GUIDE to VAX, CDC and IBM 3090

This is an on line manual in

BNLDAG::DUA0:[PARSA1]GUIDE3.VAX

[Third Edition]

May 1987

COMPILED BY

ZOHREH PARSA

BROOKHAVEN NATIONAL LABORATORY
Associated Universities, Inc.
Upton, New York 11973

GUIDE TO
VAX, CDC, and IBM 3090

This is an on line manual in

BNLDAG::DUA0:[PARSA1]GUIDE3.VAX

[Third Edition]

COMPILED BY
ZOHREH PARSA

BROOKHAVEN NATIONAL LABORATORY

This Guide was originally edited for the users of DUA0:[PARSA1] Accelerator Physics programs directory. In response to large number of requests from a broader audiance the previous edition is updated, and new sections on IBM 3090 and Network and Communications have been added. (This is a collection of notes and definitions from guides, reference manuals, and help files).

GUIDE TO
VAX, CDC and IBM3090

TABLE OF CONTENTS

	PAGE
I. COMPUTER SYSTEM	1
User's Environment	1
a. Hardware	
b. Software	
PROCESS	1
LOGIN AND LOGOUT	1
EXAMPLE	
HOW TO SET PASSWORD	2
DCL COMMANDS AND THEIR FORMAT	2
HOW TO CANCEL COMMANDS	2
SYNONYMS FOR DCL COMMANDS	3
CONTINUATION MARKER	3
EXAMPLE	3
HELP	3
EDIT/EDT	4
LINEEDITING	5
Recalling Commands	5
Editing Commands	5
USER'S ENVIRONMENT	6
Show Command	6
Set Command	6
Set Host Command	6

II. FILE SPECIFICATIONS 7

EXAMPLES 7

DIRECTORIES AND SUBDIRECTORIES 7-9

SUBDIRECTORIES 9

WILDCARDS 9

TO CHANGE DEFAULTS 10

SEARCH FOR [DIRECTORY] 11-12

ERROR MESSAGES 12-13

HOW TO CREATE AND MANIPULATE FILES WITH EDT EDITOR: 13-14

1. LINE MODE 13

2. CHARACTER MODE (or KEYPAD) 13

EXAMPLES 13

HOW TO RECOVER FILES FROM ABNORMAL EXIT 14

EDT LINE MODE COMMANDS 14-16

EDT KEYPAD EDITING COMMANDS 16

THE VT100-TYPE TERMINAL KEYPAD 17

VT100 EQUIVALENT KEYS FOR IBM-PC-KEYPAD 16-17

IBM-PC EDITING KEPAD-------VT100 KEPAD 17-18

FOLLOWING IS THE MOST FREQUENTLY USED KEYPAD COMMANDS 19-20

HOW TO CREATE A .COM FILE 20-23

HOW TO DEVELOP AND RUN A PROGRAM ON VAX/VMS 22-25

o PRINT AND COPY COMMANDS 24

To COPY A FILE FROM MAIL TO YOUR

DEFAULT DIRECTORY 25

o TO PRINT FILES ON BNLCLUSTER [BNLCL2] 25

o TO PRINT A FILE ON BNLDAG PRINTERS 26

o HOW TO STOP A JOB 27

TO DELETE A PRINT JOB 27
TO STOP A CURRENTLY EXECUTING JOB 27

TO GET THE job 27

FILE PROTECTION 28

USEFUL EXAMPLES: 29

TERMINAL CHARACTERISTICS 29

o EXAMPLES: THIS SECTION DEMONSTRATES THE USE OF COMMANDS ON BNLDAG 30-32

FUNCTIONS KEYS MOST COMMONLY USED 30-31

EXAMPLES OF HOW TO GET INFORMATION [AVAILABLE ON-LINE] 33

INDEX 33

CERNLIB 33

o GMAIL 35-41, 107

GMAIL COMMAND 37

ADDRESSING, GATEWAYS AND EXAMPLES 38

RECEIVING MAIL 39

SHOW NET 41-42

o SUMMARY OF DCL COMMANDS 43-52

USER PRIVILEDGES 52-53

o EXAMPLE2: THIS SECTIN DEMONSTRATES THE USE OF COMMANDS AND PROCEDURES ON BNLCLUSTER. 52-62

LOGIN 54

	HELP	55-57
	HELP MAIL	56-57
	NOTICE	58
	TEMPORARY FILES	59
	USER DEFINED HELP LIBRARIES	60
	CDC TO VAX FILE TRANSFER	61-62
	AMDINFO	63
	BATCH QUEUES	63
	MAGNETIC TAPE USAGE	64
	HARDWARE CONFIGURATION	64
	MFE LINK TO LLL	65
III.	INFORMATION ON IBM 3090	67-101
	TERMINAL AND KEY DEFINITIONS	68-70
	KEY, KEYBOARDS AND FUNCTION KEYS	71-72
	KEYBOARD DEFINITIONS FOR THE VT100 TERMINAL, HAZELTINE AND PCPLOT	73-78
	KEYBOARD DEFINITIONS FOR THE VT220 TERMINALS AND SMART TERM EMULATOR	79-81
o	USING YOUR VIRTUAL MACHINE	82-88
o	USING AN ANSI-CONFORMING TERMINAL	89
	VIRTUAL MINIDISKS	90
	CMS FILES	91-92
o	EXAMPLE BATCH JOB ON THE IBM	93
	MISCELIANEOUS COMMANDS	94
	HELP AND QUERY	94
	FORTRAN COMMANDS	95
	MINIDISK COMMANDS	96
	BATCH COMMANDS	97

	SPOOL COMMANDS	98
	COMMUNICATIONS COMMAND	98
o	ADDITIONAL USEFUL EXECS	99
o	COMMUNICATIONS, FTP	100-101
IV.	NETWORK AND COMMUNICATIONS	102-119
	AN OVERVIEW OF BNL-FACILITIES	102-105
	HELP MAIL	56-57
o	BNLDAG NETWORK COMMANDS	106
o	GMAIL	107, 35-57
o	LABNODE - NETWORK TABLE	108
o	BNLCLUSTER NETWORK	109
o	COMMUNICATION TO AND FROM IBM3090	109-111
	PC FILE TRANSFER	112-114
	INTERCOME	112
	KERMIT	113-114
o	COMMUNICATION ON UNIX (BNL NODE)	115-118
	UNIX MAIL	115-117
	SUMMARY OF THE TILDE ESCAPES	117-118
	CONNECTION TO IBM	119
o	COMPUTER ACCESS LINES	120
	DIAL-UP LINES	121
	CLASS DIAL SEQUENCE	122

I. COMPUTER SYSTEM

1) User's Environment consist of:
   a. Hardware* [physical computer made in the factory]
   b. Software [refer to the Instructions (programs) to be performed by the computer.]

*Hardware is devided to 3 parts:

1. Central processing unit (cpu) (where most of the work is done)
2. Main memory (used for temporary storage of inst--ructions and data.)
3. Preripheral devices (disk drives, printers, terminals, cardreaders.)

i. Process: When a user logs in, the operating system (VMS, on bnldag and bnlcluster) creates a PROCESS containing a complete description of user's environment, work, whereabout, and the information from the User Authorization File (AUF), (note also that, UIF is the User 's Identification Code). Therefore, the PROCESS is equialent to the user's environment.

ii. "The processes are created for the purpose of running programs".

iii. DCL is the digital command language.
You will know if the DCL interpreter program is executing, by presence of the DCL prompt $
(a dollar sign on the left side of your screen).

iv. LOGIN AND LOGOUT

1. Turn on the terminal press return (<return>).
2. Turn on the gandalf box (set at 40)
   [see a-1 for info. re computer access lines]
3. Type in CLASS, USER NAME, PASSWORD upon request. e.g. enter CLASS 55 (for BNLDAG), 111 (for BNLCLUSTER), 76 (for UNIX), etc.

FOR EXAMPLE:

```
$ ENTER CLASS   55
  USERNAME      parsa
  PASSWORD
$
$                  [$ is the dcl prompt, and dcl interpretor
                    is ready for a valid dcl command. e.g., logout]
$ logout           [This command deletes your process and, the
                    resources are returned to the system]
```

```
$ set password        [with this command you can change your
                     passwd, after <return>, type in the old
                        and the new password upon request.]

   <OLD PASSWORD>  task        [assume, task is the old password,
   <NEW PASSWORD>  newyork       and newyork is the new password,
   <VERIFY PASSWORD> newyork           you are changing to.]
```

Remmember, that PASSWORD DOES NOT ECHO, so you won't see the "passwords" you type in (e.g. task and newyork will not be printed).

Additionally, at login time, the system automatically executes the commands in your LOGIN.COM file, (if it exits). So, login.com file is a convenient way of reducing the regular housekeeping chores. Hence you may create a login.com file and in it define commands synonyms, defaults etc. (see examples in sec. on .COM files).

v. DCL COMMANDS AND THEIR FORMAT

Following is a partial list of DCL commands:

```
$ logout                              [$ command]
$ show device                         [$ command option]
$ show device/all                     [$ command option/qualifier]
$ type file.txt                       [$ command parameter]
$ directory/size file.txt             [$ command/qualifier parameter]
$ print file.txt/copies=4             [$ command parameter/qualifier]
$ print file1.a, file2.txt            [$ command parameter,parameter]
$ print file1.a,file2.txt/copies=3
                                      [$ command parameter,parameter/qualifier]
```

You should note that, "the first four characters of any DCL command, option or qualifier uniquely identifies it to the DCL interpreter". E.g., print (to prin), or directory (to dire or dir).
[For a complete listing see the section on Summary of DCL COMMANDS]

HOW TO CANCEL COMMANDS:

If you enter commands in response to a prompt and want to discard the entire command you have entered (not just the most recently entered line), use CTRL/C or CTRL/Y, as shown below:

```
$ PRINT/COPIES=3
$_File:     SYNCH.DAT    <ctrl>Y
```

In this example, the PRINT command was entered correctly, and the system prompted for the file name to be printed. However, while entering the file name, the user decided to discard the command completely, and used CTRL/Y to cancel the command entirely.

SYNONYMS FOR DCL COMMANDS:

You may define synonyms for commands or strings (in two ways):
[for e.g., to use list instead of directory (as a short hand)
enter either of the followings:

EXAMPLE 1:

```
1.    $ list="directory"  [i.e. use = and "string"] or
2.    $ list:=directory   [i.e. use := and string without " "]
```

EXAMPLE 2: to use xx istead of yyyyyy as a shorthand,
enter either of the followngs:

```
1.    $ xx="yyyyyy"  [or,]
2.    $ xx:=yyyyyy
```

EXAMPLE 3: to use pd as a shorthand in place of
DELETE SYS$PRINT/ENTRY= ;enter the followng:

```
      $ pd="delete sys$print/entry="
```

therefore,

```
      $ pd 300    is the same as (but much shorter than)
      $ delete sys$print/entry=300
```

Hence, in your LOGIN.COM file (for convenience) you may
define command synonyms or strings that you may often use.

(see how to create .COM file, after the section on edt.)

You should also note that, at login time, the system auto-
matically executes the commands in your login.com file

CONTINUATION MARKER:

- (hyphen) can be placed at the end of the line and press
<return>, then continue the input after the
$_ on the next line.

FOR EXAMPLE:

```
$  print file1,file2,file3-  <return>
$_ file4,file5.....  <return>
```

vi. HELP:

HELP command gives on line information on topics and
subtopics. For example, type the following.

```
$ HELP SHOW SYSTEM
$ HELP EDITOR
```

```
$ HELP PRINT/COPIES

[To exit from help or other commands, press <ctrl>c, or
<ctrl>y.]

Note the followng (enter):

$ help

HELP
      ..........................

     Press the RETURN key one or more times to exit from HELP.

   You  can  abbreviate any  topic name, although ambiguous abbreviations
   result in all matches being displayed.

  Additional information available:

  :=          =           @           ACCOUNTING  ALLOCATE
ANALYZE     APPEND      ASSIGN      ATTACH      BACKUP
CANCEL      CLOSE       CONNECT     CONTINUE    CONVERT     COPY
CREATE      DEALLOCATE  DEASSIGN    DEBUG       DECK
DEFINE      DELETE      DEPOSIT     DIFFERENCES
DIRECTORY   DISCONNECT  DISMOUNT    DUMP        EDIT
ENCRYPT     EOD         EOJ         Errors      EXAMINE
EXCHANGE    EXIT        FDL         FORTRAN     GOTO        HELP
Hints       IF          INITIALIZE  INQUIRE
Instructions            JOB         Lexicals    LIBRARY
Line_editing            LINK        LOGIN       LOGOUT      MACRO
MAIL        MERGE       MESSAGE     MONITOR     MOUNT       ON
OPEN        PASCAL      PASSWORD    PATCH       PHONE
PRINT       PURGE       Queues      READ        RECALL
RENAME      REPLY       REQUEST     RMS         RTL_Routines
RUN         RUNOFF      SEARCH      SEND        SET
SHOW        SORT        SPAWN       Specify     START       STOP
SUBMIT      Symbol_Assign           SYNCHRONIZE
System_Services         TYPE        UNLOCK      WAIT        WRITE


 For next example, enter:

 $ help edit/edt

 EDIT

   /EDT

      Invokes the EDT interactive text editor.  The /EDT qualifier is
      not required, because EDT is the VAX/VMS default editor.

       Format:

          EDIT file-spec


       Additional information available:

      Parameters Command_Qualifiers
     /COMMAND   /CREATE    /JOURNAL   /OUTPUT     /READ_ONLY /RECOVER
```

```
$ HELP LINE_EDITING      <return>

LINE_EDITING

Your terminal has special keys that allow you to recall and edit
previous commands.  These keys are described below.  Note that some
of these keys may not be available on your keyboard.

Recalling Commands
------------------
The system saves and allows you to access up to 20 of your most recently
entered commands.  It saves less than 20 if your commands are very long.
Use the UP-ARROW or CTRL/B keys to recall previously entered commands.
Use the DOWN-ARROW key to reexamine a command that you had retrieved
using UP-ARROW.

If you are supplying commands or data to a program or utility, the
system saves and allows you to access only your single most recently
entered command.


Editing Commands
----------------
To enable the editing keys, execute the command

        SET TERMINAL/LINE_EDITING

Append /OVERSTRIKE or /INSERT to the command to specify the default
editing mode that you prefer.  /OVERSTRIKE causes a character that you
type to be written over the character that the cursor is currently
pointing at.  /INSERT causes the character to be inserted before the
cursor.

Edit Key     Function
--------     --------
CTRL/A or    Switches between overstrike mode and insert mode. The
F14          default mode is reset at the beginning of each line.

CTRL/D or    Moves the cursor one character to the left.
LEFT-ARROW

CTRL/E       Moves the cursor to the end of the line.

CTRL/F or    Moves the cursor one character to the right.
RIGHT-ARROW

CTRL/H or    Moves the cursor to the beginning of the line.
BACKSPACE or
F12

CTRL/J or    Deletes the word to the left of the cursor.
LINEFEED or
F13

CTRL/U       Deletes characters from the beginning of the

             line to the cursor
```

vii. USER'S ENVIRONMENT:

1) to get information about user's environment, you can use the SHOW command:

```
$ SHOW SYSTEM          [list of all processes on the system]
$ SHOW PROCESS/ALL     [information about own process]
$ SHOW STATUS          [current statistic on own process]
$ SHOW DEFAULT         [current position for device and directory]
$ SHOW TIME            [current system date and time]
$ SHOW TERMINAL        [characteristic of own terminal]
$ SHOW DEVICE          [characteristic of other devices]
```

2) To change the user's environment, use the SET command:

```
$ SET PASSWORD                  [to change password]
$ SET DEFAULT [PARSA1]          [to change the default dir to [parsa1]]
$ SET TERMINAL/DEVICE=VT100     [to change to vt100 terminal]
$ SET TERMINAL/WIDTH=132        [to change width of the line
$ SET TERMINAL/WIDTH=80          on terminal, to 132 or 80]
```

Remmember that, you can abreviate the commands, e.g. you can use term in place of terminal:

```
$ SET TERM/WIDTH=80        ;or
$ SET TERM/VT100           , etc.
```

YOU MAY USE THE SET HOST COMMAND TO CHANGE NODE, AS WELL AS DEVICE, DIRECTORY, ETC.:

For example you are loged in on BNLDAG [class 55, see section on LOGIN], by using the SET HOST command you can log into another NODE, e.g. bnlcl2 [cluster node 2]:

```
$ SET HOST BNLCL2 <return>    ![type in class, user name, password
$                              upon request]
$ USERNAME  zohreh
$ PASSWORD
$
$                             ![when you are finished with the
$                              node, BNLCL2, enter LOGOUT;


$ logout <return>
$ ZOHREH LOGGED OUT .....     ![now, you are logged out of BNLCL2
                              !and have been returned to your
                               original, login node; in this
                               example, the BNLDAG.]
$
$                             ![so this is the BNLDAG $ prompt]
```

II. FILE SPECIFICATIONS

A full VAX/VMS file specifications has the following format:

NODE::DEVICE:[DIRECTORY]FILE.EXT;VERSIONnumber

Note that; 1) spaces are not allowed between file specifications and, 2)punctuations must be used exactly as indicated above.

Following are examples of file specifications on two different nodes, 1) on BNLDAG and 2) on the BNLCLUSTER;

1) BNLDAG::DUA0:[PARSA1]HARMON.FOR;1
2) BNLCL2::2DUA7:[ZOHREH]QAZ.TXT;4

In the example 1) above:

i.	BNLDAG	;is called the NODE on VAX system; [which is analogous to a building, (e.g.HEF) in Brookhaven National Lab]
ii.	DUA0	;is the DEVICE [analogous to aroom, e.g, room #70]
iii.	[Parsa1]	;is the DIRECTORY [analogous to a filing cabinet]
iv.	HARMON	;is the FILENAME [analogous to a file folder]
v.	FOR	;is the FILETYPE [FOR = FORTRAN]
vi.	1	;is the VERSION NUMBER .

You may note that, the name for each portion of the file specification [except the version number] may contain digits or characters, (but must begin with a character) and contain certain length limited to;

1) NODE: 1-6 characters
2) DEVICE: 1-15 CHARACTERS
3) DIRECTORY 1-9 CHARACTERS
4) FILENAME: 1-39 CHARACTERS
5) FILETYPE: 0-39 CHARACTERS
6) VERSIONNUMBER: 1-5 DIGITS.

*** Check the variation in length for different VMS operating system.

DIRECTORIES AND SUBDIRECTORIES

A directory file is a file which containes (list of) names of other files.

A master directory file named 000000.DIR, resides on each disk volume. For example (type the following and see)

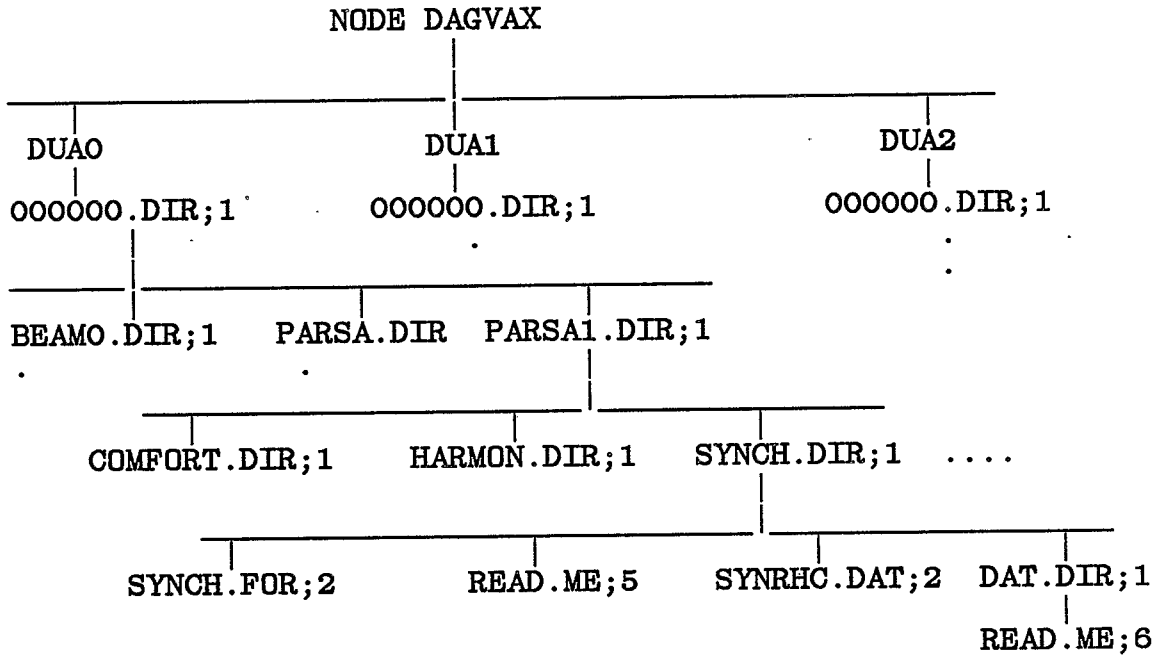
```
$ DIR DAGVAX::DUA0:[000000]

  beam0.dir;1  parsa.dir;1  parsa1.dir;1
```

You should note that:

1) Directory command always outputs in alphabetical order.
2) Subdirectories are directory files which contain a list of file names, some of which can be directories.

Following diagram is an illustration of VAX file structures:



Note that, the dir [PARSA1] containes subdirectory [SYNCH] which is referred to as dir [PARSA1.SYNCH] or DIR [.SYNCH], since [PARSA1] is the default directory.

Furthermore, all the directory files are in the form of NAME.DIR, but they must be, specified as [NAME] in the file specification. For example, the directory file PARSA1.DIR;1 must be specified as [PARSA1] in a file specification. i.e. to get the contents of PARSA1.DIR file you must type after VAX $ prompt:

```
$ DIR [PARSA1]
```

And to get the contents of a subdirectory, e.g. SYNCH, (which is a SUBDIRECTORY of [PARSA1]) you must type:

```
$ DIR [PARSA1.SYNCH]          ;or
$ DIR [.SYNCH]          (Since the default is PARSA1)
```

Here, by using period you did not have to enter the name PARSA1. You should also note that:

1) The number of directory files in any directory file is not limited;
2) However, only SEVEN levels of directories may be defined from the top.

For example, in the file structure diagram above,

```
PARSA1.DIR;1    is 1st level directory;
SYNCH.DIR;1     is 2nd level directory;
DAT.DIR;1       is 3rd level directory; etc.
```

SUBDIRECTORIES:

To create subdirectories use DCL command:

```
$CREATE/DIRECTORY [NAME]
```

For example:

```
$CREATE/DIRECTORY [PARSA1.SYNCH]
or,
$CREATE/DIRECTORY [.SYNCH]
```

Note, the period is used in place of PARSA1, which is the default directory.

To specify files in subdirectories, note that if the file name (e.g. READ.ME) is listed in a 3rd level subdirectory, the top level name and 2nd level name must be specified 1st to provide a search path for the system.

For example:

```
    DAGVAX::DUA0:[PARSA1.SYNCH.DAT]READ.ME;6
```

WILDCARDS:

Period (.) , * and % are wild cards.

The period is used within the directory (file) specification. For example:

```
$ DIRECTORY [PARSA1.SYNCH]
$ DIRECTORY [.SYNCH]
```

Here by using period, you did not have to enter PARSA1. Since the system takes the current (default) directory name [parsa1] and puts it before the period. Hence, to search for the requested file, the completed file

specification is used.

* can be used in place of any file specification.
For example:

```
$ DIR [PARSA1]*.FOR        [Will  list names of all files
                            whose type is FOR in dir PARSA1]

$ TYPE [.SYNCH]SYNCH.*     [Will type all the files named
                            SYNCH. e.g. SYNCH.FOR,
                            SYNCH.EXE,  etc.]
```

% is a single character wild card.

TO CHANGE DEFAULTS:

Use: $ SET DEFAULT [NAME] command.

For example: Device and directory names recorded for (Accelerator physics data base) PARSA1 are DUA0 and [PARSA1], respectively (see the dir diagram above).

When a user BILL on DUA1 logs in, the system sets his default device and directory.
For example; DUA1:[BILL]

To print the file NAME.FOR in the BILL directory you have to use the command:

```
$ PRINT DUA1:[BILL]NAME.FOR
```

If Bill wants to read or copy a file, e.g. SYNCH.FOR from [PARSA1] directory, he can type:

```
$ SHOW DEFAULT  <return>     ;[to check the current default]
  dua1:[Bill]

$ SET DEFAULT DUA0:[PARSA1] <return>
                            ;[To change the default to parsa1]

$ SHOW DEFAULT <return>      ;[to check that default was changed]
  dua0:[parsa1]
```

Now type directory [this gives parsa1 directory]:

```
$ DIR  <return>
  Harmon.dir;1  mad.dir;1   synch.dir;1
  .....         .....       .......
  .....         .....         ......
```

If you like to see the contents of the SYNCH.DIR;1 in [PARSA1] enter;

```
$ SET DEFAULT [PARSA1.SYNCH]  ;(or [.SYNCH]) then enter,
```

```
   $ DIR <RETURN>      ;[this gives you contents of SYNCH.DIR;1
                       which is a subdirectory of [PARSA1]]
    read.me  synch.for ....
    synch.exe  synch.obj ....
    .......    ......    ....
```

Now to copy, enter:

```
 $ COPY SYNCH.FOR;2 DUA1:[BILL]*.*    ; Which is the same as;

 $ COPY DUA0:[PARSA1.SYNCH]SYNCH.FOR;2  <return>
 $_ to:  DUA1:[BILL]*.*  <return>
```

Here *.* means to copy SYNCH.FOR and call it the same name in the dir [BILL]. You may note, the prompt $_to: after the <return> above, which you got since the full information was not supplied.

Now to check that, the default is still [PARSA1] enter:

```
  $ SHOW DEFAULT       <return>
    dua0:[parsa1.synch]
```

To change the default back to [BILL], enter:

```
  $ SET DEFAULT DUA1:[BILL] <return>
```

To check that the default is changed from PARSA1 to BILL, enter;

```
   $ SHOW DEFAULT <return>
      dua1:[Bill]          [which you expected to see].
```

CONCLUSION: To work in a subdirectory, you may SET DEFAULT [.SUBDIRECTORY-NAME], and when you are finished, you may set the default back to the main directory or logout.

SEARCH FOR [DIRECTORY]:

We already have uded two wild cards, period [.] and *. We can use two more wild cards ellipses [...] and hyphen [-].

1. Use [...] to SEARCH DOWN THROUGH DIRECTORY STRUCTURE

   Enter:

```
    $ DIRECTORY [...]  ;lists all files in your current
                        directory and all subdirectories
                        bellow your default directory.
```

2. Use [-] to BACK UP ONE DIRECTORY LEVEL, e.g.

   Enter:

```
$SET DEF [PARSA1.SYNCH]
```

Now, to list files in [PARSA1], (back up) enter:

```
$DIRECTORY [-]
```

Hence, use [-] to search up the hierarchy, and back up one level; or use [--] to back up two levels etc.

For example:

```
$ DIRECTORY [--.NAME]  backs up two level and then
                       down to its subdirectory [NAME].
```

3. To list files in the SYNCH subdir and all files bellow it (assuming the default directory is [PARSA1]), enter:

```
$ DIRECTORY [.SYNCH...]
```

4. If the default is set to [PARSA1.SYNCH], to list all files in [PARSA1] and all the files in rest of the structure, (use [-...]) enter:

```
$ DIRECTORY [-...]
```

5. To find out your login default directory (and disk) or the node that you loged in on, enter:

```
$ SET DEF SYS$LOGIN
```

This command is specially usefull if you default, to different directories and/or subdirectories, or (by $SET HOST NODENAME), work on different (computers) nodes, and suddenly want to find out what node, disk and /or directory you loged in to.

6. `[*...]` ;searches all directories and subdirectories [down as many as eight levels of directory name if exist] on the given (disk) volume.

ERROR MESSAGES:

Format:

```
%FACILITY-L-IDENT,TEXT
```

% FACILITY ;is the name of the system program or utility that generated the error message, e.g. DCL.

L ;is Level of the error. there are five levels;

1. S = SUCCESSFUL
2. I = INFORMATIONAL
3. W = WARNING

```
                4. E = ERROR   ;[there is error but prog tries
                                to continue.]
                5. F = FATAL    [severe error, program aborts].

   IDENT        ;is the abbreviation of the message text.

   TEXT         ;is descriptive message that tells, what
                 the problem is.
```

HOW TO CREATE AND MANIPULATE FILES WITH EDT EDITOR:

You can create and modify files by invoking the EDT EDITOR, eg.

```
    $ EDIT FILENAME.EXT        [EXT is extension or file type]
```

1. When a file is created, it is assigned the version number of 1.

2. If the editor is used to modify an old file (existing file), the editor will open the file of the name given which has the highest version number.

EDT editor can be used in two modes:

```
1.  LINE MODE     ;the * is the line mode prompt.
                   and DCL commands cannot be used
                   after the (*, editor) prompt.
                   (see next page for a list of EDT
                    commands).

2.  CHARACTER MODE   ;(or KEYPAD), on line mode is
                     changed to character mode by
                     command change (or c) after *
                     prompt. [This works only in VT100
                     mode;  use
                     SET TERM/VT100 command]
```

EXAMPLE:

```
$EDIT AVE.FOR  <return>
  *              [type change or c]
  * c <return>

    your text
      ....
  <ctrl>z
  *EXIT
  dua0:[parsa1]ave.for 30 lines
```

Note that, after you type C, you are in character mode and won't see the * prompt anymore.

To get back to line mode use <contrl>Z and you will get the * (line mode) prompt back. Then type

EXIT (or quit, if you don't want to save the changes), to get out of the editor.

You may note that;

```
$ EDIT/COMMAND [=FILENAME.EDT]      ;[EDT will read an initial
                                      command file (filename.edt)
                                      before prompting at the
                                      terminal (the default is
                                      the file EDITINI.EDT).]
```

Furthermore, to save the changes in a specific output file, type;

```
$ EDIT/OUTPUT=FILE.EXT
```

For example:

```
$ EDIT AVE.FOR/OUTPUT=NEWFILE.FOR
```

Here, /OUTPUT defines the specification of the file edited. If you don't specify the output file will have the same name as the input file, with a version number (one) higher than the existing file.

HOW TO RECOVER FILES FROM ABNORMAL EXIT:

Use;

```
$ EDIT/RECOVER AVE.FOR
```

Here /RECOVER invokes the EDT for recovering from an abnormal (e.g. computer goes down, line gets discnonected, etc.), exit during previous session. That is EDT reads the the AVE.JOUrnal (from the interupted session and then will continue with the normal editing session.

If the Journal file type is not JOU or the file name is not the same as the input file name, you must include /RECOVER/JOURNAL, e.g.;

```
$ EDIT/RECOVER/JOURNAL=GRADE.YYY AVE.DAT
```

EDT LINE MODE COMMANDS

Following is a list of EDT line mode commands. Type the commands after * (EDT) prompt and, press <return>.

```
* CHANGE                    [OR *C, will change to
                             character mode]

* COPY 5 to 20              [copies lines 5 to 20]

* COPY 5:20                 [same as above]
```

```
* COPY 5:20 TO 8          [copies lines 5 to 20
                           before line 8]

*DELETE 10                [delete line 10]

* DELETE 10:15            [deletes line 10 through 15]

* EXIT                    [or *ex, exits the editor,

* N1:N2                    [lists lines N1 through N2; e.g.
                           *2:8, lists lines 2 through 8]

*N1:END                   [lists lines N1 through end]

* %END                    [lists the last line]

* BEGIN                   [1ST line ot the buffer]

* FIND                    [locates specific line]

* HELP                    [invoke's EDT'S helpl facility]

* INSERT                  [or *I, Insert texts BEFORE
                           current position or line
                           number specified.  To return
                           to * prompt (out of insert mode)
                           press <ctrl>Z]

* INSERT 5                  [or *I5, adds text before line
    New text                5]
    ....
    <ctrl>Z
*
* INCLUDE                   [copies specified file into the
                            buffer]

*

* MOVE 10 TO 5             [moves line 10 to BEFORE line 5]

* MOVE 3:5 TO 11           [or MO, moves lines 3 through 5
                           to before line 11]

* QUIT                      [exits editor without saving
                           the changes]
* REPLACE 10                [deletes line 10 and replaces
                           it by the new text]

*R 10                      [R same as REPLACE]
 1 line deleted
 add new text
.........
.......
<ctrl>Z

* RESEQUENCE               [or *RES, renumbers all lines in
                           steps=1]

* Rest                     [lines including and after current
                           line]
```

```
*
* %WHOLE                          [or *%WH, types the whole file;
                                   all the lines in the buffer]
* TYPE ALL                        [or T A, prints the whole file
                                   all the lines, same as above]

* SUBSTITUTE /OLDTEXT/NEWTEXT/     [or *S]

* S /OLDTEXT/NEWTEXT/%WH          [/%WH, prints the whole file
                                   after the newtext substitution]

* S /OLDTEXT/NEWTEXT/5:15         [list lines 5 to 15 after newtext
                                   substitution]

* S /OLDTEXT/NEWTEXT/WHOLE        [makes the substitution through
                                    out the whole file]

* "string"                        [finds the string]

* WRITE                           [copies a specified range from
                                   the buffer to the specifies file]
```

You may use the help facility, while in the editor.

****

EDT KEYPAD EDITING COMMANDS:

Most keypad keys have 2 editing functions:

a. To use the primary or upper function, press the indicated key. For example; press the (key 7) for PAGE (fuction)

b. To use the secondary or lower fuction of the key, first press GOLD key and then that specific (keypad) key; For example; press GOLD key then, the (key 7) COMMAND, for the command fuction.

UP	DOWN	LEFT	RIGHT
PC-F7 12	PC-F8 13	PC-F5 15	PC-F6 14

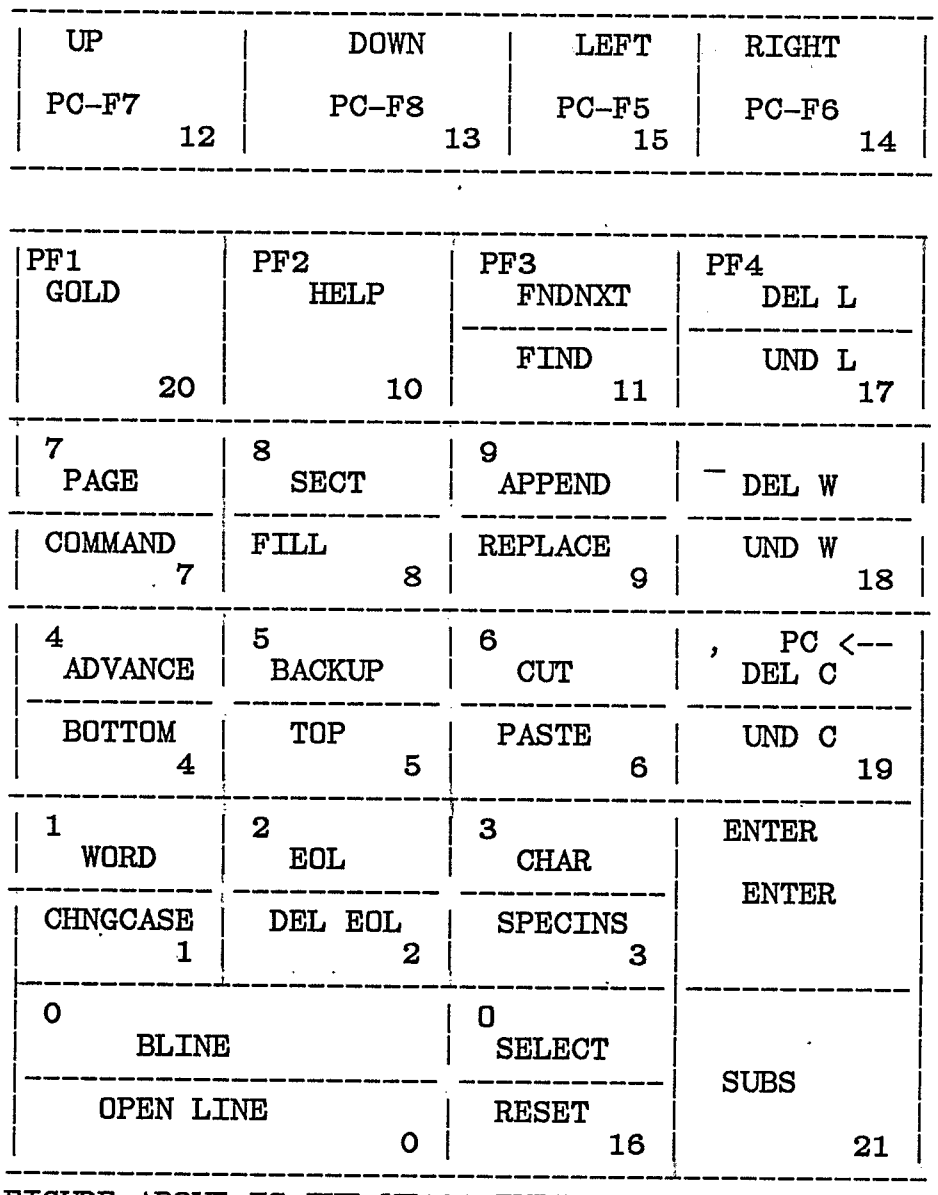


FIGURE ABOVE IS THE VT100-TYPE TERMINAL.;
THE EQUIVALENT KEYS ON PC-KEYPAD ARE INDICATED.
I.E. FOR AN IBM-PC (WITH VT100-EMULATOR) THE KEYPAD
IS THE SAME AS ABOVE, EXCEPT FOR FEW KEYS THAT ARE
LABLED (IN THE FIGURE) ACCORDINGLY. (see below)

****
IBM-PC EDITING KEPAD-------VT100 KEPAD

KEYS LABLED 7 8 9 -
4 5 6
1 2 3 +
0 . [on the PC keypad are identical]'

The backspace arrow of PC (next to NUM LOCK) is DELCHR/UNDLCHR
of VT100.

F1,F2,F3,F4 of PC is the same as PF1,PF2,PF3,PF4 of VT100.
F5 (of PC) = left arrow of VT100
F6 (of PC) = right arrow of VT100

F7 (of PC) = up arrow of VT100
F8 (of PC) = down arrow of VT100.

The return key (PC main keyboard) will give the full keypad editing diagram.

****

FOLLOWING IS THE MOST FREQUENTLY USED KEYPAD COMMANDS:

ADVACE	[moves the cursor in forward direction]
APPEND	[deletes, select range of current buffer, and appends it to the end of the PASTE buffer]
BACKUP	[moves the cursor in backward direction]
BOTTOM	[moves the cursor to the bottom of the buffer]
CHAR	[moves the cursor one character in the current direction]
CHNGCASE	[changes the case of specified characters]
COMMAND	[allows the use of line mode command, without leaving the keypad. Type in response to the COMMAND:]
CUT	[deletes specified test from text from the main buffer and stores it in the paste buffer]
DEL C	[deletes the character at the cursor]
DEL EOL	[deletes text from the cursor to the end of the current line]
DEL L	[deletes the text from the cursor through the end of the current line, moving the following up to the cursor]
DEL W	[deletes text up to the first character of the next word]
DOWN	[down arrow; moves the cursor down to the same position on the next line]
ENTER	[sends commands to EDT for processing. It also completes the ctrl/K key definition process.]
EOL	[moves the cursor to the end of the current line]
FIND	[LOCATES SPECIFIED TEXT]
FNDNXT	[locates the next occurrence of specified text]
HELP	[invokes EDT's keypad help facility]
LEFT	[left arrow; moves the cursor one character to the left]
LINE	[moves the cursor to the beginning of the next line]
PAGE	[moves the cursor to the beginning of thenext line]
PASTE	[insert the contents of the paste buffer at the

	cursor's position]
RESET	[cancels GOLD,SELECT, or any key sequence]
SECT	[moves the cursor across 16 lines of text]
SELECT	[marks one end of a select range. When you move the cursor again, the characters that the cursor passes over become the select range.]
SUBS	[deletes specified text and inserts the contents of paste buffer]
TOP	[moves the cursor to the top of the current buffer]
UND C	[restores the character deleted by the last DEL C]
UND L	[restores the line deleted by the last DEL L]
UND W	[restores the word deleted by the last DEL W]
UP	[up arrow; moves the cursor up the same position on the previous line]
WORD	[moves the cursor one word]
ctrl/K	[DEFINE KEY; activates EDT's define key facility]
ctrl/W	[refereshes the screen, same as ctrl/R]
ctrl/Z	[changes from kepad mode to line editing]

****

HOW TO CREATE A .COM FILE:

You create a .COM file, to avoide the extra work and repetition of entering the same sequence of commands that, you frequently use. In that, you must name the command procedure file, LOGIN.COM. I.e. when you login to the system, the system searches and locates (if such file exits) the LOGIN.COM file, and authomatically executes the commands within it. For example; a login.com file may contain:

```
$ ROOT==" SET DEF BNLDAG::DUA0:[PARSA1]"
$ ASSIGN [PARSA1.HARMON]HARHC.DAT FOR003
$ SET TERM/VT100
```

NOTE: using one = sign, defines a symbol locally.
using two ==sign, defines the symbol globally;

So, use the EDT editor to create a login.com file.
Note the following examples of .COM FILE:

```
$ EDIT PROCESS.COM  <return>
  input file does not exist
  [eob]
*
* i  <return>
    $ FORTRAN/LIST FILE.FOR
    $ PRINT FILE.LIS
    $ LINK FILE.FOR
    $ RUN FILE.FOR

    <CTRL>Z
* EXIT
$
```

(Note, I=insert, puts you in the insert mode, where you don't see the EDT * prompt. After you finished with entering text, you press <cntrl>Z, this gets you out of insert mode, and you'll see the * prompt again. Then you type EXIT, to get out of the editor, and see the $ (VAX) prompt.

Now you can execute all of these commands (in the PROCESS.COM file) by entering:

```
$ @ PROCESS                [each of the above commands will
                            be executed in order defined]
```

1. @ = assumes file type .COM;
2. If you include $SET VERIFY in .COM file you'll see the commands as they are excuted.
3. With SET VERIFY you can use output file

```
$ @PROCESS/OUTPUT=PROCESS.LIST
```

So , the OUTPUT from commands and errors can be saved.

Try the following examples:

```
$ EDIT SHOW.COM
*
*INSERT <return>
     $ SET VERIFY
     $ SHOW TIME
     $ SHOW USERS
     $ SHOW SYSTEM
<ctrl>Z
*
*EXIT
$
```

Now to run the above commands in the file show.com type:

```
  $ @SHOW/OUTPUT=DO.LIS
```

```
Next example,  $ EDIT TEST.COM
               * I <return>
                  $ RUN WEEKCALC
                  $ RUN TIMECARDS
                  $ PRINT WEEKCALC,TIMECARDS
                  <CTRL>Z
                * EXIT  <return>
                $                    [now type:]
                $ @TEST.COM           [or @TEST]
```

TO BATCH THIS JOB, ENTER

```
$ SUBMIT TEST
```

[See next section for SUBMIT command]

HOW TO DEVELOP AND RUN A PROGRAM ON VAX/VMS:

To develop a program on VAX/VMS written in a programming language (default file type), for example BASIC (.BAS); COBAL (.COB); FORTRAN (.FOR); MACRO (.MAR); PASCAL (.PAS); PLI (.PLI); etc.;

1. Create a text file with EDT EDITOR, e.g.;

   files HARMON.FOR, CUPTICS.MAR

2. Compile/ assemble the source program (s);

```
$ FORTRAN HARMON   (this produces HARMON.OBJ)
$ MACRO CUPTICS    (this produces CUPTICS.OBJ)
```

[Note that, you may use qualifiers /LIST, /DEBUG, /CROSS_REFERRENCE, etc. with the compiler /assembler. That is /LIST causes a listing file to be produced as well as object file. For example;
$ FORTRAN/LIST/CROSS_REFERENCE HARMON].

3. Link the compiled program (s) to create an executable image. For example;

```
$ LINK AVE          ;(since default file tyoe is .OBJ, or)
$ LINK AVE.OBJ
```

Another example is;

```
$ LINK HARMON,CUPTICS     (this produces harmon.exe,
                            in which HARMON.FOR and
                             CUPTICS.MAR are linked)
```

You may also, use qualifiers with link command, e.g.;

/MAP (produces a file containing a list of the symbols and data used in the program and their location in the memory);

/CROSS_REFERENCE
(gives list of each global symbols used in the program, its value, the name of the 1st module its defined, and name of each module in which it is referenced);

/DEBUG (causes the linker to i) generate a debug symbol table; ii) gives control to the debuger when the image is run); etc.

4. To execute a program (interactively), e.g. HARMON.EXE enter command RUN;

```
$ RUN HARMON
```

Note, the RUN command assumes the file type of .EXE.

Here, you may use:

```
$ HELP ERROR                          [or;]
$ HELP ERROR SYSTEM ERROR_CODE,
```

to identify the errors outputed at run time, which may not have been identified by compiler /assembler or linker.

5. You may execute a program with a submit command (batch). To do this you must have file.exe, input data file(s) as well as file.com;

Example:

```
$ DIRECTORY [PARSA1.HARMON]

for009.dat;2    harmon.exe;1   harmon.com;5
harmon.dat;2    input.dat;2    instruct.har;5

$ TYPE HARMON.COM

  $ assign INPUT.DAT for003
  $ assign HARMON.DAT for005
  $ run [parsa1.harmon]harmon

$ SUBMIT/NOPRINT/NAME=outputname  HARMON
```

Note that, /NOPRINT and /NAME= , are optional.

1. If /NAME is not used the output will go to HARMON.LOG file in the main directory, here [PARSA1].

2. If /NAME=OUTPUTNAME is used, the output will go to the file OUTPUTNAME.LOG.

3. IF /NOPRINT is not used output will go to the printer.

4. The ERROR file for this program is FOR009.DAT.

PRINT AND COPY COMMANDS:

Copy command can ,"copy an input file to an output file; concatenate two or more input files to a single file; or copy a group of input files to a group of output files."

The format is; $ COPY FROM TO
i.e., $ COPY INPUTFILE OUTPUTFILE

Most general format, includes : nodes, devices, directory and input, output file specs. For example; enter the following and <return>:

```
$ COPY      <return>
from_ NODE1"USER PASSWD"::DEVICE1:[DIR1]INPUTFILE.EXT;N  <RETURN>
to_ NODE2 "USER2 PASSWD2"::DEVICE2:[DIR2]OUTPUTFILE.EXT;N2
```

Note that, you may not need all of the above file specs, since some may be supplied by the system as the default.

Examples:

You are loged in, on bnldag, and like to copy a file from a directory on another node; e.g. bnlcluster2 [i.e. BNLCL2] enter:

```
$ COPY BNLCL2 "USER PASSWD"::$2$DUA7:[ZOHREH]QAZ.TXT *.*

$ COPY QAZ.TXT BNLDAG::DUA0:[PARSA1]*.*
```

Here *; indicates that your outputfile-spec is the same as your input file-spec.

You may also note that, the second COPY command is given from the default directory on BNLDAG, where as the first was from BNLCL2 [for the same file].

You may use qulifiers with the copy command, such as:

/BACKUP /BEFORE /CONCATENATE /EXCLUDE /LOG
/MODIFIED /SINCE /REPLACE, etc.

For example;

/LOG ;verifies that files were copied; it is very usefull, when you copy multiple input and output files to verify.

/REPLACE Replaces an existing version of the output file with the new file. e.g.;

To COPY A FILE FROM MAIL TO YOUR DEFAULT DIRECTORY:

Use EXTRACT out-pufile-name, (SEE INFORMATION ON MAIL);
for example:

```
MAIL> read 2
.......
.....
EXTRACT output-file-name  <return> ;[e.g. o-f-name=NEW.MAI]

[or use;EXTRACT/NOHEADER o-f-name]

MAIL>
MAIL> QUIT
$
```

In the above example, a copy of mail #2, was saved
under the name NEW.MAI;1 in your default directory.

****

TO PRINT FILES ON BNLCLUSTER [BNLCL2]:

(NOTE: The following commands must be
given from the BNLCLUSTER;
Printers are in the Computer Center Bldg.515)

```
a. $ PRINT FILE.EXT;2      [this will print at the center]
b. $ PRINTNL FILE.EXT;2    [this will print on unruled paper]
c. $ PRINTL FILE.EXT;2     [this will print on ruled paper]
d. $ VP FILE.EXT;2         [this will print on VERSATEC printer]
e. $ PRINT/QUEUE=LVA0      [same as (d) above]
f. $ PRINT/QUEUE=LZTA      [prints on the 8 PPM (ready room) printer]
g. $ PRINT/QUEUE=LZT24A    [prints on the 24 PPM (ready room) printer]
h. $ PRINT/QUEUE=LP515     [prints on LP515 printer at the center]
```

** PPM = page/minute

TO PRINT A FILE FROM THE BNLDAG SYSTEM
---

1. $ PRINT file.ext;2impact printer 600 lpm bldg.940
queue name=SYS$PRINT
device name=LP940:

2. $ COPY file.ext;2 LP940:outputs to same 600 line impact printer above

3. $ PRINT/queue=DAGPLOT file.datoutputs to 200nib Versatec in bldg.940, print or plot
queue name=DAGPLOT
device name=LPA0:

4. $ COPY file.ext;2 BNLHEP::LPA0:copy over to BNLHEP Versatec 200 nib printer/plot
queue name=SYS$PRINT
device name=LPA0:

5. $ PRINT/queue=LPAGSQ file.extoutput bldg.911 2nd floor 600 lpm impact printer
queue name=LPAGSQ
device name=LPAGS

6. $ PRINT/queue=LZ911Q file.extoutput bldg.911 2nd floor LN03 DEC LASER PRINTER
queue name=LZ911Q
device name=LZ911:

HOW TO STOP YOUR JOB:

1. TO DELETE YOUR PRINT JOB:

   i. To delete your job while it is printing:

   ```
   $ STOP/ABORT LPA0:
   ```

   ii. To delete your job from a queue:

   ```
   $ DELETE/ENTRY=JOB# LPA0:
   ```

2. To check and fix the printer (e.g. change paper)

   ```
   $ STOP/QUEUE/NEXT LPA0:
   ```

3. To delete the job [on BNLDAG, or with operator priviledge on BNLCLUSTERS]

   ```
   $ DELETE/QUEUE LPA0  (to delete job)
   $ STOP/QUEUE LPA0 (to change paper)
   $ START/QUEUE LPA0
   ```

4. TO STOP A CURRENTLY EXECUTING JOB:

   ```
   $ STOP/QUEUE/ENTRY=JOB# SYS$BATCH     ;or
   $ DELETE/ENTRY=JOB# SYS$BATCH
   ```

5. To get the JOB#, use:

   ```
   $ SHOW QUEUE    ;[this shows all YOUR job in all QUEUES]
   $ SHOW QUEUE/ALL
   $ SHOW QUEUE SYS$BATCH
   $ SHOW QUEUE SYS$PRINT
   ```

FILE PROTECTION

The system provides a default protection code for files you create. But you can define the protection you want [for your (private) files] to be applied. There are four classes : SYSTEM (S), OWNER (O, or user), GROUP (G), and WORLD (W). And 4 categories; READ (R), WRITE (W), EXECUTE (E), and DELETE (D).

1. To check the current protection on a specific file or files, use:

```
$DIR/PROTECTION FILE.EXT
```

2. To show the system default protection use:

```
$ SHOW PROTECTION
```

3. To change the protection and give RWED permission to all, use:

```
$ SET PROTECTION = (SYSTEM:RWED,OWNER:RWED,GROUP:RWED,
  _WORLD:RWED)
```

Or for short use (same as above):

```
$ SET PROTECTION = (S:RWED,O:RWED,G:RWED,W:RWED)
```

You may deny any of the RWED permission's to any of the four classes; S,O,G,W (described above), by deleting the specific category, or class.

For example;

```
$SET PROT=(SYSTEM:REW,OWNER:RWED,GROUP:RE,WORLD)-
- AVE.FOR
```

allows the SYSTEM to READ, WRITE, EXECUTE but not delete the file AVE.FOR, whereas the OWNER have ALL types of access to AVE.FOR. Here the GROUP can READ and EXECUTE but can't delete or write. Finally, the WORLD have NO ACCESS to the file AVE.FOR. To check this current default protection type, use:

```
$ SHOW PROTECTION  (you should see)

 system=rwe  ,  owner=rwed,  group=re,  world=noaccess
```

or if you use the followng command, you should see:

```
$ DIRECTORY/PROTECTION AVE.FOR
  directory dua0:[parsa1]
  ave.for;4  (rwe,rwed,re,)
  total of 1 file, 6 blocks.
```

4. To specify the protection code to be applied to all

files that you subsequently create during a terminal session or batch job, use:

$ SET PROTECTION/DEFAULT command.

****

USEFUL EXAMPLES:
Following is an example of purge command:

```
$ dir
Directory DUA0:[PARSA1.GUIDE]
GUIDE.VAX;16    GUIDE.VAX;15      GUIDE.VAX;14    GUIDE.VAX;13
GUIDE.VAX;12    GUIDE.VAX;11      GUIDE.VAX;10    GUIDE.VAX;9
GUIDE.VAX;8     GUIDE.VAX;7       GUIDE.VAX;6     GUIDE.VAX;5
GUIDE.VAX;4     GUIDE.VAX;3       GUIDE.VAX;2     GUIDE.VAX;1
SHOW.LIS;2      SHOW.LIS;1        TEST.LIS;4      TEST.LIS;3
TEST.LIS;2      TEST.LIS;1
Total of 22 files.

$PURGE/KEEP=3 GUIDE.VAX        ;[keeps 3 (highest version) copies]

$ DIR
Directory DUA0:[PARSA1.GUIDE]

GUIDE.VAX;16    GUIDE.VAX;15      GUIDE.VAX;14      SHOW.LIS;2
SHOW.LIS;1      TEST.LIS;5        TEST.LIS;4        TEST.LIS;3
TEST.LIS;2      TEST.LIS;1
Total of 10 files.
```

****

EXAMPLES: This section demonstrate some of the commands used through out this manual.

```
$ SET VERIFY

$ SHOW TIME
14-OCT-1985 21:30:42

$ SET TERM/VT100/INSERT/NOWRAP
```

****

TERMINAL CHARACTERISTICS:

How to check and set the the terminal characteristics?

The operating characteristics varies for differint types of terminals. But you can change some of these features, depending on your need, with he SET TERMINAL command. To check the current

setting of your terminal use SHOW TERMINAL command.

```
$ SHOW TERMINAL

  Terminal: _TTA3:     Device_Type: VT100       Owner: PARSA1

   Input: 9600      LFfill: 0  Width: 80   Parity: None
   Output: 9600     CRfill: 0     Page:   24

   Terminal Characteristics:

       Interactive        Echo           Type_ahead      No Escape
       Hostsync           TTsync         Lowercase       Tab
       No Wrap            Scope          Remote          No Holdscreen
       No Eightbit        Broadcast      No Readsync     No Form
       Halfdup            No Modem       No Local_echo   No Autobaud
       No Hangup          No Brdcstmbx   No DMA          No Altypeahd
       Set_speed          No ANSI_Crt    No Regis        No Block_mode
       No Advanced_video  No Edit_mode   No Dec_Crt
```

Note that, most of the parameters displayed by the SHOW TERMINAL command can be changed by corresponding qualifiers for the SET TERMINAL command. For example, to change from uppercase to lowercase letters enter:

```
$ SET TERMINAL/LOWERCASE
```

now, the lowercase alphabetic characters will be displayed (if your terminal has capability of displaying lowercase characters).

Note, the changes you make using the SET TERMINAL command will not be saved, and will only apply to your current session.

Following, is the brief description of the TERMINAL FUNCTIONS KEYS most commonly used:

KEY	FUNCTION
RETURN	Transmits the current line to the system for processing (On some terminals, the RETURN key is labeled CR.) Before a terminal session, initiates login sequence.
Control Keys	Define functions to be performed when the CTRL key and another key are pressed simultaneously. All CTRL/x key sequences are echoed on the terminal as ^x.
CTRL/C and CTRL/Y	During command entry, cancels command processing

	Before a terminal session, initiates login sequence
	Interrupts command or program execution and returns control to the command interpreter
CTRL/I	Duplicates the function of the TAB key
CTRL/K	Advances the current line to the next vertical tab stop
CTRL/L	Form feed
CTRL/O	Alternately suppresses and continues display of output to the terminal
CTRL/Q	Restarts terminal output that was suspended by CTRL/S
CTRL/R	Retypes the current input line and leaves the cursor positioned at the end of the line
CTRL/S	Suspends terminal output until CTRL/Q is pressed
CTRL/T	Momentarily interrupts terminal output to display a single line of statistical information about the current process
CTRL/U	Discards the current input line
CTRL/X	Discards the current line and deletes data in the type-ahead buffer
CTRL/Y	(See CTRL/C)
CTRL/Z	Signals end-of-file for data entered from the terminal
DELETE	Deletes the last character entered at the terminal and backspaces over it (On some terminals, the DELETE key is labeled RUBOUT.)
ESCAPE	Has special uses to particular commands or programs, but generally performs the same function as RETURN (On some terminals, the ESCAPE key is labeled ALTMODE or ESC (SEL).)
TAB	Moves the printing element or cursor on the terminal to the next tab stop on the terminal. The system provides tab stops at every eighth character position on a line.

```
$ SHOW USERS
  Username      Process Name      PID      Terminal
  766X1         766X1           000000AB   TXD1:
  HOFF          _TXD0:          000000A6   TXD0:
```

```
PARSA1        PARSA1          00000210  TTB5:


$ SHOW PROCESS
14-OCT-1985 21:30:42.98   TTB5:              User: PARSA1
Pid: 00000210   Proc. name: PARSA1           UIC: [370,373]
Priority:   4   Default file spec: DUA0:[PARSA1.GUIDE]
Devices allocated: TTB5:


$ SHOW SYSTEM
VAX/VMS V4.1  on node BNLDAG 14-OCT-1985 21:30:44.35 Uptime  0 11:20:11
Pid    Process Name    State  Pri   I/O       CPU        Page flts Ph.Mem
00000080 NULL          COM  0        0   0 01:10:09.37         0       0
00000210 PARSA1        CUR  4      410   0 00:00:08.54      1971     229
00000199 CDCLINK       HIB  9     1150   0 00:00:10.82       691     200
00000231 BATCH_399     COM  1     1588   0 03:33:16.43      2182     634  B

$ SHOW TIME
14-OCT-1985 14:01:18

$ SHOW USERS
      VAX/VMS Interactive Users  14-OCT-1985 14:01:19.20
  Total number of interactive users = 14
  Username      Process Name      PID      Terminal
  766X1         766X1           000000AB  TXD1:
  PARSA1        PARSA1          000001C5  TXA3:
  POPKEN        POPKEN          000001C3  TXB4:


 $ SHOW SYSTEM
 VAX/VMS V4.1  on node BNLDAG 14-OCT-1985 14:01:19.58 Uptime  0 03:50:46
 Pid    Process Name  State  Pri      I/O       CPU     Page flts Ph.Mem
 0000013D _TTB0:       LEF      9     1004   0 00:00:19.22   1006     137
 000001C3 POPKEN       LEF      9      134   0 00:00:03.51    793     141
 000001C5 PARSA1       CUR      4      170   0 00:00:04.12    963     205


  $ SHOW LOGICAL
   (LNM$PROCESS_TABLE)

   "SYS$COMMAND" = "_TXA3:"
   "SYS$DISK" = "DUA0:"
   "SYS$ERROR" = "_TXA3:"
   "SYS$INPUT" [super] = "_DUA0:"
   "SYS$INPUT" [exec] = "_TXA3:"
   "SYS$OUTPUT" [super] = "_DUA0:"
   "SYS$OUTPUT" [exec] = "_TXA3:"
   "TT" = "TXA3:"

   (LNM$JOB_801D5AF0)

   "SYS$LOGIN" = "DUA0:[PARSA1]"
   "SYS$LOGIN_DEVICE" = "DUA0:"
   "SYS$SCRATCH" = "DUA0:[PARSA1]"

   (LNM$GROUP_000370)
```

EXAMPLES OF HOW TO GET INFORMATION (AVAILABLE ON LINE):

$TYPE BNL$MANUAL:INDEX.MAN

```
INDEX.MAN  INDEX TO ALL OLDF HELP FILES06/24/8 VERSION 11

THE FOLLOWING HELP FILES PROVIDE OLDF INSTALLATION SPECIFIC
INFORMATION. SOME ARE COMPUTER SPECIFIC AND MAY NOT BE ON THE
COMPUTER YOU ARE ON. THESE FILES ARE ON BNL$MANUAL: DISK, SO TO
TYPE OR PRINT THE FILES, FOR EXAMPLE, DO

.TYPE BNL$MANUAL:TERMINAL.MAN!OR
.PRINT BNL$MANUAL:VAX10.MAN


1)VAX HELP FILES
BITNET.MANBITNET network
BITNETNODES.MAN List of all nodes on the BITNET network
BNLDAG.MANBNLDAG CONFIGURATION, RESOURCES,ETC.
CDCLINK.MANSEE VAXCDC.MAN
CERNLIB.MANCERN PROGRAM LIBRARY
COMDECK.MANPDP10 VERSION OF CDC'S COMDECK
CYBERVAX.MANDECNET/HYPERCHANNEL LINK BETWEEN VAXES AND CDC7600
GMAIL.MANGATEWAY MAIL TO BITNET AND NETWORKS GATED TO BITNET
HANDYPAK.MANSLAC'S HISTOGRAM/SCATTER-PLOT PRINTER-GRAPHIC PACKAGE
INDEX.MANTHIS FILE
KERMIT.MANINTERCOMPUTER FILE TRANSFER (TEST AND BINARY) PROGRAM
LEXIDA.MANLEXIDATA GRAPHICS PACKAGE
NSC16000.MANNSC16000 MICROCOMPUTER CROSS SOFTWARE PACKAGE
PHYSNET.MANNATIONAL DECNET NETWORK OF PHYSICS COMPUTERS
PHYSNETNODES.MAN list of all nodes on the PHYSNET network
POLICY.MANOLDF FACILITY COMPUTER POLICIES
SENDGATE.MANOBSOLETE, SEE GMAIL.MAN
SIR.MANSIR SCIENTIFIC DATA BASE MANAGEMENT SYSTEM
TERMINAL.MANBNL DIAL-UP AND GANDALF TERMINAL SYSTEM
UGSYS.MANSLAC'S UNIFIED GRAPHICS DISPLAY PACKAGE
VAX10.MANBNLDAG AND PDP10-SYSTEM-A FILE TRANSFER LINK (OBSOLETE)
VAXCDC.MANBNLDAG AND CDC6600/7600 LINK--OBSOLETE
VAXNET.MANFILE TRANSFER OVER ASYNC COMM LINES BETWEEN ANY COMPUTERS
XMLINK.MANSPECIAL PDP11-VAX REAL TIME LINK
```

$TYPE BNL$MANUAL:CERNLIB.MAN

Organization of the CERN Program Library

The CERN Program Library has recently been reorganized at CERN for ease of maintainance and portability. We will set it up in the same way, keeping to the CERN naming conventions as far as possible.

The "Program Library" is seen as a multi-layered object:

Level 1 - A compact nucleus library containing small self-contained utility subroutines (e.g., UZERO, UCOPY, DATIME). This library is viewed as an extension of the FORTRAN language into the domains of mathematics, data handling and operating system interfaces.
File name: BNL$CERNLIB:KERNLIB.OLB

Level 2 - A library containing the frequently used large utility packages (e.g., HBOOK, GD3). Members of this library may only resolve external references from the nucleus library.
File name: BNL$CERNLIB:PACKLIB.OLB

Level 3 - The remaining general library routines (e.g., phase space routines, specialized random number generators).
File name: BNL$CERNLIB:GENLIB.OLB

You may wish to add the following lines to your LOGIN.COM file:

```
$ ASSIGN BNL$CERNLIB:GENLIB  LNK$LIBRARY
$ ASSIGN BNL$CERNLIB:PACKLIB LNK$LIBRARY_1
$ ASSIGN BNL$CERNLIB:KERNLIB LNK$LIBRARY_2
$ ASSIGN BNL$CERNLIB:HYRUNLIB LNK$LIBRARY_3
```

This will cause LINK to automatically search the CERN libraries before searching the system libraries. (See section 3.3 in the LINK reference manual for full details, or type HELP LINK /USERLIB.)

Also available is a COMmand file containing these ASSIGNments, so you may instead use:

```
$ @BNL$CERNLIB:CERNLIB
```

Likewise, you may wish to add the following COMmand file containing PATCHY assignments to your LOGIN.COM file:

```
$ @BNL$CERNLIB:PATCHY
```

Composition of CERN Libraries -- Feb. 85

KERNLIB:	KERNVAX	2.10
	KERNFOR	4.09
	KERNNUM	2.09
PACKLIB:	HBOOK	3.37
	HPLOT	4.10
	ZBOOK	2.13
	MGD3	3.02
	EPIO	1.51
	FFREAD	2.00
	KAPACK	2.00
	+HIV	2.07
	+ZCDEDEX	2.51
GENLIB:	GEN	1.14
	MINUIT	1.04
HYDRA:	HYCDE.PAM	
	HYRUNLIB.OLB	
	HYRUNJQ81.OBJ	
	HYRUNJQB.OLJ	
	HYRUNJQE.OBJ	
	HYRUNRQ.OBJ	
	HYRUNZQ.OBJ	

PATCHY Version 4.10

$TYPE BNL$MANUAL:GMAIL.MAN

GMAIL.MAN	Gateway Mail	Send mail through Bitnet gateways
	ri v1.10	20-Jun-1986

1. ABSTRACT

GMAIL is a utility, implemented by a DCL command file, designed primarily to support the SENDing of mail messages to nodes on networks other than those directly supported by VMS MAIL and its jnet (or other) extension for BITNET. The user interface for GMAIL is patterned closely after (the sending functions of) VMS MAIL. GMAIL fully supports address lists and distribution files, and can handle a mix of 'foreign-network' addresses together with addresses which can be handled by VMS MAIL. (GMAIL invokes VMS MAIL to deal with those addresses which are VMS MAIL addresses.)

GMAIL runs only on BITNET nodes, so BNLDAG currently is the only computer at the Lab to implement GMAIL.

2. INTRODUCTION

The GMAIL utility supports both VMS MAIL addresses (local,

DECnet, jnet%, or any other VMS MAIL defined protocols) and GMAIL addresses (mail to be sent thru BITNET via a gateway to another network). GMAIL will analyze each address that it is given and, depending upon the address syntax, either (1) use VMS MAIL to send the letter to that address or (2) prepare the letter with the proper header and send it to the appropriate gateway (BITNET node and 'user') to be forwarded to another network.

Since GMAIL is implemented with DCL, it is NOT blazingly fast.

Some features of GMAIL

- Includes a comprehensive HELP file.
- Can be used with 'jnet' or Argonne NJE RSCS network software. Has capability of a user-supplied sending procedure as well.
- Supports many gateways (see HELP topic 'Gateways') and it is fairly easy to add additional gateways (in most cases) by defining appropriate logical names. Routing to gateways often is determined by the 'domains' in an address, rather than being explicit.
- Fully supports all address-specification forms (logical names, lists, distribution files) of VMS Mail. See HELP topic 'Addresses General'.
- Options generally controllable both by logical names and (DCL-stlye) command qualifiers (see HELP topic 'Flags_Modifiers').
- Method of REPLYing to foreign-mail received by VMS MAIL (see HELP topic 'REPLY_TO_$').
- Optional check of letter for format unsuitable for sending via BITNET (see HELP topic 'SEND /CHKLETTER' or 'GMAIL /CHKLETTER').
- Both 'command' mode and 'DCL' mode, in analogy with VMS MAIL (see HELP topics 'GMAIL' and 'Modes').
- Optional inclusion of 'Personal Name' in mail header (see HELP topic 'Flags_Modifiers PERSONAL_NAME').
- A 'test' mode, where no letter is actually sent (see HELP topic 'SEND /TESTMODE' or 'GMAIL /TESTMODE') and the option of preserving 'work' files, so that you can see what was (or would have been) sent (see HELP topics 'SEND /DELETEFILES' and 'GMAIL /DELETEFILES').
- Optional use of an editor when composing a letter (see HELP topic 'SEND /EDIT' or 'GMAIL /EDIT').
- Self-copy option (see HELP topic 'SEND /SELFCOPY' or 'GMAIL /SELFCOPY').
- Optional inclusion of distribution list with letter text (see subtopics '/DISLIST' and '/DISFILE' under HELP topic 'SEND' or 'GMAIL').

- Execution of any DCL command from GMAIL command prompt (see HELP topic 'DCL_command').

GMAIL is intended to be a successor to SENDGATE.COM. If you are a user of that procedure, refer to HELP topic 'Release_notes SENDGATE' for some conversion tips.

3. GMAIL COMMAND

The DCL command GMAIL invokes the GMAIL facility, which is used primarily to send letters to users on other systems which are NOT accessible by either DECnet or BITnet. However, it can be used to send to DECnet and BITnet addresses as well, hence distribution lists which include both classes of addresses are supported.

Format:

```
$ GMAIL [command-qualifiers] [file-spec] [address-spec]
```

Parameters

file-spec

Specifies an optional file containing message text to be sent to the specified user(s). If you omit the file type, the default file type is TXT.

If you omit the file-spec parameter, GMAIL is invoked to process GMAIL commands interactively (provided your process is INTERACTIVE).

Neither wild card characters nor lists are allowed in the file specification.

address-spec

Specifies one or more users to receive the message. An address-spec is a user address or a list of such addresses. Addresses (or lists of addresses) may be denoted by logical names. See topic 'Addresses' for a description of the the various formats supported for user addresses.

If you omit this parameter, you will be prompted for it (provided your process is INTERACTIVE).

As an alternative to listing the user addresses, you can specify a distribution list of user addresses. Simply precede the distribution list file specification with an at sign (@) and enclose this construction in quotation marks ("). The file you specify should contain the user addresses; multiple addresses may be entered on separate lines or collected on one or more lines with comma separators. If you omit the file type, the default file type is DIS. No wild card characters are allowed in the distribution list file specification. Distribution lists may be nested.

Your are prompted for everything. So the more typical use of GMAIL would be to type in GMAIL and then type in one of the GMAIL commands: SEND, HELP, REPLY,EXIT, or QUIT. For example,

```
$ GMAIL
Gmail> SEND
```

.
.

4. ADDRESSING

GMAIL support two addressing schemes

- o NODE::USER for DECnet/PHYSnet addressing
- o USER@NODE.DOMAIN for foreign/bitnet addressing

The address syntax that GMAIL supports for 'foreign-network' addresses is most often in the form 'user@host.domain', for instance, 'USERX@NODEY.ARPA'. Such mail is handled by combining your letter with a suitable header, and sending it (using a form of the jnet SEND/FILE command or equivalent) via BITNET to a 'gateway' node, from where it is forwarded to its ultimate destination.

Ideally, if you receive a letter from a user on another network, and the letter is indicated as being from 'xxxx@yyyy', then you should be able to reply by specifying "xxxx@yyyy" as the address. Unfortunately, there may be many exceptions to this ideal case. Consequently, GMAIL offers you a number of alternatives to express a correct address. See HELP topic REPLY_TO_$.

5. GATEWAYS

GMAIL supports gateways to many networks such as ARPANET, MILNET, UUCP, MFENET, JANET, MAILNET, CCNET, CSNET. See HELP topic GATEWAYS. The BITNET network server, NICSERVE will send you a list of BITNET gateways by executing the JNET/BITNET command:

```
$ SEND NICSERVE@BITNIC SEND BITNET GATES
```

GMAIL conforms to most of the recommendations in this file, but not all. With each gateway is listed a 'contact' address--if you have problems with a gateway you may be able to get help by sending an inquiry to this address. This file may also give you suggestions as to alternative ways to send to certain addresses.

6. EXAMPLES

Here are examples using GMAIL interactively, allowing it to prompt for information.

1.

```
$ GMAIL
gMail>
```

This GMAIL command invokes GMAIL to process commands interactively.

2.

```
gMail> SEND GEOMCON
gTo:   myers@sparta.uucp
gSubj: Geometric Concepts
 ...
gMail>
```

This example shows how to send a copy of the file GEOMCON.TXT to a user named Myers on node 'sparta' in the UUCP network.

3.

```
gMail> SEND/SELF
gTo:    Ahab@Pacific.Ocean.mil
gSubj:  Good Harbor
gEnter your message below. Press CTRL/Z when complete, or CTRL/C to quit:
```

This example shows how to send a mail message to a user named Ahab on node Pacific.Ocean.mil (on Arpanet). The /SELF qualifier instructs GMAIL to send a copy of the same message back to you. The subject of the message is Good Harbor.

4.

```
gMail> SEND
gTo:    BAKER,CHARLIE,@SUPERVISORS
gSubj:  Handling Stress
gEnter your message below. Press CTRL/Z when complete, or CTRL/C to quit:
```

This example shows how to send a mail message to two users (BAKER and CHARLIE) and a distribution list (SUPERVISORS).

Here are examples where the entire mailing is done in the GMAIL command line. This is used in noninteractive situations such as from batch jobs or spawned jobs or command file executions.

1.

```
$ GMAIL/SUBJECT="New Project" PROJECT.DOC -
$_"HAYDEN,""SAVIO@FSM.BERKELEY.EDU"""
```

This GMAIL command specifies that the file named PROJECT.DOC is to be sent to users HAYDEN and SAVIO, with a subject description of New Project in the heading.

2.

```
$ GMAIL/SUBJECT="Vacation Policy Change" NEWSLETTR "@LOSERS"
```

This GMAIL command invokes GMAIL to send the file NEWSLETTR.TXT to all the users named in the file LOSERS.DIS. The subject description is Vacation Policy Change.

.7 RECEIVING MAIL

You read mail using the normal VMS MAIL utility. For example:

```
$ MAIL
MAIL> READ
```

You can have all your mail on every computer on DECNET sent to your account on one computer by using the SET FORWARD command. For example, to forward all mail to the BNLHEP, you (say JONES) would log into all the other VAXes and do

```
$ MAIL
MAIL> SET FORWARD BNLHEP::JONES
MAIL> EXIT
```

To rescind the forwarding do SET NOFORWARD. A negative thing about SET FORWORD is that you lose mail if the target node is down.

Both the MAIL and GMAIL commands enable you to append a field to the end of the "From:" field of mail messages sent out. You can fill this field with your full name or any other information adding up to 127 characters. Use the Mail utility to permanently store this field, for example,

```
$ MAIL
MAIL> SET PERSONAL_NAME "Jill Jones; Brookhaven Nat Lab; Phone 999-99
9-9999; Bitnet: JONES@BNLDAG; Physnet: BNLDAG::JONES;"
MAIL> SHOW PERSONAL_NAME
```

To rescind the personal name do SET NOPERSONAL_NAME.

8. REFERENCES

- See this document BNL$MANUAL:GMAIL.MAN and if you are even more ambitious see the GMAIL command file itself in DUA0:[GMAIL]GMAIL.COM.

- GMAIL's HELP command allows you to obtain information about the GMAIL Utility. To obtain information about all of the GMAIL commands, enter the following command:

  ```
  gMail> HELP *...
  ```

  To obtain information about individual commands or topics, enter HELP followed by the command or topic name.

  ```
  gMail> HELP [topic [subtopic...]]
  ```

  If this is your first introduction to GMAIL, then topic 'Getting_started' is a good place to begin. Note that you can output this HELP file (or a part of it) to a printable file by using the following sequence of commands:

  ```
  gMail> $DEFINE/USER SYS$OUTPUT your-file
  gMail> HELP *...
  Topic? ^Z
  gMail> $PRINT/DELETE your-file
  ```

  Notice that GMAIL allows execution of DCL commands within itself if you prefix the DCL command with a dollar sign, "$".

- NETSERV is a standard file server and user directory service which exists on a number of key BITNET nodes. For a list of these servers, (and much, much more) request the BITNET SERVERS file from one of them. For example (using the jnet SEND command):

  ```
  $ SEND NETSERV@BITNIC SEND BITNET SERVERS   ! (USA)
  $ SEND NETSERV@DEARN  SEND BITNET SERVERS   ! (Germany)
  ```

  To find out how to use a NETSERV server, send the HELP command. For example:

  ```
  $ SEND NETSERV@BITNIC  HELP
  ```

  The BITNIC node (USA BITNET Network Information Center) has some additional sources of information. Find out more about two of these with:

  ```
  $ SEND NICSERVE@BITNIC  HELP
  $ SEND DATABASE@BITNIC  HELP
  ```

  For the UUCP network, there is a server at Penn State which can supply you with information about a specific node.

  ```
  $ SEND @PSUVAX1 UUPATH node     Gives path (if known) from PSUVAX1
                                    to the specified node.
  $ SEND @PSUVAX1 UUHOSTS node    Gives miscellaneous information about
                                    the specified node.
  ```

- See BNL$MANUAL:BITNET for information on the JNET software that supports BITNET. It is used by GMAIL to send mail to

the gateway nodes.

- To contact the creator and maintainer of GMAIL see
  Ed Miller, SLAC
  Physnet: TWGM::ESMP09
  BITNET: ESMP09@SLACVM

$ SHOW NET

VAX/VMS Network status for local node 43.30 BNLDAG
on 22-MAY-1987 16:53:32.82

The next hop to the nearest area router is node 43.71 BNLRS2.

Node		Links	Cost	Hops	Next Hop to Node		
43.30	BNLDAG	0	0	0	(Local)	-> 43.30	BNLDAG
43.10	BNLAMM	0	3	1	UNA-0	-> 43.10	BNLAMM
43.11	BNLCL1	1	3	1	UNA-0	-> 43.11	BNLCL1
43.12	BNLCL2	0	3	1	UNA-0	-> 43.12	BNLCL2
43.13	BNLCL3	1	3	1	UNA-0	-> 43.13	BNLCL3
43.14	BNLCL4	0	3	1	UNA-0	-> 43.14	BNLCL4
43.16	BNLAMD	0	3	1	UNA-0	-> 43.16	BNLAMD
43.31	BNLHEP	1	3	1	UNA-0	-> 43.31	BNLHEP
43.32	BNLMPS	0	3	1	UNA-0	-> 43.32	BNLMPS
43.34	BNL787	0	3	1	UNA-0	-> 43.34	BNL787
43.35	BNLMP1	0	3	1	UNA-0	-> 43.35	BNLMP1
43.37	BNLK01	0	3	1	UNA-0	-> 43.37	BNLK01
43.41	BNL791	0	3	1	UNA-0	-> 43.41	BNL791
43.42	BNLKME	2	3	1	UNA-0	-> 43.42	BNLKME
43.50	BNL814	0	3	1	UNA-0	-> 43.50	BNL814
43.59	NYUHEP	0	4	2	UNA-0	-> 43.61	BNLD0R
43.60	BNLD01	0	3	1	UNA-0	-> 43.60	BNLD01
43.61	BNLD0R	0	3	1	UNA-0	-> 43.61	BNLD0R
43.62	BNLD02	0	3	1	UNA-0	-> 43.62	BNLD02
43.63	BNLD03	0	3	1	UNA-0	-> 43.63	BNLD03
43.70	BNLRS1	0	3	1	UNA-0	-> 43.70	BNLRS1
43.71	BNLRS2	0	3	1	UNA-0	-> 43.71	BNLRS2
43.101	SBNUC1	1	7	3	UNA-0	-> 43.71	BNLRS2
43.111	SBNSLA	0	7	3	UNA-0	-> 43.71	BNLRS2
43.112	SBNSLB	0	7	3	UNA-0	-> 43.71	BNLRS2
43.113	SBNSLC	0	7	3	UNA-0	-> 43.71	BNLRS2
43.114	SBNSLD	0	7	3	UNA-0	-> 43.71	BNLRS2
43.120	SBHEP	0	4	2	UNA-0	-> 43.71	BNLRS2
43.121	SBHEP1	0	7	3	UNA-0	-> 43.71	BNLRS2
43.122	SBHEP2	0	4	2	UNA-0	-> 43.71	BNLRS2
43.133	BNLX26	0	3	1	UNA-0	-> 43.133	BNLX26
43.134	BNLX1	0	3	1	UNA-0	-> 43.134	BNLX1
43.136	BNLX17	0	3	1	UNA-0	-> 43.136	BNLX17
43.137		0	3	1	UNA-0	-> 43.137	
43.170	NEVIS	1	4	2	UNA-0	-> 43.71	BNLRS2
43.171	NEVIS1	0	7	3	UNA-0	-> 43.71	BNLRS2
43.180	JHUP	0	4	2	UNA-0	-> 43.71	BNLRS2
43.181	JHUP1	0	14	3	UNA-0	-> 43.71	BNLRS2
43.200	UPENN1	0	4	2	UNA-0	-> 43.61	BNLD0R
43.202	UPENN3	0	7	3	UNA-0	-> 43.61	BNLD0R

43.203	UPENN4	0	7	3	UNA-0	-> 43.61	BNLDOR
43.204	UPENN5	0	7	3	UNA-0	-> 43.61	BNLDOR
43.210	BRHEP1	0	4	2	UNA-0	-> 43.61	BNLDOR
43.213	MIC3	0	8	3	UNA-0	-> 43.61	BNLDOR
43.220	LUKE	0	8	3	UNA-0	-> 43.61	BNLDOR
43.221	DARTH	0	8	3	UNA-0	-> 43.61	BNLDOR
43.226	LOLA	0	8	3	UNA-0	-> 43.61	BNLDOR
43.250	CUSB85	0	4	2	UNA-0	-> 43.61	BNLDOR
43.251	LNS61	0	7	3	UNA-0	-> 43.61	BNLDOR
43.252	LNS62	0	7	3	UNA-0	-> 43.61	BNLDOR
43.253	CUSB2A	0	7	3	UNA-0	-> 43.61	BNLDOR
43.254	CUSB2B	0	7	3	UNA-0	-> 43.61	BNLDOR
43.255	CUSB2C	0	7	3	UNA-0	-> 43.61	BNLDOR
43.405	BNLNDC	0	4	2	UNA-0	-> 43.70	BNLRS1
43.408	BNLAKE	0	14	3	UNA-0	-> 43.70	BNLRS1
43.409	BNLPDB	0	14	3	UNA-0	-> 43.70	BNLRS1
43.410	BNLCHM	0	4	2	UNA-0	-> 43.70	BNLRS1
43.411	BNLITHC	0	14	3	UNA-0	-> 43.70	BNLRS1
43.412	BNLCHA	0	14	3	UNA-0	-> 43.70	BNLRS1
43.414	BNLITHM	0	14	3	UNA-0	-> 43.70	BNLRS1
43.415	BNLNC1	0	14	3	UNA-0	-> 43.70	BNLRS1
43.416	BNLCHW	0	14	3	UNA-0	-> 43.70	BNLRS1
43.417	BNLPB1	0	14	3	UNA-0	-> 43.70	BNLRS1
43.418	BNLPB2	0	14	3	UNA-0	-> 43.70	BNLRS1
43.419	BNLPB3	0	14	3	UNA-0	-> 43.70	BNLRS1
43.420	BNLBIO	0	4	2	UNA-0	-> 43.70	BNLRS1
43.422	BNL421	0	8	3	UNA-0	-> 43.70	BNLRS1
43.440	BNLHYP	0	4	2	UNA-0	-> 43.71	BNLRS2
43.442	BNL734	0	3	1	UNA-0	-> 43.442	BNL734

Total of 69 nodes.

******** SUMMARY OF DCL COMMANDS *********

FOLLOWING SECTION, GIVES A BRIEF DESCRIPTION OF DCL COMMANDS:

COMMAND	FUNCTION
Assignment statements	Assign strings as synonyms for all or a portion of a DCL command
=	General purpose assignment - equates a local symbol name to a character string expression or an integer expression
==	General purpose assignment - equates a global symbol name to a character string expression or an integer expression
:=	String assignment - defines a local symbol name as a synonym for a character string
:==	String assignment - defines a global symbol name as a synonym for a character string
@ (execute procedure)	Executes a command procedure or places data from a command file into the input stream
ACCOUNTING	Selects, sorts, lists, and summarizes accounting data
ALLOCATE	Reserves a device for use by a single user and, optionally, assigns a logical name to the device
ANALYZE/CRASH_DUMP	Invokes the System Dump Analyzer (SDA) to examine the specified dump file
ANALYZE/DISK_STRUCTURE	Invokes the VAX-11 Verify Utility to check the readability and validity of Files-11 Structure Level 1 and Files-11 Structure Level 2 disk volumes
ANALYZE/IMAGE	Provides a description of the contents of an image file
ANALYZE/OBJECT	Provides a description of the contents of an object file
ANALYZE/RMS_FILE	Inspects and analyzes the internal structure of a VAX-11 RMS file
ANALYZE/SYSTEM	Invokes the System Dump Analyzer (SDA) to examine the running system
APPEND	Adds the contents of one or more files to the end of another file
ASSIGN	Defines a file specification or a device name to be associated with a logical name for subsequent use in commands or programs

ASSIGN/MERGE	Removes the jobs from one queue and places them in another queue
ASSIGN/QUEUE	Assigns a logical queue to a device
ATTACH	Permits you to switch control of your terminal from your current process to another in your job
BACKUP	Invokes the BACKUP Utility to save or restore files from tape or disk
BASIC	Invokes the VAX-11 BASIC compiler to enter and compile BASIC language source statements
BASIC/RSX11	Invokes the PDP-11 Basic-Plus-2 compiler to begin a BASIC session
BLISS	Compiles one or more BLISS-32 or common BLISS source programs
CANCEL	Halts periodic execution of an image scheduled for execution in a process
CC	Invokes the VAX-11 C compiler to compile C source programs
CLOSE	Cancels an input or output path to sequential file or device
COBOL	Compiles COBOL language source statements
COBOL/C74	Compiles COBOL language source statements
CONTINUE	Resumes execution of an interrupted command, program, or command procedure
CONVERT	Copies records from one file to another, changing the organization and format of the input file
CONVERT/RECLAIM	Makes empty buckets in Prolog 3 indexed files available so new records can be written in them
COPY	Copies one or more files to one or more additional files
CORAL	Compiles one or more CORAL 66 source language programs
CREATE	Creates a file from data entered at the terminal or in the input stream
CREATE/DIRECTORY	Defines a new directory or subdirectory for cataloging file
CREATE/FDL	Uses the specifications of an FDL file to create a new, empty data file

DBO	Invokes the VAX-11 DBMS Data Base Operations (DBO) Utility
DDL	Invokes the VAX-11 DBMS Data Definition Language (DDL) Utility
DEALLOCATE	Relinquishes use of a previously allocated device, thus making the device available to other users
DEASSIGN	Cancels a logical name assignment made with the ALLOCATE, ASSIGN, or DEFINE command
DEASSIGN/QUEUE	Deassigns a queue from a device
DEBUG	Invokes the VAX-11 Symbolic Debugger to begin or continue interactive debugging
DECK	Marks the beginning of records to be read as the input data stream for a command (required only when data contains a dollar sign ($) in the first position of any record)
DEFINE	Equates character strings with file specifications or logical names
DELETE	Removes a directory entry for a file or files and makes any data in the files(s) inaccessible
DELETE/ENTRY	Deletes an entry from a printer or batch job queue or stops processsing of the current job
DELETE/QUEUE	Deletes batch queues and printer queues
DELETE/SYMBOL	Deletes one or more symbol names from the local or global symbol tables for the process
DEPOSIT	Replaces the contents of a location in virtual memory with new data or instructions
DIFFERENCES	Compares the contents of files and reports the differences between them
DIRECTORY	Displays information about a file or a group of files
ISMOUNT	Releases the connection between a user and a disk or tape volume that is currently mounted on a device
DUMP	Displays or prints the contents of a file or volume in ASCII, hexadecimal, octal, or decimal format
EDIT/EDT	Begins an interactive editing session with the EDT editor to create or modify a file
EDIT/FDL	Creates and modifies File Definition Language (FDL) files

EDIT/SLP	Provides input to the batch-oriented editor, SLP
EDIT/SOS	Begins an interactive editing session with the SOS editor to create or modify a file
EDIT/SUM	Invokes the SUMSLP batch-oriented editor to update a single input file with multiple files of edit commands
EOD	Marks the end of input data stream begun with the DECK command
EOJ	Signals the end of a batch job submitted through a card reader
EXAMINE	Displays the contents of a location in virtual memory
EXIT	Terminates an image or command procedure processing at the current level
FORTRAN	Compiles FORTRAN language source statements
GOTO	Transfers control to another statement in a command procedure
HELP	Displays information on the current output stream device from the system HELP files or any help library you specify
IF .. THEN	Compares expressions consisting of symbolic or literal values, or command or program status values, and performs a stated action based on the result of the test
INITIALIZE	Deletes all existing data, if any, on a mass storage volume, writes a label on the volume, and readies the volume for new data
INITIALIZE/QUEUE	Creates batch queues and output queues
INQUIRE	Requests interactive assignment of a variable value for a symbol name
JOB	Marks the beginning of a batch job submitted through a card reader
LexicaL Functions	Represent symbols or expressions that return information about character strings and attributes of the current process
LIBRARY	Creates or modifies various kinds of libraries
LIBRARY/RSX11	Creates or modifies RSX-11M macro libraries or object module libraries
LINK	Binds one or more object modules into an executable or shareable program image

LINK/RSX11	Invokes the RSX-11M Task Builder to link one or more object modules into an RSX-11M task
Login Procedure	Initiates communication between a user and the system
LOGOUT	Terminates communication between a user and the system
MACRO	Invokes the VAX-11 MACRO assembler to assemble a VAX-11 assembly language program
MACRO/RSX11	Invokes the MACRO-11 assembler to assemble a PDP-11 assembly language program
MAIL	Sends messages to other users of the system
MCR	Passes a command line to the RSX-11M Application Migration Executive, or places the terminal in MCR command mode
MCR BAD	See RUN SYS$SYSTEM:BAD
MCR DSC1	See RUN SYS$SYSTEM:DSC1
MCR DSC2	See RUN SYS$SYSTEM:DSC2
MERGE	Combines 2 through 10 similarily sorted input files and creates a single output file
MESSAGE	Compiles one or more files of message definitions
MONITOR	Collects or displays system performance data
MOUNT	Makes a disk or tape volume available for the reading and writing of files
ON ... THEN	Defines the action to be taken when a command or program incurs errors of particular severity levels, or when the CTRL/Y function key is used
OPEN	Establishes a path to a file or a device for input or output operations
PASCAL	Compiles one or more PASCAL source programs
PASSWORD	Provides a password associated with a job entered through a card reader
PATCH	Patches an executable image, shareable image, or device driver image
PHONE	Allows you to talk with other users on your system or on any other VAX/VMS system connected to your system
PLI	Compiles one or more PL/I source programs
PRINT	Queues a file for printing or on a specified device

PURGE	Deletes old versions of a specified file or files
READ	Reads the next record from a file or device and assigns the contents of the record to a symbol name
RENAME	Changes the name of a file or a group of files
REPLY	Allows the operator to communicate with system users, selectively enable and disable operator status, and examine the log file
REQUEST	Displays a message at an operator's terminal
RUN (Image)	Places an image into execution in the current process
RUN (Process)	Creates a separate process to execute a specified image
RUN SYS$SYSTEM:BAD	Locates and counts the bad blocks contained on Files-11 disks
RUN SYS$SYSTEM:DSC	Transfers files contained on Files-11 Structure Level 1 disks to tapes or disks for back-up and storage
RUN SYS$SYSTEM:DSC2	Transfers files contained on Files-11 Structure Level 2 disks to tapes or disk for back-up and storage
RUN SYS$SYSTEM:INSTALL	Installs or deletes known images
RUN SYS$SYSTEM:SYE	Creates an error log report from a binary formatted file
RUNOFF	The DIGITAL Standard Runoff (DSR) text formatter formats one or more ASCII files
SEARCH	Searches one or more files for a specified string and lists all the occurences of that string
SET ACCOUNTING	Selectively enables or disables the recording of particular kinds of accounting information
SET CARD_READER	Defines the translation mode for a card reader
SET COMMAND	Invokes the Command Definition Utility to add commands defined in a command description file to your process command set or to a command tables file
SET CONTROL	Enables the use of the CTRL/Y and CTRL/T function keys
SET DAY	Overrides the default day type specified in the user authorization file (UAF)
SET DEFAULT	Changes the directory and/or disk device used by default to locate and catalog files

SET DEVICE	Establishes the spooling and error logging status on a specified device
SET DIRECTORY	Modifies the characteristics of a directory
SET FILE	Modifies the characteristics of a file
SET HOST	Establishes a virtual communication link between a terminal and a network node to which the terminal is not directly connected
SET LOGINS	Establishes the maximum number of users able to log into the system
SET MAGTAPE	Defines the density of a magnetic tape device or rewinds a tape
SET MESSAGE	Overrides or supplements the system messages
SET NOCONTROL	Disables the use of the CTRL/Y and CTRL/T function keys
SET NOON	Disables previously declared ON conditions and prevents the command interpreter from taking any action on errors issued during command processing
SET NOVERIFY	Suppresses display of command lines in subsequently executed command procedures
SET ON	Restores command interpreter error actions in a command procedure
SET PASSWORD	Allows users to change their own passwords
SET PRINTER	Establishes the characteristics of a specified line printer
SET PROCESS	Changes execution characteristics of a process
SET PROCESS/PRIORITY	Changes the base priority for a process
SET PROTECTION	Changes the protection applied to a file or a group of files, restricting or allowing access to the file by different catagories of user
SET PROTECTION/DEFAULT	Establishes the default protection for all files subsequently created during the terminal session or batch job
SET PROTECTION/DEVICE	Establishes the protection for non-file-structured devices
SET QUEUE/ENTRY	Changes the current status or attributes of a file that is queued for printing or for batch job execution, but not yet processed by the system
SET RMS_DEFAULT	Defines default multiblock and multibuffer counts, the default extend quantity, and the default prologue for VAX-11 RMS file operations

SET TERMINAL	Defines the characteristics of the terminal
SET TIME	Resets the system clock to the specified value
SET UIC	Establishes a new user identification code as the process UIC
SET VERIFY	Causes all command lines in command procedures subsequently executed to be displayed at the terminal or printed in the batch job log file
SET VOLUME	Modifies the characteristics of one or more mounted Files-11 volumes
SET WORKING_SET	Establishes a default working set size for images executed in the current process
SHOW DAYTIME	Displays the current date and time of day on the current output device
SHOW DEFAULT	Displays the current default directory and disk device
SHOW DEVICES	Displays the status of devices in the system
SHOW ERROR	Displays the error count for the CPU, memory, and all the physical devices other than terminals
SHOW LOGICAL	Displays the current assignments of logical names and equivalence names made by the ASSIGN, ALLOCATE, DEFINE, or MOUNT command
SHOW MAGTAPE	Displays characteristics of a magnetic tape device
SHOW MEMORY	Displays the availability and use of memory resources
SHOW NETWORK	Displays the availability of the local node as a member of the network and the names of all nodes currently accessible by the local node
SHOW PRINTER	Displays the characteristics of a line printer
SHOW PROCESS	Displays information about the current process, including subprocesses, privileges, quotas, and accounting information
SHOW PROTECTION	Displays the default protection applied to new files created
SHOW QUEUE	Displays the names, job numbers, and status of current and pending jobs in the printer and batch job queues
SHOW QUOTA	Displays the current disk quota that is authorized and used by a specific user on a specific disk
SHOW RMS_DEFAULT	Displays the current multiblock and multibuffer counts, the default extend quantity, and the default prologue for VAX-11 RMS operations

SHOW STATUS	Displays information about the image currently executing in the process
SHOW SYMBOL	Displays current local or global symbols and the strings or values assigned to them
HOW SYSTEM	Displays the current status of processes in the system
SHOW TERMINAL	Displays the current characteristics of the terminal
SHOW TIME	Displays the current date and time on the current output device (see SHOW DAYTIME)
SHOW TRANSACTION	Searches all three logical name tables for a logical name and displays the equivalence name of the first match found
SHOW USERS	Displays the terminal name, user name, and process-id of specific interactive users or of all interactive users on the system
SHOW WORKING_SET	Displays the current working set default and limits
SORT	Invokes the VAX-11 SORT Utility to sort the records in a file based on one or more key fields within each record
SORT/RSX11	Invokes the PDP-11 SORT Utility to sort the records in a file based on one or more key fields within each record
SPAWN	Creates a subprocess of the current process
START/QUEUE	Starts batch queues and printer queues
STOP	Halts execution of a command procedure, program, or a subprocess or detached process
STOP/ABORT	Stops the printing of a job that is currently being printed
STOP/ENTRY	Stops the execution of a batch job that is currently running
STOP/QUEUE	Suspends batch queues and printer queues
STOP/REQUEUE	Stops the printing of a job that is currently being printed and requeues that job at the end of the queue
SUBMIT	Enters one or more command procedures in a batch job queue
SYNCHRONIZE	Places the process issuing this command in a wait state until a specified batch job completes
TYPE	Displays the contents of a file or files on the

	current output device
UNLOCK	Allows access to file that was not properly closed
WAIT	Places the current process in a wait state for a specified period of time
WRITE	Writes a single record consisting of one or more character strings or evaluated symbols to a sequential file or device

USER PRIVILEDGES:

Following are the list of the user priviledges; that are normally established at login time [for your terminal session]:

ACNT	Create a process for which no accounting records are made
ALLSPOOL	Allocate spooled devices
ALTPRI	Increase the base execution priority for any process
BUGCHK	Make bug check error log entries
BYPASS	Bypass UIC protection
CMEXEC	Change mode to executive
CMKRNL	Change mode to kernel
DETACH	Create detached processes
DIAGNOSE	Issue diagnostic I/O requests
EXQUOTA	Exceed resource quotas
GROUP	Control execution of other processes in the same group
GRPNAM	Enter names in the group logical name table
LOG_IO	Perform logical I/O functions
MOUNT	Execute a mount volume I/O function
NETMBX	Create a network device
OPER	Perform operator functions
PFNMAP	Create or delete sections mapped by page frame number

PHY_IO	Perform physical I/O functions
PRMCEB	Create permanent common event flag clusters
PRMGBL	Create permanent global clusters
PRMMBX	Create permanent mailboxes
PSWAPM	Change process swap mode
SETPRV	Grant a created process any privileges
SHMEM	Create or delete data structures in shared memory
SYSGBL	Create system global sections
SYSLCK	Requests locks on system-wide resources
SYSNAM	Enter names in the system logical name table
SYSPRV	Access files and other resources as if the user has a system UIC
TMPMBX	Create temporary mailboxes
VOLPRO	Override protection on a volume
WORLD	Control the execution of any process in the system

##########**********CDC**********##########

BNL_VAX_CLUSTER
****

EXAMPLE 2:

Here we illustrate the use of DCL commands on BNLCLuster [BNLCLn, n=1,2,3,or 4] as we showed in the previous example for the BNLDAG node.

HOW TO LOGIN ON THE CLUSTER; [BNLCLn, n=1,2,3,4]
---

1. Turn on the terminal press <return>
2. Turn on the gandalf box (set class at 40)
   [see section on computer access lines]
3. Type in CLASS, USERNAME, LOCAL, USERNAME, PASSWORD
   upon request:

```
ENTER CLASS 111 <return>
class 111 start
Digital Ethernet Terminal Server.......

Enter Usernmae> ZOHREH   <return>

Local>                    ![while in the Terminal Server, you can
                            enter:help, and get information on
                            different topics; e.g File transfer.

Local>HELP <return>
     ......................
     ..................

Local> C BNLCLn           ![in response to Local prompt, you enter
                          the C BNLCLn (1,2,3, or 4), i.e.,the node
                          you would like to log into, e.g. BNLCL2.]

Local> C BNLCL2  <return>

USERNAME: zohreh  <return>
PASSWORD:

$
$
$
$ show time
  24-OCT-1985 23:29:09
```

```
$ help

 HELP

 The  HELP  command  invokes  the  VAX/VMS  HELP
 Facility  to  display  information about a VMS
 command or topic.  In response to the "Topic?"
  prompt, you can:

  o Type  the  name of the command  or topic for which
    you need help.

  o Type  INSTRUCTIONS  for more detailed  instructions
    on how to use HELP.

  o Type  HINTS  if you are not  sure of the  name  of
    the command or topic for which you need help.

  o Type a question mark (?) to redisplay the most
    recently requestedtext.

  o Press the RETURN key one or more times to exit
    from HELP.

   You  can  abbreviate any  topic name, although
   ambiguous abbreviations  result in all matches
   being displayed.


   Additional information available:

   :=      =       @             ACCOUNTING ALLOCATE
   ANALYZE       APL             APPEND     ASSIGN
   ATTACH        BACKUP          BCREF      BLISS
   CANCEL        CC      CLOSE   CMS        CONNECT
   CONTINUE      CONVERT         COPY       CREATE
   DEALLOCATE DEASSIGN DEBUG     DECK DEFINE
   DELETE        DEPOSIT         DIFFERENCES
   DIRECTORY     DISCONNECT DISMOUNT    DUMP
   EDIT    ENCRYPT  EOD     EOJ          Errors
   EXAMINE       EXCHANGE        EXIT       FDL
   FORTRAN       GOTO            HELP
   Hints         IF              INITIALIZE  INQUIRE
   Instructions                  JOB        Lexicals
   LIBRARY       Line_editing               LINK
   LOGIN         LOGOUT          MACRO      MAIL
   MERGE         MESSAGE         MMS        MONITOR
   MOUNT         ON              OPEN       PASCAL
   PASSWORD      PATCH           PHONE      PRINT
   PURGE         Queues          READ       RECALL
   RENAME        REPLY           REQUEST    RMS
   RTL_Routines                  RUN        RUNOFF
   SEARCH        SET             SHOW       SORT
   SPAWN         Specify         START      STOP
   SUBMIT        Symbol_Assign              SYNCHRONIZE
   System_Services               TYPE       UNLOCK
   WAIT          WRITE
```

```
$ HELP MAIL
-----------

MAIL

    Invokes the VAX/VMS Personal Mail Utility (MAIL), which is used to
    send messages to other users of the system.  For a complete
    description of the VAX/VMS Personal Mail Utility, including
    information about the MAIL command and its qualifiers, see the
    VAX/VMS Mail Utility Reference Manual.

    Format:


      MAIL  [file-spec] [recipient-name]


  Additional information available:

  Parameters Command_Qualifiers
  /SUBJECT   /EDIT      /SELF
  Examples

SUBTOPIC? EXAMPLES
------------------

MAIL

  Examples

      1.   $ MAIL
           MAIL>

      This MAIL command invokes MAIL to process commands interactively.

      2.   $ MAIL/SUBJECT="New Project" PROJECT.DOC JONES,SMITH,ADAMS

      This MAIL command specifies that the file named PROJECT.DOC is to be
      sent to users JONES, SMITH, and ADAMS, with a subject description of
      New Project in the heading.

      3.   $ MAIL/SUBJECT="Vacation Policy Change" NEWSLETTR "@USERS"

      This MAIL command invokes MAIL to send the file NEWSLETTR.TXT to all
      the users named in the file USERS.DIS.  The subject description is
      Vacation Policy Change.


SUBTOPICS? /EDIT
----------------------

MAIL

  /EDIT
   /EDIT=[(send,reply=extract,forward)]

   Sets the default to /EDIT for the SEND and REPLY commands.
```

SUBTOPICS? /subject

MAIL

/SUBJECT
/SUBJECT=text

Specifies the subject of the message for the heading. If the text consists of more than one word, enclose the text in quotation marks (").

You must include a file specification on the command line to enable this qualifier.

If you omit this qualifier, the message is sent without a subject notation.

Enter NOTICE to get new information:

$ notice <return>

NEW ACCELERATOR PHYSICS DATA BASE FEATURING
PROGRAMS FOR AGS, NSLS, RHIC AND SSC RESEARCH
IS NOW AVAILABLE FOR ALL USERS ON
DAGVAX::DUAO:[PARSA1]
FOR FURTHER INFO CONTACT PARSA VIA MAIL
TO ZOHREH ON THE AMDVAXCLUSTER OR PARSA
@BNLDAGVAX.

An introductory course in the SIR data base system is being offered by SIR inc. at BNL NOV. 18-19. The cost is $ 500.00 . Anyone interested can contact Kurt Fuchel , at ex.4116.

7600 background job pre-input queue and its associated procedures (BG76, etc) are no longer available. Submit 7600 lower tier jobs by only using BG on the job statement and adding 100000 to your problem number. See AMD_INFO RESOURCE 7600_2Tiered_Pri for more information.

New Terminal Server V2.0 and Decnet Router Server V1.1 on all nodes.

Temporary Files:

There are two scratch devices available for temporary files. They can be used, for example, when generating files which would normally exceed your disk quota. To use this facility, you can type:

```
SET DEFAULT $2$DUA15:[SCRATCH]
```

Files created here will not be backed up and will be DELETED AFTER TWO DAYS from the date of creation. You may also type:

```
SET DEFAULT $2$DUA11:[XSCR]
```

Files created here will not be backed up and will be DELETED AFTER FOUR DAYS from the date of creation.

NOTE: A wild card (*.*) copy does NOT change the creation date. Be aware of this when moving files to [SCRATCH] or [XSCR] from another directory.

Users that are sometimes frustrated by the fact that their subdirectory of the scratch directory 2dua11:[XSCR] frequently needs to be recreated can create a small command procedure that will create the subdirectory whenever it is found to be missing and set the directory default to it. A typical example follows:

```
$ IF F$SEARCH("$2$DUA11:[XSCR]yourname.DIR") .EQS. "" THEN -
                        CREATE/DIR $2$DUA11:[XSCR.yourname]
$ SET DEFAULT  $2$DUA11:[XSCR.yourname]
```

When editing large files on either device, the message "disk quota exceeded" may appear. This occurs because the editor attempts to open a work file on the users DEFAULT device. To direct the editor's work files to a scratch device, type one of the commands below (appropriate to the scratch device you are using) followed IMMEDIATELY by the EDIT.

```
DEFINE/USER_MODE SYS$SCRATCH  $2$DUA15:[SCRATCH]
                        or
DEFINE/USER_MODE SYS$SCRATCH  $2$DUA11:[XSCR]
```

The following is an example of a command procedure which will assign the scratch disk to dua15 and enter the editor automatically.

```
$ if p1 .eqs. "" then inquire p1 "File"
$ assign $2$dua15:[scratch] sys$scratch
$ define/user sys$input sys$command:
$ edit 'p1'
$ deassign sys$scratch
$ exit
```

USER DEFINED HELP LIBRARIES:

VAX/VMS supports user defined help libraries. They are very easy to create and can be very useful for keeping track of all the files one collects in their directories. The first step is the most difficult one but as they say:

A Journey of 1,000 miles begins with the first step.

1) Create a text file of descriptions of your files. For the purpose of illustration, name the file MYHELP.HLP. Leave column one blank.

2) Preface each file description with a numeral 2 and a ONE WORD program name.

Ex: 2 GAUSS.FOR

3) Organize the descriptions into groups that are to your liking and preface each group with a numeral 1 and a ONE WORD group description.

Ex: 1 MONTE_CARLO_PROGRAMS

3) Execute the following library command to create the help file.

LIBRARY/HELP/CREATE:KEYSIZE=31 MYHELP MYHELP

4) Execute the following help command to use the new library:

HELP/PAGE/LIBRARY= yourdisk:[yourdirectory]MYHELP/NOINSTR

Note: This should be enough to get you started. You can create as many levels as you like (ie: level 3 for sample inputs etc.). See the Library Utility Reference Manual for more information.

CDC_to_VAX_File_Xfer

CTV (CDC-TO-VAX) is a file-transfer procedure which fetches a file from the CDC 830 computer, or a file resident on an MFZ PF archive tape and moves it to the VAX cluster. CTV uses MDX, MASSNET Data Exchange, which is a MASSNET program supporting peer file transfer and remote job entry capabilities between host computers within a MASSNET-based network.

If the file is an UPDATE oldpl, it will be converted, via UTOH, to a HISTORIAN PLUS compatible library, which can be used on either the Vax cluster or the IBM 3090. (See also AMDINFO help topic Software_Packages, subtopic HISTORIAN_PLUS.) If the file is a CDC source file, it will be converted to a VAX source file. CTV can be used in either the interactive mode, or the single-line command mode.

Interactive_mode

To invoke CTV in the interactive mode, type CTV. The procedure will prompt for:

- CDC computer -- mfC, Dumpf
- Dumpf tape VSN (for archived file)
- CDC permanent file name
- Cycle number (<CR> defaults to highest cycle)
- CDC file id
- CDC file type -- Oldpl, Source
- CDC oldpl type -- Random, Sequential (for archived oldpl)
- VAX file name
- VAX login password

Single_line_command_mode

To invoke CTV in the single line command mode, type CTV followed by either 6 or 8 parameters. CTV will prompt for missing parameters.

Command Syntax:

```
CTV   cc   p[,c]   cfi   cft   vfn   vlp  [tp#]  [olt]

  cc    =  CDC computer ( mfC or Dumpf )
  p[,c] =  CDC pfn[,cy #] (e.g. myfile,16  or  myfile )
  cfi   =  CDC file id
  cft   =  CDC file type ( Oldpl or Source )
  vfn   =  VAX file name (up to 10 characters)
  vlp   =  VAX login password
  tp#   =  CDC 7600 dumpf tape number (when cc=Dumpf)
  olt   =  CDC Oldpl type ( Random or Sequential) (when cc=Dumpf)
```

Restrictions_etc.

- In order to retrieve and convert a random or sequential oldpl which has been dumped to tape, the pfn, id, and cycle number (as well as the tape number) are required.
- Oldpls residing on permanent files must be random oldpls to be converted by CTV.
- Only sequential text files are supported for transfer using CTV.
- Data files to be transferred must be no greater than 512

bytes per record.

- For transfer of very large files, truncation may occur without an error message from MDX. A user who needs to transfer large files to the scratch disk should contact R. Evans x2851.

```
$ AMD_INF
----------

  Information available:

  7600_File_Migration   Batch_Queues
  CAI_Courses           CDC_to_VAX_File_Xfer
  Cluster_Software      Common_Symbols
  Cybervax_To_7600      Disk_Quota_Policy
  File_Backup_Scheme    FPS364_Processor
  Hardware_Config       Hints_and_Kinks
  Holiday_Schedule      Large_Masstor
  Laser_Printing        Mag_Tape_Usage
  Maintenance_Schedule  Manuals_for_V4.
  Mfa_Permanent_Files   MFC_Cyber_830
  Mfe_Link_to_LLL       N_Tape_LIbrary
  Questions_Answers     Resource_Accounting
  Security_Access_Syst. Software_Packages
  TCPIP_Connection      Temporary_Files
  Terminal_Server       Upgraded_Software
  Upton_Decus_LUG       Where_To_Find_It

  Additional help libraries available (type @name for topics):

  AMDINFO  MFENET  FPS364

subtopic? batch_queue
---------------------
Batch_Queues

   Batch jobs are run on the AMD CLUSTER by using the SUBMIT command and
   specifying which Batch queue (see list below) you wish to use.

   QUEUE NAME       NODE NAME    CHARACTERISTICS
   ------------------------------------------------------------------------
   AMDCL1_20M       BNLCL1       Pri=2  Job Limit=3  Cpu Time Limit = 20 min.
   AMDCL1_BATCH     BNLCL1       Pri=2  Job Limit=3  Cpu Time Limit = nolimit
   AMDCL1_SLOW      BNLCL1       Pri=0  Job Limit=1  Cpu Time Limit = nolimit
   AMDCL2_20M       BNLCL2       Pri=2  Job Limit=3  Cpu Time Limit = 20 min.
   AMDCL3_20M       BNLCL3       Pri=2  Job Limit=3  Cpu Time Limit = 20 min.
   AMDCL3_BATCH     BNLCL3       Pri=2  Job Limit=3  Cpu Time Limit = nolimit
   AMDCL3_SLOW      BNLCL3       Pri=0  Job Limit=1  Cpu Time Limit = nolimit
   AMDCL4_20M       BNLCL4       Pri=5  Job Limit=3  Cpu Time Limit = 20 min.
   AMDCL4_BATCH     BNLCL4       Pri=4  Job Limit=6  Cpu Time Limit = nolimit
   AMDCL4_SLOW      BNLCL4       Pri=0  Job Limit=1  Cpu Time Limit = nolimit

   The queue AMDCL4_BATCH can be accessed by the generic queue name SYS$BATCH.
   Jobs submitted to SYS$BATCH (or submitted without specifying a queue name)
   will automatically be directed to node BNLCL4, which is parameterized to
   favor batch jobs.

   The queues AMDCLn_SLOW have been set up to run batch jobs at a reduced rate
   (see RESOURCE_ACCOUNTING for charges). Jobs submitted to SYS$SLOW will be
   directed to the "SLOW" queue with the least activity. Jobs in these queues
   may require a longer time to completion, due to the lower queue priority.

   If you plan to submit a Batch job which will run for several days, make
   sure you consider the AMD CLUSTER operating schedule. When the cluster is
   taken down for maintenance, currently running batch jobs will be lost.
   (Batch checkpointing is promised in a future release of VMS)
```

Additional batch queues have been defined for system and operator jobs.

Additional information available:

SYSTEM_QUEUES

topic? mag_tape_usage

MAG_TAPE_USAGE

Users wishing to use the AMD CLUSTER magnetic tape(s) must first determine if the device(s) is available. To do this type, SDMT<cr>. The response to this command will indicate whether the tape drive is IN USE or AVAILABLE. If the drive is available issue your mount command to reserve the drive for your tape. You can call Operations at x4173 and ask them to put your tape on the drive or use the /COMMENT qualifier in your mount command to communicate any special instructions. A response message will be sent to your terminal when your request has been honored. Any messages from the operator can be answered by using the REQUEST command.

The system is configured so that all nodes have access to the TA78 drives (6250/1600 bpi) - device type HSC00n$MUAx (where n=0,1;x=0,1). The TU77 drives (1600/800 bpi) are accessible only on BNLCL2 and BNLCL4 and the TK50 (cartridge tape) drive is accessible only on BNLCL3. These drives should be referenced by their specific addresses, for example:

```
ALLOCATE BNLCL2$MTA0 lfn        to allocate a TU77 drive   or
ALLOCATE BNLCL3$MUA0 lfn        to allocate a TK50 drive
```

where 'lfn' refers to logical file name.

If you have no need for a specific drive, refer to the subtopic 'Tape_Allocation' which shows how the system can select it for you.

Additional information available:

Blank_Tape_Initialization Conversion_Utilities Labeled_Tapes_For_The_VAX

subtopic? <return>

topic? hardware_config

HARDWARE_CONFIG

Hardware configurations as of July, 1986:

CPUS	- one	11/785 with floating point accelerator and 10 megabytes of memory
	- three	11/785's with floating point accelerators and 8 megabytes of memory. (one of the above CPUS's also has access to an FPS364

Advanced Processor)

PERIPHERALS	- 14	RA81, 456 MB disks
	- two	RA60, 205 MB removeable disks
	- two	RX02 floppy disk drives
	- two	HSC50, intelligent controllers which allow the sharing of the disks and the 6250 tape drives
	- one	TK50, tape transport (6667/bpi)
	- two	TU78, tape transports (6250/1600 bpi)
	- two	TA78, tape transports (6250/1600 bpi)
	- two	TU77 tape transports (1600/800 bpi)
	- two	LP27, line printers (1200 lpm)
	- one	Talaris Laser printer (8 pages/minute)
	- four	IA120, Dec III consoles
	- one	SC008, Star Coupler
COMMUNICATIONS	- four	DECSA Terminal Servers, each of which supports 32 async lines
	- two	DMZ32 Unibus terminal interfaces
	- one	DZ11 Unibus terminal intertace
NETWORKS	- two	Decnet routers, (each having 8 slots) which will support other nodes on Ethernet
	- four	PI-13 HYPERchannel links

Additional information available:

ACCESS

subtopic? access

HARDWARE_CONFIG

ACCESS

- Gandalf class code 111, autobaud to 9600
- X7021 dial-up 300/1200 baud
- outside users gain access via PACX through the Security Access System.

topic? Mfe_Link_to_LLL

Mfe_Link_to_LLL

The MFE link to the CRAY at Lawrence Livermore Laboratory is available on node BNLCL2. You must CONNECT to BNLCL2 in order to use it.

Symbols needed for its use are automatically defined at LOGIN time. Details are available in the MFENET help file which can be accessed with the DCL command HELP @MFENET (or return to the primary HELP display and type @MFENET).

Information On IBM 3090

TERMINALS AND KEY DEFINITIONS *#

The IBM operating system and the full screen editor XEDIT assume that the user has an IBM 3270 series terminal. Gandalf class 71 provides emulation of such terminals on DEC VT100 terminals and equivalents. The VT100 must be set for 7 bits, even parity, and full duplex. This can be done both on DEC VT220 and VT240 terminals and in PC INTERCOM on the IBM PC. It is preferable to use a VT220 in its VT100 mode because the key layout is more logical.

The following chart shows the key definitions for the VT100 keypad. The alternate function on the second line is obtained by first pressing KP0.

pf1 PF1 PF13	pf2 PF2 PF14	pf3 PF3 PF15	pf4 PA1
kp7 PF4 PF16	kp8 PF5 PF17	kp9 PF6 PF18	kp- PA2
kp4 PF7 PF19	kp5 PF8 PF20	kp6 PF9 PF21	kp, PA3
kp1 PF10 PF22	kp2 PF11 PF23	kp3 PF12 PF24	enter CLEAR
kp0 Alternate PF Keys		kp. INS	

* From F. Paige and M. Murtagh (Physics Dept. Computer Committee; # = H. Berry and other users.

The following chart shows the PF keys defined for use in CMS by SETKEYS EXEC on PAIGE minidisk 191. All of these commands are immediate except for RETRIEVE BACKWARD and RETRIEVE FORWARD, which just display the command.

pf1 HELP	pf2 BATCH Q M	pf3 BATCH Q *	pf4 CP/VM
kp7 FILELIST	kp8 RDRLIST	kp9 Q DISK	kp- PA2
kp4 RETRIEVE B	kp5 RETRIEVE F	kp6 Q DASD	kp, PA3
kp1 HT	kp2 HI	kp3 HX	enter CLEAR
kp0 Alternate PF Keys		kp. INS	

The following chart shows the function keys defined for use in XEDIT by PROFILE XEDIT on PAIGE minidisk 191:

pf1 HELP HELPKEYS	pf2 LINEADD RESET	pf3 QUIT SAVE	pf4 PA1
kp7 TABKEY WFIND	kp8 SCHANGE6 WERASE	kp9 ? TOP	kp- PA2
kp4 BACKWARD HALFUP	kp5 FORWARD HALFDOWN	kp6 = BOTTOM	kp, PA3
kp1 RGTLEFT BOL	kp2 SPLTJOIN EOL	kp3 CURSOR INSHERE	enter CLEAR
kp0 Alternate PF Keys		kp. INS	

Several of these commands are new XEDIT macros which are on minidisk PAIGE 191:

BOL	Beginning of cursor line
EOL	End of cursor line
HALFDOWN	Scroll forward 1/2 page
HALFUP	Scroll backward 1/2 page
HELPKEYS	Display VT100 keypad for XEDIT
INSHERE	Insert after cursor line
WERASE	Erase next word on current line
WFIND	Find next word on current line

The following additional prefix commands from SLAC are defined:

PT[T]	[Block] Put
PD[D]	[Block] Put and delete
G	Get
L[L]	[Block] LOWERCAS
U[U]	[Block] UPPERCAS

Keyboard Definitions for the VT100 Terminal, PC/InterComm & SmarTerm

Control and Cursor Movement Keys

IBM FUNCTION	VT100	PC/InterComm & SmarTerm
Enter	RETURN	
Clear	ENTER[1]	+[1]
Redisplay	CTRL–V	CTRL–V
Erase EOF[3]	ESC DELETE	ESC DEL
Delete Character[3]	DELETE	DEL
Toggle Insert Mode	. (period)[1]	. (period)[1]
Field Tab	TAB	
Field Backtab	ESC TAB	ESC
Column Tab	ESC →	ESC →
Column Backtab	ESC ←	ESC ←
Indent	ESC ↑	ESC ↑
Undent	ESC ↓	ESC ↓
PA1	PF4 or ESC , (comma)[2] or ESC <	F4 or ESC , (comma)[2] or ESC <
PA2	– (hyphen)[1] or ESC > or ESC . (period)[2]	– (hyphen) or ESC > or ESC . (period)[2]
PA3	, (comma)[1] or ESC / (slash) or ESC ? (question mark)	, (comma)[1] or ESC / (slash) or ESC ? (question mark)
Newline	LINE FEED	CTRL–←
Home	BACK-SPACE	↑ or F7
Cursor Up	↑	↓ or F8
Cursor Down	↓	→ or F6
Cursor Right	→	← or F5
Cursor Left	←	

[1] This key can be found only on the additional numeric keypad on the right side of the VT-100 or the IBM PC.

[2] This key can be found only on the main keypad. Do not select a similar key on the additional numeric keypad on the right side of the VT-100 (or the IBM PC) keyboard.

[3] For the SmarTerm VT100 emulator Erase EOF is ESC CTRL–← and Delete Character is CTRL–←.

Program Function Keys

IBM FUNCTIONS	VT100	PC/InterComm & SmarTerm
PF1	ESC 1[2] or PF1	ESC 1[2] or F1
PF2	ESC 2[2] or PF2	ESC 2[2] or F2
PF3	ESC 3[2] or PF3	ESC 3[2] or F3
PF4	ESC 4[2] or 7[1]	ESC 4[2] or 7[1]
PF5	ESC 5[2] or 8[1]	ESC 5[2] or 8[1]
PF6	ESC 6[2] or 9[1]	ESC 6[2] or 9[1]
PF7	ESC 7[2] or 4[1]	ESC 7[2] or 4[1]
PF8	ESC 8[2] or 5[1]	ESC 8[2] or 5[1]
PF9	ESC 9[2] or 6[1]	ESC 9[2] or 6[1]
PF10	ESC 0[2] or 1[1]	ESC 0[2] or 1[1]
PF11	ESC – (hyphen)[2] or 2[1]	ESC – (hyphen)[2] or 2[1]
PF12	ESC = or 3[1]	ESC = or 3[1]
PF13	0[1] PF1 or ESC q[3]	0[1] F1 or ESC q[3]
PF14	0[1] PF2 or ESC w[3]	0[1] F2 or ESC w[3]
PF15	0[1] PF3 or ESC e[3]	0[1] F3 or ESC e[3]
PF16	0[1] 7[1] or ESC r[3]	0[1] 7[1] or ESC r[3]
PF17	0[1] 8[1] or ESC t[3]	0[1] 8[1] or ESC t[3]
PF18	0[1] 9[1] or ESC y[3]	0[1] 9[1] or ESC y[3]
PF19	0[1] 4[1] or ESC u[3]	0[1] 4[1] or ESC u[3]
PF20	0[1] 5[1] or ESC i[3]	0[1] 5[1] or ESC i[3]
PF21	0[1] 6[1] or ESC o[3]	0[1] 6[1] or ESC o[3]
PF22	0[1] 1[1] or ESC p[3]	0[1] 1[1] or ESC p[3]
PF23	0[1] 2[1] or ESC _ (underscore)	0[1] 2[1] or ESC _ (underscore)
PF24	0[1] 3[1] or ESC +	0[1] 3[1] or ESC +

Local Reset and Control Functions

IBM FUNCTION	VT100	PC/InterComm
Master Reset (Reinitialize Terminal)	CTRL–G	CTRL–G
Character Error Reset (Clear Parity Error)	CTRL–R	CTRL–R
Keyboard Unlock (IBM Reset Function)	CTRL–T	CTRL–T
Type-ahead Purge	CTRL–X	CTRL–X
Pacing Start	CTRL–S	CTRL–S
Pacing Stop	CTRL–Q	CTRL–Q

[1] This key can be found only on the additional numeric keypad on the right side of the VT-100 (or the IBM PC).

[2] This key can be found only on the main keypad. Do not select a similar key on the additional numeric keypad on the right side of the VT-100 keyboard (or the IBM PC).

[3] This character may be typed in as uppercase or lowercase.

Setup Functions

IBM FUNCTION	VT100	PC/InterComm & SmarTerm
Set Column Tab	ESC ` TAB	ESC ` CTRL-I
Delete Column Tab[3]	ESC ` DELETE	ESC ` DEL
Set Left Margin	ESC ` LINE FEED	ESC ` CTRL-
Set Home Line	ESC ` BACKSPACE	ESC ` ←
Delete All Column Tabs, Home Line and Left Margin	ESC ` ENTER[1]	ESC ` +[1]
Improved Null Processing	ESC ` N	ESC ` N
3270 Null Processing	ESC ` n	ESC ` n
Zones Mode On	ESC ` z	ESC ` z
Zones Mode Off	ESC ` Z	ESC ` Z
Reverse Enter/Newline Keys	ESC ` e	ESC ` e
Restore Enter/Newline Keys	ESC ` E	ESC ` E
Reverse Column and Field Tab Keys	ESC ` c	ESC ` c
Restore Column and Field Tab Keys	ESC ` C	ESC ` C
Alpha in Numeric-Only Field	ESC ` v	ESC ` v
3270 Numeric Fields	ESC ` V	ESC ` V
3278 Insert Mode	ESC ` i	ESC ` i
3277 Insert Mode	ESC ` I	ESC ` I
Suppress Pacing	ESC ` P	ESC ` P
Restore Pacing	ESC ` p	ESC ` p
Keyboard Initiated Line Drop	ESC ` .(period)[2]	ESC ` .(period)[2]
Return to ENTER TERMINAL TYPE Message	ESC ` ,(comma)[2]	ESC ` ,(comma)[2]
Dvorak Keyboard Arrangement	ESC ` q	ESC ` q
Primary Keyboard Arrangement	ESC ` Q	ESC ` Q

[1] This key can be found only on the additional numeric keypad on the right side of the VT-100 (or the IBM PC).

[2] This key can be found only on the main keypad. Do not select a similar key on the additional numeric keypad on the right side of the VT-100 keyboard.

[3] For the SmarTerm VT100 emulator Delete Column Tab is ESC ` CTRL- ←.

Keyboard Definitions for the VT100 Terminal, Hazeltine & PCPlot

Control and Cursor Movement Keys

IBM FUNCTION	Hazeltine	PCPlot[1]
Enter	RETURN	↵
Clear	ENTER[2]	AF8 or AF10
Redisplay	CTRL–V	CTRL–V
Erase EOF	ESC DEL	ESC DEL
Delete Character	DEL	DEL
Toggle Insert Mode	. (period)[2]	AF9
Field Tab	CTRL–I	
Field Backtab	ESC CTRL–I	ESC
Column Tab	ESC →	ESC →
Column Backtab	ESC ←	ESC ←
Indent	ESC ↑	ESC ↑
Undent	ESC ↓	ESC ↓
PA1	F4 or ESC , (comma)[3] or ESC <	AF2 or ESC , (comma)[3] or ESC <
PA2	– (hyphen) or ESC > or ESC . (period)[3]	AF4 or ESC > or ESC . (period)[3]
PA3	, (comma)[2] or ESC / (slash) or ESC ? (question mark)	AF6
Newline	LF	CTRL–
Home	BACK SPACE	←
Cursor Up	↑	↑
Cursor Down	↓	↓
Cursor Right	→	→
Cursor Left	←	←

[1] Under PCPlot, AFn means to depress the ALT key with the Fn key.

[2] This key can be found only on the additional numeric keypad on the right side of the VT-100, Hazeltine or IBM PC.

[3] This key can be found only on the main keypad. Do not select a similar key on the additional numeric keypad on the right side of the VT-100, Hazeltine or IBM PC keyboard.

Program Function Keys

IBM FUNCTIONS	Hazeltine	PCPlot[1]
PF1	ESC 1[3] or F1	ESC 1[3] or F1
PF2	ESC 2[3] or F2	ESC 2[3] or F2
PF3	ESC 3[3] or F3	ESC 3[3] or AF1
PF4	ESC 4[3] or 7[1]	ESC 4[3] or F3
PF5	ESC 5[3] or 8[1]	ESC 5[3] or F4
PF6	ESC 6[3] or 9[1]	ESC 6[3] or AF3
PF7	ESC 7[3] or 4[1]	ESC 7[3] or F3
PF8	ESC 8[3] or 5[1]	ESC 8[3] or F6
PF9	ESC 9[3] or 6[1]	ESC 9[3] or AF5
PF10	ESC 0[3] or 1[1]	ESC 0[3] or F7
PF11	ESC – (hyphen)[3] or 2[2]	ESC – (hyphen)[3] or F8
PF12	ESC = or 3[2]	ESC = or AF7
PF13	0[2] F1 or ESC q[4]	F9 F1 or ESC q[4]
PF14	0[2] F2 or ESC w[4]	F9 F2 or ESC w[4]
PF15	0[2] F3 or ESC e[4]	F9 AF1 or ESC e[4]
PF16	0[2] 7[2] or ESC r[4]	F9 F3 or ESC r[4]
PF17	0[2] 8[2] or ESC t[4]	F9 F4 or ESC t[4]
PF18	0[2] 9[2] or ESC y[4]	F9 AF3 or ESC y[4]
PF19	0[2] 4[2] or ESC u[4]	F9 F3 or ESC u[4]
PF20	0[2] 5[2] or ESC i[4]	F9 F6 or ESC i[4]
PF21	0[2] 6[2] or ESC o[4]	F9 AF5 or ESC o[4]
PF22	0[2] 1[2] or ESC p[4]	F9 F7 or ESC p[4]
PF23	0[2] 2[2] or ESC _ (underscore)	F9 F8 or ESC _ (underscore)
PF24	0[2] 3[2] or ESC +	F9 AF7 or ESC +

Local Reset and Control Functions

IBM FUNCTION	Hazeltine	PCPlot[1]
Master Reset (Reinitialize Terminal)	CTRL–G	CTRL–G
Character Error Reset (Clear Parity Error)	CTRL–R	CTRL–R
Keyboard Unlock (IBM Reset Function)	CTRL–T	CTRL–T
Type-ahead Purge	CTRL–X	CTRL–X
Pacing Start	CTRL–S	CTRL–S
Pacing Stop	CTRL–Q	CTRL–Q

[1] Under PCPlot, AFn means to depress the Alt key with the Fn key.

[2] This key can be found only on the additional numeric keypad on the right side of the VT-100, Hazeltine or IBM PC.

[3] This key can be found only on the main keypad. Do not select a similar key on the additional numeric keypad on the right side of the VT-100, Hazeltine, or IBM PC keyboard.

[4] This character may be typed in as uppercase or lowercase.

Setup Functions

IBM FUNCTION	Hazeltine	PCPlot[1]
Set Column Tab	ESC ` CTRL-I	ESC ` CTRL-I
Delete Column Tab	ESC ` DEL	ESC ` DEL
Set Left Margin	ESC ` LF	ESC ` CTRL-
Set Home Line	ESC ` BACKSPACE	ESC ` ←
Delete All Column Tabs, Home Line and Left Margin	ESC ` ENTER[2]	ESC ` AF8 or ESC ` AF10
Improved Null Processing	ESC ` N	ESC ` N
3270 Null Processing	ESC ` n	ESC ` n
Zones Mode On	ESC ` z	ESC ` z
Zones Mode Off	ESC ` Z	ESC ` Z
Reverse Enter/Newline Keys	ESC ` e	ESC ` e
Restore Enter/Newline Keys	ESC ` E	ESC ` E
Reverse Column and Field Tab Keys	ESC ` c	ESC ` c
Restore Column and Field Tab Keys	ESC ` C	ESC ` C
Alpha in Numeric-Only Field	ESC ` v	ESC ` v
3270 Numeric Fields	ESC ` V	ESC ` V
3278 Insert Mode	ESC ` i	ESC ` i
3277 Insert Mode	ESC ` I	ESC ` I
Suppress Pacing	ESC P	ESC ` P
Restore Pacing	ESC p	ESC ` p
Keyboard Initiated Line Drop	ESC ` .(period)[3]	ESC ` .(period)[3]
Return to ENTER TERMINAL TYPE Message	ESC ` ,(comma)[3]	ESC, ` ,(comma)[3]
Dvorak Keyboard Arrangement	ESC ` q	ESC ` q
Primary Keyboard Arrangement	ESC ` Q	ESC ` Q

[1] Under PCPlot, AFn means to depress the Alt key with the Fn key.

[2] This key can be found only on the additional numeric keypad on the right side of the VT-100, Hazeltine or IBM PC.

[3] This key can be found only on the main keypad. Do not select a similar key on the additional numeric keypad on the right side of the VT-100, Hazeltine or IBM PC keyboard.

Keyboard Definitions for the VT220 Terminal and SmarTerm Emulator

Control and Cursor Movement Keys

IBM	VT220	SmarTerm
Enter	RETURN or ENTER	
Clear	NEXT SCREEN	F10
Redisplay	CTRL–V	CTRL–V
Erase EOF	F19	ALT–4
Delete Character	REMOVE	F7
Toggle Insert Mode	INSERT HERE	F6
Field Tab	TAB	
Field Backtab	FIND TAB	F5
Column Tab	FIND →	F5 →
Column Backtab	FIND ←	F5 ←
Indent	F18	ALT–3
Undent	F17	ALT–2
PA1	FIND , (comma) or FIND <	F5 , (comma) or F5 <
PA2	FIND . (period) or FIND >	F5 . (period) or F5 >
Newline	F13	ALT–F8
Home	F14	ALT–F9
Cursor Up	↑	↑
Cursor Down	↓	↓
Cursor Right	→	→
Cursor Left	←	←

Program Function Keys

IBM	VT220	SmarTerm
PF1	PF1	F1 or ESC 1
PF2	PF2	F2 or ESC 2
PF3	PF3	F3 or ESC 3
PF4	PF4	F4 or ESC 4
PF5	CTRL–5[1]	ESC 5
PF6	F6	ALT–F1 or ESC 6
PF7	F7	ALT–F2 or ESC 7
PF8	F8	ALT–F3 or ESC 8
PF9	F9	ALT–F4 or ESC 9
PF10	F10	ALT–F5 or ESC 0
PF11	F11	ALT–F6 or ESC – (hyphen)
PF12	F12	ALT–F7 or ESC =
PF13	FIND PF1	F5 F1 or ESC q
PF14	FIND PF2	F5 F2 or ESC w
PF15	FIND PF3	F5 F3 or ESC e

[1] This key can only be found on the main keypad. Do not select a similar key on the additional numeric key pad.

PF16	FIND PF4	F5 F4 or ESC r
PF17	FIND CTRL-5[1]	ESC t
PF18	FIND F6	F5 ALT-F1 or ESC y
PF19	FIND F7	F5 ALT-F2 or ESC u
PF20	FIND F8	F5 ALT-F3 or ESC i
PF21	FIND F9	F5 ALT-F4 or ESC o
PF22	FIND F10	F5 ALT-F5 or ESC p
PF23	FIND F11	F5 ALT-F6 or ESC [
PF24	FIND F12	F5 ALT-F7 or ESC]

Local Reset and Control Functions

IBM	VT220	SmarTerm
Master Reset (Reinitialize Terminal)	CTRL-g	CTRL-g
Character Error Reset (Clear Parity Error)	CTRL-r	CTRL-r
Keyboard Unlock (IBM Reset Function)	CTRL-t	CTRL-t
Type-ahead Purge	CTRL-x	CTRL-x
Pacing Start	CTRL-s	CTRL-s
Pacing Stop	CTRL-q	CTRL-q

Setup Functions

IBM	VT220	SmarTerm
Set Column Tab	SELECT TAB	F8
Delete Column Tab	SELECT DEL	F8 DEL
Set Left Margin	SELECT F13	F8 ALT-F8
Set Home Line	SELECT F14	F8 ALT-F9
Delete All Column Tabs, Home Line and Left Margin	SELECT NEXT SCREEN	F8 F10
Improved Null Processing	SELECT N	F8 N
3270 Null Processing	SELECT n	F8 n
Enter APL	SELECT a	F8 a
ASCII in APL	SELECT m	F8 m
Exit APL	SELECT A	F8 A
Zones Mode On	SELECT z	F8 z
Zones Mode Off	SELECT Z	F8 Z
Reverse Enter/Newline Keys	SELECT e	F8 e
Restore Enter/Newline Keys	SELECT E	F8 E
Reverse Column and Field Tab Keys	SELECT c	F8 c
Restore Column and Field Tab Keys	SELECT C	F8 C
Alpha in Numeric-Only Field	SELECT v	F8 v
3270 Numeric Fields	SELECT V	F8 V
3278 Insert Mode	SELECT i	F8 i
3277 Insert Mode	SELECT I	F8 I
Alternate display attributes (color)	SELECT d	F8 d
Normal display attributes	SELECT D	F8 D

Line Drop (hang up modem)	SELECT . (period)[1]	F8 . (period)
Return to ENTER TERMINAL TYPE Message	SELECT , (comma)[1]	F8 , (comma)
Dvorak Keyboard Arrangement	SELECT q	F8 q
Primary Keyboard Arrangement	SELECT Q	F8 Q

[1] This key can only be found on the main keypad. Do not select a similar key on the additional numeric keypad.

USING AN ANSI-CONFORMING TERMINAL WITH THE IBM 3090 †

Set your terminal to use 7 bit EVEN parity.

Use the GANDALF class code IBM, or the class code 71.

In response to ENTER TERMINAL TYPE: vt100

The following is a subset of the IBM function keys and the equivalent VT100 key sequence. The ASCII code sequence is sometimes useful for duplicating the VT100 key sequence.

IBM FUNCTION	VT100 Key Sequence	ASCII Code Sequence
Enter	RETURN	CR
Clear	ENTER	ESC O M
Cursor movement	Arrow keys	
Up		ESC O A
Down		ESC O B
Right		ESC O C
Left		ESC O D
Delete character	DELETE	DEL
Insert Mode Toggle	keypad period (.)	ESC O n
Erase EOL	ESC DELETE	ESC DEL
Newline	LINE FEED	LF
Home	BACKSPACE	BS

Programmable function keys:

PFK1	PF1	ESC 1
PFK2	PF2	ESC 2
PFK3	PF3	ESC 3
PFK4	Keypad 7	ESC 4
PFK5	Keypad 8	ESC 5
PFK6	Keypad 9	ESC 6
PFK7	Keypad 4	ESC 7
PFK8	Keypad 5	ESC 8
PFK9	Keypad 6	ESC 9
PFK10	Keypad 1	ESC 0
PFK11	Keypad 2	ESC -
PFK12	Keypad 3	ESC =
PA1	PF4	ESC ,

Acts as a toggle between CP and CMS.

PA2	Keypad hyphen (-)	ESC .

In CMS this clears the screen except for the command line.

Local Reset and Control Functions:

Master Reset	CTRL and g
Keyboard Unlock	CTRL and t
Type-ahead Purge	CTRL and x
Pacing Start	CTRL and s
Pacing Stop	CTRL and q

† From H. Berry's (AMD) Lectures.

VIRTUAL MINIDISKS

PERMANENT MINIDISKS	Disks that persist across logons. These disks are defined in your CP directory.
TEMPORARY MINIDISKS	Disks that are automatically destroyed at LOGOFF.

The operating system CP identifies a minidisk LINKed to your Virtual Machine by a 3-digit hexidecimal number called the minidisk VIRTUAL ADDRESS. For example, every user should have a permanent Read/Write minidisk at Virtual Address 191. To view all the minidisks LINKed to your Virtual Machine use the command:

```
Query DASD
```

The operating system CMS identifies a minidisk ACCESSed to your Virtual Machine by a FILEMODE letter (A to Z). The filemode defines the CMS search order for locating a file on one of the user's minidisks. Normally, the user's 191-disk is accessed at filemode A (but the filemode of a disk can easily be changed). To view all the minidisks ACCESSed by your Virtual Machine use the command:

```
Query DISK
```

EXAMPLE: A typical minidisk configuration.

label	CUU	M	status	cyl
BER191	191	A	R/W	3
CMS190	190	S	R/O	28
CMS59E	59E	T	R/O	150
CMS59F	59F	U	R/O	100
CMS19E	19E	Y/S	R/O	100

The minidisk label is assigned to one's own permanent minidisks using the FORMAT command. The column marked CUU refers to the Virtual Address for each minidisk. The column marked M identifies the filemode for each disk. All the disks with status R/O are system disks accessed in Read Only mode. The size of each minidisk is measured in cylinders (.6 M.bytes/cylinder).

The GIME command is useful for getting read access to someone's permanent minidisks, and for creating temporary scratch disks:

EXAMPLE: To get read access to Berry's 191-disk, issue the command:

```
GIME BERRY
```

EXAMPLE: To create a 10 cylinder temporary minidisk:

```
GIME 10
```

CMS FILES

FILEID: A file is identified by the three parts:

	FILENAME	FILETYPE	FILEMODE	
Examples:				
	MYFILE	FORTRAN	A1	A FORTRAN source file
	MYFILE	TEXT	A	Object code file produced by compiler
	MYFILE	LISTING	A	Listing file produced by compiler
	MYFILE	MODULE	A	Executable image file
	MYLIB	TXTLIB	B	TEXT library
	FILE	FT02F001	A	Default FORTRAN fileid for unit 2
	ELASTIC	DATA	*	Wildcard applies only to input files
	PROFILE	EXEC		User file is executed automatically at LOGON (or whenever IPL CMS is issued).
	PROFILE	XEDIT		User created file is executed automatically whenever the user edits a file.

IBM does NOT support:

(1) Logical names
(DDNAME acts somewhat like a logical name)

(2) Hierarchical directories
(each minidisk has its own directory)

(3) File version numbers, or cycle numbers

Example: To compile, load and execute the file: TEST FORTRAN C
where the minidisk with filemode C is a Read/Write disk.

```
FOR  TEST                 Creates object file TEST TEXT C
LOAD TEST (START          Load and execute
```

or

```
FOR  TEST
LOAD TEST
START
```

The GLOBAL command is used to declare CMS libraries and their search order. At BNL the user's default PROFILE EXEC issues the following GLOBAL commands:

```
GLOBAL TXTLIB  VSF2FORT CMSLIB TSOLIB
GLOBAL LOADLIB VSF2LOAD
```

VSF2FORT contains the standard IBM service and mathematical subroutines. VSF2LOAD is needed to dynamically load certain I/O routines at execution time. The remaining libraries are needed for some system operations. It may occur that the default GLOBAL libraries are destroyed by a severe error condition in which case one needs to reenter the above GLOBAL statements.

Additional user TXTLIB libraries can be appended by use of the command:

```
ADDLIB txtlib1 txtlib2 . . .
```

or use

```
GLOBAL TXTLIB VSF2FORT CMSLIB TSOLIB txtlib1 txtlib2 . . .
```

EXAMPLE: TO COMPILE, LOAD AND EXECUTE A FORTRAN PROGRAM

Program MMC2PW uses the following files:

```
      UNIT   RELAVANT FORTRAN STATEMENTS

       2     WRITE(2,xxxx)

       5     READ  *
             READ  xxxx,

       6     PRINT *
             PRINT xxxx,

       3     OPEN(UNIT=3, FILE='ELASTIC', STATUS='OLD', ERR=1000)
             READ(3,xxxx)
             READ(3,*)
```

VAX METHOD:

```
PUBLIC  IMSL
FOR     MMC2PW
LINK    MMC2PW, IMSLIBS/LIB
DEFINE  ELASTIC ELASTIC.DAT
DEFINE/USER  SYS$INPUT  MMC2PW.INP
RUN     MMC2PW
```

IBM METHOD:

```
GIME    PUBLIC
ADDLIB  IMSLS
FOR     MMC2PW
FILEDEF 2  DISK MMC2PW OUT A (PERM LRECL 133
FILEDEF 5  DISK MMC2PW INP *
FILEDEF ELASTIC DISK ELASTIC DAT *
LOAD    MMC2PW  (START
```

NOTES:

(1) The PUBLIC minidisk contains the mathematics libraries.
(2) ADDLIB appends the single precision IMSL TXTLIB library to the GLOBAL library list.
(3) The file MMC2PW FORTRAN contains the file to be compiled. It may exist on any one of the accessed minidisks.
(4) Unit 2 (output) is associated by the FILEDEF command with the file MMC2PW OUT on the A-disk. PERM prevents the file definition from being cleared by a future compilation, but NOT by an ABORT condition. LRECL 133 makes the record length 133 characters (the default is 80).
(5) Unit 5 (input) is associated with the disk file MMC2PW INP. The asterisk is a wild card for the filemode. All accessed disks are searched for the file in the order A to Z.
(6) Unit 6 (output) defaults to the terminal screen.
(7) Unit 3 (input) is associated by the FORTRAN OPEN statement with the name ELASTIC. The FILEDEF command associates the name ELASTIC with the file ELASTIC DAT located on one of the accessed disks.
(8) LOAD creates an executable program in memory from the object file MMC2PW TEXT created by the FORTRAN compilation. The GLOBAL libraries are searched for unsatisfied externals. The option START executes the program.

The file MMC2PW EXEC A contains the following REXX statements:

```
/* EXEC TO RUN THE MMC2PW PROGRAM */
'GIME PUBLIC'
'ADDLIB IMSLS'
'FILEDEF 2 DISK MMC2PW OUT A (LRECL 133'
'FILEDEF 5 DISK MMC2PW INP *'
'FILEDEF ELASTIC DISK ELASTIC DAT *'
'LOAD MMC2PW (START NOMAP'
EXIT
```

The EXEC is executed interactively by typing the command:

MMC2PW

To execute the EXEC as a batch job, issue the command:

BATCH SUBMIT (options) MMC2PW

NOTES:

1. A REXX exec must begin with a comment statement: /* comment */
   Any statement within quotes is passed to CMS without interpretation by the command language interpreter.
   The EXIT statement is an optional REXX command to terminate the file.

4. Some important differences between interactive and batch execution:
   (a) A batch job is executed by another virtual machine called the Batch Worker. The disk configuration for the submitter's machine is not the same as for the Batch Worker's machine. In particular, the submitter's 191-disk is automatically accessed as the Batch Worker's B-disk in Read Only mode. Thus, the reference to the two files:

   ```
   MMC2PW  INP *
   ELASTIC DAT *
   ```

   will succeed if the files appear on the submitter's 191-disk. The asterisk is a wild card used to specify the filemode. CMS will search for the files on all of the batch worker's disks to locate them, presumably at filemode B.

   (b) The batch worker's A-disk is available for output files created by the batch job. Any files that remain on this disk when the job completes will be returned to the submitter's Reader. The example case creates the output file:

   ```
   MMC2PW OUT A
   ```

   on the batch worker's A-disk, and hence will be returned to the user's Reader when the job completes.

   (c) The batch job writes the information that is normally produced interactively to a Console file using the submitter's userid as the filename. This file also contains details pertaining to the batch worker's performance. This file will also be returned to the submitter's Reader.

   (d) The Batch Worker executes the submitter's PROFILE EXEC file before executing the file MMC2PW EXEC, but the option NOPROF can be used to suppress the PROFILE.

   (e) Limits are placed on the batch worker's use of machine resources. These limits (such as CPU time and Printer and Punch file sizes) can be specified as options.

Logon userid	To identify a user to the system, and to permit access to the system.
LOGout [HOLD]	To terminate a session. HOLD retains the connection on a switched line so that it is not necessary to reconnect in order to logon again.
Xedit fn ft fm	The CMS editor.
DEFine DEF STOR 4M	Makes changes to your virtual machine configuration (i.e., to increase your virtual memory or add virtual devices.) When you change virtual memory it is then necessary to IPL CMS before you can continue.
IPL CMS	Simulates an Initial Program Load of CMS.
DISConn [HOLD]	Disconnects your terminal from the system while the virtual machine continues operation. Temporary files are retained (but not if the system should crash).
CP command	Transmits commands directly to the Control Program (CP).
CMS command	Transmits commands directly to the Conversational Monitor System (CMS).
HI HX HT	Immediate command to halt an EXEC. Immediate command to halt execution and return to CMS. Immediate command halts terminal output, but continues execution.
FTP node	Allows you to transfer files to/from other machines connected to Ethernet using the TCP/IP protocal.
TELNET node	Allows you to logon to other machines connected to Ethernet using the TCP/IP protocal.
SETPFN	Sets the PF function keys for the CMS environment. PF7 and PF8 retrieves previous commands forward/backward.

HELP and QUERY

AID keyword	An index that suggests how to find information related to a given keyword.
Help	Provides information on VM commands, Execs, Error messages, Libraries, etc.
HELPRINT	Prints the Help response to a central printer, or stores the information on disk.

Query DISK	Displays the status of your accessed disks. That is, all disks which are known to CMS.
Query DASD	Displays the status of your linked disks. That is, all disks which are known to CP.
	Other Query commands of interest are: Query files, Query Reader, and Query Virtual.
USERS	Displays information about logged-on users.
WHOS user	Gives information about a VM userid.
CONSPIRE	Bulletin board and conferencing system. Useful for asking questions or getting information from other users.
DISKLIST $gg	Lists all minidisks that belong to a user or a group.

FORTRAN COMMANDS

FORtvs2 name	Compiles the first file found with fileid: name FORTRAN *
LOAD name	Loads into memory the object file: name TEXT *
INCLUDE name	Includes the object file: name TEXT * as part of a load sequence.
START	Starts execution of a loaded program.
Genmod name	Creates a nonrelocatable (MODULE) file from a loaded program.
GLobal TXTLIB lib1 lib2 . . . libn	Declares libraries to be searched for unsatisfied externals. A member of a library consists of one or more routines.
TXTlib GEN libname fn1 fn2 . . .	Creates a TXTLIB library using the given TEXT files.
ADD libname fn1 fn2 . . .	Adds TEXT files to a library.
DEL libname m1 m2 . . .	Deletes members from a library.
MAP libname	Produces a map listing of a library.
TXTADD libname fn1 fn2 . . .	Adds TEXT files to a TXTLIB library but makes each ROUTINE a separate member. TXTLIB makes each FILE a separate member.
FIledef ddname DISK fn ft fm	Establishes the connection between program I/O and a disk file. ddname may be the unit number if there is no OPEN statement, or it is the name specified by FILE = 'ddname' in the OPEN statement.

FIledef	When no arguments are given this command displays a list of current filedef definitions.
Query TXTLIB	Displays a list of the current TXTLIB libraries to be searched for unsatisfied externals during a Load sequence.

MINIDISK COMMANDS

FILEList fn ft fm	Uses the Xedit environment to display information about CMS files residing on accessed disks. You can issue CMS commands for the files directly from the displayed list.
GIME 10 GIME 10 E	To create a scratch minidisk of 10 cylinders. The scratch minidisk is assigned filemode E.
GIME BERRY GIME BERRY 195 GIME BERRY 195 F	Accesses BERRY'S 191-disk in read only mode. Accesses BERRY'S 195-disk in read only mode. Accesses BERRY'S 195-disk in read only mode at filemode F.
LINK	Links a minidisk to your virtual machine. This makes the minidisk known to CP.
ACCESS	Accesses a linked minidisk. This makes the minidisk known to CMS. GIME is equivalent to the LINK, ACCESS sequence.
DROP E	Removes the minidisk with filemode E from your virtual machine, so the disk is neither linked nor accessed.
DETach 205	When the disk at address 205 is linked to your machine but not accessed, this is the way to get rid of it.
COPYfile	Copies and/or modifies disk files.
DIFFER	Compares two files and reports differences to the user. More flexible and powerful than IBM's COMPARE command.
ERASE	Deletes one or more files from a read/write disk. DISCARD will do the same thing.
PRint	Prints a file on one of the central printers.
PUnch	Punches a disk file to your virtual punch.
RECOVER	To restore part or all of a minidisk from a backup tape. If you accidentally destroy a file this is the way to recover yesterday's version.
Rename	Changes the fileid of one or more files on a read/write disk.
Type	Displays all or any part of a file at the terminal.

ARCHIVE	Use this command to store files in archival storage for long term retention. ARCHIVE LIST will produce a listing of all your archived files.
SPACE	Creates permanent minidisks and temporary disks that survive logout, and makes other directory updates not normally available to the general user. Only a Space Administrator for a group can make permanent minidisks for members of the group.
FORMAT	After a permanent minidisk is created it is necessary to format the disk before it can be ACCESSed to your virtual machine. Default blocksize is 4K. To follow convention user BLACK would use the label BLA193 for his 193-disk. Example: FORMAT 193 C (BLA193
DIRMaint	Gives information about your own VM directory entry, or makes directory updates. You can use this command to set WRITE passwords for one of your minidisks. You should leave your READ password as ALL. Disks that are READ protected are not backed up by the nightly backup.
INSTALL	Allows a group of people to share the use of specially designated minidisks. The operating system does not allow two people to have write access to a minidisk at the same time; using INSTALL to maintain files on the disk is a way of avoiding this conflict.

BATCH COMMANDS

BATCH	A utility for submitting jobs to be executed independently of your own virtual machine. Only batch jobs are permitted access to the tape drives.
SETUP	Acquires real tape drives and mounts the first tape on each drive. This command must be part of a batch job.
MOUNT	To perform subsequent TAPE volume mounting and verification.
TAPE	Dumps CMS formatted files from disk to tape, loads previously dumped
TAPEIT	An Exec for writing source programs, and other coded files to tape in a format suitable for transport to non-IBM computers.
DISKIT BLOCKIT	These two Execs combined do the reverse of TAPEIT. Can be used to read coded tape files created by non-IBM computers.
CARD DUMP	Dumps a file to the Virtual Punch which then gets sent back to the submitter's Reader.

SPOOL COMMANDS

RDRList	To list the files in your virtual Reader. Uses the Xedit environment. Subcommands exist to RECEIVE files to disk or PEEK at their contents.
TRANsfer	To transfer your closed spool files to a specified user or queue, or to reclaim closed spool files that you created.
CHange	To modify attributes of closed spool files.
PURge	To remove a closed spool file from the system before it is printed on a real device or before it has been read by a user.
SPool	To modify the spooling options in effect for a virtual device. May be used to redirect a file to a remote location.
QSPOOL	Allows any user to see which spool files exist for another user or for the entire system.
PRTSTAT	Show the status of print queues.

COMMUNICATIONS COMMANDS

NEWS	Displays news items. Maintains a pointer to last item read.
NOTE	To send a short communication to one or more people on this computer or to a remote site using the Remote Spooling Communications Subsystem (RSCS) network. Uses the Xedit editor to compose the note.
NAMES	To display a menu from which you can create, change, and remove entries in a "userid NAMES" file. The NAMES file is referenced by both the NOTE, SENDFILE and TELL commands.
SENDFILE	To send files to one or more people on this computer or on remote sites using the RSCS network. SENDFILE and NOTE both reference a "userid NAMES" file. By creating a NAMES file, you can identify recipients by using their nickname which is automatically converted into a node and userid.
TELL	Sends a message to a logged-on VM user. Can be used with the NAMES file to address a person by nickname.

ADDITIONAL USEFUL EXEC'S:

ADDLIB: Add a library to a GLOBAL list. If the library is already in the list, then it is deleted. Usage:

ADDLIB <name> {TXTLIB | LOADLIB | MACLIB}

The new library is placed first so that it is first in the search order.

AFFON: Make a <name> AFFON file for use with the debugger with all subroutines turned off by default. Fortran library subroutines are not included -- hopefully they do not contain bugs. Usage:

AFFON <name>

To turn on debugging for a subroutine, edit the <name> AFFON file and delete the * in column 1 for that subroutine. It is essential to debug only a few routines to obtain reasonable efficiency. Note that one must specify

GLOBAL TXTLIB TSOLIB ...

to use the debugger.

FTNLIB: Make a sensible TXTLIB from a Fortran source so that only routines actually needed are loaded. The IBM default is to load all the subroutines from a file if one is used(!). Usage:

FTNLIB <Fortranname> <Libraryname> (<options>

where <options> are any options for FORTVS2 except those related to printing, which is suppressed. If <Libraryname> TXTLIB exists, then the routines from <Fortranname> FORTRAN are added to it. This EXEC splits <Fortranname> FORTRAN into individual routines by looking for END statements and compiles each separately.

GETFM: Get the file mode for a file and stack it LIFO. If more than one file with <fname> <ftype> exists, the first is found. Usage:

GETFM <fname> <ftype>

If the file does not exist, then the nonzero RC from LISTFILE is returned and nothing is stacked.

SCRATCH: Make a temporary scratch disk if one does not exist and flag it with a file <username> <flag>. Usage:

SCRATCH <ncyl> <flag>

where <ncyl> is the desired number of cylinders (1 cylinder = .6 MB). The <flag> should be DISKnnn, DISKSCR,.... A file <username> DISK191 must exist. Flagging disks with files in this manner seems to be the best way to avoid errors (especially in EXEC's) arising from the fact that the FMODE assigned to a disk can change.

COMMUNICATIONS

1.BITNET This is now available on the IBM. The node to use is BNLVMA.

2.TCPIP is the protocol between the VAX CLUSTER and the IBM. It can transfer files in both directions. There are on-line HELP files available under HELP FTP or HELP TELNET. In all cases enter LOCAL_FILE in the form FILENAME.EXT. To use it:

VAX → IBM

LOG onto VAX CLUSTER (any node)

FTP BNLVMA↩

ftp> It will ask you to LOG onto the IBM

—

—

SEND LOCAL_FILENAME REMOTE_FILENAME

—

—

ftp>QUIT↩ to leave FTP

DO NOT DO *cntrlY* OR YOU WILL BE LEFT LOGGED ONTO THE IBM

If you do you must call the computer operators and ask them to

FORCE you out of the IBM.

IBM → VAX

FTP BNLCL1(2,3,4)↩

it will ask you to log onto the CLUSTER

ftp:

PUT LOCAL_FILE <REMOTE_FILE>↩

ftp:

QUIT↩

Network & Communications

I. Introduction

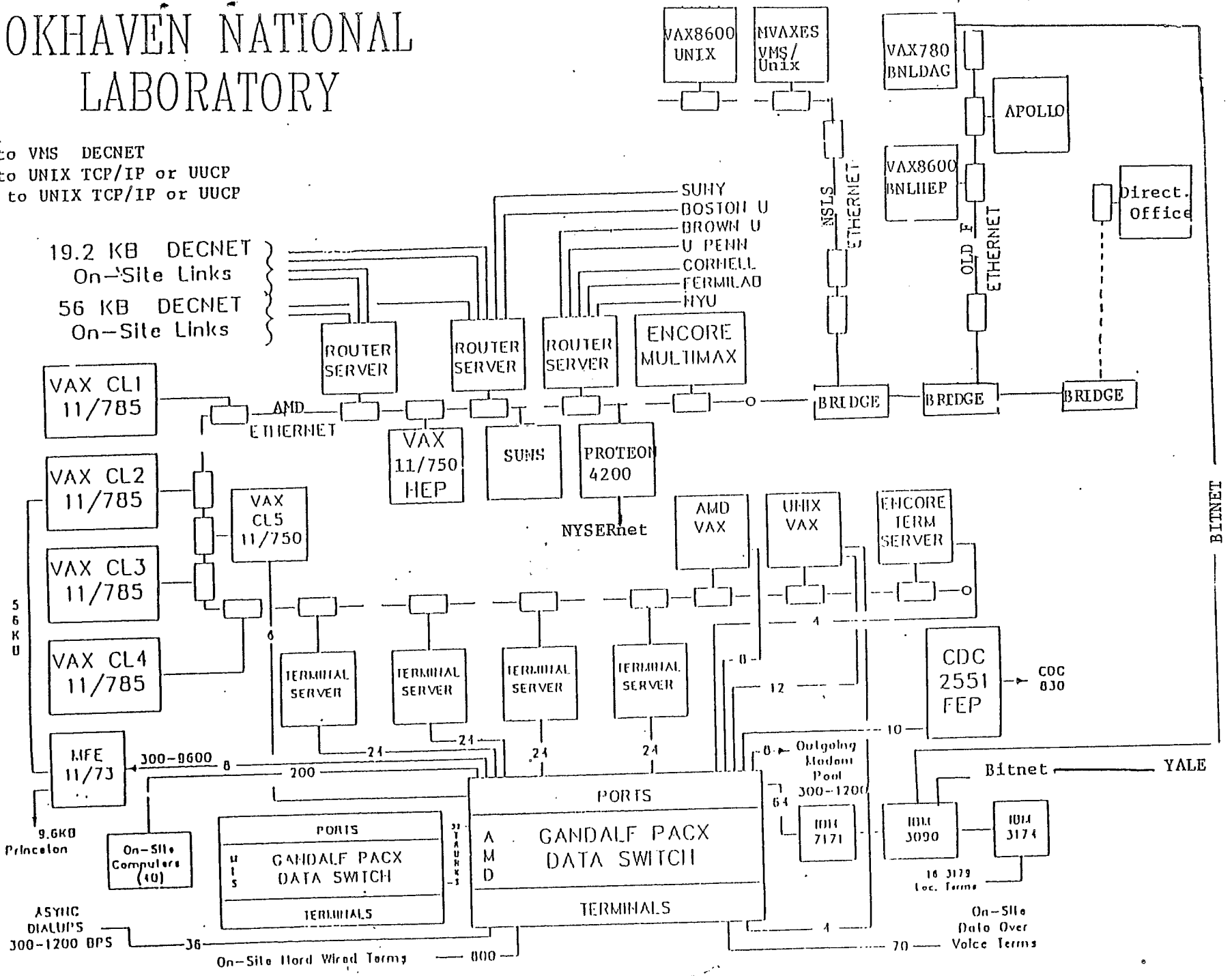
An overview of the network facilities is given in Section ii. Section iii describes the BNLDAG Network . Informations on the Bnlcluster is given in section iv , IBM 3090 in section v, PC file transfer in section vi. and UNIX VAX in section vii.

ii) OVERVIEW OF BNL - COMPUTER FACILITIES

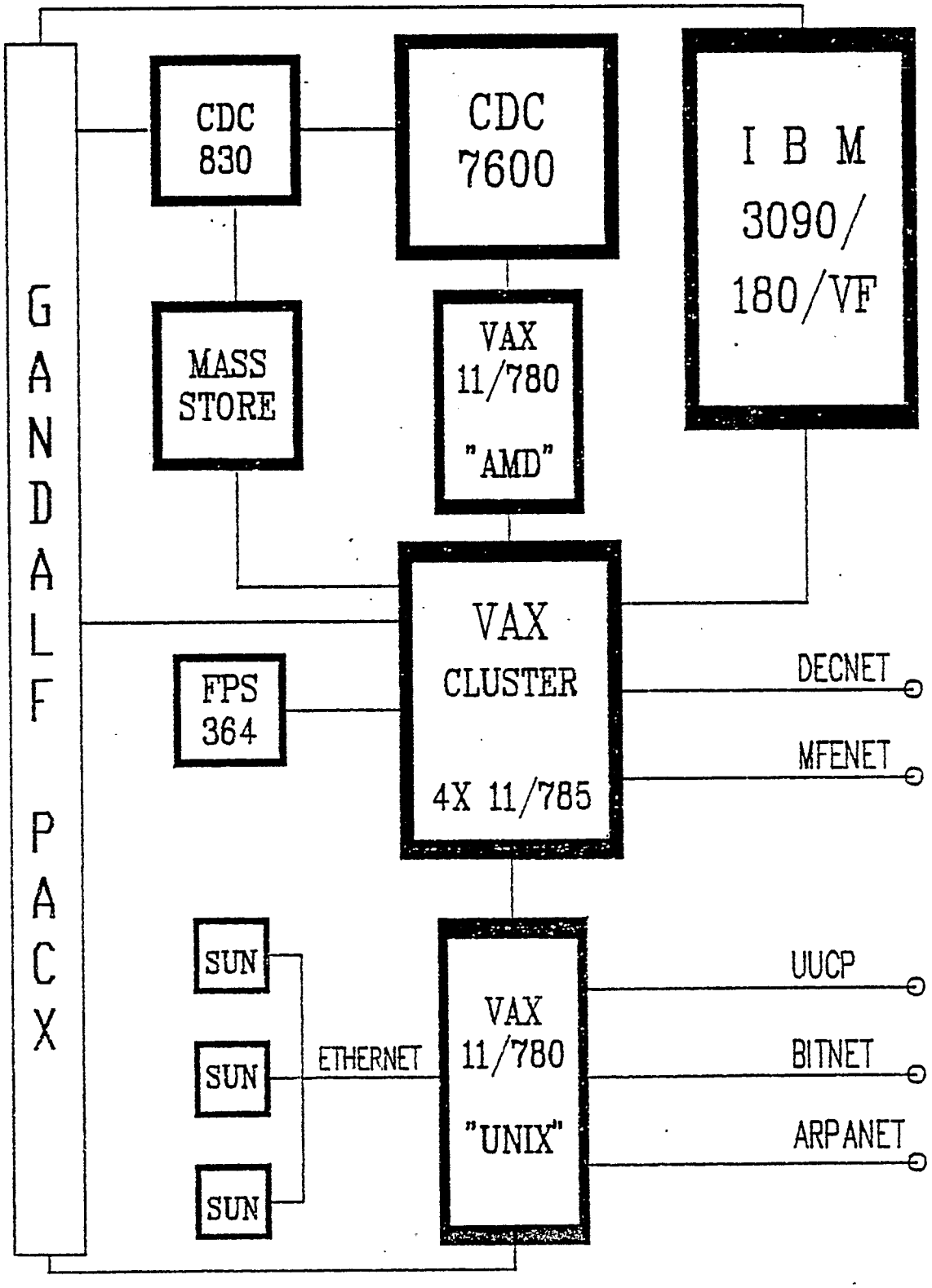
Following shows the link between the major facilities available at Brookhaven National Laboratory. These includes the NSLS computing facility the Central Scientific Computing facility (AMD) and the AGS Computing Facility (see sec. ii), etc.

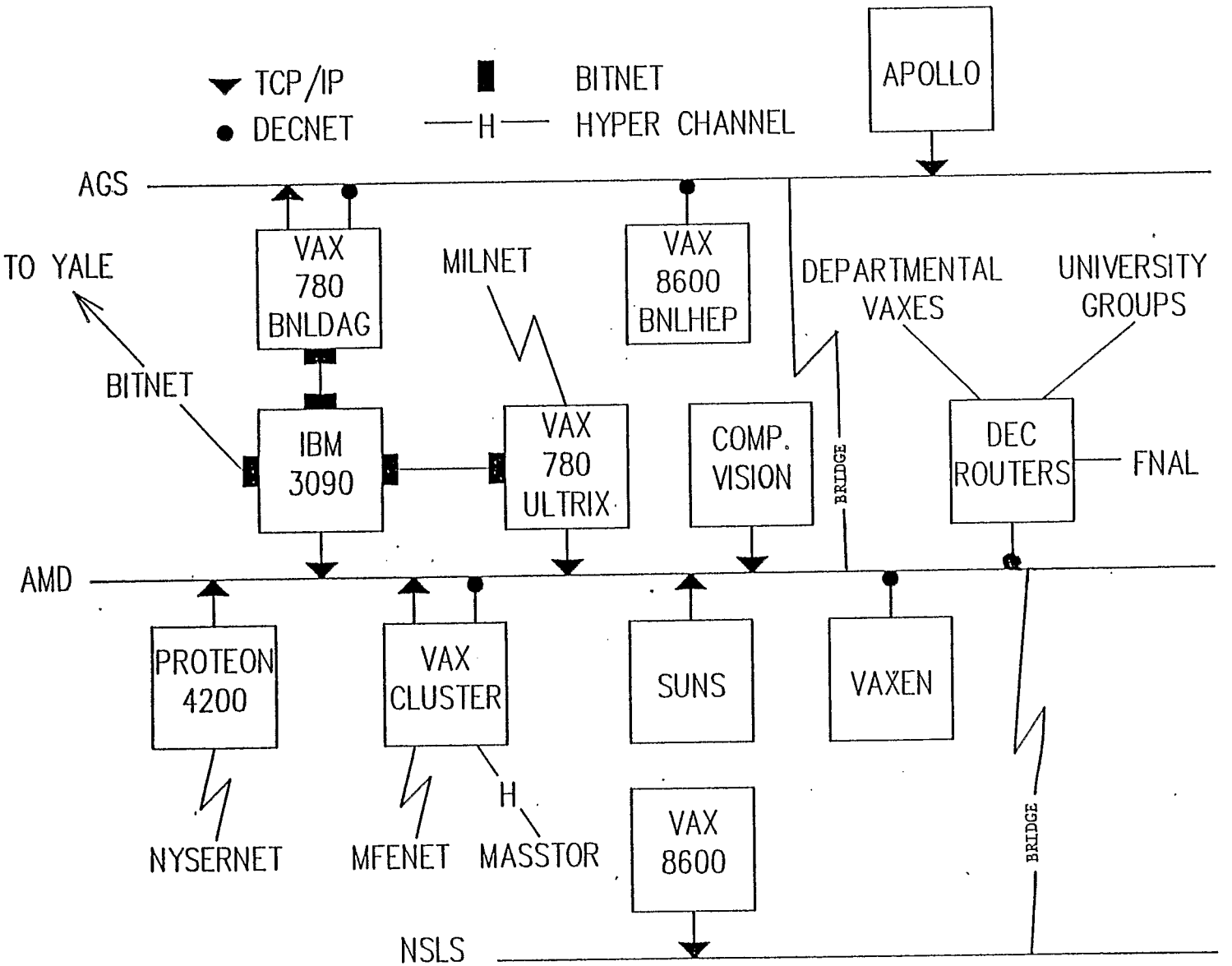
** We thank R. Imossi for the informations on BNLDAG, R. Thomas for UNIX, P. Kessler and A. Natoli for the BNLCluster and D. Stampf for KERMIT.

BROOKHAVEN NATIONAL LABORATORY



CENTRAL SCIENTIFIC COMPUTING FACILITY




TCP/IP
DECNET
BITNET
H HYPER CHANNEL
APOLLO
AGS
TO YALE
VAX 780 BNLDAG
MILNET
VAX 8600 BNLHEP
DEPARTMENTAL VAXES
UNIVERSITY GROUPS
BITNET
IBM 3090
VAX 780 ULTRIX
COMP. VISION
BRIDGE
DEC ROUTERS
FNAL
AMD
PROTEON 4200
VAX CLUSTER
SUNS
VAXEN
BRIDGE
H
NYSERNET
MFENET
MASSTOR
VAX 8600
NSLS

SECTION ii

I. Networks Introduction:

There are four major world-wide networks:

TCP-IP
BITNET
DECnet
and
UUCP

1. TCP-IP: Internet, Arpanet, Milnet, Nysernet, NSFnet, CSNET, X25net, EDVNET. Also there are programs that use TCP-IP such as TELNET, FTP, etc.

2. Bitnet: RSCS, VNET, Netnorth, EARNnet, JNET, KNET.

3. DECnet: Hepnet, Physnet, Span, CCNET.

4. UUCP: Usenet.

II. Network Capabilities:

There are many network capabilities such as task to task communication, remote job submittal, record level access, etc., however, the main functions are file transfer, mail, and remote login. Remember even though your computer may be connected to a particular network, it may have only one of the possible network functions. This is true of many networks only mail is possible.

III. BNLDAG Network Capabilities.

The BNLDAG VAX is directly connected to all networks except UUCP which is a Unix specific network, however, even UUCP is indirectly accessible through a gateway - (see GMAIL).

IV. BNLDAG Network commands:

Nodes	remote login	file transfer	mail *
DECnet	$ SET HOST node	$ COPY ...	$ MAIL
Bitnet	not available	$ SEND/FILE **	$ MAIL > SEND To:BITNET%"..."

TCP-IP	$ TELENET node	$ FTP node	$ MAIL > SEND To:EXOS%"..."
UUCP	not available	not available	$ GMAIL > SEND To: ...UUCP

...=username@node, e.g. parsa@bnlcl1

* Note: It is recommended that GMAIL be used for all network mailing instead of mail.

** Note: BITNET can transfer only one way, out.

V. GMAIL:

GMAIL (Gateway Mail) is an intelligent mailing program on VAX computers. It's equivalent on IBM computers is SENDGATE. GMAIL is patterned closely after VMS MAIL, but it takes the worry and confusion away from a user on how and what network to use when sending mail to a friend on a remote computer. All four major networks are handled. For example:

```
$ GMAIL
> HELP                          ! get help in general
> HELP GATEWAYS                 ! get help on gateways
> SEND
To username@node.uucp           ! UUCP mode
    ........
> SEND
To username@node.mil            ! Milnet mode
    ........
> SEND
To BNLCL1::username             ! DECnet mode
    ........
> SEND
To user@BNLMVA.BITNET           ! BNL IBM 3090
```

VI. References:

1) DECnet:

VMS Mail Utility manual
VMS Phone Utility manual
VMS DCL Dictionary

Interactive help:

```
$ HELP COPY
$ HELP MAIL
$ HELP SET HOST
```

$ HELP PHONE

2) Bitnet:

BNLDAG::BNL$MANUAL:BITNET.MAN
BNLDAG::BNL$MANUAL:GMAIL.MAN
JNET Users Guide

Interactive help:

$ HELP SEND
$ HELP RECEIVE

3) TCP-IP:

BNLDAG::BNL$MANUAL:INTERNET.MAN
EXOS TCP/IP Network Software Reference Manual

4) UUCP:

See the section on Unix-Vax

VII. Labnode - Network Table:

Lab Nodes		Network			
node	type	DECnet	Bitnet	TCP-IP	UUCP
BNLCL1	VAXVMS	X		X	
BNLCL2	VAXVMS	X		X	
BNLCL3	VAXVMS	X		X	
BNLCL4	VAXVMS	X		X	
BNLDAG	VAXVMS	X	X	X	
all other lab VAXes		X			
BNLVMA	IBM 3090		X	X	
BNL-GW	VAX/UNIX		X	X	X
BNL-APPOLLO	Appollo node			X	
SUN work stations				X	X

Section iv

BNLCluster Network

Protocol	Node	Remote Login	File Transfer	Mail
DECnet	All	SET HOST node	COPY	Mail
TCP/IP	All	TELENET node	FTP node	Mail >SEND TO:EXOS%"User@node"
MFEnet	BNLCL2	Yes	Yes	Yes
Kermit	All	No	Yes	No?
Bitnet	None	-	-	-

For example, to send a message to BNL (UNIX VAX) node to username PARSA

```
$ MAIL
  MAIL> SEND
  TO:   EXOS%"PARSA@BNL"
  SUBJ: ......

  ...........
  .........
  ctrl Z      (to end)
or ctrl C     (to cancel)
```

More detailed information is given in section iii, (BNLDAG) that may be used.

SECTION v

COMMUNICATION TO AND FROM IBM 3090

COMMUNICATIONS

1.BITNET This is now available on the IBM. The node to use is BNLVMA.

2.TCPIP is the protocol between the VAX CLUSTER and the IBM. It can transfer files in both directions. There are on-line HELP files available under HELP FTP or HELP TELNET. In all cases enter LOCAL_FILE in the form FILENAME.EXT. To use it:

VAX → IBM

LOG onto VAX CLUSTER (any node)

FTP BNLVMA↩

ftp> It will ask you to LOG onto the IBM

—

—

SEND LOCAL_FILENAME REMOTE_FILENAME

—

—

ftp>QUIT↩ to leave FTP

DO NOT DO *cntrl*Y OR YOU WILL BE LEFT LOGGED ONTO THE IBM

If you do you must call the computer operators and ask them to

FORCE you out of the IBM.

IBM → VAX

FTP BNLCL1(2,3,4)↩

it will ask you to log onto the CLUSTER

ftp:

PUT LOCAL_FILE <REMOTE_FILE>↩

ftp:

QUIT↩

SECTION vi

PC FILE TRANSFER

There are several ways of file transfer to and from Pcs and main frames, (e.g. IBM-PCs to (and from) VAXes). One of the commonly used programs is the IBM-Pc-Intercome (ic) program (which emulates the PC to a VT100 terminal), and another is the program KERMIT.

EXAMPLES:

I. FILE TRANSFER VIA INTERCOME (Version 3.+, some of the keys differ in older versions)

1. Use "ic"
2. Login
3. Press F9 {om PC, to get the MENUE}
4. Press F10 {twice and turn on the GANDALF}

5. Login to the Main Frame (e.g. VAX)

6. Press F9 {om PC, to get the MENUE} then
   Press F5 {file function key} and then
   Press F2 {start collecting text, key}
   or Press F3 {for appending text to an existing file}

7. Press F10 {on PC twice}' to get back to the HOST computer (e.g. VAX) and give commands, e. g.
   $ type VAXfilename (you want to transfer).
8. Press F9 {on PC to get back the MENUE} then
   Press F5 {file function, key} and then
   Press F6 {save collected text, key}

9. Press F10 {on PC twice}, to get back to the HOST computer to logout, if the transfer is completed, or continue with other works.

II. File transfer via KERMIT

Kermit is the name of a communications protocol that allows files to be transferred between two computers. It's primary use is to transfer files between a microcomputer such as an IBM PC and a larger multiuser system. Other protocols are usually preferable for other situations.

This protocol is capable of transferring binary as well as text files and does so in an error free fashion. A typical session is shown below where files are transferred between an IBM PC and the Ultrix Vax. The usage is quite similar for other computer types.

In the following, my annotation is preceded by a semi-colon. The computer output has been mapped to upper-case, the user input is shown in lower case.

```
A> kermit                    ; A version of this protocol is available from
                             ; the cscf for the pc.  It is run by typing
                             ; kermit.

CUCCA IBM-PC KERMIT-86 VERSION 1.20
KERMIT-86> set port com1     ; Pick com1/2 on the PC - whichever
                             ; is connected to a modem or gandalf
                             ; switch

KERMIT-86> set baud 9600     ; Modify as appropriate.
KERMIT-86> connect
[CONNECTING TO HOST, TYPE CONTROL-]C TO RETURN TO THE PC
BAUD RATE IS 9600, CONNECTING OVER PORT COM1]
                             ; This program is now behaving as a
                             ; simple terminal emulator.  Toggle
                             ; the gandalf switch, or turn on your
                             ; modem.  This assumes a gandalf.


ENTER CLASS 76               ; Ultrix Vax
CLASS START

ULTRIX-32 V1.2 (BNL) LOGIN AS HELP TO REGISTER

LOGIN: drs
PASSWORD:

                             ; The message of the day appears

BNLVAX<1> kermit             ; Same name on the vax.
C-KERMIT, 4C (057) 31 JUL 86, 4.2BSD
TYPE ? FOR HELP
C-KERMIT> send utilh.c       ; We decide to send a file from the
                             ; Vax to the PC.  If you wait a few
                             ; seconds, you will see some gibberish
                             ; on the screen as the program on the
```

```
                                ; VAX tries to "shake hands" with the
                                ; PC program.

ctrl-]C                         ; Back the the PC
KERMIT-86> rec                  ; and tell it to receive a file.  It
                                ; will have the same name as the VAX
                                ; file and will be place in the
                                ; default directory.  A bell will ring
                                ; when the transfer is complete.

KERMIT-86> conn                 ; re-connect to the VAX
C-KERMIT> set file type binary  ; Prepare to transfer a binary file
C-KERMIT> rec                   ; now receiving a file.

ctrl-]C                         ; Back to the PC
KERMIT-86> send command.com     ; As an example...
                                ; Again, a bell will ring when complete.
KERMIT-86> conn                 ; Back to the VAX
C-KERMIT> exit
BNLVAX<2> logout

ctrl-]C                         ; Back the the micro

KERMIT-86> exit

A>
```

That is all there is to it. Remember that kermit is a communications protocol and not a program. The kermit programs that exist are all different and have been written by different people at different times. None of them are exactly like any other, and not all offer the same features. (I have known one or two that offered no features including file transfer!) Also, none of them was written at BNL. Still every implementation should be able to communicate with every other versions.

SECTION vii

I. Communication on UNIX (BNL node)

This section provides some information on the UNIX mail. including examples and useful commands.

1. netexec will execute a command on the specified networked system, for example bnldag.
   Either netexec bnldag 'show users'
   or netexec bnldag show users will work.
   netexec bnldag 'show time' gives the time.

2. netwrite will let you write on a user's terminal on a networked system.

3. query lets you know whether the connection to bnldag is up, just type
   q sy

4. vlpr and tlpr appear to route print files, via uux, to the VAX cluster print queues for the versatec and talaris printers, respectively.

In order to send mail to someone on the cluster, use:

mail USERNAME@bnl-cl1.ARPA

For example

mail ZOHREH@bnl-cl1.ARPA

II. Following are the general information on the UNIX MAIL

mail - send and receive mail

SYNTAX

```
mail [ -v ] [ -i ] [ -n ] [ -s subject ] [ user ... ]
mail [ -v ] [ -i ] [ -n ] -f [ name ]
mail [ -v ] [ -i ] [ -n ] -u user
```

DESCRIPTION

The mail utility is an intelligent mail processing system, which has a command syntax reminiscent of ed with lines replaced by messages.

The -v flag prints the mail message. The details of delivery are displayed on the users terminal. The -i flag causes tty interrupt signals to be ignored. This is particu-

larly useful when using mail on noisy phone lines. The -n flag inhibits the reading of /usr/lib/Mail.rc.

SENDING MAIL: To send a message to one or more persons, evoke mail with arguments which are the names of people receiving your mail. Then type in your message, followed by an EOT (control-D) on a separate line. A subject may be specified on the command line by using the -s flag. (Only the first argument after the -s flag is used as a subject; be careful to quote subjects containing spaces.)

REPLYING MAIL In normal usage MAIL is given no arguments and checks your mail out of the mail directory, then prints out a one line header of each message there. The current message is initially the first message (numbered 1) and can be printed using the print command (which can be abbreviated p). You can move among the messages much as you move between lines in ed either with the commands + and - to move forward or backwards with simple numbers.

DISPOSING OF MAIL After examining a message you can delete (d) the message or reply (r) to it. Deletion causes the MAIL program to forget about the message. This is not irreversible; the message can be undeleted (u) by giving its number, or the MAIL session can be ended by giving the exit (x) command. If you end a mail session, you cannot retrieve deleted messages.

SPECIFYING MESSAGES Commands such as print and delete can be given a list of message numbers as arguments to apply to a number of messages at once. Thus the command, delete 1 2, deletes messages 1 and 2, while the command delete 1-5 deletes messages 1 through 5. The special name * addresses all messages, and $ addresses the last message; thus the command top which prints the first few lines of a message could be used in top * to print the first few lines of all messages.

REPLYING OR ORIGINATING MAIL You can use the reply command to respond to a message, sending it back to the person who sent it to you. Text you then type in, up to an end-of-file, defines the contents of the message. While you are composing a message, MAIL treats lines beginning with the character '~' (tilde) in a special way. For instance, typing ''~m'' alone on a line will place a copy of the current message into the response shifting it to the right by a tabstop. Other escapes will set up subject fields, add and delete recipients, and allow you to escape to an editor to revise the message or to a shell to run some commands. (These options are given in the summary below.)

ENDING A MAIL PROCESSING SESSION You can end a MAIL session with the quit (q) command. The messages go to your MBOX file unless they have been deleted. Unread messages go back to the mail directory. The -f option causes MAIL to read in the contents of your MBOX (or the specified file) for processing; when you quit, MAIL writes undeleted messages back to this file. The -u flag is a short way of doing mail -f /usr/spool/mail/user.

PERSONAL AND SYSTEMWIDE DISTRIBUTION LISTS It is also possible to create a personal distribution list so that you can send mail to to a group of people. Such lists can be defined by placing a line similar to this

alias users thomas dell steve kessler parsa@bnldag

in the file .mailrc in your home directory. Cohorts is the name of this distribution list. A list of current aliases can be displayed with the alias (a) command in mail System wide distribution lists can be created by editing /usr/lib/aliases, see ALIASES and SEND MAIL these are kept in a different syntax. In mail you send, personal aliases will be expanded in mail sent to others so that they will be able to reply to the recipients. System wide ALIASES are not expanded when the mail is sent, but any reply returned to the machine will have the system-wide alias expanded as all mail goes through SEND MAIL.

III. summary of the tilde escapes

Here is a summary of the tilde escapes, which are used when composing messages to perform special functions. Tilde escapes are only recognized at the beginning of lines.

~!command Execute the indicated shell command, then return to the message.

~? Prints a brief summary of tilde commands.

~: Executes the mail commands. (e.g., The command ~:10 prints out message number 10 while ~:- prints out the previous message.

~c name ... Add the given names to the list of carbon copy recipients.

~d Read the file ''dead.letter'' from your home directory into the message.

~e Invoke the text editor on the message collected so far. After the editing session is finished, you may continue appending text to the message.

~f messages Read the named messages into the message being sent. If no messages are specified, read in the current message.

~h Edit the message header fields by typing each one in turn and allowing the user to append text to the end or modify the field by using the current terminal erase and kill characters.

~m messages Read the named messages into the message being sent, shifted one tab space to the right. If no messages are specified, read the current mes-

sage.

~p Print out the message collected so far, prefaced by the message header fields.

~q Abort the message being sent, copying the message to ''dead.letter'' in your home directory if save is set.

~r filename Read the named file into the message.

~s string Cause the named string to become the current subject field.

~t name ... Add the given names to the direct recipient list.

~v Invoke an alternate editor (defined by the VISUAL option) on the message collected so far. Usually, the alternate editor will be a screen editor. After you quit the editor, you may resume appending text to the end of your message.

~w filename Write the message onto the named file.

~|command Pipe the message through the command as a filter. If the command gives no output or terminates abnormally, retain the original text of the message. The command fmt(1) is often used as command to rejustify the message.

~~string Insert the string of text in the message prefaced by a single ~. If you have changed the escape character, then you should double that character in order to send it.

Options are controlled via the set and unset commands. Options may be either binary, in which case you should see whether or not they are set, or string, in which case the actual value is of interest. The binary options include the following:

FOR ADDITIONAL INFORMATION SEE 'The Mail Reference Manual', OR

type man mail

WHICH GIVES YOU THE UNIX MANUAL PAGES ON MAIL FACILITIES.

IV. Connection to IBM3090 (vm/cms),

With tn3270 - a full-screen remote login to IBM VM/CMS

SYNOPSIS

tn3270 [-d] [-n filename] [-t commandname] [sysname [port]]

DESCRIPTION

Tn3270 permits a full-screen, full-duplex connection from a UNIX machine to an IBM (or compatible) machine. Tn3270 gives the appearance of being logged in to the remote machine from an IBM 3270 terminal. Of course, you must have an account on the machine to which you connect in order to log in. Tn3270 looks to the user in many respects like the Yale ASCII Terminal Communication System II. Tn3270 is actually a modification of the Arpanet TELNET user interface (see telnet(1)) which will, in certain circumstances, interpret and generate raw 3270 control streams.

The flags to tn3270 are as follows:

-d Turn on socket-level tracing (for super-user only)

-n filename
Specify a file to receive network trace data output (from commands "toggle netdata" and "toggle options", see telnet(1c)); the default is for output to be directed to the standard error file.

-t commandname
Specify a UNIX command to process IBM 4994 style transparent mode data received from the remote IBM machine.

sysname
The name of the remote system. If the remote name is NOT specified, the user will be prompted for a command (see below).

port The port to connect to on the remote system. Normally, tn3270 attempts to connect to the standard TELNET port (port 23) on the remote machine.

For additional information use the following command:

man tn3270

this gives the full information on how to login remotely on IBM3090 from UNIX VAX.

COMPUTER ACCESS LINES *

GANDALF PACX (Private Automatic Computer eXchange)

Respond to Enter Class with either numeric or mnemonic code as follows:

Numeric Code	*Mnemonic Code*	*S E R V I C E*
–	ST	Machine Status and PACX Directory
111		AMD VAX Cluster auto baud
101	MFC	AMD Cyber 830 300-1200 baud
30	MFC48	AMD Cyber 830 4800 baud
27	MFC96	AMD Cyber 830 9600 baud
	MFE	MFE Lab. CRAY
75	BNL44	UNIX PDP 11/44
76	UNIX	UNIX VAX
15		Meteorology VAX 11/750 autobaud
55		BNLDAG VAX 11/780 autobaud
52		BNLDAG Versatec Printer, Bldg. 510
46		BNLDAG Laser Printer, Bldg. 902
57		BNLHEP VAX 8600 autobaud
51		BNLHP1 VAX 11/780 autobaud
53		BNLHP1 Laser Printer EX802
47		BNL734 VAX 11/780 autobaud
61		BNLMPS VAX 11/750 autobaud
63		BNL787 VAX 11/750 autobaud
43		KL10 AGS Online System 2400 baud
44		KA10 AGS Development System 2400 baud
11		Chemistry VAX
04	IPAP	MIS HP68 auto baud
51		HEP VAX auto baud
47		734 VAX auto baud
03	MIS3	MIS HP 2400 baud
17		CBA 11/60 9600 baud
35		NSLS Data General 9600 baud
45		911 PDP 11/45 9600 baud
77	LOOP	Loop Back
70		PC or terminal interconnect(1)
	DIAL	Autodial modems (516, 718, 212, & FTS)
	CDIAL	Autodial modems Commercial net (Dial 9 + number)

(1) - *Terminal interconnect:* One terminal, whose terminal ID is *nnnn*, selects PACX class 70. Another terminal can then respond to the Enter Class prompt with T-*nnnn*, and the two terminals are connected. To determine the ID number of ones terminal, select class ID.

* see Computing Newsletter for updates

DIAL-UP LINES (On-site only):

Machine	*Type/Duplex*	*Baud*	*Ext.*
PACX		300 – 1200[1]	7021[2]
		2400	2124

[1] – VADIC 34XX, 212 and 103
[2] – Access to MFC and VAX Cluster – Use Class Codes as above.

Off-site access numbers supplied to users upon registration.

AUTODIAL MODEMS ON PACX

There are two autodial modem pools available as a PACX resource. These modems may be accessed by a PACX user and requested to dial a remote computer. The two classes are:

DIAL: for calls to area codes 516, 718, 212 and FTS.

CDIAL: Commercial DDD network only (no FTS access). You must dial 9 + number.

Class DIAL should meet the neds of nearly all users. When FTS service to an area is so poor that 1200 baud data transmission is impossible, or if calls to Canada are required, use Class CDIAL. All calls are recorded both by PACX and GTE.

The modems in the Class DIAL pool are new, permit tone dialing, and give more feedback regarding call progression. The CDIAL modems are pulse dial only and have limited call progression features.

The modems protocols, while quite similar, have some subtle differences:

1. If you make a mistake in the number and the modem is already dialing, the call may be cancelled with a $< CR >$ on the DIAL modems.
2. The DIAL modems send the user call progression information (RINGING, BUSY and ANSWER TONE) in addition to DIALING and FAILED CALL.
3. The number is echoed back as it is entered on the DIAL modems while the CDIAL modems display the entire number after the $< CR >$.
4. If the CDIAL modems do not respond do dial commands, they must be forced to IDLE status by the command I$< CR >$, and then restarted with the command E$< CR >$.
5. CDIAL modems usually need the addition of KKK after the number to prevent timeout on long distance calls.
6. DIAL modems sent a beep when the remote modem is connected.
7. Although both modem types allow up to 9 retries of the same number, the DIAL modems prompt the user with: "Number of retries?"
8. CDIAL modems can call VADIC only 1200 baud modems; DIAL modems cannot. As nearly all auto-answer modems are 212-compatible, this is of little consequence.

The Class DIAL and CDIAL sequences are shown on the next page:

Class **DIAL** Sequence

PACX or MODEM	*USER*
Enter Class:	DIAL < CR >
Class DIAL Start	Ctrl E < CR >
Hello, I'm ready	D < CR >
Number?	number < CR >
DIALING...	
RINGING	
ON LINE (Audible beep)	
... or ...	
BUSY or FAILED CALL	R < CR >
Number of retries?	n < CR > ($n \leq 9$)
... loop back to DIALING ...	

Class **CDIAL** SEQUENCE

PACX or MODEM	*USER*
Enter Class:	CDIAL < CR >
Class CDIAL Start	Ctrl E < CR >
Hello, I'm ready	D < CR > (D not echoed back)
Number?	number < CR >
[number is displayed]	< CR >
DIALING	
ON LINE	
... or ...	
FAILED CALL	R < CR > (R not echoed back)
	n < CR > ($n \leq 9$) (number of retries)
... loop back to number is displayed...	

The phone number is 4 to 14 digits long with dashes omitted; for example:

Enter 92125551212 for 9-212-555-1212
Enter 86175551212 for 8-617-555-1212 (FTS line)

If an error is made, a recovery can be attempted by entering: Ctrl E<CR>.