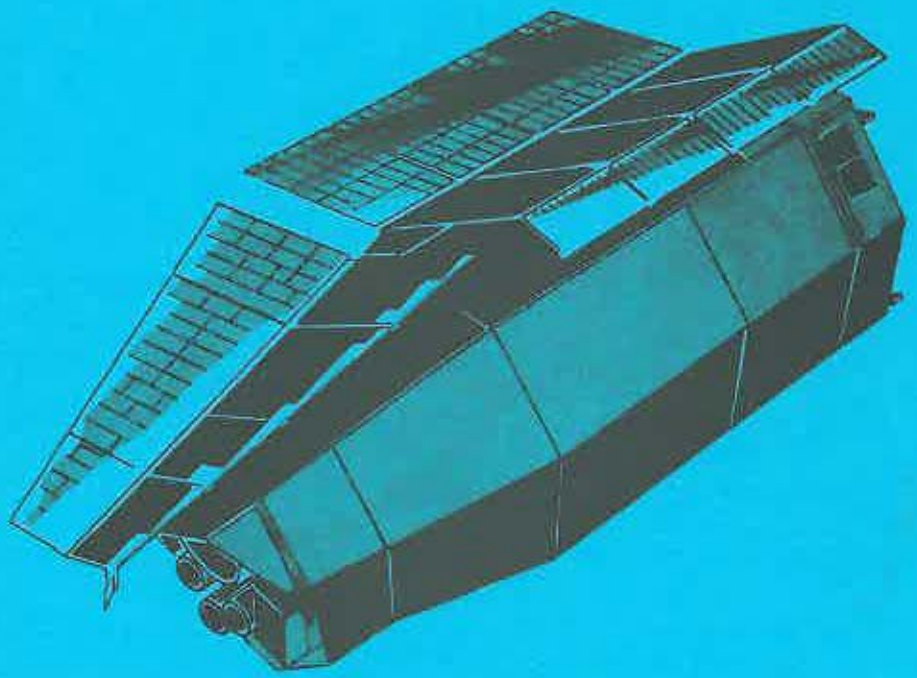
EINSTEIN OBSERVATORY
REVISED USER'S MANUAL



HARVARD-SMITHSONIAN CENTER FOR ASTROPHYSICS

EINSTEIN OBSERVATORY

REVISED USER'S MANUAL

JULY 1984

D.E. Harris, Editor
D. Irwin, Editorial Assistant

EO REVISED USER'S MANUAL

Table of Contents

Chapter 1: INTRODUCTION

1 March 1984

Contents of Chapter I

1.1 History

The Einstein Observatory (known as HEAO-B before launch) was placed in near earth orbit on 13 Nov 1978. The mirror system produced an X-ray image at the focal place with a scale of 1 mm = 1 arcmin. Details concerning the mirror may be found in an article by Van Speybroeck (SPIE, Vol 184, Space Optics, p2, 1979). The focal plane detectors included two Imaging Proportional Counters (IPC) with angular resolution of ~ 1.5 arcmin and energy resolution of $\Delta E/E \sim 1$; three High Resolution Imagers (HRI) with angular resolution ~ 4 arcsec; an Objective Grating Spectrometer (OGS); a Focal Plane Crystal Spectrometer (FPCS); and a Solid State Spectrometer (SSS). The SSS was operated by Goddard Space Flight Center and the FPCS was managed by MIT; neither of these instruments will be discussed in this manual. The Monitor Proportional Counter (MPC) was a separate detector co-aligned with the mirror pointing direction.

Initial Testing

During Nov and Dec 1978 the various focal plane instruments were tested and scientific observations were started. A high background splotch was found on one of the HRIs ("HRI-2"), so that detector was no longer used. "HRI-1" was never used (it had a thick window) and thus "the" HRI for most of the mission was "HRI-3". Only one IPC was used; the other was known from pre-flight tests to be inferior with bad local distortions. It served as a back-up, but was never needed.

JAN 1979 - AUG 1980

This period represents the bulk of the EO mission. In Jan 1980, the gas flow into the IPC was reconfigured in order to use the porous plug in the back-up IPC to recover the desired controlled leakage. This reduced the extreme temperature sensitivity of the IPC gain, and obviated the need for high voltage adjustments during IPC observations.

AUG 1980 - DEC 1980

During a recovery from a "Lost" state, one of the satellite's gyros did not restart. There ensued three months of various attempts to regain an operational status. Early in December this effort was successful with the proper functioning of the needed gyro. However, a large amount of attitude control gas was expended during this period; and only limited scientific data were obtained.

DEC 1980 - APRIL 1981

From recovery until 25 April when the depletion of attitude control gas terminated the mission, the EO was fully operational although target selection was constrained somewhat in order to minimize gas usage.

Data Reduction and Documentation

A data reduction pipeline produced images and other information with a 6 - 16 week delay. The EO User's Manual was revised in July 1980. After the mission terminated, it was decided to reprocess the entire data stream, since much higher quality could be obtained from the experience gained in the first pass. Since the HRI data were less complex than those from the IPC, the HRI ("REV1") production was done first (Jan - Oct 1982). Starting in May 1983 and continuing until late 1984, is the "REV1 IPC" effort.* Because we anticipate a significant number of new users in connection with the EO Data Bank, and because a large fraction of the software for IPC REV1 is entirely new, it was decided to write this Revised User's Manual (RUM).

1.2 Guest Observers

Users of the CFA reduction facilities should be aware that all the HRI and IPC data are processed in a standard manner and a great deal of information is presented in the resulting hard copy ("PRD"). A considerable effort was made to document the results in the actual PRD. However, more detailed descriptions of the PRDs are given in chapters 4 and 5.

The off-line software for further data analysis has been substantially expanded and revised. Chapter 8 contains the documentation for the programs used most frequently. There is also a COOKBOOK available at CFA which is organized in a topical way to help you decide which programs are useful to obtain particular results. Finally, the Guest Observer should be familiar with the information presented in Chapter 2 of this document which explains the structure and use of the EO computer system.

1.3 CFA Scientific Staff

*Concurrently, the MPC data are being reprocessed since the behavior of the background count rate is now better understood.

associated with EO data reduction.

	Scientific Interests	Technical Expertise
ELVIS, Martin	Active Galactic Nuclei, Seyferts	Unification of off-line scientific software
FABBIANO, Pepi	Normal, Peculiar and Radio Galaxies	Management of Production Processing
FABRICANT, Dan	Clusters of Galaxies	IPCs (EO and future)
FORMAN, Bill	Clusters of Galaxies	Extended source analysis
GIOIA, Isabella	AGN; Medium Survey; radio galaxies	Documentation of Rev1 IPC Software specifications
GOLUB, Leon	Stars	
GORENSTEIN, Paul	SNR, Clusters	IPCs (EO and future)
GRINDLAY, Josh	Globular Clusters and galactic sources	MPC
HARNDEN, Rick	Pulsars; SNR; stars	IPC spectral properties; timing analysis
HARRIS, Dan	Non-thermal processes; radio sources and clusters of galaxies	User's Manual; Cookbook; GO program
JONES, Christine	Clusters of Galaxies	HRI calibration; extended source analysis
MACCACARO, Tommaso	Medium surveys, quasars; BL Lacs	MDETECT for Rev1
MURRAY, Steve	Deep Survey	
PRIMINI, Frank	Deep Survey; AGN	Rev1 software Evaluation
SCHWARTZ, Dan	BL LAC and other AGN	
SCHWARZ, Joe	Early type galaxies;	Management of EO

	clusters of galaxies	computers and software group
SEWARD, Fred	SNR and pulsars	Management of GO program; OGS calibration and software
TANANBAUM, Harvey	Quasars	
TRINCHIERI, Ginevra	galaxies	MPC Rev1 processing
VAN SPEYBROECK, Leon	normal galaxies	Program development for off-line analysis; telescope design.

1.4 Updating Procedure, LOG, and pagination

Some care has been exercised to allow updating of this document. Whether this will be done in units of chapters or in smaller sections will be decided when necessary. Users will find the date of last revision in the Table of Contents. For revisions of entire chapters, it will also appear on the first page of that chapter. Any further details will appear here, in chronological order.

Revised sections will be available at CFA on request.

What might appear as a rather cumbersome pagination scheme has been devised to accommodate updates and to make it easy to find sections. The page number will be the last (sub)section number which appears on that page. However, >4 decimal subsections (e.g. 8.3.2.1) will be ignored when selecting page numbers. For pages without new subsections, the page number will be that of the preceeding page, followed by a slash and a running serial number (e.g. 8.3.2/2).

Figures generally have numbers which associate them with given text sections, as do "Listings".

CHAPTER TWO: The EO Computer System

1 March 1984

2.0 PURPOSE AND GLOSSARY

This chapter is intended to provide new users with a rough knowledge of the internal structure of the Einstein M600 and enough information to run programs described in Chapter 8. More extensive documentation is available in the computer room for users with more advanced experience or needs.

For convenience, we list here a glossary of terms used by Data General or invented for our own operating system.

ACL	access control list - a file attribute to control which user(s) have access to the file
BATCH	batch streams are available for users to run jobs "away" from the console
CLI	command line interpreter - Data General's program for user interface with the operating system is waiting for input when you see the)
CLEAN	Einstein program to delete unused files
DIRECTORY	A file which contains files...a local disk storage area
FILE	a unit of disk storage containing data or software
MACRO	a series of CLI commands stored in a file...readily executed by the CLI
PATHNAME	a chain of directory names which shows the location of a file on the disk
PERMANENCE	a file attribute which renders it immune from deletion (under some conditions -- see CLEAN section 2.3.4)
SEARCHLIST	list of directories to be scanned if access to a particular file is desired and the file is not found in the current working directory

2.1 SYSTEM STRUCTURE

2.1.1 Hardware Configuration

The Einstein M600 User Computer System is designed to support three types of activity: scientific software development, SAO data analysis, and Guest Observer data analysis. To achieve this aim, the hardware on the M600 is configured as follows:

a. 3 200-megabyte Zebra disk drives arranged as follows:
- System disk
- User disk
- Data disk

b. 15 terminals distributed as follows:
- 5 computer room terminals for general use
- 1 t.v. room terminal

- 1 measuring engine terminal
- 6 programming terminals
- 1 300-baud modem terminal line
- 1 1200-baud modem terminal line

c. 3 dual-density (800/1600 b.p.i.) tape drives

d. 2 Lexidata 3400 image display systems (HIP and HIP34)

e. 1 Versatec V-80 printer/plotter

2.1.2 Command Line Interpreter (CLI)

There are a number of concepts worth reviewing concerning the M600 for those not familiar with Data General computers.

The Command Line Interpreter, or CLI, is the program which you will use to communicate with the Data General operating system. The CLI allows you to perform a host of functions such as create, delete, or display files, or execute programs.

The CLI is an interactive program which is active whenever you are logged onto the system. It is recognizable by the "prompt" character). Whenever you see this prompt, the CLI is waiting for you to give a command. For example:

```
) TIME
```

TIME is a CLI command which requests the system to display the current time. The CLI contains approximately 100 commands, some of which are documented in section 2.2. For a more complete list of CLI commands, refer to the Data General CLI manual.

2.1.3 Directories and Pathnames

The M600 uses its 3 200-megabyte disks to store files, i.e. collections of information. These files are contained in directories, which are themselves actually files which contain and catalog information used to access other files. One can think of files stored in these directories in the same way that papers are stored in a file cabinet.

The system directory structure can be thought of as an "inverted tree", with the root at the top and the branches extending downward. The "root", in this case, is the directory named simply ":", while the "branches" are the subdirectories of : and their associated files.

Every file on the system has a unique "pathname". A pathname is a chain of directory names which are followed to a particular file. Remember that a directory is also a file, and it may be specified by reference to its pathname. The pathname shows where a file (or directory) exists in reference

to the overall "inverted tree" directory structure.

You can use the CLI to make your way around this inverted tree directory structure. The first concept to understand is the idea of being "in" a directory. The CLI always has some "working directory", or a directory to which you have local access. To display your current working directory, use the CLI command:

```
) DIR
```

The pathname of your current working directory is displayed.

The first directory you will encounter is your "user" directory. This will be a space on the disk reserved for you. This user directory will in turn be contained in the "user directory directory", or UDD. UDD itself is a sub-directory of the root directory. Thus, for example, if your username is SAM, your user directory pathname may be specified as :UDD:SAM. Your user directory is always initialized as your working directory when you log on.

The files in your working directory are ready for access, meaning that you do not have to specify full pathnames to identify them. For example, if the :UDD:SAM directory were your working directory, the files contained in it could be referenced directly using just the filenames. You may obtain a list of files in your working directory using the CLI FILESTATUS command (abbreviated "F"). Thus, just entering "F" will list all files to which you have local access.

Files in your working directory may be accessed using templates. Templates are special characters which tell the CLI which filenames you wish to specify. For example, to search for all files ending in ".IMG", you would type:

```
) F +.IMG
```

The template, in this case is the "+". This tells the CLI to match any character string to the "+". The F command will list only filenames which match this template. The F command with no arguments assumes the "+" template alone. Thus the commands "F +" and "F" will perform the same function. Other templates exist:

-	(hyphen) match any character string without periods
*	(asterisk) match any single character

You may also use the DIR command to change your working directory, by supplying an argument specifying the desired working directory. For example, if you wish to move to another user's directory whose username is SPADE you would type:

```
) DIR :UDD:SPADE
) DIR
 :UDD:SPADE
```

Here, the DIR command was used twice to change and display the working directory.

2.1.4 The SEARCHLIST

Another essential concept is that of the "searchlist". The CLI has a feature called the searchlist, which is a list of directories which are scanned when access to a file is desired, and the file does not exist in the current working directory. The searchlist is initialized to include two special directories when you log on:

:UTIL - contains generally used system software
:HUTIL - contains generally used HEAO software

You may change your searchlist when you wish to access files that are not contained in your current directory. The CLI SEARCHLIST command (abbrev. SEA) is used to display or change the current searchlist. (Note: all commands may be abbreviated as long as the abbreviation is unique.)

Scenario: you have :UDD:SAM as your working directory. Your searchlist is set to :HUTIL :UTIL. You wish to access files in another directory, say :UDD:SPADE, without changing your working directory. If you set your searchlist to include :UDD:SPADE the CLI will be able to access any file contained there. Suppose a file named FOO exists in :UDD:SPADE.

```
) SEA :UDD:SPADE :HUTIL :UTIL     - set the searchlist
) PATH FOO
  :UDD:SPADE:FOO                  - find the file FOO
```

As you can see, the file FOO was accessed by name only, without specifying its full pathname. Remember that the CLI searches your working directory first, before checking the directories contained in the searchlist. More examples of searchlist usage will follow as we explore the M600 system structure.

2.1.5 M600 Disk configuration

The M600 system is made up of three 200-megabyte disks, each having a special function:

System disk - contains system software, utility programs, data bases
User disk - contains user directories for user software
Data disk - contains data directories for user data

2.1.5.1 SYSTEM disk

The SYSTEM disk is the most important of the three M600 disks, because it contains all the files needed to run the operating system. Also contained on the system disk are the utility directories, :UTIL and :HUTIL. These

utility directories contain often used macros and programs, and are by default included in all user's searchlists. Data bases and program archives are also stored in directories on the system disk. This disk is intended to be primarily a "read only" disk, as apparent by the nature of the files stored there. Some important directories contained on the system disk:

:UTIL	general system utility software
:HUTIL	HEAOB utility software, scientific programs
:DB	data base storage directory
:HSOURCE	program source archive directory
:COMPILER	high-level language compilers (F5,F77,PASCAL)

These directories are READ ONLY areas, to be modified ONLY through standard archival procedures.

2.1.5.2 USER disk

The USER disk contains the directory UDD (or the user directory directory). This directory contains all of the user directories for everyone with an M600 account. When you log on to the M600, the system initializes your working directory to be your user directory. Thus, if you are user SAM, your initial working directory will always be :UDD:SAM.

User directories are provided to CFA users and Guests for the development of software. Each CFA and Guest user directory is presently alloted a maximum of 5000 disk blocks (512 data bytes per block). These directories are backed up by copying them to magnetic tape weekly, but users may wish to do their own tape backups from time to time. (Data aides are available to instruct users in the use of tape drives.)

User directories should be kept as clean as possible at all times. If there are seldom used files in your user directory, they should not be kept on the disk, as a full disk will impact system performance. You may check your disk space usage by using the CLI SPACE command. Just issue the command SPACE in your user directory. The maximum, current, and remaining disk blocks in the directory will be displayed. If a directory has grown to its maximum size, this error will be reported:

CONTROL POINT DIRECTORY MAX SIZE EXCEEDED

To correct this situation, some file(s) will have to be deleted. Use the DELETE command to delete files:

) DELETE/V <file templates>

2.1.5.3 DATA disk

The DATA disk is the area of the system which is used to store user data. Data files should be stored here, NOT ON THE USER DISK. The data disk is known by name as Z3, and its pathname is :Z3.

There are three subdirectories of Z3, each with a special purpose. Their pathnames are:

:Z3:IMAGES	image data directory
:Z3:HBDATA	user's data directories
:Z3:SPEC	reserved for special processing

The IMAGES directory contains two subdirectories, REV0 and REV1. These, in turn, contain imaging data for two different processing revisions of the Einstein system. Usually, REV0 will contain old versions of image data and REV1 will contain the most recent versions. The directories IMAGES:REV0 and IMAGES:REV1 contain numerous subdirectories, named for the observation sequence number corresponding to the data contained within them. For example, to access the image file for the IPC revision 1 sequence number 9004, the complete pathname would be:

:Z3:IMAGES:REV1:9004:I9004.IMG

Using this full pathname, you could access this file from any working directory on the system. A better way to access an image file is by adding the directory containing the file, :Z3:IMAGES:REV1:9004, to your searchlist.

No user is allowed to write files into the IMAGES directory. Users should make image LOAD requests for image data needs. These image LOAD requests are filled daily by the data aide staff. This restriction has been made to insure that definitive image files remain definitive. Users should refrain from trying to modify files in IMAGES in any way, either by changing existing files or by adding new ones. Please keep non-definitive data files in other data directories.

The HBDATA directory is a directory containing user's data directories. CFA users have an HBDATA directory of their own (limited to 5000 blocks). This is where you should keep currently used data files. For example, user CFA will have a directory :Z3:HBDATA:CFA to match his :UDD:CFA directory. These two directories, totalling a maximum of 10,000 disk blocks per user, comprise a user's "private space" on the M600.

Data directories in HBDATA are not normally backed up. If users wish to save data files from their HBDATA directory they should make their own tape backups. This is especially true in the case of files which will not be accessed for several days. Such files may be purged from the disk by the CLEAN program, which searches nightly for unaccessed files on the data disk. See section 2.3.4 for details on CLEAN.

Also in HBDATA, there exists a large data directory, called EINSTEIN, where any user may store data on a temporary basis. This is for users who need more than the 5000 blocks in their own data directory.

The SPEC directory is not for use by general M600 users. It is space reserved for special processing, mostly of MPC data. Please do not attempt to create files in the SPEC directory.

2.1.6 Access Control Lists (ACL)

Each file on the Einstein system has an associated Access Control List, or ACL. This ACL is used to specify which users will have access to the file. The CLI command ACL is used to set or display a file's ACL. To display ACLs:

) ACL/V <file templates>

This command will display all filenames matching the templates with their ACLs. To set a file ACL:

) ACL <file template> <access list>

The access list for ACLs has this format:

user,access,user,access....

There are five access codes which may be assigned to an ACL:

O - the user has "ownership" of the file, meaning that he/she may change the ACL or delete file

W - the user may write to the file, that is, change its contents

A - "append" access, meaningful for directory files only, means the user may create files in the directory

R - user has read access to the file, or, if it is a directory file, examine filename list in the directory (via F command)

E - user may execute the file, or, if it is a directory file, use the filename in a pathname

For example, if we are to set the ACL of a file such that user CFA has all access types:

) ACL FILE CFA OWARE

To set the ACL such that user CFA has all access types, and all other users have read and execute access:

) ACL FILE CFA OWARE + RE

The + template will match all usernames. If ACLs are not set to allow you

the type of access you desire, you will get one of these error messages:

```
FILE ACCESS DENIED
DIRECTORY ACCESS DENIED
READ ACCESS DENIED
WRITE ACCESS DENIED
```

If you are having file access problems, you should check the ACLs of all files you are trying to access.

Each CLI has a Default ACL assigned to it. This DEFACL is set when each user logs on to <username> OWARE + RE. The DEFACL is applied to files which are created , that is, new files will carry the ACL defined by the DEFACL. To display the DEFACL, just type DEFACL. To change it, use arguments to define the DEFACL you want. Files in your user directory should always have a + RE in their ACL, since otherwise they will not be readable by the tape backup routine, and hence will not be backed up.

2.1.7 HELP files

The HELP command exists to enable quick access to documentation on the various commands. Just type HELP to obtain a list of topics, after which you can type HELP <topic> to see the desired topic. The command:

```
) HELP *COMMANDS
```

will list all CLI commands available. You can also type HELP with a command as argument to display help information about that command. Try it!

2.1.8 CLI macros

A CLI macro is simply a file which contains a list of CLI commands to be performed. Usually CLI macro files have the suffix .CLI on their filenames. CLI macros are executed by typing the name of the macro, without its CLI suffix. Macros (or any other text file) may be displayed by using the TYPE command. For example, if we have a macro called FOO.CLI, we could display it:

```
) TYPE FOO.CLI

TIME
DATE
DIR
SEARCHLIST
```

This is a simple macro to display the time, date, directory, and searchlist. To execute FOO.CLI, just type FOO. The commands in FOO.CLI will be executed by the CLI, and a prompt will be returned when FOO is finished. You can

write your own more useful macros very easily, using the CREATE command. The CREATE command takes as an argument the name of the file you wish to create. With a /I switch you may enter text directly to the new file as it is created. This is done in the following way:

```
) CREATE/I FOO.CLI
))                              - a double prompt is displayed
)) TIME                         - you type the text for FOO.CLI
)) DATE
)) DIR
)) )                            - enter a ) when finished
)                               - normal prompt is returned
```

You have just created a macro which is ready for execution. The editor should be used to create more complex macros.

There are a number of macros which are regularly used on the Einstein computer system. A list of standard Einstein macros and their usage follows in section 2.2.

2.2 USEFUL MACROS and COMMANDS

2.2.1 FILE LISTING COMMANDS

```
**********************************************************************************
Command: FILESTATUS ( F )
Call:
        F <file templates>
Description:
        Obtain list of files in working directory matching templates.
Switches:
        Details on available switches may be obtained using "HELP/V F".

**********************************************************************************
Macro:  FD - Display Directories
Call:
        FD [<file1>,<file2>,...]
Description:
        This command worked identically to the CLI "FILESTATUS" command,
        except that it finds only directories.  It will normally display
        the file type, creation time, creation date, and file length.

**********************************************************************************
Macro:  FND -  Display Non-Directory files
```

```
Call:
        FND [<file1>,<file2>,...]
Description:
        This command worked identically to the CLI "FILESTATUS" command,
        except that it finds files which are NOT directories.
**************************************************************************
Macro:  IMF - display the contents of a specified IMAGES directory
Call:
        IMF <sequence number> [<file1>,<file2>,...]
Description:
        This command works like the FILESTATUS command except that it
        displays the FILESTATUS for the specified files in the IMAGES
        sequence number directory.  Templates may be used for filenames.
**************************************************************************
```

2.2.2 Directory Change/Display Commands

```
**************************************************************************
Command: DIRECTORY ( DIR )
Call:
        DIR <filename>
Description:
        Without argument, displays current working directory.
        With argument, changes current working directory as specified.

        For more details use "HELP/V DIR".
**************************************************************************
Macro:  DDIR - Puts the user into a Data Directory
Call:
        DDIR [<user>]
Description:
        The data directory corresponding to the specified user is made
        the current working directory.  If no user is specified then
        the user's own data directory is assumed.

**************************************************************************
Macro:  UDIR - Puts the User into a Development (user) directory
Call:
        UDIR ([user])
Description:
        The development (UDD) directory corresponding to the specified
        user is made the user's current working directory.  If no user
        is specified then the user's initial directory (:UDD:<user>) is
        assumed.  Note: DIR/I will cause the user's "log-on" directory
        to become his/her current directory.

**************************************************************************
```

```
Macro:  IMDIR - Puts the user into an IMAGES directory
Call:
        IMDIR <sequence number>
        IMDIR/0 <sequence number>
        IMDIR/1 <sequence number>
Description:
        The IMAGES directory corresponding to the specified sequence
        number is made the user's current working directory.  The REV0
        IMAGES directory is assumed if no switch is present, and the REV1
        IMAGES directory will is assumed if IMDIR/1 is specified.  If no
        sequence number is specified then IMAGES:REV0 or IMAGES:REV1 will
        become the current directory (depending on the switch value).
        NB: The IMAGES directory is a READ ONLY data base.  Any attempt to
            write into it will result in the error WRITE ACCESS DENIED.
```

****

2.2.3 File Transfer Commands

```
Command: MOVE
Call:
        MOVE <destination directory> <file templates>
Description:
        The MOVE command moves copies of files from a working directory
        to another directory, as specified by the first argument.  The
        second and subsequent argument(s) specify the file templates
        which will be moved.  No second argument assumes that all files
        in and subordinate to the current working directory will be moved.
        Use this command with care!

        More details may be obtained via "HELP/V MOVE".
```

****

```
Macro:  DMOVE - Move files from current directory into a data directory
Call:
        DMOVE <user> [<file1>,<file2>,...]
Description:
        This command moves copies of the specified files from the current
        working directory to the specified user's data directory.  File
        templates may be used.
```

****

```
Macro:  UMOVE - Move files from current directory into a user directory
Call:
        UMOVE <user> [<file1>,<file2>,...]
Description:
        This command moves copies of the specified files from the current
```

working directory to the specified user's user directory. File templates may be used.

****

Macro: GET - Get files from specified directory to current directory
Call:
GET <directory name> [<file1>,<file2>,...]
Description:
This command moves copies of the specified files from the directory specified to the current working directory. File templates may be used.

****

Macro: DGET - Get files from specified data directory to current directory
Call:
DGET <user> [<file1>,<file2>,...]
Description:
This command moves copies of the specified files from the data directory corresponding to the user specified to the current working directory. File templates are allowed.

****

2.2.4 Searchlist Change/Display Commands

****

Command: SEARCHLIST (SEA)
Call:
SEARCHLIST <directory names>
Description:
Without arguments, displays the current searchlist.
With arguments, sets the searchlist as specified. The directory names specified must pertain to directories which exist on the disk.
For more details, use "HELP/V SEA".

****

Macro: DSEA - sets the current SEARCHLIST to include the specified data directory
Call:
DSEA <user>
Description:
Adds the :HBDATA:<user> directory to the current searchlist. This is useful when a user wants to run programs from a development directory and access data files which are in a data directory. To restore the previous SEARCHLIST use the CLI "POP" command.

****

Macro: USEA - sets the current SEARCHLIST to include the specified user directory
Call:

```
        USEA <user>
Description:
        Adds the :UDD:<user> directory to the current searchlist.
        This is useful when a user wants to run programs from a data
        directory and access program files which are in a user directory.
        To restore the previous SEARCHLIST use the CLI "POP" command.
**********************************************************************
Macro:  IMSEA - sets the current "SEARCHLIST" to include the specified IMAGES
                directory
Call:
        IMSEA</0,/1> <sequence>
Description:
        This is useful when a user wants to run programs from a directory
        and access data files which are in IMAGES.  IMAGES:REV0 or IMAGES:REV1
        are specified by the switch value.  The REV0 directory is assumed if no
        switch is present.
        To restore the previous searchlist use the CLI "POP" command.
**********************************************************************
Macro:  ADDSEA - Set current SEARCHLIST to include the specified directories
Call:
        ADDSEA <dir1,dir2,...>
Description:
        This is useful when a user wants to add directories to his/her
        searchlist.  To restore previous searchlist, use the CLI "POP"
        command. (This should be done as a matter of course when the
        user is finished "using" the added directories.)
        Note: Maximum number of directories allowed in a searchlist: 8
**********************************************************************
Macro:  HSEA - Reset the SEARCHLIST to the default HEAO-B SEARCHLIST
Call:
        HSEA
Description:
        Resets the searchlist to the logon default of :HUTIL,:UTIL,:PER
**********************************************************************
```

2.2.5 Miscellaneous Macros/Commands

```
**********************************************************************
Command: BYE - Log off the system
Call:
        BYE
Description:
        This command will terminate your CLI and log you off the system.
**********************************************************************
Command: CREATE - (CR) create a file
Call:
        CREATE <filename>
Description:
        This command will create a file as specified.  Only one argument
        is allowed, and it must be a full filename (no templates).
        For details on CREATE options, use "HELP/V CREATE".
**********************************************************************
```

```
Macro:  CDIR - CREATE a permanent directory
Call:
        CDIR <filename>
Description:
        This command works like the CLI CREATE command, except that directory
        type files are created, with ACL <user> OWARE + RE and the PERMANENCE
        attribute on.  The default maximum directory size is set to 5000
        disk blocks, but this can be changed by using a /MAX= switch.
********************************************************************************
Command: DATE - display the current date.
Call:
        DATE
Description:
        This command just displays the current system date.  If you notice
        that the system date is not correct, please tell someone on the
        computer staff.
********************************************************************************
Command: DELETE - (DEL) Delete one or more files from the disk ( abbrev. DEL )
Call:
        DELETE <file templates>
Description:
        Eliminates files matching the specified templates from the disk.
        Please use care in deleting files.  It is very easy to delete
        files accidentally.  Before using DELETE with templates, it is
        better to use the FILESTATUS command with the same templates to
        see which files will be deleted.
        For more DELETE details, use "HELP/V DEL".
********************************************************************************
Macro: LCXP - print a text file in rotated page format
Call:
        LCXP <filename>
Description:
        This macro calls a program, LCXP, which translates a text file to
        rotated print format.  Use this macro to generate "book page" printout.
        (The page you are now reading was generated in this manner.)
********************************************************************************
Macro:  MYACL - Set Access Control List to <user> OWARE + RE
Call:
        MYACL <file templates>
Description:
        This macro sets the ACL of the files specified by the templates
        to <your username> OWARE,+,RE.
********************************************************************************



********************************************************************************
Macro:  YOURACL -  set Access Control List to (arbitrary) <user> OWARE + RE
Call:
        YOURACL <user> <file templates>
Description:
        This macro is useful when a user wants to set the ACL of a file
```

```
        to another user.
**************************************************************************
Command: PERMANENCE - (PERM) set,change or display a file's permanence attribute
Call:
        PERM <file template> <ON or OFF>
Description:
        Each file has a permanence attribute which may be set or unset
        using this command.  The permanence attribute, when "ON", renders
        thee file immune from deletion until the permanence is turned "OFF".
        The command, with one argument, will only show permanence status.
        The second argument, when used, must be either "ON" or "OFF".
        More help is available via "HELP/V PERM".
**************************************************************************
Macro:  PERMED - Edit a Permanent File
Call:
        PERMED <filename>
Description:
        This macro is useful in editing permanent files.  Note that after
        editing, the file is left permanent, regardless of its original
        state.
**************************************************************************
Macro:  PERMDEL - Delete a Permanent File
Call:
        PERMDEL [<file1>,<file2>,...]
Description:
        This macro works identically to the CLI "DELETE" command, except
        that it sets the file's PERMANENCE attribute to "OFF" before deleting.
**************************************************************************
Macro:  NOBRK - delete all break files
Call:
        NOBRK
Description:
        This macro should be used to delete BRK files caused by TRAPs, etc.
        It is a good idea to TYPE the macro NOBRK.CLI before using it to
        see which file templates will be deleted.
**************************************************************************
Macro:  PRINT - print an ASCII text file on the system printer
Call:
        PRINT <file templates>
Description:
        This command queues the file(s) specified to the system printer.
        You may have to wait a short while to get your output if the
        queue is full.  Check the queue by typing "QD".
        BE SURE THE FILE YOU ARE PRINTING IS AN ASCII TEXT FILE!
**************************************************************************
Command: RENAME - change a file's filename
Call:
        RENAME <old filename> <new filename>
Description:
        Changes a filename.  This command takes two and only two arguments.
        The file remains the same, only the name has been changed.
**************************************************************************
```

```
**************************************************************************
Command: SPACE - (SP) display or change a directory's maximum space allowance
Call:
        SPACE</V> <directory templates> <new max. number of disk blocks>
Description:
        With no arguments, SPACE (abbrev. SP) returns the maximum, current,
        and remaining disk blocks in use by the current working directory.
        With one argument, the space statistics for the named directories
        are returned.  The /V is used to verify the directory names.
        The SPACE command with two arguments will change the maximum allowable
        directory space for the named directories.
**************************************************************************
Macro:  SPY - Display disk space of specified directories
Call:
        SPY [<file1>,<file2>,...]
Description:
        Returns the SPACE used either in the specified arguments (if any)
        or in all CPD subdirectories of the current working directory.
        Also gives the space statistics for the root (:) directory.  Useful
        in determining which directories need cleaning up.
**************************************************************************
Command: TIME - display the current system time
Call:
        TIME
Description:
        Returns the current system time.  If you notice that the system time
        is not correct, i.e., more than 15-30 minutes off, please tell someone
        on the computer staff.
**************************************************************************
Command: TYPE - (TY) display at the screen an ASCII text file
Call:
        TYPE <file templates>
Description:
        This command will display the contents of a file in ASCII format on
        the console screen.  If you supply a template matching a nonASCII
        file, the CLI will attempt to display this file also, resulting in
        unintelligible garbage on the screen. (Use the ESC key to get out
        of this mess.)  Please try to TYPE only ASCII text files.
**************************************************************************
Macro:  WHOALL - display all current system processes
Call:
        WHOALL
Description:
        This command executes the CLI "WHO" command for all processes on
        the system, displaying the Process ID (PID), username, simple
        process name, and program name.  Useful for seeing who is doing
        what.
**************************************************************************
Command: XEQ or EXECUTE - execute a program file ( abbrev. X or EX )
Call:
        <EX or X> <program name>
```

Description:
This command is used to run other programs from the CLI.
CLI execution is suspended until the program is finished.

****

2.3 SYSTEM OPERATION

2.3.1 consoles

Consoles allocated for scientific use are located in room B306. All consoles are Digital VT100's, and on each console is a tag which refers to how it is identified by the computer (e.g. CON3 CON4, etc.). Any of the five consoles in B306 are available for general use.

You should use the console sign-up sheet in B306 to reserve a console during daily working hours. Just initial the time block you wish to reserve on the sign-up sheet. This sheet is usually located on the bulletin board just inside the door of B306.

There is also a console in the "TV room" (B302) across the hall from B306. This console also has a sign-up sheet of its own on the door. The TV console is reserved for users who wish to use the HIP image display, but it is available for general use if it has not been signed for.

The measuring engine in room B343 also has an M600 console. This console has a sign-up sheet as well, and is reserved for users of the measuring engine. If no names appear on the sign-up sheet, this console is also available for general use.

2.3.2 logging on

Once you have procured a console, you may log on to the Einstein system. If your console is not already logged on, you should see this:

*** SAO DEVELOPMENT / TYPE NEW-LINE TO BEGIN LOGGING ON ***

This message is the mark of an idle console. To log on, just type RETURN. The computer will request your USERNAME and PASSWORD, and if correct, will log you onto the system. When you see the CLI prompt ")", the system is awaiting your input.

Special console characters exist to give you more control over your console's input and output. The "no scroll" key will stop output to the console when pressed once, and will continue output when pressed again. The "delete" key will erase the last character input to the console. A

CTRL-U sequence will erase the entire last line input to the console. (To type a CTRL character, just press the CTRL key, and while holding it, type the desired letter.) The ESC key is used to "escape" from a CLI instruction or macro. The sequence CTRL-C CTRL-B is used to terminate a program which is in control of the console. Be careful of using CTRL-C CTRL-B, since it will log you off the system if used in CLI.

If you are ready to log off the system, use the CLI command BYE. This terminates your CLI process, and frees your terminal for the next user.

2.3.3 BATCH streams

Users can also run programs in the BATCH streams. A BATCH request is made by the user, and this request is entered to the batch queue, where it is processed, in turn, with other batch requests. This allows the user to make use of the console while his/her job is running in the batch stream.

To make a BATCH request, use one of the following BATCH macros:

) EBATCH <job>

This is the "express" batch stream, to be used ONLY for short jobs, such as Fortran compiles and links. A CPU time limit of 2 minutes has been imposed on this batch stream, meaning that jobs trying to exceed this limit will be terminated before completion.

) BATCH <job>

This batch stream, is used for somewhat longer jobs which must be run during the day. Large compiles and links, should be run in this stream, leaving the EBATCH stream open for shorter jobs. CPU time limit for BATCH is 4 minutes. Longer jobs and scientific analysis programs should not be BATCHED during the day, but rather should be run from a console.

) NBATCH <job>

This is the "night" batch stream. This stream will accept queue entries during the day, but the queue requests are executed only between the hours of 12 a.m. - 8 a.m. The night batch stream is for lengthy jobs which are too large for EBATCH or BATCH. There is a CPU time limit of 8 minutes of NBATCH.

) ELBATCH <job>

This is the "elongated" batch stream, to be used for very large jobs only. The idea here is to leave the NBATCH stream open for medium-sized jobs. The ELBATCH stream has no CPU time limit, so

a job could very well (and often does) run all night.

Applying a "/I" to any of the above batch commands will allow you to input more than one line of input to the batch stream (similar to CREATE/I).

To display the current queue entries, use the CLI command "QDISPLAY", abbreviated "QD". The list of queue entries will show some useful information. Each queue entry is accompanied by a username and a sequence number (not to be confused with the Einstein observation sequence numbers). Jobs which are currently active will be marked by an asterisk. You should be able to recognize your job in the queue display (if you have made a batch request).

If you make a mistake in a batch request, you can effectively remove the queue entry by using the "QCANCEL" command:

) QCAN <sequence number>

where "sequence number" is obtained from the queue display. QCAN also will terminate an active batch job.

2.3.4 The CLEAN program

The CLEAN program runs nightly on the M600's data disk. This program is meant to delete files which have not been accessed for a specified period of time. The CLEAN rules apply to the following directories:

:Z3:HBDATA - This is the directory which contains all users' data directories. Most files in these directories have a life span of 14 days unaccessed. DECON files will be deleted after 2 days unaccessed. Permanent files will be kept for 30 days unaccessed before they are deleted. No cleaning here on Saturday or Sunday.

:Z3:IMAGES - This is the image file directory. The files contained here will be deleted after 10 days unaccessed. Permanent files will not be deleted.

:Z3:HBDATA:EINSTEIN - This is a special data directory, to be used as everyone's scratch space. Files in this directory are deleted after 2 unaccessed days. Permanence makes no difference in EINSTEIN. No cleaning here on Saturday or Sunday.

The CLEAN program leaves a log, in :CLEAN.LOG, which may be typed to see which files were deleted the previous night.

2.3.5 The editor (ED) (Short Summary)

To edit a text file, use the screen editor ED. The command to call the editor is:

) ED <filename>

The console display will now be entirely controlled by the editor program. On the first line of the display is the current working directory, followed by the input file name, followed by a comma, followed by the output file name. If either of the files does not exist, its name is replaced by a "*". Following the input and output file names is the current input page number, followed by the current output page number. The editor displays 8 lines of text on either side of the current cursor position. The cursor will be displayed somewhere in the text. At the bottom of the display will be a dotted line, below which editor commands will appear as you type them. Use the directional arrows on the keyboard to position the cursor wherever you like.

Any editor command is terminated (and sent to execute) by pressing the LINEFEED key twice. The system echoes this as "$$". A single LINEFEED as the first character of a command will repeat the last command.

Some basic editor commands:

I — inserts text at the cursor position
For example, the instruction ICOUNT=0.0$$ will insert the string "COUNT=0.0" into the text at the cursor position and the cursor will be positioned after the string. Note that a carriage return is considered a character.

nD — Deletes the n characters to the right of the cursor. If n is omitted, 1 is assumed.

nK — Deletes all characters to the right of the cursor through the nth carriage return. If the cursor is positioned at the beginning of a line, the next n lines will be deleted. If n is omitted, 1 is assumed.

nL — Moves the cursor n times. If n is 0, the cursor is placed at the beginning of the current line.

Sstr — Searches for the string "str" and positions the cursor after the string (if found).

Cstr$rts — Searches for the string "str" and, if found, replaces it with the string "rts". String arguments are delimited by a single LINEFEED.

B — Move the cursor to the beginning of the text.

Z Move the cursor to the end of the text.

J Writes out all the changes you have made and re-opens the file. Preserves changes made so far in case editing is aborted for any reason.

W Writes out and closes the file you are currently editing.

GX Closes the file you are currently editing without writing out the changes you have made.

Ofilename Opens the named file for editing

H Exits the editor. After a delay of a few seconds (at most), the CLI prompt will reappear.

There are also some "instant" commands, which are used without the LINEFEED command delimiter. The "direction" keys (at top right of keypad on VT100s), are used to position the cursor to the left or right of the current position, or to the next line up or down from the current line. The PF1-4 keys at the extreme right of the keyboard will move the cursor 6 positions up,down,right, or left.

Buffer commands are useful as well, mostly for moving chunks of text from one place to another within the same file, or to another file. Refer to the editor manual to find out about these (red books in B306).

2.3.6 BASIC GUIDE TO EDITING ON THE S230 AND THE M600 (Longer Description)

****

This writeup covers basic procedures and possibilities available to the AOS editor. For more complex problems, please see the SAO DISPLAY TEXT EDITOR MANUAL.

****

CONVENTIONS USED IN THIS WRITEUP

Commands are shown in LARGE TYPE.

Arguments are shown in parentheses, in small type. DON"T INCLUDE THE PARENTHESES WHEN TYPING IN THE ARGUMENT!

Screenedit control characters (as described on the terminals) do not operate in the editor.

) designates the prompt in normal CLI mode.

$ designates the LINEFEED key. When you type LINEFEED, the $ symbol will appear on the screen. Hereafter it will be assumed that $ refers to LINEFEED. (In speaking about the editor, some folks will call the LINEFEED key "escape" for some reason. This adds confusion to an already complex situation!)

Control Key: holding down the key marked CTRL and simultaneously striking a letter, is used for various short-hand instructions (see below). For example, CTRL N puts you in INSTANT INSERT MODE.

CP stands for CHARACTER POINTER. The character pointer is the cursor. In the editor, the cursor indicates the origin point of commands. It shows you where you are in the text. The cursor is always at the point after (TO THE RIGHT OF) the current character typed.

****
****

MISCELLANEOUS VALUABLE INFORMATION

BLUNDERS. If you make a mistake, the display will usually vanish (very disconcerting!) and an error message will be displayed. Most errors are no problem to correct. Hit a carriage RETURN. Your text should reappear. If your mistake caused a real disaster, see the section in this writeup on ERRORS. Otherwise, correct whatever you did and try a new version of your command. If the text doesn"t reappear, perhaps you should ask a data aide for help. Some errors really upset the editor. It can be difficult to get a file safely out of the editor.

TO MAKE SURE YOU DON'T LOSE A LOT OF WORK!!!!! If you are doing a lot of work on a file, type J$$ from time to time. This command will make the changes you have typed in since you entered the editor (or since the last time you typed J$$) permanent. The CP will go to the begining of the

file (or page, if the file has pages). If you make a horrendous mistake or even if the system crashes, you have a good chance of saving most of your work. To be CERTAIN you have a recent copy of your file, go to the beginning of the file and type FO(any-new-filename)$$. This sends a new copy of your file, called (any-new-filename), out of the editor into the directory you were working in. This only works with files which are not divided into pages. For longer, divided files, exit the editor, copy the file and re-enter.

Upper and lower case letters are treated as identical by the editor, except, of course, when they are referred to specifically by commands. For example: if you give a command to search for the word "IMPORTANT", The editor will not find "important" or "Important". You can, however, type the command either as SIMPORTANT$$ or as sIMPORTANT$$.

Spaces between words or at the end of lines are counted as entities by

the editor. This is important when you are constructing files which will have columns. You can use the M (move) command on a file which is empty of characters but has lines of blanks set up. Forinstance, if you want to fill in elements in a column 14 spaces from the left margin you"d type:

14M$I(whatever-you-want)$$

(Without blanks, the CP would move instantly to the end of the file.)

There is some danger of creating very long lines and not realizing you have done so. If you set up a file using blanks, remember to delete the number of blanks you replace with your entries. For example:

14M$I(whatever-you-want)$19D$$

Thus you have the same number of total spaces as before you entered (whatever-you-want). This may be a bit fancy...

LINE TOO LONG Occasionally you will bring a file into the editor and the message LINE TOO LONG will appear. This may happen for the reason cited above, or because you have accidentally appended one line to the next during editing. It is sometimes difficult to locate long lines. Try typing TT$$. TT$$ cuts all lines which extend beyond the screen. The portion which was hidden is printed below the begining of the line. You will have to look through the file to find such lines. This can be a pain in a long file, so try to avoid the problem! To restore the display to normal, type TT$$ again. (TT$$ doesn't change the file, only how it is displayed.)

****

ENTERING THE EDITOR

You can either bring a file into the editor or you can create a file once you are there.

To bring a file into the editor, type:
)ED (filename)

The current display will clear. A line of information will appear at the top of the screen. The first 18 lines of the file will be displayed below that, followed by a row of dots. When you type commands, they will appear under the row of dots. YOU AUTOMATICALLY CREATE AN OUTPUT FILE BY USING THIS FORMAT.

The line of information at the top of the screen shows the following: CURRENT-DIRECTORY INPUT-FILE,OUTPUT-FILE CURRENT-PAGE NEXT-PAGE RADIX* TIME
*The radix is decimal by default but it can be changed. See the MANUAL.

If there is an asterisk in either the input or the output file's space, you either didn't bring a file into the editor or haven't created an output file.

CREATING OR BRINGING A FILE INTO THE EDITOR

To create a file in the editor, type:

)ED

The display will clear and you will be in editor mode as described above.

Once in the editor, you can bring in any file from any directory to which you have access (The file must be in your current searchlist). You can only edit textfiles. To bring in a file, type:

FI(filename)$$

CREATING AN OUTPUT FILE

If you have no output file, you can create one by typing:

GW(filename)$$

You can use any new filename. If you name an output file some name which is already in use, you will get the error message FILENAME ALREADY EXISTS.

$$

INFORMATION ABOUT $$ (LINEFEEDS)

A single $ is used to separate elements in a command line. The command line is activated by a second $. Any command which requires arguments MUST be followed by a $. Any command MAY be followed by a $. If you want to be sure your command line gets executed, separate each command in the command line with a $ and type two $$ at the end!

You can reiterate any command line by typing another $. The third $ will appear briefly before the command is executed.

$$

CONTROL OF THE CP - MOVING AROUND IN THE TEXTFILE

The display is redrawn after most commands. The following cursor control commands are all exceptions. Unless the CP moves off the top or bottom of the screen, the image remains.

down-arrow key (cursor down key) MOVE CURSOR TO BEGINNING OF NEXT LINE
up-arrow key (cursor up key) MOVE CURSOR TO BEGINNING OF LINE OR ONE LINE UP
right-arrow key (cursor right key) MOVE CURSOR AHEAD 1 CHARACTER
left-arrow key (cursor left key) MOVE CURSOR BACK 1 CHARACTER
THE KEY LABELED PF1 MOVE CURSOR UP SIX LINES
THE KEY LABELED PF2 MOVE CURSOR DOWN SIX LINES

THE KEY LABELED PF3 MOVE CURSOR BACK SIX SPACES
THE KEY LABELED PF4 MOVE CURSOR FORWARD SIX SPACES
CTRL U MOVE CURSOR TO BEGINNING OF TEXT BUFFER
CTRL B MOVE CURSOR ONE LINE DOWN
CTRL R MOVE CURSOR BACK 4 CHARACTERS

The following commands may or may not redraw the screen.

CTRL Z MOVE TO END OF FILE (OR CURRENT PAGE*)
CTRL B MOVE TO BEGINING OF FILE (OR CURRENT PAGE*)
*More about page division later.

To move the CP a certain number of lines, type No. of-lines-desired followed by L. You can move toward the begining of a file by typing a negative number.

12L (moves you twelve lines toward the end of your file.)
-12L (moves you twelve lines toward the begining...)

****
****

INSTANT INSERT MODE

Probably the single most useful aspect of the editor is the INSTANT INSERT mode. As its name implies, it allows you to type changes directly into the text. To get into INSTANT INSERT, type:

CTRL N

The words INSTANT INSERT will appear under the line of dots to show you you are in that mode.

You can move the CP using the usual controls. Insert or delete whatever you wish directly. You may want to go in and out of INSTANT INSERT. Some tasks are much easier to accomplish using editor commands. To get out of INSTANT INSERT, type another CTRL N.

****
****

BASIC TEXT EDITING COMMANDS

S Search - this requires an argument. S(whatever-you-want)$$

C Change - this requires arguments. C(whatever)$(to-whatever-you-want)$$

I Insert - requires an argument, obviously! I(whatever)$$

D Delete - you can delete any number of characters.
Examples: D$$ (causes the character under the CP to vanish)
1D$$ (does the same as D$$)

12D$$ (erases 12 characters, starting with the one under the CP, moving forward.
-12D$$ (erases 12 characters, starting with the one under the CP, moving backward. (Don't forget that blank spaces between words count as characters.)

If you type D$$ when the CP is at the first possible location of a line, THE LINE IS APPENDED TO THE LINE ABOVE. THE CHARACTER DELETED IS THE LAST CHARACTER OF THE PREVIOUS LINE. This can lead to the error message LINE TOO LONG if you don't realize what happened.

K Kill! - you can kill any number of lines. The same rules which apply to D apply to K. If the CP is in the middle of a line when you use the K command, the line will be killed from the CP to the carriage return and THE NEXT LINE WILL BE APPENDED TO THE BEGINING OF THE LINE YOU KILLED. This can also lead to LINE TOO LONG.

This aspect of the D and K command may seem confusing. It is, in fact, a great help in re-arranging the margin during editing. If you delete sections of a line, it is easy to bring the next section into the space. Then move the CP to a convenient place to end the line. Hit a RETURN and the remainder of the line will be chopped off and moved to the line below.

MOVING OR COPYING LARGE CHUNKS OF A FILE

If you want to move a paragraph or section of text count the number of lines it contains. Move the CP to the begining of the section and type:

(number-of-lines)XM$$

The section will vanish (OH NO!!!) Luckily it will be waiting (in a buffer) for you to move the CP where you really wanted the section to go, and to type the command:

XG$$

This command brings ALL THE LINES in the buffer into the file at the point of the CP. If you wanted the section to be REPEATED 5 TIMES you would type:

5XG$$

To copy a section of text into the buffer and put a copy into the text at another location use the following:

(number-of-lines)XC$$ - This copies the text. Your original is still in its original place.
Move the CP to where you want the copy to be.

XG$$ - This inserts the copy.

MORE EXOTIC STUFF

If you have fairly complex command lines to use, chances are you will get tired of retyping them. You can put command lines into a buffer. Type the command as usual. Execute it. Then type a CTRL F. Your command will appear as if it were a file being edited. You can edit it if you need to. If you want to put it in a buffer, use the (#-of-lines)XM$$ command. Then type another CTRL F. Do whatever you need to do until you need the command line(s), type CTRL F, XG$$, CTRL F and you have the stored command line(s) back.

GLOBAL CHANGES

If you have written a paper in the editor and have misspelled the word satellite every time you wrote it, you will want to make a global change! To do so, type the command in square brackets. In the example given above, you'd type:

[Csatelite$satellite$]$$

You will get the message MISSING on an otherwise blank screen. That means that the command line has been executed each time it found "satelite" until it reached the end of the file and didn't find another one. Hence the word MISSING. Just hit a RETURN and the CP will return to the begining of the file.

To insert a margin, you can type:

[S(carriage-return)
$I(tab-or-spaces)$]$$

NOTES ABOUT FILE FORMATS

There are 64 lines per printed page on the M600 versatec. If you LCXP a file (LCXP is a command which you give outside the editor. It causes a file to be printed lengthwise on the page.) you'll only get 62 lines because the title of the file will appear at the top left, followed by a blank line.

The computer will automatically make pages if your file is more than 64 lines long. These will be marked with page numbers if you LCXP the file but not if you just print it.

TO EDIT A VERY LARGE FILE

If you know the file you are editing is very large, begin editing by typing

)ED/C=35 (filename)

This creates the largest possible buffer space in the editor. You may still get a CORE FULL error message. Type a RETURN if that happens. You will need to create an output file. Type:

GW(new-filename)$$

If your file becomes too large, you will also get CORE FULL. This means that the normal buffer space allowed for editing has been exhausted.

CREATING PAGES

One way to deal with this problem is to create "pages" as you go. When you have, say, 60 lines of text, you can give the command

I(CTRL)L$$

This inserts a page delimiter. The 60 lines will still be on the screen. There will be a L at the CP. This will not appear in the text when you print or display it outside the editor. The spacing will be your original spacing. Now you can go on to create other pages. The text of earlier pages is no longer in the buffer. Pages can be longer than 60 lines. The printer will only print 62 or 64 lines per page, however.

You exit from the editor as usual (W$H$$) Don't worry, your file will make it safely out of the editor in one piece! The next time you edit the divided file, however, you will have to edit it one page at a time. When you finish with page 1, the CP will not advance until you give the command

R$$

This "turns the page". Now you must stay on page 2 unless you go on via the R command.

MOVING AROUND IN A FILE WHICH IS DIVIDED INTO PAGES

GXO(name-of-file)$$ - this will return the CP to the beginning of page 1 of the file. NB: This is GXO, not GXzero!

N(whatever-you-want)$$ - this will search for whatever-you-want through all the lines and pages AFTER the CP.

You can be sneaky and put singular characters on each page. Then, if you want to get to a page quickly, you can use the GXO(filename) command followed by Npage 5$$ (or whatever you set as your unique element for the fifth page).

If you want to rejoin 2 pages, use the command:

A$$ - this appends the page you are on to the previous page.