes file2)
FWFILE3$$          (creates file3)
P$$                (writes all of buffer into file3)
H$$
YES
```

SPLIT FILES IN SPEED

```
SPEED
FRfilename$$       (opens filename)
500WM$$            (sets window to 500 lines)
R$$                (reads in first 500 lines)
FWout.1$$          (creates file named "out.1")
1LSXXX$L1T$$       (search for "XXX")
0,.PW$$            (writes from beginning to cursor into
                     "out.1")
0,.K$$             (kills from beginning to cursor)
1LS$L1T$$          (searches for next occurrence of "XXX")
                    -- search string not found in this example
FNWout.2$$         (opens second output file "out.2")
0,ZPW$$            (writes rest of buffer to "out.2")
0,ZK$$             (kills rest of buffer)
R$$                (reads in next 500 lines)
S$L1T$$            (search for next occurence of "XXX")
0,.PW$$            (writes from beginning of buffer to cursor
                     -- appends to "out.2")
0,.K$$             (kills beginning to cursor)
....               (etc.)
FC$$               (closes all files)
H$$
YES
```

COMMENTS

1. Once a search string is typed, only "S<ESC>" is required for subsequent searches.

2. Don't use a <LINE FEED> character in speed. Always use the <CARRIAGE RETURN> character.