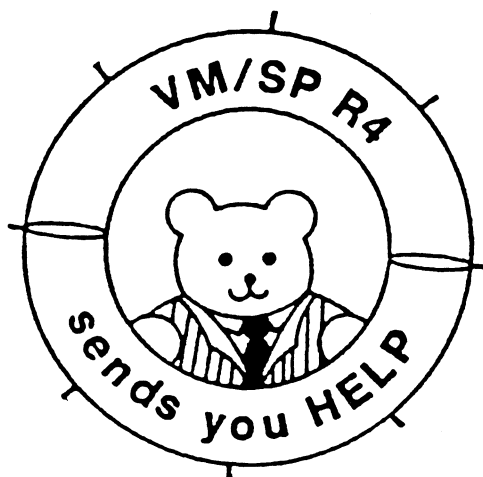


## VM-CMS TUTORIAL



Second Edition, October 1987

This writeup is also available in French: see DD/US/14.

CERN LIBRARIES, GENEVA



CM-P00060791

"An elephant is a mouse with an IBM operating system."

Anon

## Preface

These notes contain the material accompanying the VM-CMS course part 1, given for the first time in April 1986 as part of CERN's Technical Training programme. It is essentially a self contained tutorial, with both text and exercises. Some knowledge of computers would be helpful before starting this course. This second edition contains corrections which reflect changes to the Computer Centre; the description of the MAIL system is also changed.

The material in Chapters 1-5 contains the minimal background for anyone wishing to use VM-CMS. Chapters 6-7 contain advanced topics and the remaining chapters may be read in any order.

To those starting to learn VM I should like to say the following. In many respects learning VM is like learning how to drive a car – one has to acquire many skills at the same time to be able to safely follow one's course. In this sense this tutorial will probably not be entirely clear after the first reading as many forward references had to be made to concepts and features that are explained later in the course. However, during a second reading most things should be much clearer. Like learning how to drive, learning VM does not come without practice, and the exercises in the text should be carried out as you go along. Finally, remember that anyone can learn how to drive and that the same certainly applies to VM although it may seem quite complicated in the beginning.

Please do not consider this the equivalent of a manual; although the syntax of many commands is correctly given, you should always check in the appropriate IBM documentation or the on-line Help (FIND) for extra options and a more complete description. The IBM CMS Primer [1] is strongly recommended as alternative reading material. The emphasis here is on concepts rather than individual commands. Do not hesitate to contact me if either the course material or the tutorial is not clear.

E. van Herwijnen, Tel. 5010 or ERIC at CERNVM.

---

## Contents

<b>Preface</b> .....	<b>ii</b>
<b>Chapter 1: What is VM?</b> .....	<b>1</b>
1.1 VM/SP .....	1
1.2 The Concept of a Virtual Machine .....	2
1.3 The Composition of a Virtual Machine .....	3
1.4 CMS (Conversational Monitor System) .....	5
<b>Chapter 2: Using your virtual machine</b> .....	<b>6</b>
2.1 Starting your Virtual Machine .....	6
2.1.1 Setting your Logon Password .....	11
2.2 Special Features of the Keyboard .....	11
2.3 Screen Layout .....	14
2.4 Status Messages .....	14
2.5 Exercises for Chapters 1 and 2 .....	16
<b>Chapter 3: The Editor (Xedit)</b> .....	<b>17</b>
3.1 What is a File? .....	17
3.2 How to Create a CMS File Using the Editor (Xedit) .....	18
3.2.1 Entering the Editor or Starting an Xedit Session .....	18
3.2.2 Quitting the Editor or Ending your Xedit Session .....	18
3.2.3 The Xedit Screen Layout .....	19
3.3 Adding Data to a File .....	21
3.3.1 The Input Command. ....	21
3.3.2 The Add Command .....	22
3.3.3 POWerinput .....	23
3.4 Deleting and Recovering Lines .....	23
3.5 Exercises for Chapter 3 .....	24
<b>Chapter 4: More Xedit</b> .....	<b>25</b>
4.1 Moving around in a file .....	25
4.2 Prefix commands .....	27
4.3 Some important Xedit Concepts .....	29
4.3.1 The Current Line and Current Column .....	29
4.3.2 Targets .....	30
4.3.3 The Zone .....	35
4.3.4 Finding the Current Settings .....	35
4.4 Changing Data in a File .....	36
4.5 Some Examples of Selective Changes .....	37
4.6 Editing Multiple Files .....	37
4.7 Some useful features in Xedit, function keys .....	38

4.8	Creating your own Xedit Environment .....	40
4.9	The Profile Xedit .....	41
4.10	Exercises for Chapter 4 .....	41
 <b>Chapter 5: Properties of your Virtual Machine .....</b>		<b>45</b>
5.1	The State of your Virtual Machine .....	45
5.2	Minidisks .....	47
5.3	Manipulating Mini-disks .....	49
5.3.1	The Gime Command .....	50
5.3.2	Linking and Accessing Minidisks .....	50
5.3.3	Dangers of linking to other people's disks .....	51
5.4	Swapping mode letters .....	52
5.5	Manipulating files .....	52
5.5.1	Displaying a list of files .....	52
5.5.2	The filemode and CMS search order .....	53
5.5.3	Useful commands when inside Filelist .....	55
5.5.4	Use of '=' sign inside commands .....	55
5.6	Spool Files .....	56
5.6.1	Spooling of virtual devices .....	59
5.7	Exercises for Chapter 5 .....	60
 <b>Chapter 6: HELP .....</b>		<b>62</b>
6.1	FIND .....	63
6.2	HELP MENUS .....	64
6.3	HELP TASKS .....	65
6.4	Error Messages .....	65
6.5	Exercises for Chapter 6 .....	66
 <b>Chapter 7: Adapting CMS to Your Taste .....</b>		<b>68</b>
7.1	Synonyms for CMS commands .....	68
7.2	Programmable Function Keys (PF-keys) .....	69
7.3	Some general CMS settings .....	70
7.4	Creating and Using EXEC Procedures (Files) .....	72
7.4.1	The PROFILE EXEC .....	73
7.5	Setting Defaults .....	73
7.6	Halting output to the terminal .....	74
7.7	Exercises for Chapter 7 .....	74
 <b>Chapter 8: Communicating with other Computer Users and Transferring Files .....</b>		<b>76</b>
8.1	The Names File .....	76
8.1.1	Interrogating the Names File using Namefind and Whois .....	78
8.1.2	The Tell Command .....	79
8.1.3	Notes .....	80
8.1.4	The Sendfile Command .....	81

---

8.1.5 Mail .....	82
8.2 EMDIR .....	83
8.3 Automatic forwarding of MAIL .....	84
8.4 Transferring Files from Wylbur to VM and vice versa .....	85
8.4.1 Use .....	85
8.4.2 Getfile .....	86
8.4.3 Import .....	86
8.4.4 Export .....	87
8.5 Transferring Files from the VAX to VM and vice versa .....	87
8.6 Submitting Jobs on Networked Machines .....	88
8.7 Exercises for Chapter 8 .....	88
<b>Chapter 9: Some Local CERN Facilities .....</b>	<b>89</b>
9.1 How to print files .....	89
9.2 Text processing facilities available from VM .....	92
9.3 Recovering Backup Files and Archiving .....	93
9.4 Use of magnetic tapes .....	95
9.4.1 Dumptape .....	95
9.4.2 Tapedisk .....	95
9.4.3 Disktape .....	96
9.5 The Space Machine .....	97
9.6 Exercises for Chapter 9 .....	99
<b>Chapter 10: Creating and running programs .....</b>	<b>100</b>
10.1 Invoking the Fortran compiler using the VFORT command .....	100
10.2 Running a compiled Program .....	102
10.3 Defining Input/Output Files for Programs .....	103
10.4 The Batch Machine .....	103
10.5 Exercises for Chapter 10 .....	104
<b>Appendix A: How to use LTERM .....</b>	<b>107</b>
<b>Appendix B: Minidisk passwords and access modes .....</b>	<b>108</b>
B.1 Minidisk link modes .....	108
B.2 Minidisk passwords .....	109
B.3 Setting default links and passwords using DIRM .....	109
<b>Appendix C: The Wylbur Bridge .....</b>	<b>112</b>
<b>Appendix D: ACE, the Automatic Calculating Engine .....</b>	<b>113</b>

<b>Appendix E:</b>	<b>Terminal Keyboards</b> .....	114
E.1	Terminal Types .....	114
E.2	Keyboard Layouts .....	114
E.2.1	D2 terminals, right hand end of the keyboard .....	115
E.2.2	PA terminals, right hand end of the keyboard .....	116
E.2.3	P7,PG and D1 terminals, right hand end of the keyboard .....	116
E.3	Locally added support for line mode terminals (TTY) .....	117
E.4	Use of Pericom's as VM Terminals .....	117
E.4.1	Pericom 6803 – standard 'old' Pericom (PA) .....	118
E.4.2	Pericom "ALPHA" .....	118
E.4.3	Pericom "GRAPH" .....	118
E.4.4	Pericom "GRAPH PAC" .....	119
E.5	Use of the DMT2200 as a VM terminal .....	119
E.6	Use of the Tandberg as a VM terminal .....	120
E.7	Use of the Macintosh Plus as a VM terminal .....	120
E.7.1	VersaTerm Settings .....	120
E.7.2	Keyboard layout .....	121
E.7.3	File Transfer .....	121
<b>Appendix F:</b>	<b>FILE TRANSFER USING WISCNET</b> .....	122
<b>Appendix G:</b>	<b>References</b> .....	124
<b>Appendix H:</b>	<b>Index</b> .....	125

## Figures

1.	Example of a Computer Running VM with 4 Virtual Machines .....	2
2.	A Sample Configuration with 2 (Virtual) Machines .....	3
3.	Definition of Minidisks .....	4
4.	The CERNVM Logo .....	8
5.	The NEWS system .....	9
6.	A news item .....	9
7.	The DMT2200 Keyboard (terminal type: D2) .....	12
8.	Example of a CMS Screen Layout after Logging in .....	14
9.	A rough Map of Environments in your Virtual Machine .....	16
10.	The Relationship between Virtual Storage and Disk Files .....	19

---

11.	Layout of the Xedit Screen .....	20
12.	Illustration of the Input Command with an Argument .....	22
13.	Xedit Screen for TUTORIAL SCRIPT in Input mode .....	23
14.	Example of the Use of POWERinput .....	24
15.	How to Move the Viewing Window inside Xedit .....	26
16.	How to move the Current Line with Locate .....	30
17.	How CLocate moves the Current Column .....	31
18.	How to set the Current Line/Column with a Numeric Target .....	32
19.	Locate using a Complicated String Target .....	33
20.	Result of the Command ALL /logo/ .....	34
21.	Result of ALL .....	35
22.	Result of the Command Query All .....	45
23.	Standard Virtual Machine Configuration .....	46
24.	Disk Configuration in a Virtual Machine .....	48
25.	The DISKLIST command .....	51
26.	FILEList display .....	53
27.	The Query SEARCH command .....	54
28.	The Q RDR ALL Command to determine the Spoolid .....	56
29.	The RDRLIST display .....	57
30.	The PRTLIS command .....	58
31.	Issuing the command CP SPOOL CONSOLE CLOSE STOP .....	59
32.	A Console Log file .....	60
33.	The PUNCH HELP File .....	62
34.	The result of FIND RDR .....	63
35.	The list of categories for HELP .....	64
36.	The list of HELP-items in component CMS .....	65
37.	HELP TASKS in the Xedit and CMS environments .....	66

---

38.	Dependence of function keys on the environment .....	71
39.	The NAMES Command .....	77
40.	The NOTE command .....	80
41.	The SendFile Command .....	81
42.	Sending a MAIL .....	83
43.	The MAILBOOK command .....	84
44.	The Use Command .....	85
45.	The Xprint command .....	90
46.	A List of Printers in Building 513 from Xprint .....	90
47.	The XSCRIPT Command .....	92
48.	The SGML command .....	93
49.	The Result of VMBLIST PROFILE EXEC ( RESTORE .....	94
50.	Restoring PROFILE EXEC with VMBLIST .....	95
51.	The Tapedisk command .....	96
52.	The Disktape command .....	97
53.	A simple Fortran Program .....	100
54.	Response from the VFORT command .....	101
55.	Loading and Executing a Program .....	102
56.	Setting a default link to a minidisk .....	109
57.	The CP Directory Entry with Default Links to Minidisks .....	110
58.	Setting a Read or Write Password on a Minidisk .....	111
59.	ACE, the Automatic Calculating Engine .....	113
60.	The D2 terminal right hand end of the keyboard .....	116
61.	The PA terminal, right hand end of the keyboard .....	116
62.	The P7,PG and D1 terminals, right hand end of the keyboard .....	117
63.	Connecting to the remote system. ....	122
64.	Supplying login information. ....	123



## Tables

1.	Set up of the DMT2200 terminal .....	6
----	--------------------------------------	---

## 1. What is VM?

### Summary of Chapter 1.

At the end of this chapter, you should;

- Have an idea what VM is.
- Know some features of VM.
- Understand the concept of a virtual machine.

### 1.1 VM/SP

To manage the hardware of a computer (i.e. the memory, the disks, the tapes, the processing unit etc.), various tasks have to be carried out such as mounting tapes, reading disks, unloading the printer, starting/stopping jobs and if it is an old fashioned computer, reading/punching cards.

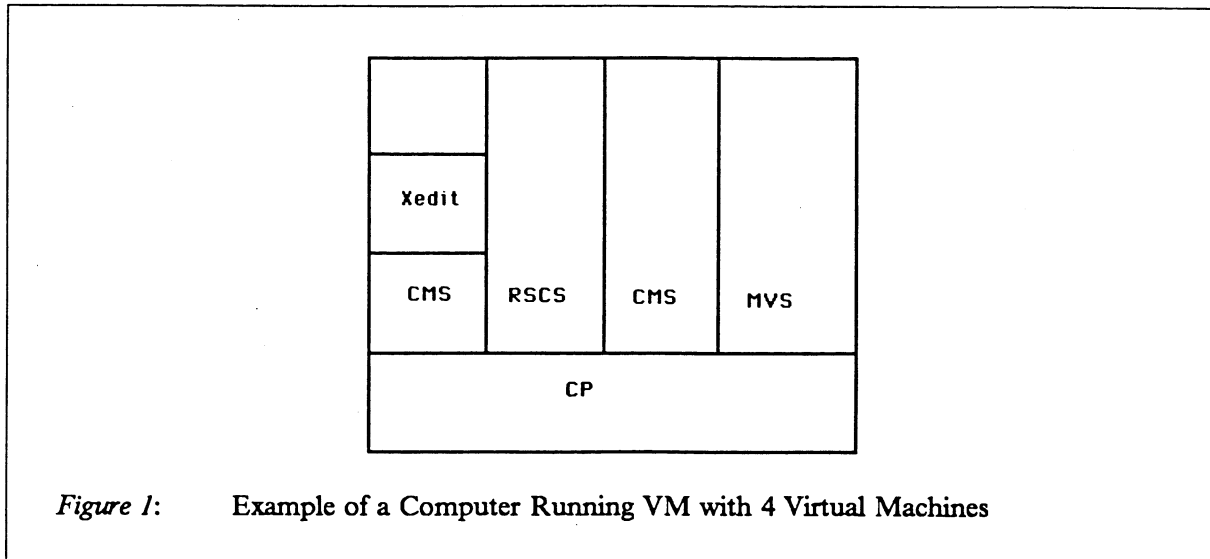
These tasks enable the resources to be shared efficiently between many users and programs so that they do not have to worry about the details of how the system works. In old computers, these tasks were performed manually by the operators. In more modern computers a continuously running program called the **operating system** takes care of resource management.

VM or VM/SP (Virtual Machine/System Product) is an operating system for medium to large computer systems. At CERN several computers run the VM system (all these VM systems are slightly different), among which is the IBM 3090 housed in the computer centre. This course is intended for potential users of the VM system running on the 3090 in the computer centre. Other operating systems you may have heard of are MVS (with Wylbur), NOS-BE (with Intercom), VMS, DOS, UNIX and CMS. Most, but not all of these, are aimed at sharing the resources of the computer as efficiently as possible between many users. VM does this in a unique way by providing an environment which simulates a real computer system to each user or program. These simulated environments are called **Virtual Machines**. Compare this approach to that of MVS (**Multiple Virtual Storage**) which makes it appear to the user or program that they are alone in the machine. In MVS in principle every job has access to the real devices whilst in VM one is restricted to virtual devices. Components of VM which will be considered here are:

- *CP (Control Program)*: a program which manages the real system and creates the virtual machines.
- *CMS (Conversational Monitor System)*: the operating system running in each individual virtual machine. One could equally well use UNIX, or MVS, and indeed such configurations exist.
- *Xedit (the editor)*: a program that allows you to modify and create other programs (files). This runs inside a virtual machine running CMS.
- *REXX (the interpreter)*: a command language that allows the execution of a set of CP, CMS and Xedit commands in an easy way. Equivalent to Wylbur Exec language, CCL (Cyber Control Language), C-shell scripts (UNIX) and DCL (DEC Control Language).

- *RSCS (Remote Spooling and Communication Subsystem)*: a program that enables different virtual and physical machines to communicate.

The relationship between these components is shown in Fig. 1.



## 1.2 The Concept of a Virtual Machine

A virtual machine is an independent copy of a real computer system, i.e. it has virtual disks (so-called minidisks), virtual storage, a virtual printer, punch and card reader etc. It is called virtual because there are many such systems in a given real computer, all sharing the real facilities simultaneously.

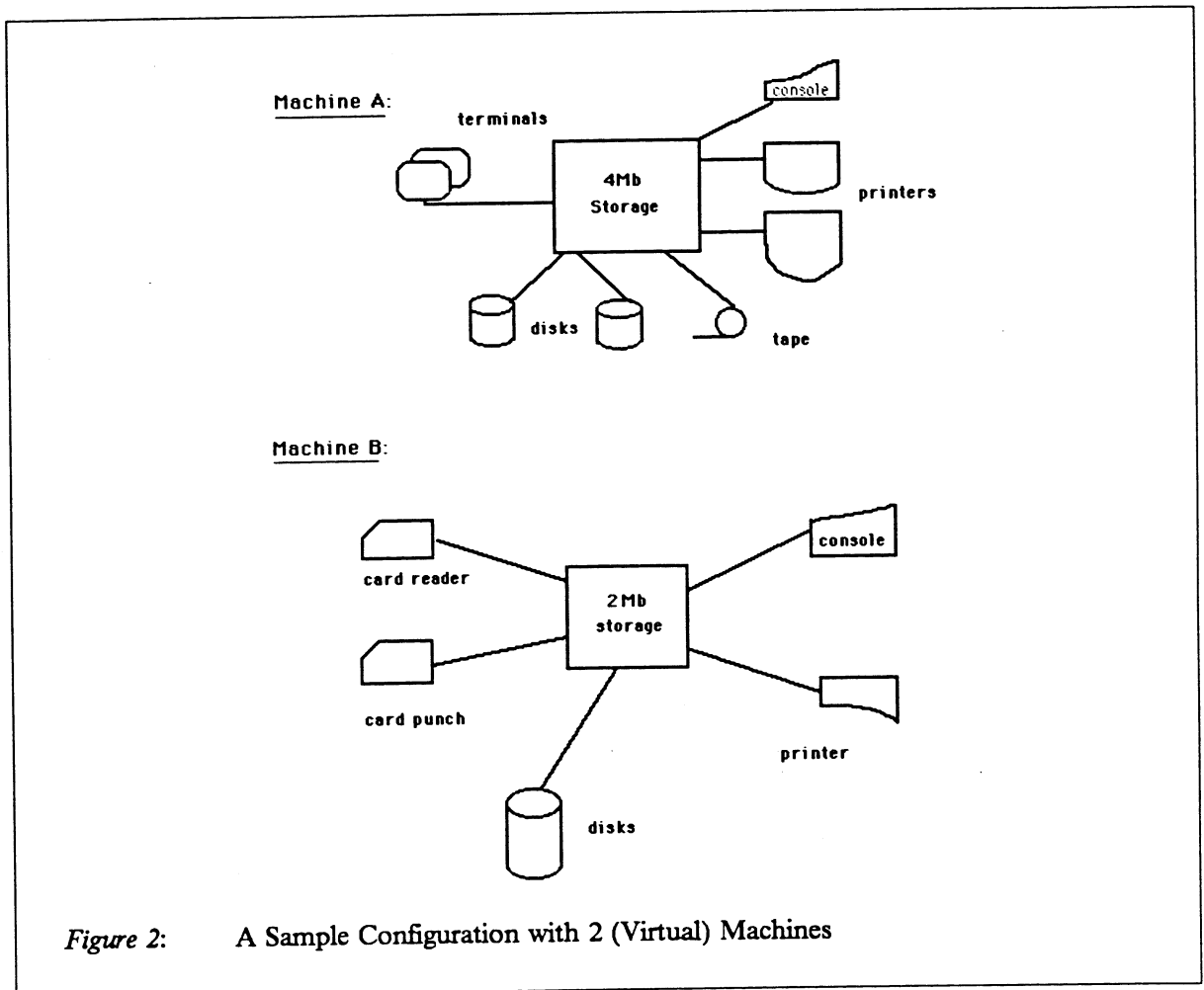
The "real" organization (sharing of disks, processor time i.e. multi-tasking, storage and I/O devices) is done by the **Control Program, CP**. CP assures the independence of the machines under its control by not allowing any direct communication between them. The details about each Virtual Machine are kept in the **directory**. The concept of a virtual machine is most easily understood by the following properties of its user-environment;

- It looks like a PC, or a personal workstation, i.e. it is interactive.
- It is a self contained, secure system.
- One cannot directly access the memory of another virtual machine or influence what goes on there.
- Many tasks can be carried out simultaneously.
- There is an easy transfer of data between virtual machines and other connected systems at CERN, such as Wylbur, Nord, VAX etc.
- There is great flexibility in defining the properties of each machine.

To summarize, the interface created by VM can be described as a "poor man's personal workstation". The software may not always be as user-friendly as that of a personal workstation, but the computing power of a 3090 allows many other things which would be impossible in a real personal workstation.

### 1.3 The Composition of a Virtual Machine

Roughly speaking, to each real IBM device (printer, terminal, tape unit, disk, card reader etc.) corresponds a virtual device, and as for real computers, each virtual machine can be configured with different types of virtual devices. In many cases, the user is free to define these. Figure 2 contains an example of two totally different virtual machines, that could in fact correspond to two different real computer systems.



The following is a list of the most important virtual devices which you will encounter;

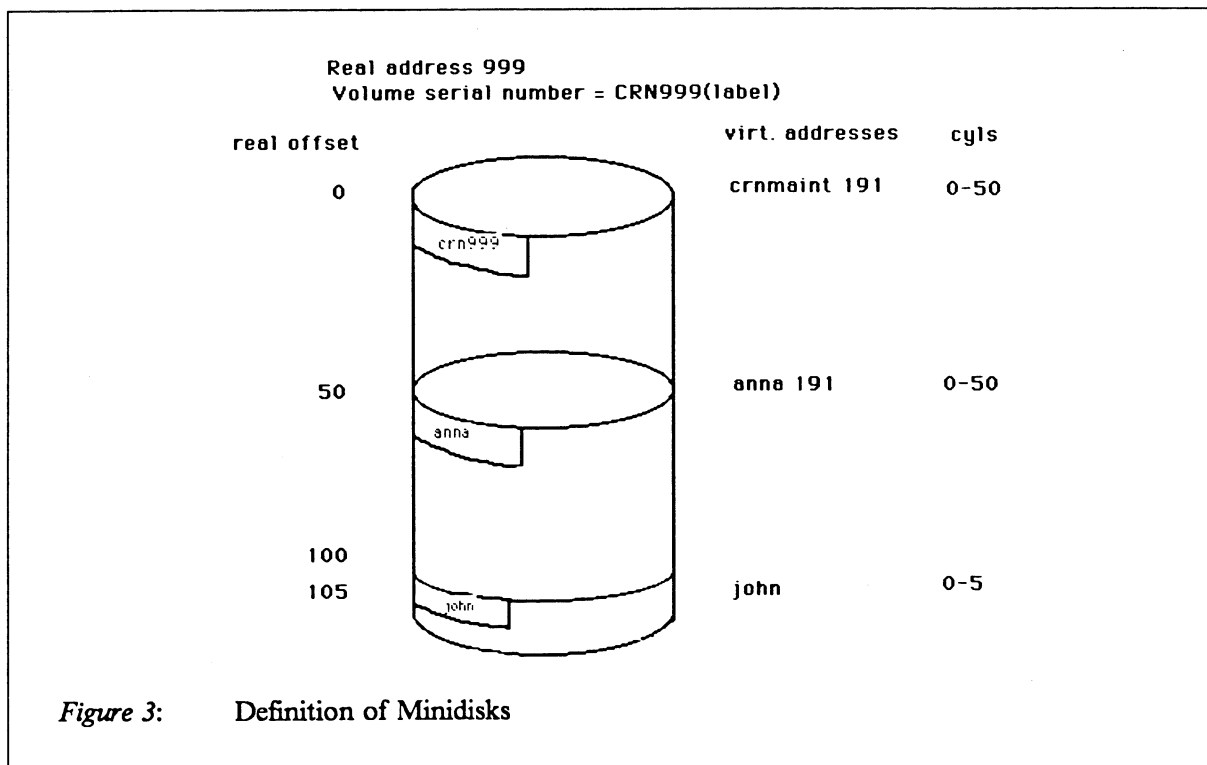
#### Terminals

Your terminal will be seen by your virtual machine as the operator's **console**. Normally each real terminal will correspond to one Virtual console. However, at CERN a facility called **LTERM** has been developed, allowing one physical terminal to be defined as virtual console to several virtual machines, using a

function key to switch between the various sessions. An example of how to use LTERM will be given in Appendix A on page 107. Machines can run without a terminal, in so-called **disconnected** or **batch** mode. This is the only difference between a "batch" machine and an "interactive" machine. Disconnected machines are sometimes also called **service machines**.

**Tapes** Tapes are not shared and can only be used by one machine at a time. At CERN the use of tapes is reserved for a special machine which runs in disconnected mode. This machine is called the **batch worker**. The batch machine can only be accessed via a special command, which will be discussed later.

**Punch/Reader/Printer** These are also called Unit Record Devices. CP shares these devices through the spool (Simultaneous Peripheral Operation On Line). Use of any of these devices results in a new **spool-file** on disk, which can be printed<sup>1</sup>, read, discarded or sent across the network. The spool is the main way of communication between virtual machines. For example sending someone a file goes via the spool.



**Disks** are shared among users by CP via the concept of a **mini-disk**. The size of a mini-disk can be anything from a few blocks to an entire disk volume. Any virtual machine can be configured with many minidisks at a given time. These

<sup>1</sup> At the moment there are no (public) real printers connected to VM at CERN, with the exception of the APA6670 and IBM 3812 laserprinters which are only accessible via SCRIPT. They are to be used for text processing only. Thus all line-printing must be performed by transferring files via the spool and RSCS to MVS.

need not all be permanent; those that are permanent have to be defined as such in the users directory. Each disk is labelled with a letter from A to Z, and up to 26 disks may be identified to a machine at any given time. By default, users get an "A"-disk of 2 cylinders at CERN. In Figure 3 on page 4 we can see how a real disk can be split into several virtual minidisks. Note how the address of a minidisk is independent of the real address.

#### **1.4 CMS (Conversational Monitor System)**

CMS is an operating system specifically for use in a virtual machine. It is **single-user environment**: a simpler situation is hard to imagine. CMS has a very powerful filing system which is good and efficient at handling many separate entities. The filing system does not have a tree-like structure such as Wylbur, UNIX or VMS and despite the fact that this feels somewhat clumsy at first, the flexible nature of VM-CMS allows it to be used very efficiently. CMS is interactive and contains a good full screen text editor, Xedit. It allows flexible management of the content of your virtual system via simple and powerful commands.

## 2. Using your virtual machine

### Summary of Chapter 2.

In this chapter, you will learn;

- How to use your virtual machine (start it and stop it).
- The meaning of some special keys on the keyboard (e.g. character insert/delete, clear screen, erase end of line, cursor movement keys).
- How to set a password for logging on.
- The layout of your screen under CMS and how to interpret it.

### 2.1 Starting your Virtual Machine

Before you start your virtual machine ("logging on") it may be useful to know the status of the system. This can be done from Wylbur via HELP TVSCREEN<sup>2</sup>. The operators (5011) or the UCO (4952) can also provide information. Normally, however, the VM system will be running.

If the terminal you are using is a DMT2200, you should first check its setup (see Table 1). To change it press the keys marked with arrows until the required field is highlighted and press the key marked ENTER to obtain the desired values.

*Table 1: Set up of the DMT2200 terminal*

SETUP A	
MAIN PORT:	Transmit 4800
	Receive Transmit
	Character 7S
	Stop bit 1
	local echo off
	interface RS232C
	duplex FDX A
KEYBOARD:	break enable
GENERAL:	emulation VT200(7)

Starting your virtual machine is a routine process that may seem laborious at first but soon will become automatic. The process consists of the following steps (carry them out as you are reading along):

---

<sup>2</sup> On VM, try the command TVSCREEN.

1. If you are using a PC (like the Olivetti M24 used in the Technical Training lab.), insert the floppy disk in the disk-drive, switch the drive on (the switch is at the back on the right handside of the drive) and wait until the content have been loaded. The floppy disk contains a program that "emulates" an IBM terminal for the computer on which we shall be working. Type in **Connect to CMS**.
2. If you are not using a PC, but an ordinary terminal such as a Pericom (PA) or DMT2200 (D2) you start at this point in the sequence. Choose 00 on the blue "Gandalf" index box and turn on the black switch until the "Ready" light comes on. Hit RETURN.
3. When the

```
enter class
```

prompt appears on the screen you type 72 for line mode access, class 101, 103 or 125 for full screen terminal access (from some buildings it may be necessary to type class 1 or 2 to obtain a cross connection). First, type 125 (for class 101, 103 and 125 the procedure is identical; we shall discuss class 72 later).

4. The system responds with
5. The system responds with the prompt:

```
class start
```

You hit RETURN.

```
ENTER TERMINAL TYPE:
```

- If you are using a PC (such as the M24), you reply **PC**<sup>3</sup> and hit RETURN (for a description of this key, see section 2.2 on page 11 point 4). For a Pericom you would reply: PA, for a DMT2200: D2.
6. If all is well, the VM Logo (see Figure 4 on page 8) now appears on the screen. If your terminal is hard wired into the machine, you follow the procedure from this point onwards and ignore the points above.
  7. You may choose for how many virtual machines you wish your terminal to act as console. If you wish to use more than one virtual machine, consult Appendix A on page 107. Otherwise clear the logo by hitting RETURN .
  8. To login, use the command

```
Login userid [password]
```

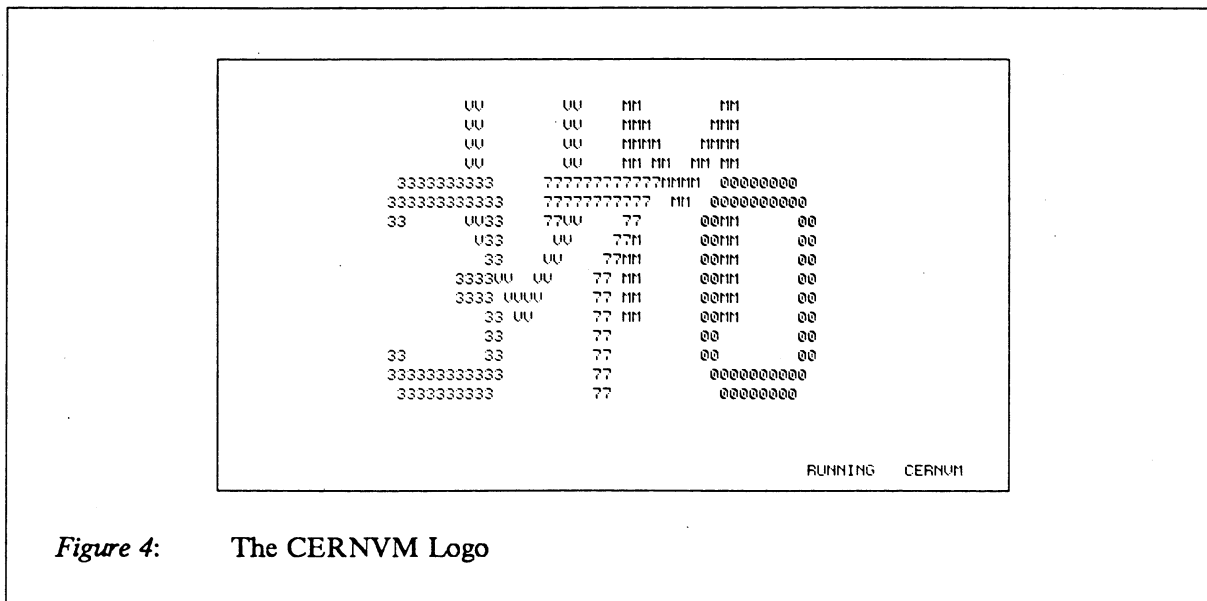
Note that we follow the IBM convention of capitalizing that part of the command which is its minimal abbreviation. Thus L is a valid abbreviation of Logon. If the password is not given, the user will be prompted for it. You are now logged on to your virtual machine in full screen mode.

9. The system replies with the following messages:

---

<sup>3</sup> It you are using an IBM PC with the ambassador emulator, type AMBASS.





*Figure 4:* The CERNVM Logo

```
FILES: xxx RDR, xxx PRT, xxx PUN
LOGON AT hh:mm:ss GVA xxxDAY mm/dd/yy
CERN CMS R4.x V.3
possibly other "LOG" messages
```

and in the bottom right hand corner you will have the message "VM READ". You respond by hitting RETURN (after having read the "LOG" messages if any were present; if you missed them you can read them later using the command "Q LOGMSG" ).

10. The system may reply with a panel which displays the news (see Figure 5 on page 9). To look at any news items, place the cursor next to it and hit RETURN. All news items should be read carefully as they contain important information (see Figure 6 on page 9). To quit hit function key 3 (PF3) twice.
11. The system replies with:

```
G(197) R/O
R; T=0.09/0.13 hh:mm:ss
```

In following this procedure, you will have IPL-ed (IPL= Initial Program Load) CMS in your machine, i.e. it has loaded its operating system and is ready for use. This is indicated by the "Ready message":

```
R; T=xx/yy hh:mm:ss
```

where xx/yy is the time your machine took for processing, and hh:mm:ss is the time. If you do not get the ready message, you machine is still running some program, or it is waiting for input (it says VM READ in the bottom right hand corner, in which case you hit RETURN).

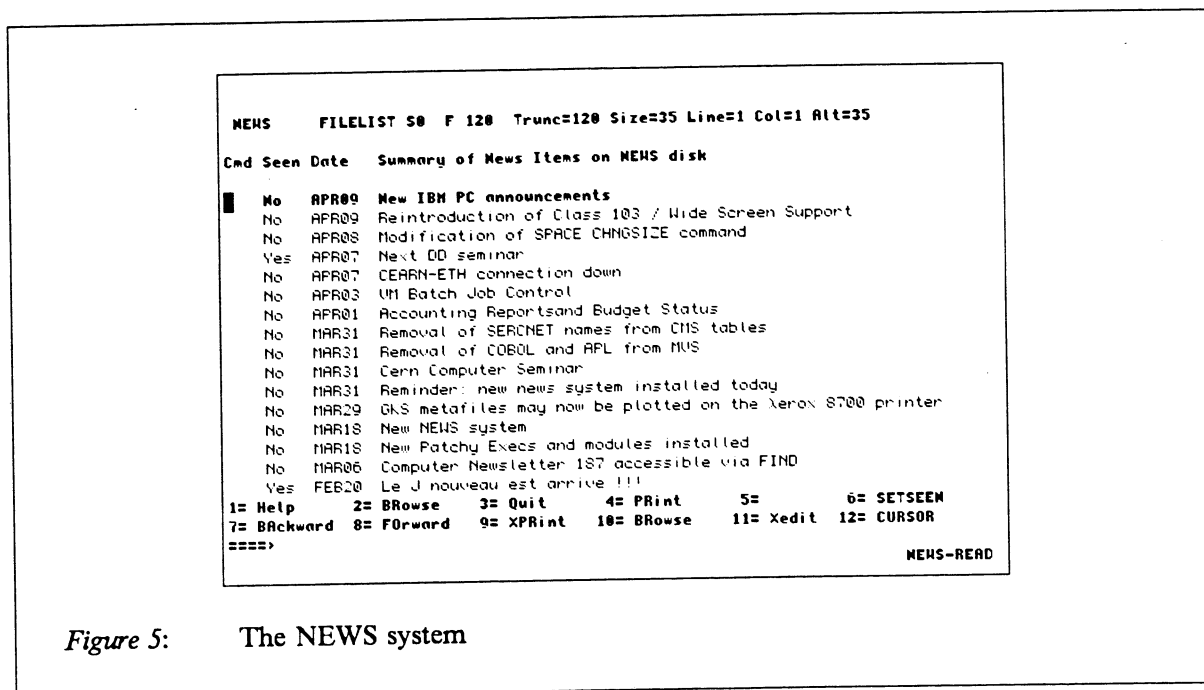


Figure 5: The NEWS system

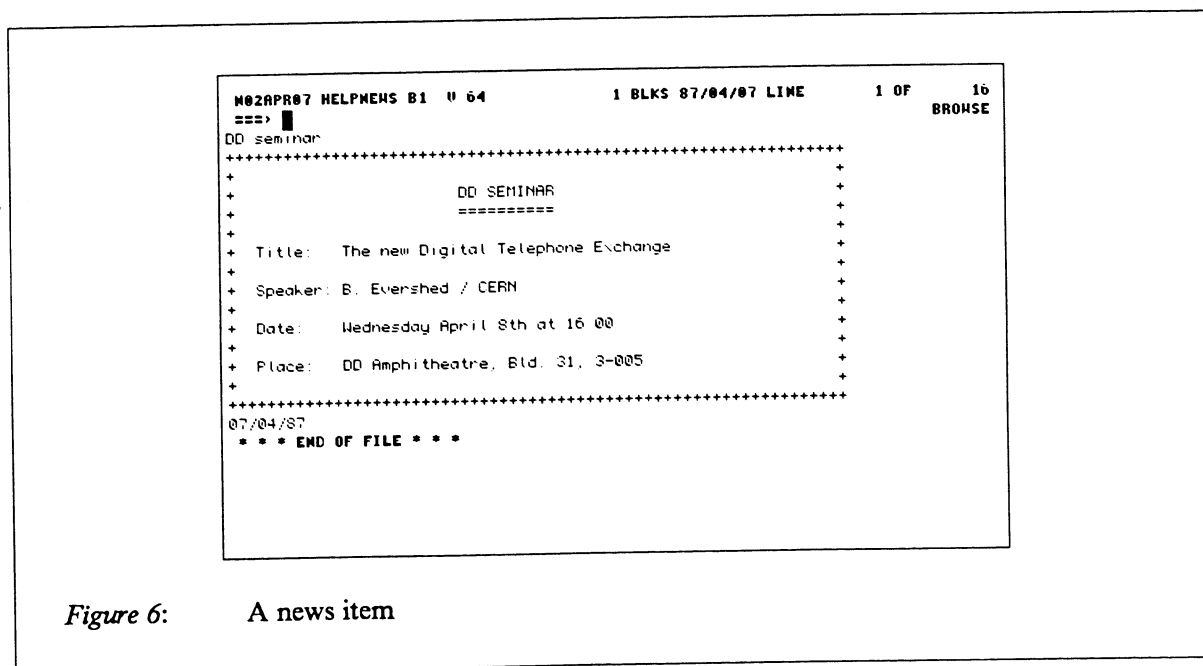


Figure 6: A news item

12. If you chose class 72, the procedure would have been different. Remember that class 72 enters the system in line-mode and is normally not recommended for doing a substantial amount of work. In the case of remoter access however, this may be the only possibility. After point 4 (class start) you would have seen the message

enter w for wylbur, v for vm

13. You reply with `v`. The system replies with

```
VM/370 ONLINE
```

where `VM/370 ONLINE` is the line mode VM Logo. Notice the following line starts with a `'`, the prompt.

14. You reply with the login command, as above:

**Login userid password**

If the login is successful, you are now logged on to your virtual machine in line mode. The system replies with the following messages:

```
FILES: xxx RDR, xxx PRT, xxx PUN
LOGON AT hh:mm:ss GVA xxxDAY mm/dd/yy
CERN CMS R4.x V.3
possibly other "LOG" messages
```

but notice the absence of the prompt.

15. You reply by hitting `RETURN`, to which the system replies with

```
maybe some news is shown (quit news by typing exit)
G(197) R/O
R; T=0.09/0.13 hh:mm:ss
```

Notice the appearance of the prompt on the line after the ready message.

To shut down your virtual machine use the command

**LOGoff**

It is important to realize that this command is completely equivalent to the shut-down of a real machine, i.e. all storage will be cleared and any temporary configuration (for example connected to minidisks) will be lost. Use the `LOG` command to logout of your full screen session and try logging on to your virtual machine in line mode.

As mentioned in Chapter 1, your virtual machine may continue working independently without having a virtual console (terminal) attached to it. This is achieved by using the

**DISConn**

command. The entire configuration of your machine will be preserved which is particularly useful in the middle of editing sessions or when you have a complicated disk set-up. To restart your machine after disconnecting, type

**Begin**

at point 9 (or 14 for class 72) in the sequence above.

A particular example of a program which you may run in your machine while it is disconnected is the "GONE" program. Among many other useful things "GONE" will keep track of any messages that come to your machine while you are away and can send others messages in turn. For example, activate GONE by simply typing

### **GONE away somewhere**

Now go to a friend's terminal and issue the command

**MSG userid-of-gone-user ARE YOU THERE?**

(see section 8.1.2 on page 79 for more information about communication between users). To this message your friend will obtain the message

```
FROM userid-of-gone-user: *GONE away somewhere
```

To restart your session (reconnect), you don't need to use B(egin), GONE does this for you. Normally GONE and DISConnect should only be used during the day; at night, LOGoff should be used.

### **2.1.1 Setting your Logon Password**

In these days of easy access to our computers across various networks it is important to prevent unauthorized use of your virtual machine by setting a logon password that is not too obvious to guess. The CERN VM system requires a logon password between 5 and 8 arbitrary characters. Set your logon password right now by using the command

#### **DIRM PW**

The DIRM command interfaces with the VM directory where CP keeps all the details of your Virtual Machine. To avoid these so-called "hackers" from finding your password and thus obtaining illegal access to our computers you are recommended to change it every 30 days.

If you have forgotten your password, please contact your **space administrator** (see sect. 9.5 on page 97 to find out who this is). He has sufficient privileges to give you a new password (see sect. 9.5 on page 97).

## **2.2 Special Features of the Keyboard**

If you have chosen class 101, 103 or 125 to log on to your virtual machine you have chosen to use VM in "full screen" mode. This means that the information is sent to the computer from your terminal one full screen at a time, i.e. after you hit RETURN. This as opposed to line-mode (typewriter-mode, class 72 or cf. also Wylbur) where the communication goes one line at a time. Fullscreen mode has many advantages but to be able to use its full power one clearly needs special features on the keyboard in comparison with an ordinary typewriter.

Some special keys on the DMT2200 Keyboard (see Figure 7 on page 12) are<sup>4</sup>.

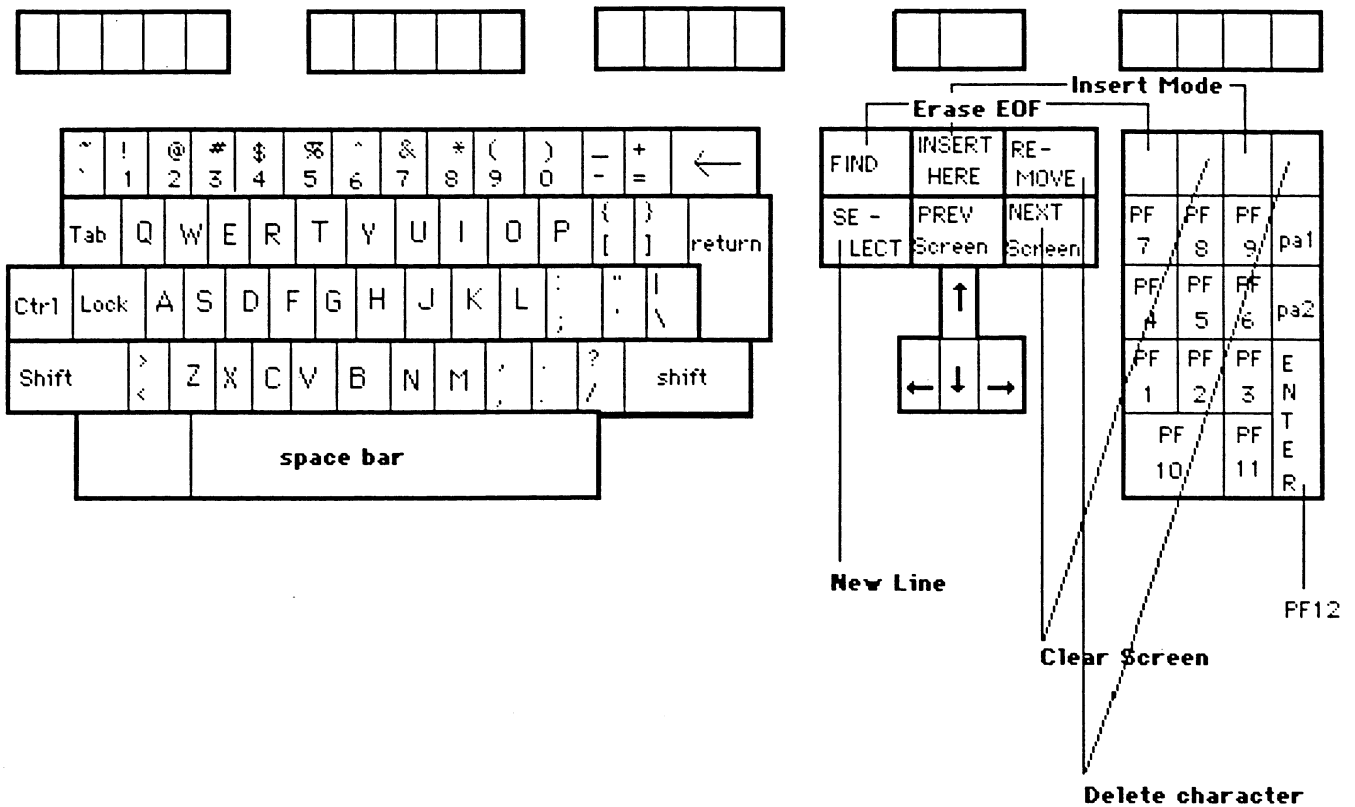


Figure 7: The DMT2200 Keyboard (terminal type: D2)

1. The **cursor movement** keys. These are marked by a left arrow, a top arrow, a bottom arrow and a right arrow respectively and are positioned to the right of the main keyboard. These keys allow you to place the cursor anywhere on the screen. Please locate these keys and try them out.
2. The **linefeed** (new line) key. Two keys can be used, one marked "Select" and the other above the key marked 7 in the numeric keypad. These keys position the cursor at the beginning of the line, or the first input area on the (next) line.
3. The **tabulator** key, marked TAB in the second horizontal row of the main keyboard. This key can be used for skipping to different (tab or input) positions on the line. Is equivalent to linefeed if no tab positions are set.
4. The **RETURN** key. On IBM terminals, this is called the ENTER key. It is placed below the backspace key on the right of your main keyboard. When this key is pressed, the entire screen is

<sup>4</sup> For the positions of these keys on the keyboards of other terminals the VM Users Guide or the VM Pocket card should be consulted. However, their functionality remains valid as described here.

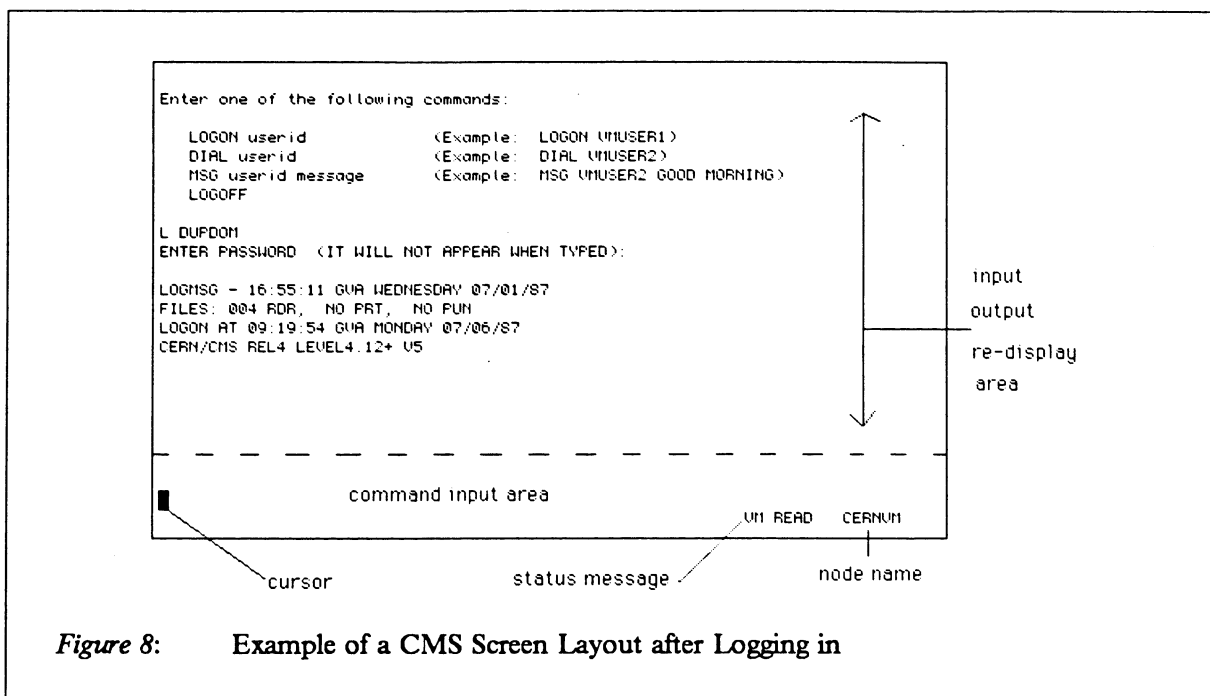
- 
- sent to the computer. All other keys may be pressed without any effect; only when the RETURN (or any PF key) key is pressed will the computer know about any other keys that were hit before. To re-instate the screen, use the CLRSCRN key (see sect. 2.4 on page 14).
5. The **space bar**. Function as on any typewriter, i.e. it moves the cursor one position along, except that it also erases what was there previously. Compare this to the cursor right movement key. For example, type something on the screen, press the linefeed key to get to the beginning of the line and then move the cursor along by hitting the space bar.
  6. The **backspace** key. This key is positioned above the RETURN key, at the right in the first horizontal row of the main keyboard. It moves the cursor back one position, but does not erase what was there. Identical to the cursor left movement key.
  7. **Capital and lowercase** letters. Either one of the two keys marked "Shift" on the main keyboard move into uppercase. The capslock key is positioned between the control key and the letter "A" in the third row of the main keyboard.
  8. The **Erase-End-Of-Field** key. Two keys can be used. One marked "Insert" and the other above the 7 in the numeric keypad. This key erases all characters from the cursor to the end of the line.
  9. The **CLRSCRN** key. Two keys can be used. One marked "Next Screen" and the second above the 8 in the numeric keypad. Pressing one of these keys inside the editor re-instates the screen as it was immediately after the last time the "RETURN" key was pressed. Outside the editor, it simply clears the screen including the command-line.
  10. The **INSERT MODE** key. Two keys can be used. One marked "Insert Here" and the second above the 9 in the numeric keypad. Pressing one of these keys once activates **insert mode**. This means that any text entered at the position of the cursor is entered in the middle of the line while any text after the cursor is moved on until the line is full. Pressing this key again or hitting RETURN de-activates insert mode, i.e. it acts as a "toggle" or "switch".
  11. The **DELETE CHAR** key. Two keys can be used. One marked "Remove", and the other above the key marked "-" in the numeric keypad at the top right. Deletes the character after the cursor.
  12. The **Programmable Function** keys. These are marked "1-11, ENTER" in the numeric keypad. They can be programmed to perform special tasks according to the environment that you are in (for programming these keys in the CMS environment see sect. 7.2 on page 69 and for the Xedit environment see sect. 4.7 on page 38).
  13. **PA1 and PA2**. These keys are marked "-" and "," respectively in the second and third row on the right of the numeric keyboard. PA1 returns your machine to CP, which may be useful if it is stuck inside a looping exec. Type **B** or hit PA1 again to return to CMS. PA2 clears the screen or displays the next screen if there is one (see below), but does not clear the command line (compare this to the CLRSCRN key).
  14. **Control G**. Sometimes your terminal will get blocked due to fast typing or other problems. You will not be able to enter any character and the terminal will beep at you if you attempt to do so. In this case, type Control G to reset the terminal. On an IBM PC or terminal, use the RESET key.

15. Repeat action. All keys on the DMT2200 keyboard will repeat themselves when they are hold down for a continuous period.

### 2.3 Screen Layout

Let's look at the layout of your screen (see Figure 8) after you have logged on. Starting from the bottom we have:

1. The input area. You are only allowed to type in the bottom two lines of your screen. This is called the **command line**.
2. In the bottom right hand corner we have the **status message area**, which normally contains "RUNNING" or "VM READ".
3. The **input/output display area**. After you hit RETURN, anything you typed on the command line will appear at the top of the screen, together with the response from the system.



Please note that inside the editor (Xedit) you may have a completely different screen layout.

## 2.4 Status Messages

The status messages in the bottom right hand corner of your screen give very useful information concerning the state of your virtual machine. There are not many of them, but it is important to learn to know them and to recognize which action to take when you encounter them:

**MORE...** There is more information than will fit on one screen. After 50 seconds the terminal will beep at you and 10 seconds later display the second screen.  
**Action:** Press CLRSCRN or PA2 (if you want to keep what is already on the command line) to see the next screen before the 1 minute is over. Press RETURN to hold the information on the screen. MORE... is then changed to HOLDING.

**HOLDING** The system is waiting for you to clear the screen before showing more information.  
**Action:** Hit PA2 or CLRSCRN as for MORE... Hit RETURN to move between the HOLDING and MORE... states.

**VM READ** CMS (or any program running in your machine) is waiting for data or a command.  
**Action:** Type in the requested reply, or simply hit RETURN. If you are in the CMS environment and CMS is waiting for you to enter a command, the system will respond to you with

CMS

**NOT ACCEPTED** Your machine is possibly trying to tell you something and you have stopped it by hitting RETURN or a PF key. You have probably ignored a MORE... or HOLDING message.  
**Action:** Wait and see what the status message changes to and take the appropriate action. Don't keep hitting the RETURN or PF key.

**RUNNING** This means that either the system is running your program, or CMS is waiting for your next command.  
**Action:** If you are not sure, hit RETURN to get VM READ, then RETURN again to see if you get the response

CMS

To find out which environment you are in when the screen is empty, simply hit RETURN and the machine will respond with

CP or  
CMS or  
CMS SUBSET

(To exit from CMS SUBSET, type RETURN (in letters) and press the ENTER key).



For a rough map showing the relationship between these different environments in your machine, see Figure 9 on page 16.

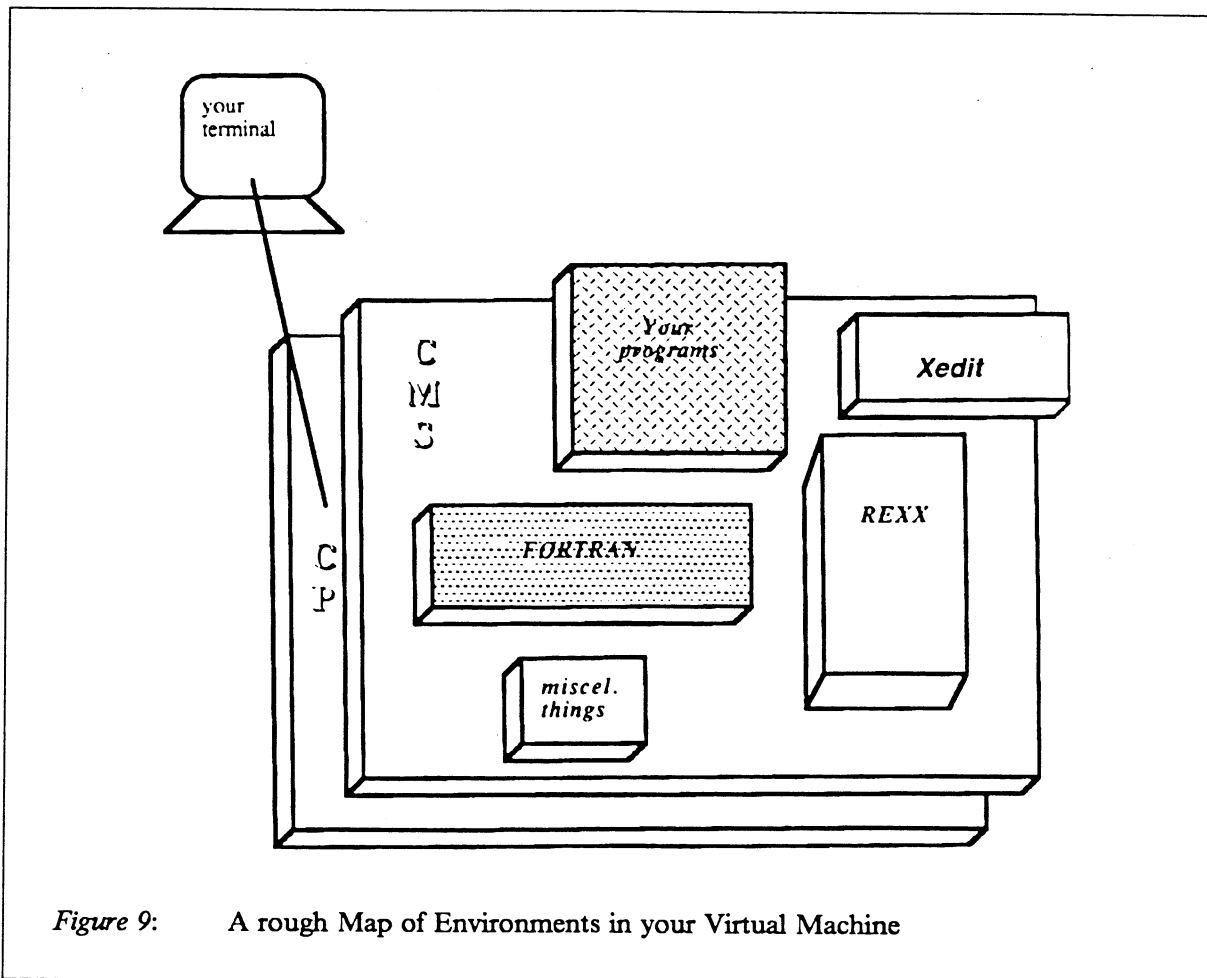


Figure 9: A rough Map of Environments in your Virtual Machine

The communication between you and the system goes through commands which you type in at the terminal. However, as can be seen from Figure 9, it is important to know in which environment you are. The lower down you are in the map, the fewer commands your environment will understand: in CP your machine will not understand CMS commands.

For example, if you are in the CMS environment, and you try typing an Xedit command, this will result in an error message. However, if you are inside Xedit and you type a CMS command, Xedit will pass the command on down to CMS (if there isn't an Xedit command with the same name). Similarly, CMS passes commands it does not understand on to CP. You can prevent CMS from doing this by giving the command

**SET IMPCP OFF**

To be certain that the command always goes to the right environment, precede it with **CP** or **CMS** from inside Xedit.

## **2.5 Exercises for Chapters 1 and 2**

As an exercise for the material covered in Chapters 1 and 2 now login to your virtual machine and type the commands:

**GIME PUBCP 201  
TUTORIAL**

Subsequently work through lessons 1 and 2 in the interactive tutorial until you feel you have mastered the material covered there.

### 3. The Editor (Xedit)

#### Summary of chapter 3.

By the end of this Chapter, you will have learnt;

- How to use/understand CMS files.
- How to create CMS files by using Xedit.
- Various functions of the editor.

#### 3.1 What is a File?

A file is a collection of data. Examples are the text of a memorandum, a program written in FORTRAN, or simply a collection of numbers. Each file consists of lines, or **records**. All CMS files are characterized by a unique 3-part label, called the **filename**, **filetype** and the **filemode** respectively. The filename and filetype can be any combination of uppercase letters or numbers provided they are no longer than 8 characters. The filemode consists of a letter (A-Z) followed by a number (0-6). The letter denotes the minidisk on which the file resides and the significance of the number will be explained later. The filetype is often used to group together certain files that have something in common. For example all files that are input to the FORTRAN compiler have filetype FORTRAN. The files that are input to SCRIPT (the textprocessor) have filetype SCRIPT, and command files for the REXX interpreter have filetype EXEC. These processors search for files that have given filetypes so that you do not have to type them in separately. You will recognise many more filetypes as you become more familiar with VM.

The 3-part label that characterises a CMS file will often be referred to as the **CMS fileid**. Some examples of fileids are

```
PROFILE EXEC A1
PROFILE XEDIT A0
JETSET FORTRAN B1
LETTER SCRIPT D1
CRNPDISK OWNERS P2
COPYFILE HELPCMS P1
```

Please note that the concept of a 'library' or 'directory' which is common to other operating systems, does not exist in VM<sup>5</sup>. Instead, files may be grouped by giving them a common filetype.

---

<sup>5</sup> MACLIBS, TXTLIBS and LOADLIBS have a library structure but they are only used in a special context.

## 3.2 How to Create a CMS File Using the Editor (Xedit)

### 3.2.1 Entering the Editor or Starting an Xedit Session

To create a CMS file you have to invoke the editor, Xedit. Xedit has its own "environment" or "mode" which you have to move "in" and "out" of when you want to create a file. This can be compared to logging "on" and "off" your machine. To enter into the editor, use the command Xedit:

**Xedit filename filetype [filemode]**

Notice that the Xedit command can be abbreviated to "X", and that the filemode can be omitted (indicated by the angular brackets). Xedit will search through your disks in alphabetical order, stopping at the first file to match the filename and filetype that you gave. This search can be restricted by specifying the filemode. If no match is found, Xedit assumes you want to create a new file on your "A"-disk unless another letter was given.

### 3.2.2 Quitting the Editor or Ending your Xedit Session

When you enter Xedit, a copy of the file is brought into the virtual storage of your machine. If the file already exists, the original on disk remains unchanged until you exit the editor, or explicitly SAVE a copy on disk by issuing the Xedit SAVE command. If the file did not exist, you are only working in your virtual storage until you explicitly SAVE or FILE a copy on disk. Figure 10 shows the relationship between what you edit and what is on disk.

To exit the editor without saving or changing the copy on disk, use the

**QQ (Quit Quickly)**

command.

To exit the editor and save a copy on disk, use the

**FILE [fn [ft [fm] ] ]**

command.

Note that you may change the filename, filetype and filemode by supplying these as arguments with the FILE command. One can also use the commands

**[SET] FName newfilename**

**[SET] FType newfiletype**

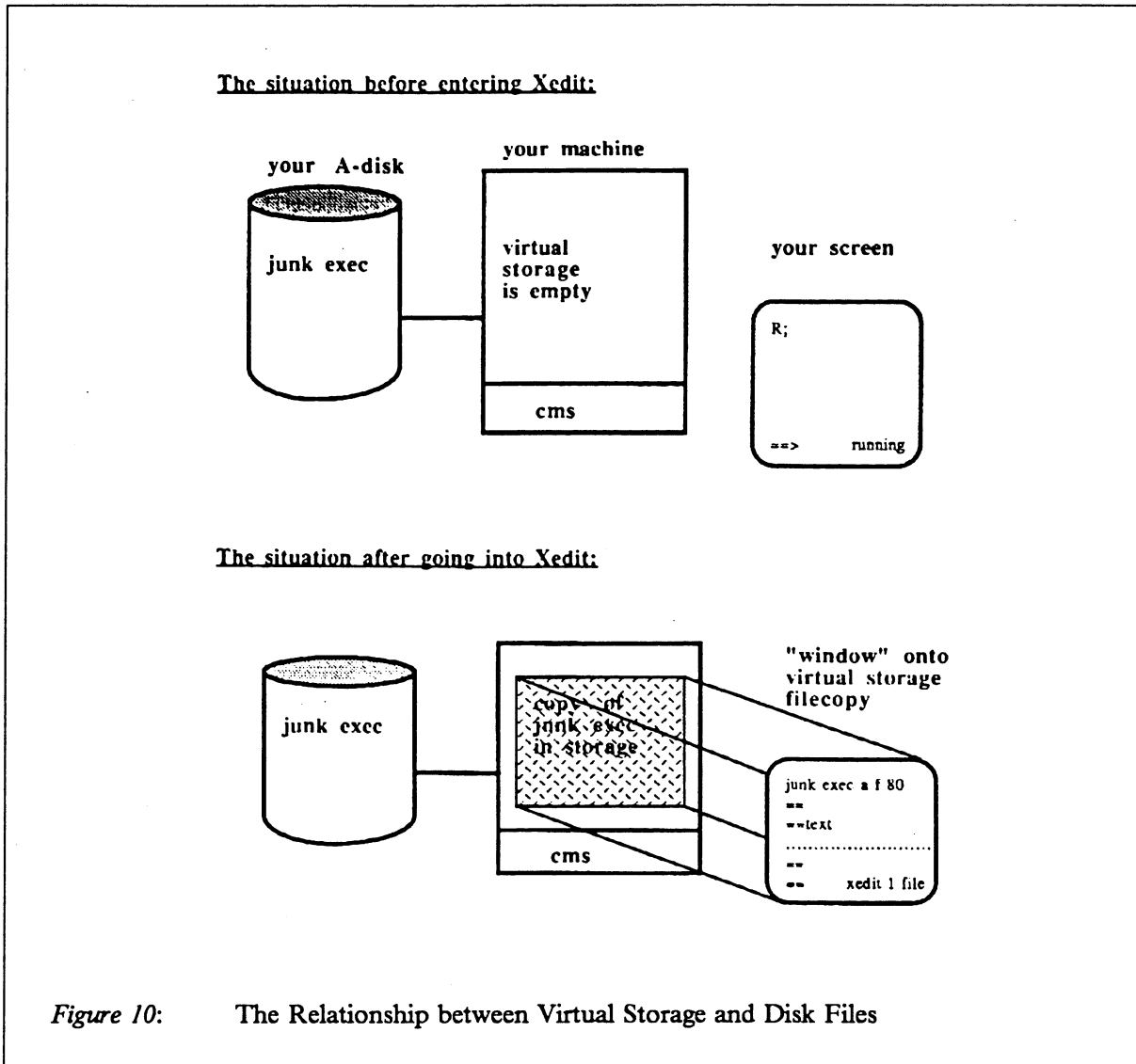
**[SET] FMode newfilemode**

inside Xedit to change the fileid of the file in storage.

To save a copy on disk, but remain inside the editor, use the

**SAVE**

command.



This mechanism allows you to make changes to an existing file in a safe manner, keeping the original on disk. However, if there are problems with the system and for some reason your virtual storage is lost, so are all your edits. For this reason the AUTOSAVE function is available. Autosave will automatically SAVE a copy on disk after every n changes. Here n is defined by the command

**SET AUTOSAVE n**

Xedit stores away the file with a filetype of AUTOSAVE and a number as filename. It is erased when you FILE, but not when you QQuit.

### 3.2.3 The Xedit Screen Layout

When you enter Xedit to create a new file, e.g. JUNK EXEC your screen will look as shown in Figure 11.

The editor organises the screen into 8 functional areas:

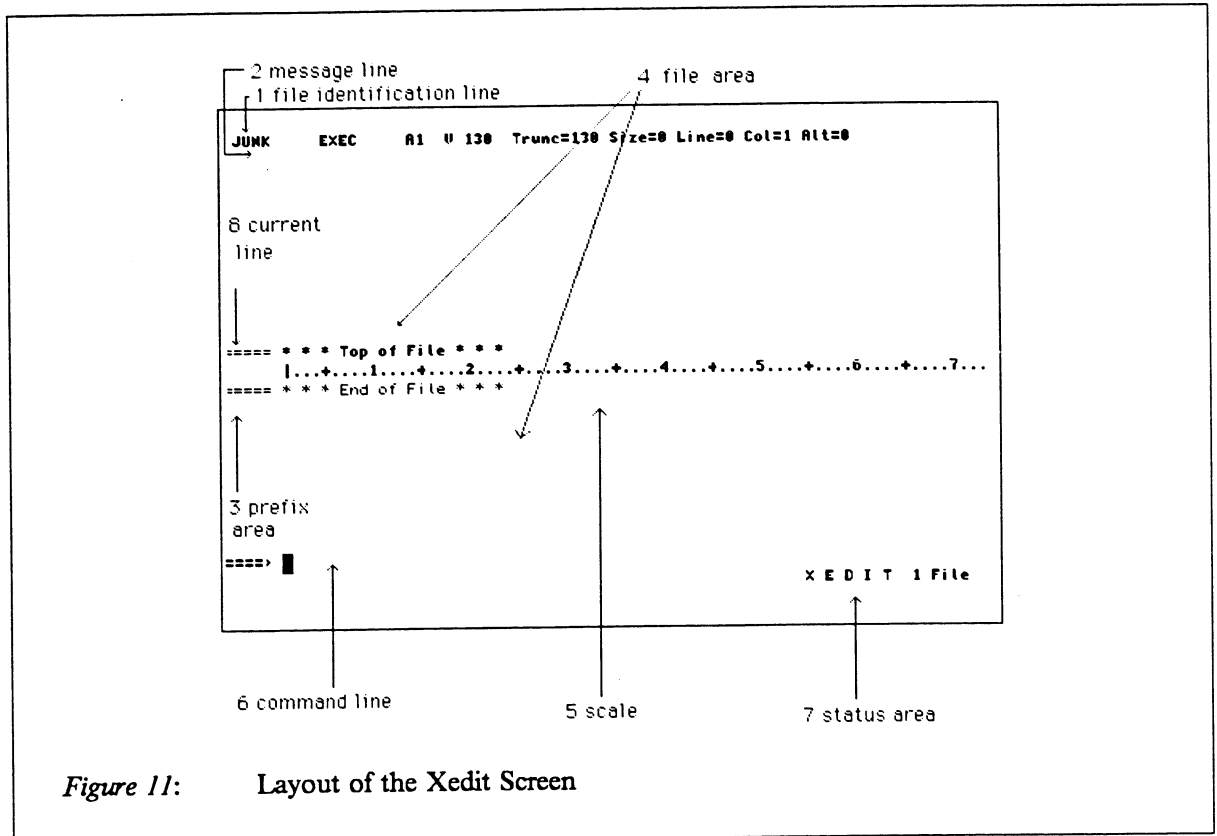


Figure 11: Layout of the Xedit Screen

1. **The File Identification Line.** This contains, from left to right:
  - a. **The fileid:** filename, filetype and filemode.
  - b. **The record format:** F or V. F means all records in the file have the same length, V means the records have variable length. CMS only understands record formats F and V.
  - c. **The record length.** For variable length files this is the length of the longest record.
  - d. **The truncation column.** Any data beyond this column will be rejected by the system. The truncation column does not exceed the record length but it can be less.
  - e. **The size of the file,** i.e. the number of lines (records) of the file in storage.
  - f. **Line.** The number of the current line (see section 4.3.1 on page 29).
  - g. **Column.** The number of the current column, also indicated by a "|" (vertical bar) in the "scale" line (see section 4.3.1 on page 29).
  - h. **Alteration count.** The number of alterations made to the file in storage compared to the file on disk or the last (AUTO)SAVED file.
2. **The message line.** This line is used for messages from the editor. Hit RETURN to get rid of them.

3. **Prefix area.** This area can be used for entering certain commands, called **prefix commands**. It will be discussed later.
4. **File area.** The lines of the file are displayed on the screen between the **\*\*\* TOP OF FILE \*\*\*** and **\*\*\* END OF FILE \*\*\*** notices. All this area may be edited.
5. The **scale line.** This line displays column numbers to help you edit the file. The current column is indicated with a **|** and the zone (if set) is marked by **"<"** and **">"**.
6. **Command line.** All communication between you and the editor occurs here. After typing in an Xedit command after the **"= >"**, hit RETURN to send it to the editor. You may give the editor more than one command at a time by using the **linend character #**. Examples of this will be shown later.
7. The **status area.** This area shows the status of your editing session, e.g. **"INPUT MODE"**, **"X E D I - > JOHN"** or how many files you currently have in the editor: **"X E D I T 2 Files"** implies you have 2 files in storage (see section 4.6 on page 37).
8. The **current line.** Normally the line above the "scale line" is the current line. Most Xedit commands perform their function from the current line downwards in the file. Many commands position to a new current line after they have been executed.

Most of this default display can be changed via a file called PROFILE XEDIT. Section 4.9 on page 41 shows how this is done.

### 3.3 Adding Data to a File

There are three different ways of adding data to a file: via the **INPUT**, **ADD** or **POWERINPUT** commands.

#### 3.3.1 The Input Command.

The **input** command can be given with or without arguments. The command

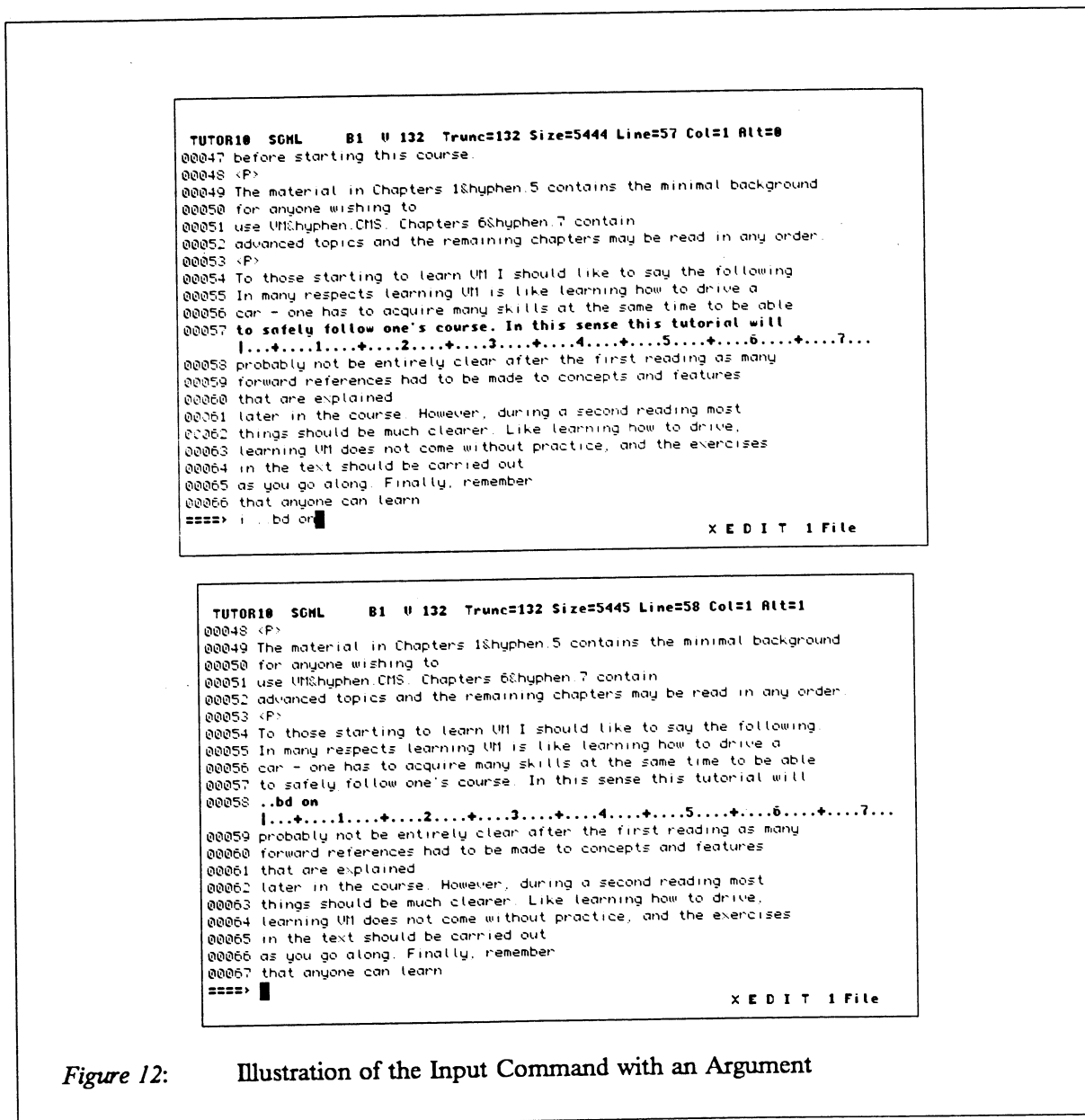
```
Input ..bd on
```

will change the screen as shown in Figure 12; it simply places the argument on the current line.

The input command with no arguments places you in an environment called **input mode**. Figure 13 on page 23 shows the changes that occur to the screen of Figure 12 after going into input mode.

Notice the following characteristics of input mode:

1. In input mode, the prefix area disappears. The message line and status area display the notice: **"INPUT-MODE"**. The command line contains **\*\*\* Input Zone \*\*\***. In other words, there is nowhere on the screen to type commands.
2. Input-mode is a sub-environment of Xedit, where that part of the display following the current line is cleared ready for any data you wish to add to the file. This area is called the **"Input Zone"**.
3. Each line typed in the input zone becomes a separate record in the file. At the end of each line, hit line-feed. When the end of file is reached, a new line is started automatically, also resulting in a new record in the file.



4. If the screen is full, pressing RETURN once clears it for more data.
5. To exit from input mode, hit RETURN twice.

### 3.3.2 The Add Command

If you wish to remain inside the normal Xedit environment while you are adding lines, type

**Add n**

where n is the number of lines you wish to add after the current line. Any empty lines will remain in the file until explicitly removed by the **DELETE** command (see section 3.4).



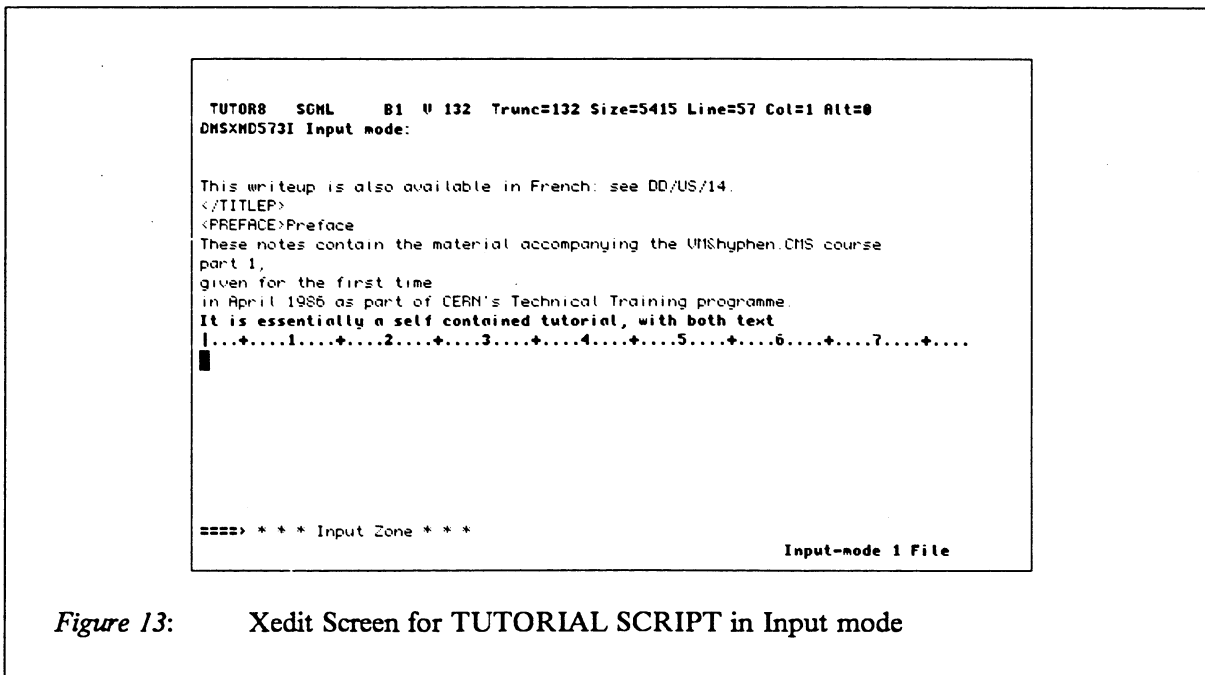


Figure 13: Xedit Screen for TUTORIAL SCRIPT in Input mode

### 3.3.3 POWerinput

If you have a large amount of text to enter into your file, use the

#### POWerinp

command. This clears all but the top two lines of the screen and allows you to type continuously, as if the input zone were one long line. When you reach the end of the line, the cursor will automatically move to the next line. After pressing RETURN, Xedit reassembles the words that are broken across the screen and split the one long record up into lines of 72 characters. To force Xedit to start a new line in the file you may use the `linend` character ("`#`" by default at CERN). This is very useful when you are entering SCRIPT input, where some words must start specifically in column 1. To exit powerinput, hit RETURN twice. An example of the use of powerinput is given in Figure 14.

### 3.4 Deleting and Recovering Lines

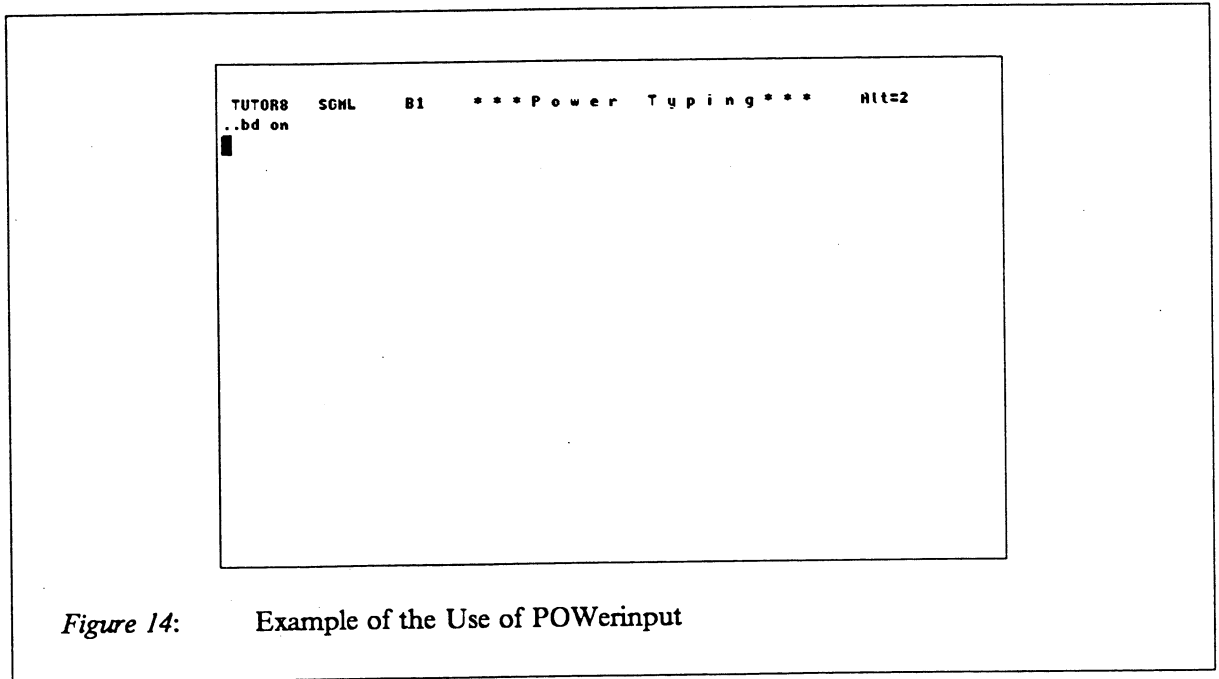
To delete `n` lines from (including) the current line downwards, use the command

#### DELeTe n

Supposing you accidentally removed some lines from a file you were editing. In this case Xedit has a very friendly command called

#### RECOVer n

This command will recover `n` lines for you and place them before the current line.



*Figure 14:* Example of the Use of POWERinput

### 3.5 Exercises for Chapter 3

Execute the command "TUTORIAL" on PUBCP 201 and continue at lesson number 3.

## 4. More Xedit

### Summary of Chapter 4.

At the end of this chapter you will have learnt;

- How to move around in a file.
- What prefix commands are and how to use them.
- More functions of the editor, such as moving and copying lines; searching for and changing of data.
- Some useful commands.
- Editing multiple files.
- What the PROFILE XEDIT is and how it can be used to adapt your Xedit environment to your own taste.

### 4.1 Moving around in a file

A terminal has a width of 80 characters by 24 lines. When inside Xedit (not in the POWerinput environment), at least 3 lines are reserved, plus, if the prefix area is on, 6 columns for the prefix area. This leaves a viewing window of 73 columns by 21 lines. However, many files will be much longer than 21 lines and have records that contain more than 73 characters (see Figure 15). The following is a list of commands which may be used to move the window over the file:

**Backward n|\*** – move up n screens

**Forward n|\*** – move down n screens

**Bottom** – move to end of file

**TOP** – move to top of file

**Up n|\*** – move up n lines

**Down n|\*** – move down n lines

**Next n|\*** – move down n lines

**RIght n** – move n columns to the right

**LEft n** – move n columns to the left

n defaults to 1 and "\*" means "as far as you can go", i.e. to the top or the bottom of the file. For example, the commands

**D \***  
**FO \***

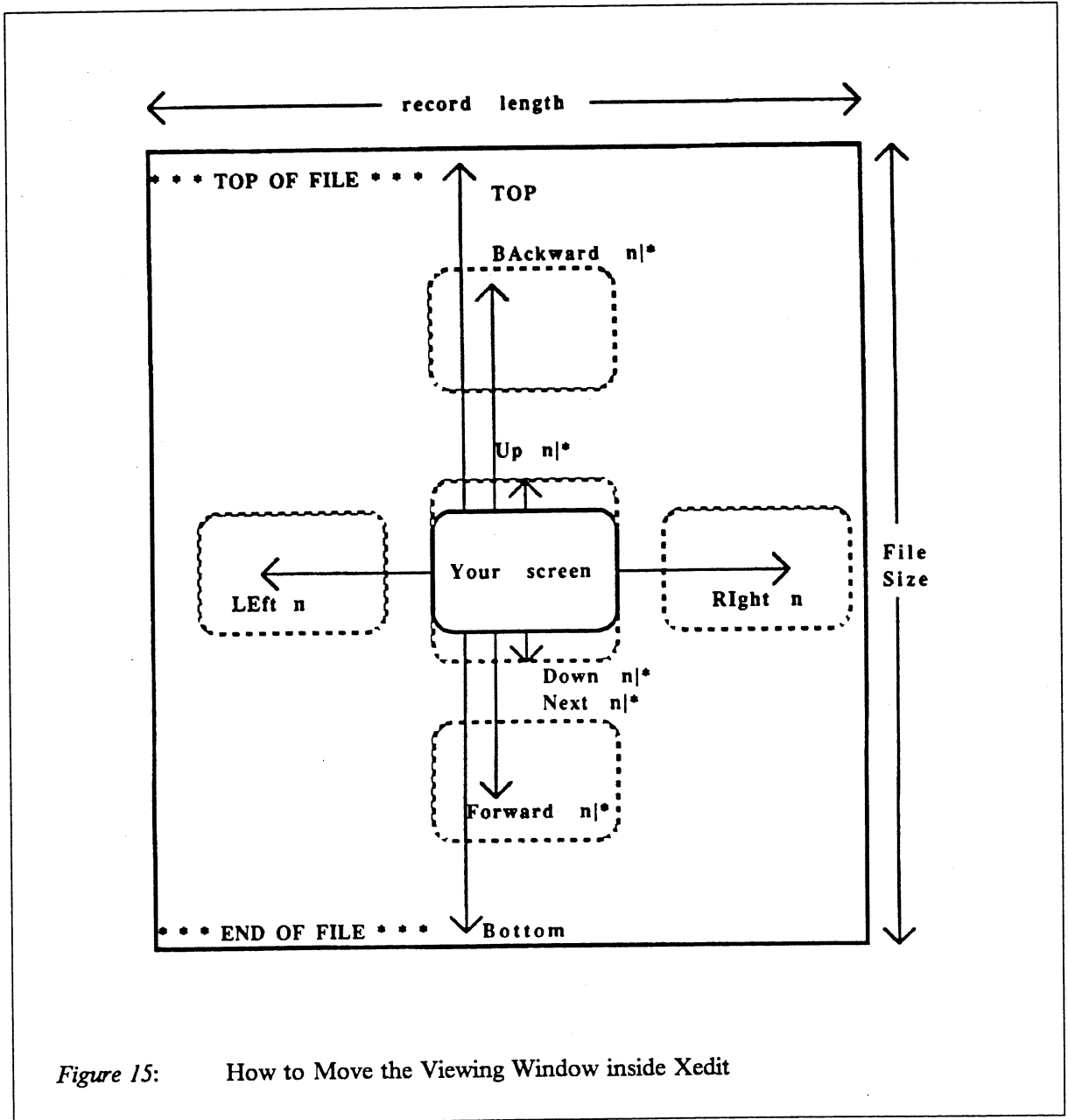


Figure 15: How to Move the Viewing Window inside Xedit

B

all have the same effect.

## 4.2 Prefix commands

Prefix commands are commands which are entered in the prefix area instead of on the command line. They are usually related to line or groups of lines. Valid prefix commands are<sup>6</sup>:

<b>An</b>	add n lines following this line
<b>AL</b>	align this line
<b>Cn</b>	copy n lines to a position defined by F or P
<b>CE</b>	centre this line
<b>Dn</b>	delete n lines starting with this line
<b>E</b>	extend this logical line to contain the next line on the screen
<b>F</b>	following; tells MOVE or COPY to place lines following this line
<b>G</b>	get lines from (temporary) virtual storage following this line
<b>In</b>	insert n lines following this line
<b>JU</b>	justify this line
<b>L</b>	translate this line to lowercase characters
<b>M</b>	move this line to a position defined by F or P
<b>P</b>	preceding; tells MOVE or COPY to place lines preceding this one
<b>PT</b>	put this line in (temporary) virtual storage
<b>PD</b>	put this line in (temporary) virtual storage and remove it from the Xedit file
<b>RA</b>	right adjust this line
<b>S</b>	show (reinstate lines lost from display via the exclude command)
<b>U</b>	translate this line to uppercase
<b>X</b>	exclude (conceal) this line from display
<b>"</b>	duplicate this line
<b>/</b>	make this line the current line
<b>&gt; n</b>	move the data n columns to the right

---

<sup>6</sup> Please do not try to memorise them all. Select which ones you think may be useful for your purposes. For a more detailed description, try FIND PREFIX or HELP PREFIX MENU or see the Xedit manual [4].

---

<b>&lt;n</b>	move the data n columns to the left
<b>SCALE</b>	display the scale line above this line
<b>TABL</b>	display the tabs line above this line
<b>.xxxx</b>	assign the symbolic name xxxx to this line

Prefix commands obey the following rules.

1. Prefix commands may be entered anywhere in the prefix area, in lower or uppercase letters. However being in INSERT mode will prevent you from entering prefix commands in the prefix area; For example

A====

and

==a==

will both add one blank line after the line on which they are typed.

2. Only one prefix command may be given per line. For instance,

C=P==

is an illegal combination.

3. Some commands (i.e. those which have the letter "n" behind them in the list above), can take a number as argument. This does not have to immediately follow the command, but can be entered anywhere in the prefix area:

A3===

and

=a3==

and

=3=A=

are all equivalent.

4. Block commands. Certain prefix commands can be duplicated to "bracket" a block of lines. This is very useful as records need not be counted. For example

```

d3=== d==== dd===
===== d=====
===== d===== dd===

```

all have the same result, i.e. deleting 3 records. Other commands that can be used like this are

```
XX < < > > "" MM CC PDD PTT UU LL CEE JUU
```

#### 5. The commands

```
C Cn CC M Mn MM
```

need to be told where to "copy" or "move" to. Use P or F to show the target line, which may be on a different logical screen. If the P or the F is omitted, a "PENDING" notice will appear in the status area. This can be removed (together with any outstanding prefix commands) with the RESet command, typed on the command line.

6. Prefix commands do not normally move the current line (except if you delete it).
7. To display the numbers in the prefix area instead of "=" signs, use the command

```
SET NUMBER ON
```

This is helpful for remembering the position of certain lines in a given file, or if one is used to a line editor such as Wylbur. Line names (assigned by typing .name in the prefix area) are also useful.

## 4.3 Some important Xedit Concepts

### 4.3.1 The Current Line and Current Column

The **current line** is a pointer which indicates to Xedit your position in a file. The **current column** indicates your position on a line.

Many commands take effect from the current line onwards. It can be found in the file identification line at the top of the screen as "LINE=nn". For example, if the current line is 5, a search for the string "VM-CMS" will commence from line 5 downwards. The command

```
Locate /VM-CMS/
```

sets the current line to the first line after line 5 containing the string "VM-CMS" (see Figure 16).

Similarly, many commands start from the current column. For example, if the current column is column 1, the command

```
CLocate /VM-CMS/
```

will search the current line starting at the current column (column one in this case); the column pointer will indicate where the string "VM-CMS" starts. See Figure 17 on page 31 for an example of how CLocate moves the current column.

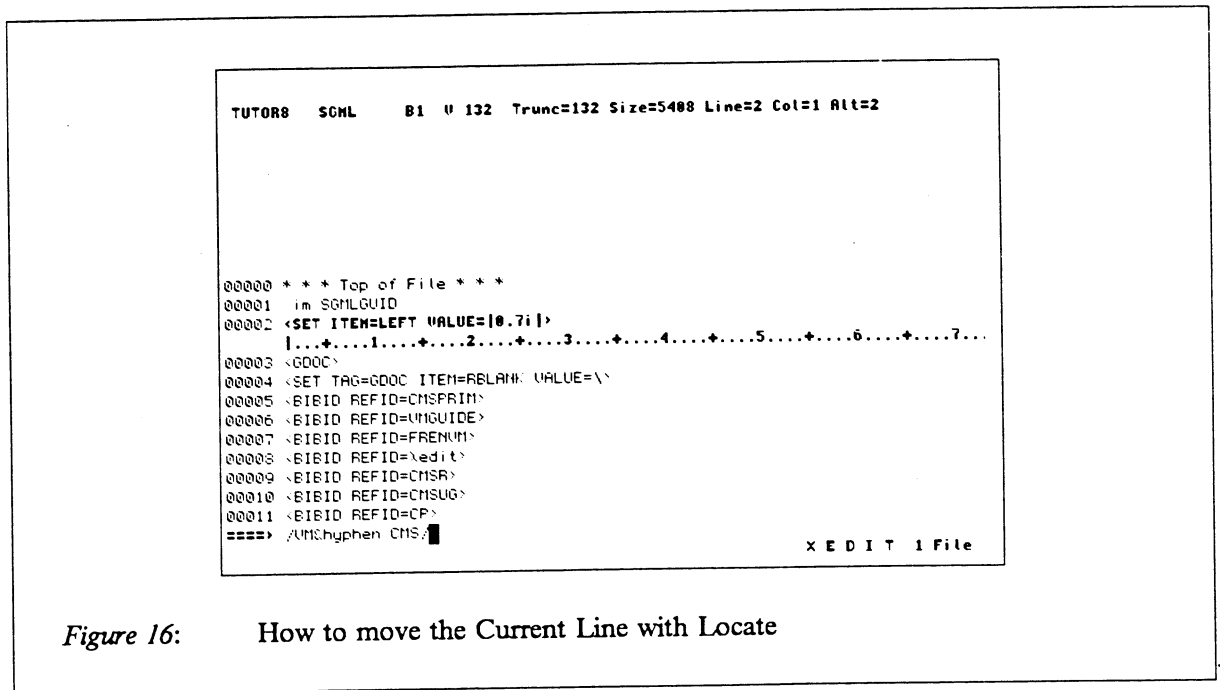


Figure 16: How to move the Current Line with Locate

There are many commands beginning with "C" that start at the current column in the current line. The current column can be found by looking for the "|" in the scale line or from "COL=nn" in the file identification line.

#### 4.3.2 Targets

The current line and current column define where a command commences its action. The target specifies where this action will stop. Normally the command will not act on the target but make its position the new current line or column.

There are two types of targets, **numeric** and **string**:

##### 1. Numeric targets.

```

:n moves the current line/column to line/column number n
+n moves the current line/column forward n lines/columns
-n moves the current line/column backward n lines/columns

```

Whether the action takes place on lines or columns depends on the command. Those prefixed by a "C" generally act on columns. For an example, see Figure 18 on page 32. Notice the use of the "#" to allow both Locate and Clocate commands to be given on the same line.

##### 2. String targets. The format, as shown below, is very similar for both lines and columns:

```

[+|-] [-] /string1 [ / | [-]/ string2 / ] ...
  a   b c   d           e

```



```

TUTOR10 SGHL      B1 U 132 Trunc=132 Size=5444 Line=1 Col=1 Alt=0

00000 * * * Top of File * * *
00001 .im SCHLGUID
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7..
00002 <SET ITEM=LEFT VALUE=[0 7i]>
00003 <SET TAG=H1 ITEM=PLACE VALUE=TOP>
00004 <GDOC>
00005 <SET TAG=GDOC ITEM=RBLANK VALUE=\>
00006 <BIBID REFID=CMSPRIM>
00007 <BIBID REFID=UMGUIDE>
00008 <BIBID REFID=FRENUM>
00009 <BIBID REFID=XEDIT>
00010 <BIBID REFID=CMSR>
====> cl /UM&hyphen.CMS:█

X E D I T 1 File

TUTOR10 SGHL      B1 U 132 Trunc=132 Size=5444 Line=18 Col=16 Alt=0
00008 <BIBID REFID=FRENUM>
00009 <BIBID REFID=XEDIT>
00010 <BIBID REFID=CMSR>
00011 <BIBID REFID=CMSUG>
00012 <BIBID REFID=CP>
00013 <BIBID REFID=CERNPAP>
00014 <BIBID REFID=MAIL>
00015 <FRONTM>
00016 ..bf roman12
00017 <TITLEP>
00018 <TITLE STITLE="UM&hyphen.CMS Tutorial">
      <...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
00019 ..tb $SYSCL r
00020 <TITLE>CERN &'05 CERN User Support Group DD/US/5
00021 <TITLE>UM&hyphen.CMS Tutorial &'05.$SYSDATE
00022 ..tb
00023 ..sp 8
00024 ..bd
00025 ..bf roman18
00026 ..ce on
00027 UM&hyphen.CMS Tutorial
====> █

X E D I T 1 File

```

Figure 17: How CLocate moves the Current Column

where:

- a. + means a forward (lines) or right (columns) search  
- means a backward (lines) or left (columns) search
- b. ¬ is the "not" symbol (locate any text that is not the given string)
- c. / is the delimiter. Any character or two identical characters may be used except ¬, +, - or . (dot). The trailing delimiter may be necessary if the string contains blanks.
- d. string is the n-th character string.
- e. | is the "or" symbol (locate any of these strings). If the string is applied to a line search, the & (ampersand, the "and" symbol) may also be given (locate the line containing both strings).

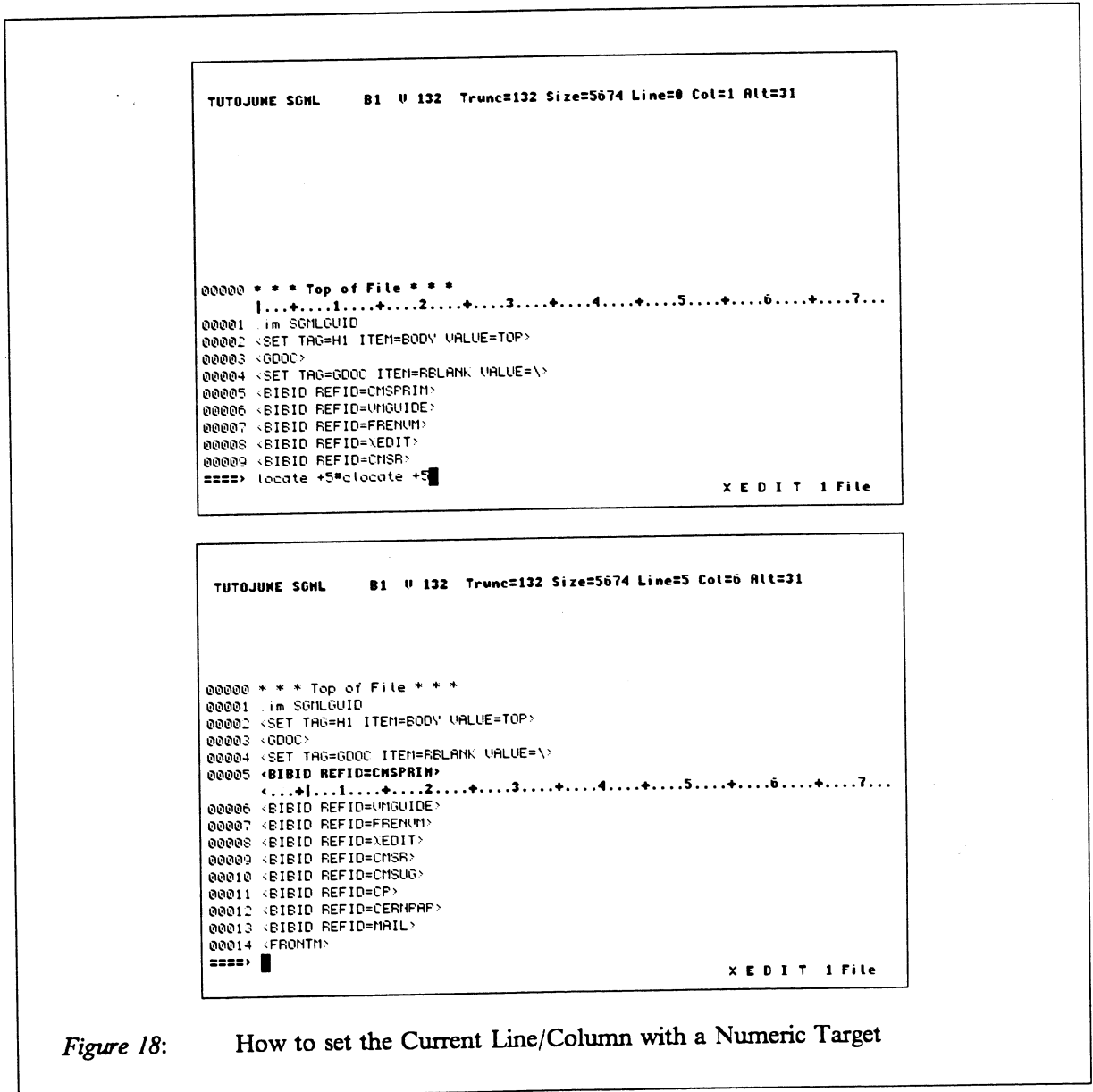


Figure 18: How to set the Current Line/Column with a Numeric Target

The most obvious application of targets is for use with line (Locate) or column (CLocate) searches (see Figure 16 on page 31 and Figure 17) and for changing strings (see below). Please note that if only a target is given, without a preceding Locate or CLocate, Locate will be assumed. An example of a more complicated target is given in Figure 19.

If a search for ALL instances of a particular string in a file is required, the ALL command can be used. The result of the command

**ALL /logo/**

(i.e. find all the occurrences of the string logo in the file called TUTOR10 SGML) is displayed in Figure 20 on page 34.

If you want to get rid of the lines telling you how many lines are not displayed between each instance, use the command

```

TUTOR10 SGML B1 U 132 Trunc=132 Size=5444 Line=35 Col=1 Alt=0
00025 .bf roman18
00026 .ce on
00027 UM&hyphen.CMS Tutorial
00028 .ce off
00029 .pf
00030 .sp 4
00031 .bf roman10
00032 "An elephant is a mouse with an IBM operating system."
00033
00034
00035
                                Anon
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
00036 .sp 5
00037 This writeup is also available in French: see DD/US/14.
00038 .sp 15
00039 </TITLEP>
00040 <PREFACE>Preface
00041 These notes contain the material accompanying the UM&hyphen.CMS course
00042 part 1,
00043 given for the first time
00044 in April 1986 as part of CERN's Technical Training programme.
====> /UM/S/like//the;█
                                X E D I T 1 File

```

```

TUTOR10 SGML B1 U 132 Trunc=132 Size=5444 Line=54 Col=1 Alt=0
00044 in April 1986 as part of CERN's Technical Training programme.
00045 It is essentially a self contained tutorial, with both text
00046 and exercises. Some knowledge of computers would be helpful
00047 before starting this course.
00048 <P>
00049 The material in Chapters 1&hyphen.5 contains the minimal background
00050 for anyone wishing to
00051 use UM&hyphen.CMS. Chapters 6&hyphen.7 contain
00052 advanced topics and the remaining chapters may be read in any order.
00053 <P>
00054 To those starting to learn UM I should like to say the following.
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
00055 In many respects learning UM is like learning how to drive a
00056 car - one has to acquire many skills at the same time to be able
00057 to safely follow one's course. In this sense this tutorial will
00058 probably not be entirely clear after the first reading as many
00059 forward references had to be made to concepts and features
00060 that are explained
00061 later in the course. However, during a second reading most
00062 things should be much clearer. Like learning how to drive,
00063 learning UM does not come without practice, and the exercises
====> █
                                X E D I T 1 File

```

Figure 19: Locate using a Complicated String Target

## SET SHADOW OFF

Please note that the lines which are not displayed, still remain in virtual storage, although prefix commands only act on lines on the screen.

To re-display the hidden lines, type ALL without any arguments.

Xedit contains several features to make complicated targets. Below are a few examples:

1. By setting the case, you can control how Xedit deals with upper and lowercase letters:

[SET] CASE [Uppercase|Mixed] [Respect|Ignore]

**Upper** translate lowercase to upper case before letters are entered into the file  
**Mixed** do not translate

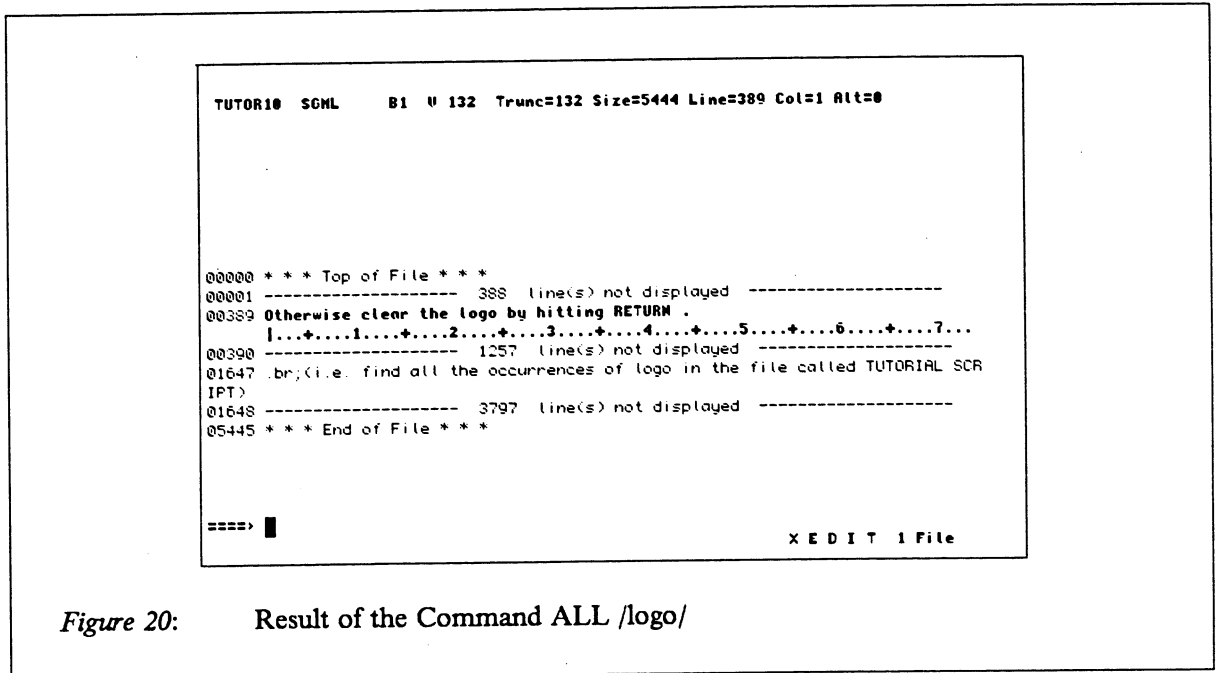


Figure 20: Result of the Command ALL /logo/

**Respect** respect case differences in a target search

**Ignore** ignore case differences in a target search

For example, after **CASE M I**

**/cern/**

will locate "cern", "Cern" and "CERN"; but after **CASE M R** it would only find "cern".

2. An arbitrary character can be defined to represent any number of characters irrelevant to the search target (a so-called wild-card). This may be set using the following command:

**SET ARBCHAR ON \$**

See Figure 21 for an illustration of the use of ARBCHAR.

3. A target may contain a variable number of blanks. If

**SET VARBLANK ON**

is in effect (OFF is the default) then a search for the string "of the line" will result in

```

"of the  line"
" of the line"
"  of the line"

```

all being found.

```

TUTOLAST SOHL      B1  U 132  Trunc=132 Size=5551 Line=296 Col=1 Alt=0

==== * * * Top of File * * *
==== ----- 295 line(s) not displayed -----
==== CERN. In <FIGREF REFID=MINDISK> we can see how a real disk can
==== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== ----- 1489 line(s) not displayed -----
==== /see$show/ with ARBCHAR ON $
==== ----- 934 line(s) not displayed -----
==== configuration see the section on how to use the
==== ----- 2830 line(s) not displayed -----
==== * * * End of File * * *

====> █

X E D I T 1 File

```

Figure 21: Result of ALL /see\$show/ with ARBCHAR ON \$

4. A target may be contained in more than one line. If

#### SET SPAN ON

is in effect, a string target may span one line and continue on the subsequent lines (OFF is the default).

#### 4.3.3 The Zone

The zone defines the columns (starting and ending position) of each line to be scanned for targets. For example, if the command

```
[SET] ZONE 10 20
```

was issued, then the search for CERN on the line

```
CERN is the European ....
```

would not locate the string CERN as this does not fall within the defined zone.

#### 4.3.4 Finding the Current Settings

To know what settings are in effect during your Xedit session, e.g. what your ARBCHAR is set to, the

#### Query

command may be used:

Q ARBCHAR  
 Q CASE  
 Q AUTOSAVE  
 Q VARBLANK  
 Q WRAP  
 Q SPAN  
 etc.

#### 4.4 Changing Data in a File

Any data on the screen can be changed by simply moving the cursor anywhere and typing over the data which is displayed. However, this soon becomes tedious if you have many changes to make and can very easily be done with the **Change** command:

**Change /string1/string2/target|\* p q**

where:

- String1** is the data to be changed.
- String2** is the new data. Clearly neither string1 nor string2 may contain the delimiter. If string2 is empty ("/") then string1 is not replaced by anything. It is deleted.
- Target** indicates when to stop changing. If the target (the same rules apply as listed above for targets) is omitted, only the current line will be changed. If the target = "\*", this is interpreted as the end of the file.
- P** is the number of occurrences to change per line (default = change only the first occurrence in the line). \* means all occurrences on a line should be changed.
- Q** is the relative number in a line of the first occurrence to be changed within a line (default = start changing from the first occurrence).

For example,

**C /CERN/European Laboratory for Particle Physics/ \* 1 2**

means, where CERN occurs more than once in a line, change the second string occurrence to "European ...", starting from the current line and terminating at the end of the file.

The same rules for targets (including ARBCHAR and CASE etc.) as described for searches also apply for the change command.

Selective changes, i.e. verify each change before it is made, can be achieved with the **SCHANGE** command. The **SCHANGE** command can be used when it has been assigned to a PF key and when a **Change** or **CLocate** command has been typed on the command line. By default, **SCHANGE** is assigned to PF5 by your **PROFILE XEDIT** (see section 4.9 on page 41).

Type any valid change command, but instead of RETURN, hit PF5. Verification for each change is now done by pressing PF6, the repeat key. Hit RETURN to quit SCHANG mode.

#### 4.5 Some Examples of Selective Changes

Changes involving a non-trivial range of lines must be done in more than one command combining the change, set zone and all commands. For example, to delete all the lines in a given file containing the string "BEETHAM" in column 7 one would use the sequence:

```
SET ZONE 7 13    restricts the search area to the columns 7-13
ALL /BEETHAM/    display all lines which have the string 'BEETHAM' in the specified zone.
DEL *           delete all lines which are displayed
ALL             makes the hidden lines visible again
SET ZONE 1 *    reset the zone to the full line length
```

All these commands may be given on one line using the linend character #:

```
SET ZONE 7 13#ALL /BEETHAM/#DEL *#ALL#SET ZONE 1 *
```

Please remember to reset the zone after any operations of this kind, otherwise subsequent string searches will fail. Notice that the same rules for targets as for locate apply to ALL, i.e. one could issue a command

```
ALL/BEETHAM/ & /JONES/
```

Deleting columns in a text may be done as follows:

```
SET ARBCHAR ON S
TOP
SET ZONE 31 40#CHA /S/      /* ##SET ZONE 1 *
SET ARBCHAR OFF
```

#### 4.6 Editing Multiple Files

Xedit allows you to edit more than one file at a time. Simply enter

```
Xedit another file
```

while inside Xedit. As many files that fit into the virtual storage of your machine can be edited at one time. The status message changes to "XEDIT - 2 Files" (except if your terminal is used through LTERM), indicating how many files there are in your virtual storage. This is called a ring. If you have no status area you can find out where you are via the command

```
Query RING
```

Move in between the files in the ring by typing X (without arguments). To exit from a ring (i.e. quit all Xedit sessions at the same time), use the command

**CANCEL**

If you wish to display any of these simultaneously, you may do so with the

**SET SCREEN n [H|V]**

where H stands for horizontal and V for vertical. This allows you to split the screen into many different windows.

It is often useful to move or copy parts of one file to another. This can be done, inside or outside an Xedit ring, by the commands **PUT** and **GET**:

**PUT [target [fileid]]**  
or prefix command **PT**

issued in the sending file puts the lines from the current line to the target in storage, or puts them in the file given by fileid.

**PUTD**  
or prefix command **PD**

does the same as **PUT** but deletes them from the current file.

**GET [fileid [n [m]]]**  
or prefix command **G**

issued in the receiving file adds lines from storage (if no arguments were given) after the current line. If a fileid was given it takes m records from fileid starting at record n.

#### 4.7 Some useful features in Xedit, function keys

Xedit has several tricks built in to make life easier. The use of the following characters may save a lot of time:

- &** Any command prefixed by an ampersand will remain on the command line after it has been executed. After this simply hit enter to reissue the command.
- ?** A question mark will repeat the last command issued, and place it in the command line except if this was a PF key function. It will not be executed. The same as PF 6.
- =** An equal sign will repeat the last command executed, including PF key functions, and place it in the command line, and execute it. The same as PF 9.

You may program your function keys (PF keys) to issue various commands that you find useful by the command

**[SET] PFn [BEFORE|AFTER|IGNORE|ONLY] command**

where;



---

<b>n</b>	specifies the number of the PF key (can be 1–24, although at CERN only 1–12 are used. 13–24 are set equal to 1–12).
<b>BEFORE</b>	specifies that the PF key contents is executed before the contents of the command line.
<b>AFTER</b>	specifies that the PF key contents is executed after the contents of the command line.
<b>ONLY</b>	specifies that the PF key contents is executed and the contents of the command line is ignored.
<b>IGNORE</b>	specifies that the PF key is ignored when a command is entered on the command line.
<b>command</b>	is any valid Xedit, CP or CMS command.

The defaults set up for you by your PROFILE XEDIT (see Section 4.9 on page 41) are:

1. **BEFORE HELP MENU.** Displays a menu of the available Xedit commands.
2. **BEFORE SOS LINEADD.** Adds a new blank line after that in which the cursor is.
3. **BEFORE QUIT.** Exit the editor without saving a copy of the file back on disk.
4. **BEFORE TABKEY.** Moves the cursor to the next tab position in the file. To set tabs inside xedit, use the command
 

**SET TABS col-1 col-2 col-3...col-n**

 where col-1...col-n are the columns of the tab position. PF4 may be then used to move the cursor from one tab position to another.
5. **BEFORE SCHANG 6.** Search and Selectively CHANGE data with PF-key 6. For example, if the command **CHANGE /stringa/stringb/** is entered on the command line, by hitting PF5 instead of RETURN, Xedit will prompt to hit PF6 to execute the change. Hit RETURN to get out of SCHANG mode; hit PF5 to search for the next instance of stringa.
6. **ONLY ?.** Re-display previous subcommand, excepted if it is the content of a key function. Repeated use shows all subcommands used in the XEDIT session, or as many as can be stored. The number of commands memorized by the editor depends of their size.
7. **BEFORE BACKWARD.** Display one screen of lines back (towards the start) in the file.
8. **BEFORE FORWARD.** Display one screen of lines forward (towards the end) in the file.
9. **ONLY =.** Repeat the previous command.
10. **BEFORE RGTLEFT.** Move the display to the right-most column position of all records, or back to the left-most again.
11. **BEFORE SPLTJOIN.** Split a line in two at the cursor position, or join the next line with this one at the cursor position. The function depends on whether or not the cursor is in the middle or the end of a line.

12. **BEFORE CURSOR HOME.** Move the cursor back and forth between command line and elsewhere.

The function keys are displayed with the command

**Q PF**

#### 4.8 Creating your own Xedit Environment

Xedit allows you to customize your screen layout to your own taste. Useful commands to know are (try these commands out as you are going along by xediting your PROFILE XEDIT file):

- **on [SET] CMDline [ON|Top|Bottom].** The OFF option should normally not be entered on the command line, as it will remove your command line.
- **[SET] CURLine ON [M] [+|-] n.** Moves the current line position.
- **[SET] PREFIX ON|OFF|NULL|RIGHT**
- **[SET] NUMBER ON|OFF.** Set linenumbers in prefix area.
- **[SET] RESERVED.** Useful for creating fullscreen panels with Xedit.
- **[SET] SCALE [ON|OFF] [+|-] n**
- **[SET] TABline [ON|OFF] [M] [+|-]**

The **SYNONYM** command allows you to add your own synonyms for any Xedit command to the editor's table of synonyms. This can be achieved by using the command

```
[SET] SYNONYM newname oldname
[SET] PREFIX Synonym newname oldname
Query SYNONYM * for a list of all defined synonyms
```

The **SYNONYM** command has many more possibilities than sketched here. Please consult the Xedit Reference manual [4] for more details.

All these and other commands may be used to create your own environment via **Xedit macros**, or to create new functions that are not supplied by Xedit macros.

The simplest form of an Xedit macro is a list of Xedit commands. It must have a filetype of Xedit, otherwise it is not recognised as a valid Xedit command. They are executed in the same way as all other Xedit commands. Using the REXX command language very powerful Xedit macros can be created, some examples of which will be shown later in the course. The Xedit macro that is best known is PROFILE XEDIT.

## 4.9 The Profile Xedit

The PROFILE XEDIT macro is automatically executed every time an Xedit command (or subcommand) is issued. It can be compared to a "LOGON" file for your Xedit session. Every new user has a standard PROFILE XEDIT on his/her A-disk. If you look at your PROFILE XEDIT, you will notice that the first uncommented command (no "/\*" or "\*/" around it) is the line

```
call 'STANPROF'
```

This macro is a systems macro that defines suitable defaults for the CERN Xedit environment (including the function keys described above, and some filetype dependent settings such as tabs for FORTRAN files etc.) This macro should be called only from within your own PROFILE XEDIT and not from an Xedit session. The STANPROF XEDIT macro should be the first XEDIT subcommand issued by the PROFILE XEDIT (the reason for this is explained below). If you do NOT want a PROFILE XEDIT to be executed for a particular editing session (useful if you suspect problems with your PROFILE), you can issue the following XEDIT command:

```
XEDIT fn ft ( NOPROFILE
```

What follows the opening parenthesis is called an **option**. Please note that the closing parenthesis may be omitted (this is a general CMS feature).

It should be noted that if your PROFILE XEDIT does not call STANPROF, then you will not benefit from any changes made to the system STANPROF until you make the corresponding changes to your own profile. You must not have your own version of STANPROF XEDIT. To tailor your own profile to your personal needs you should activate the relevant commands in your copy of PROFILE XEDIT after the call STANPROF statement. For example, to use/set a **current** Xedit file, i.e. you wish Xedit to remember which file you last edited and use that as default file next time the editor is invoked without any arguments, you may do so by changing the call STANPROF line as follows:

```
call 'STANPROF' 'ARG'(1)
```

This will pass all the arguments (filename, filetype, filemode and Xedit options) to STANPROF, which then decides whether they are sufficient (in which case they set a new current file) or not (in which case the current file will be used). This must be done before the file is loaded into storage, and therefore the call to STANPROF must be the first in the PROFILE XEDIT.

Note that the filemode of the PROFILE XEDIT is "A0". The "0" means that the file will be hidden from other people that have access to your disk, i.e. they will be prevented from accidentally executing it and thus obtaining your environment.

## 4.10 Exercises for Chapter 4

Please execute the command TUTORIAL on PUBCP 201 and complete the exercises for lessons 1, 2 and 3. You will then have obtained the necessary files for the exercises for this chapter.

You already know how to create and save a file on your disk. This exercise is to gain some experience with a wider range of Xedit commands.

You will create one small file, and change another. This will involve moving around the file and changing existing data. You will also delete and perhaps add some lines to the file.

1. Tell the system that you wish to create a new file called MYTELL EXEC on your A-disk (the TUTORIAL command will prompt whether a previous MYTELL EXEC may be erased).
2. When the XEDIT screen is presented to you, indicating that this is indeed a new file, tell XEDIT that you wish to put some line into the file. Do this with the INPUT command.
3. Type the following lines, ensuring that each line starts in column 1:

```

/* MYTELL EXEC - simple exec to send messages to users */
say "Give the userid of the person you want to send the message to
=>"
pull userid
say "Give the text of the message =>"
pull text
say "How many times ? =>"
pull times
if times= "" then times=1
do times
    "TELL" userid text
end
exit

```

4. Press RETURN, and the screen changes; the last line you typed is displayed in the data display area.
5. Press RETURN again to leave input mode.
6. Type the command TOP, and press RETURN.

You can now see the complete file as you typed it except that it may be in upper case, whereas you probably entered it in lower case. Find out what your case setting is by **QUERY CASE**. If it is Upper Respect, everything you typed will be translated into uppercase. Type **SET CASE MIXED IGNORE**. If it is Mixed Ignore, type **SET CASE UPPER RESPECT**.

7. Now change any word to lower case and hit RETURN to see the effect.
8. Store a permanent copy of this file on your A-disk. Try executing the MYTELL exec to see what it does.

Next we shall work with a file called CERNVM DATA which the TUTORIAL command will have put on your A-disk.

Should you want to make a clean start at any stage with a new copy of CERNVM DATA, simply QUIT your Xedit session and re-issue the TUTORIAL command which will give you a fresh copy of CERNVM DATA and will allow you to start again. XEDIT the file CERNVM DATA.

As you can see it contains data about the various printers at CERN, in fact it is an old data file which was used by the XPRINT facility (a command which allows you to print your files). We shall bring this file more up to date.

9. **Scroll (move) your editing window one screen forward.** Use the FORWARD command, or function key 8.
10. Use any command to make line 25 the current line. The most concise method is the following example of the LOCATE command:

```
:25
```

Use the XEDIT command

```
set num on
```

if you would like XEDIT to show you the relative number of each record in the file. The number is displayed in the prefix area.

11. Move your window up to line 15.
12. Move your window up and down as you wish.

Try to use all the commands you know for this purpose (Up, Down, Next, TOP, Bottom, Forward, Backward, / etc.)

13. Position your window at the top.

Now you are going to add and delete lines, and make some changes.

14. The 8700 Xerox printer is now also useable with class A, so we shall add a line allowing this: duplicate line 3 with the prefix command " .
15. Change the class column in the first of these lines from "C" to "A".
16. The VERSATEC printer is now only accesible via the GKSVT command, so we must remove it from the data file. Delete it via the prefix command D or the DELETE command on the command line.
17. In its place, add a line for the CDC line printer:

```
R4      CDC      GEN CDC580      513  PRINT A      7      53 NULL
```

Delete all other lines that have the string CDC in the second column. Do this by setting the zone to restrict the search (SET ZONE 9 15), use ALL/CDC/ to concentrate only on these lines, then finally delete all of them (excepting the first one!) by using the prefix command dd. Then type ALL again and reset the zone.

18. Type BOT to go to the end of the file and delete the 4 lines containing the string "CYBER".
19. The 6670 laser printer in building 299 has been moved to building 513. Change 299 to 513 using the Change command.
20. Having all building numbers prefixed by a B is inconvenient. Remove the B in all building numbers, leaving the numbers in the same place so that the alignment is not upset. Do this in one command, using the ARBCHAR and by setting a ZONE. Remember the end of line character

21. In line 82, we accidentally changed SB to S. Put the B back after the S.
22. Change the filetype of this file to NEWDATA and reissue the TUTORIAL command to see if you got it right.
23. Try HELP XEDIT MENU to find out about other Xedit commands that you may find interesting.
24. Modify your PROFILE XEDIT to contain any of the Xedit commands described in Chapter 4 to adapt your Xedit environment to your own taste.

Remember that the changes will not take effect until you next enter Xedit.

## 5. Properties of your Virtual Machine

### Summary of Chapter 5.

At the end of this chapter, you should have learnt how to;

- Determine and set various attributes of a virtual machine.
- Change the disk configuration of a virtual machine.
- Handle disk files (display a list of files, erase or rename files, compare or copy files, use another person's files).
- Handle spool files (display a list of files, discard files, look at spool files, receive spool files on a minidisk, send a spool file to someone else).

### 5.1 The State of your Virtual Machine

As explained in Chapter 1, your virtual machine has been configured by CP with a number of virtual devices. To determine how your configuration looks at any given moment, use the CP command

#### CP Query ALL

Typically, the response would be as shown in Figure 22 on page 45.

```

q all
STORAGE = 06144K
CHANNELS = SEL
CONS 009 ON GRAF 00EA TERM STOP
      009 CL T NOCONT NOHOLD COPY 001 READY FORM STANDARD
      009 FOR DUPDM DIST LOCAL FLASHC 000
      009 FLASH CHAR NDFV 0 FCB
RDR 00C CL * NOCONT NOHOLD EOF READY
PUH 00D CL A NOCONT NOHOLD COPY 001 READY FORM STANDARD
      00D FOR DUPDM DIST LOCAL
PRT 00E CL A NOCONT NOHOLD COPY 001 READY FORM STANDARD
      00E FOR DUPDM DIST LOCAL FLASHC 000
      00E FLASH CHAR NDFV 0 FCB
      00E SIZE 0F 22 -4U0GM CFS NODATCK 3800-1
DASD 190 3380 UNUS02 R/O 32 CYL
DASD 197 3380 UNUS32 R/O 15 CYL
DASD 199 3380 UNPDSK R/O 400 CYL
DASD 19A 3380 UNUS42 R/O 50 CYL
DASD 19D 3380 UNUS06 R/O 27 CYL
DASD 19E 3380 UNUS05 R/O 55 CYL
P. T=0 01/0 01 09 10 32

```

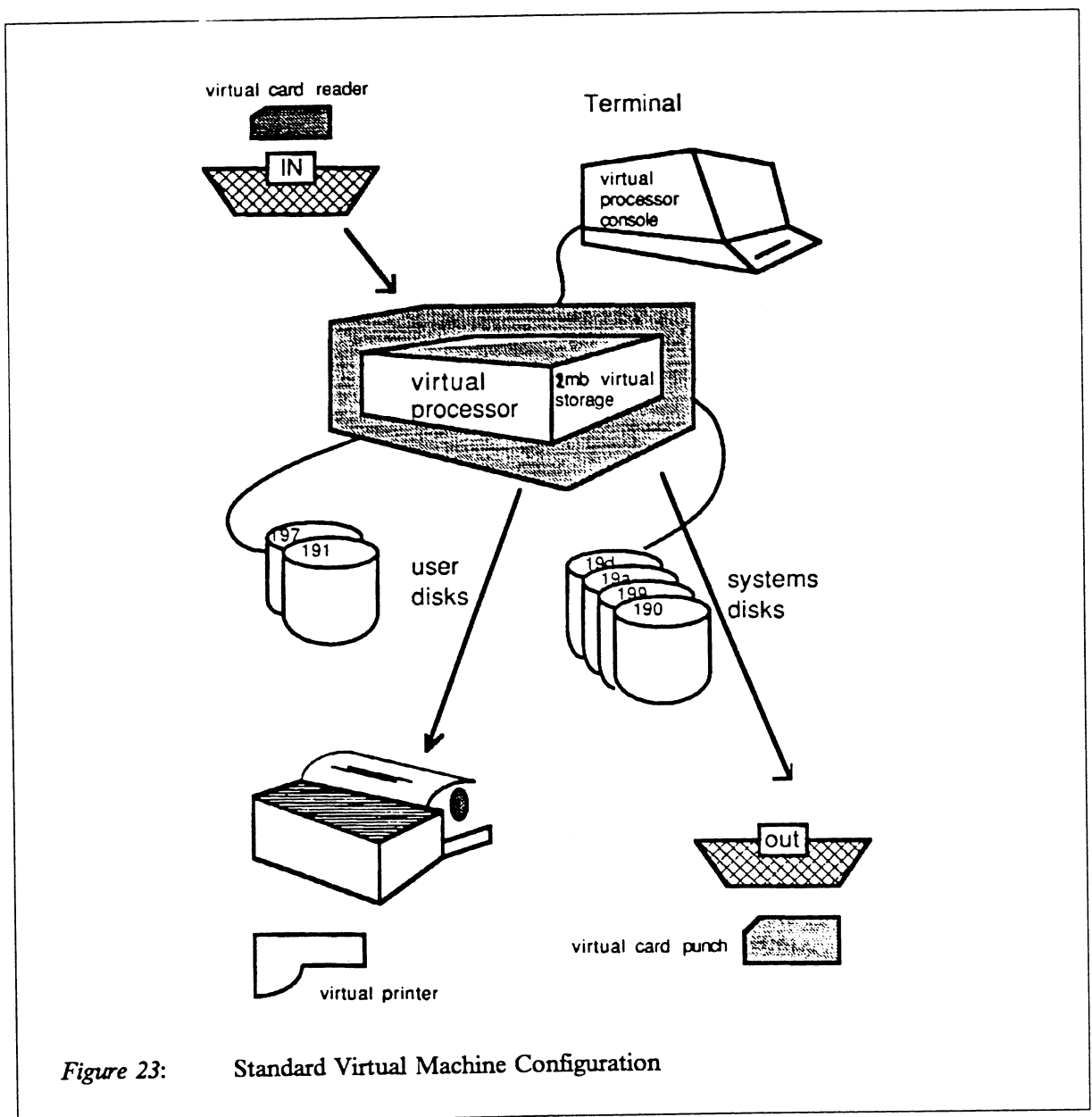
RUNNING CERNUM

Figure 22: Result of the Command Query All

Note the following points:

1. The command can be abbreviated to Q ALL.
2. The command in Figure 22 on page 46 is given in lower case. CMS translates a command to upper case before executing it.
3. CMS passed the command on to CP when it found it could not process it.

The virtual machine that produced the response to Q ALL shown in Figure 22 on page 46 has a configuration containing a virtual console, virtual printer (defined to have the characteristics of an IBM 3800 laser printer), a virtual card reader, a virtual card punch and various user and system minidisks. It can be graphically depicted as shown in Figure 23 on page 46.





Normally it is not necessary to know all the information given by Q ALL. Various other QUERY commands give a more direct response, such as

### CP Query STORAGE

This command tells you the size of your virtual storage. At CERN, by default, everyone gets a virtual storage of 2 Megabytes. Sometimes this is not enough, for example, when trying to edit a large file. To obtain more virtual storage (up to 6 Mb), use the CP command

### CP DEFINE STORAGE nM

where n is anything up to 8<sup>7</sup>. One cannot enter this command inside the CMS environment, i.e. after this command has been issued CMS must be reloaded and restarted:

### CP IPL CMS

This command does not change the size of storage forever; after logging out, the storage is reset to the figure in the machines directory entry. To change the directory entry, use the command

### DIRM STORAGE nM

This command will not change the storage immediately; to see its effect one has to log on again. Other useful query commands are (most of them are self-explanatory):

#### CP Query TIME

Query CMSLEVEL (What CMS version am I running?)

CP Query NAMES (To find out who is logged on, also try USERS)

## 5.2 Minidisks

In CMS, files are organised by disk. All new machines have a 2 cylinder A-disk. It is a good idea to allocate a minidisk to each project (or large program) that you may have, for example, one for workspace (normally the A-disk), one for SCRIPT files, another for FORTRAN files, yet another for EXEC files and system utilities and so on. New mini-disks may be added to a virtual machine by the **space administrator** of the group to which the machine is accounted. To find out who he/she is, use the command

### SPACE ADMIN QUERY

All requests for extra (or larger) mini-disks or should be addressed to the space administrator.

A minidisk can be compared to a drawer in your desk. Your desk has several drawers, each of which you use to save related objects. In the same way that no object can be larger than the drawer, it is impossible to create a file that is larger than the size of the minidisk.

The space on the disk is divided into blocks, usually of 2K (2048) bytes each. The minimum size of a file is 1 block. This blocksize can be selected when the disk is initialised, or **formatted**. Clearly it is less wasteful to format the disks with a small blocksize. However, a large blocksize is more efficient from the I/O point of view, provided your files are large and dense. You may find information about

---

<sup>7</sup> Requests for more storage should be addressed to the UCO.

the disk configuration in a virtual machine by using the

### Query DISK

command. (See Figure 24 on page 48).

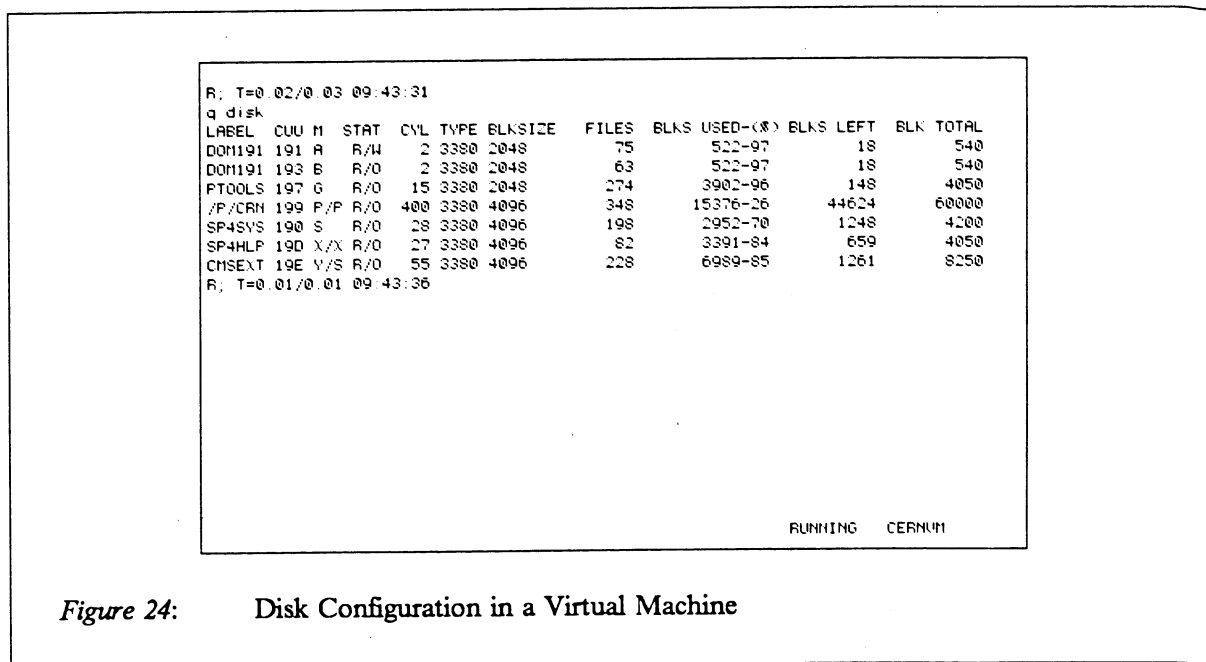


Figure 24: Disk Configuration in a Virtual Machine

Note the following:

1. Query DISK is a CMS command. Compare this to the output of the CP Q ALL command.
2. The list contains respectively:
  - a. The **disk label**. This is a 6 character identifier of the disk. Some common labels are P-CERN, PTOOLS, SP4HLP etc. The label may be changed using the  
**format virtual-address access-letter ( label**  
**command.**
  - b. The **virtual address** of the minidisk in the virtual machine's system. This can be any valid hexadecimal number of three digits (provided it is not already in use).
  - c. The **access letter** of the disk. The access letter determines the filemode of the files on that disk. At CERN a (fairly random) convention was adopted to relate the access letters to disk addresses. The relation is A:191, B:193, C:194, D:192, etc. The configuration in Figure 24 contains some non standard addresses.
  - d. The **access mode**. R/W means that you are allowed to read from and write to the disk, whereas R/O stands for read only. See appendix B on page 108 for more details about mini-

Some commands are an exception to this rule in that they only restrict themselves to the A disk. Some of them are: PRINT, ERASE TYPE, SYNONYM, FILELIST etc. To force the standard search order in this case, give a "\*" (asterisk) as filemode.

### 5.5.3 Useful commands when inside Filelist

Several actions can be carried out on files. They can all be done very conveniently from inside the filelist environment, as well as from outside filelist. Inside filelist, the command is placed in the command area (if there is not enough space in the area below CMD, the text on the line may be typed over; please note the "/" which replaces, when necessary in the syntax command, the fileid corresponding to the line where the command is written.

<b>RENAME / newfn newft newfm</b>	(rename a file to newfn newft newfm) This command can only change the digit but not the letter in the filemode.
<b>COPYFILE / newfn newft newfm</b>	(copy a file to newfn newft newfm)
<b>ERASE</b>	(erase a file from disk)
<b>XCOMPARE / fn2 ft2 fm2</b>	(compare a file to fn2 ft2 fm2)
<b>XPRINT /</b>	(print a file)
<b>SENDFILE / userid</b>	(send a copy of the file to userid)
<b>BROWSE /</b>	(look at a file)

To issue these commands outside filelist, replace the "/" by the fileid. Please note there must be a blank between the command name, the slash and what follows it. CMS uses the blank as a separator, and if it was not given, the rest would be assumed to be part of the command. All these commands have useful options, which are placed behind a bracket "(" (see the CMS User's Guide [6]).

The commands that are prefixed by an "X" are usually locally written facilities which have logically equivalent IBM counterparts. For example COMPARE and PRINT can still be used as provided by IBM.

### 5.5.4 Use of '=' sign inside commands

Commands that operate on two files such as RENAME or COPYFILE, usually allow an equal sign to be given for any part of the second fileid, indicating that the filename, type or mode is equal to that of the first file:

```
COPYFILE PROFILE EXEC A NEWPROF = =
```

would create a file NEWPROF EXEC A. The = may also be used inside FILELIST or RDRLIST. There it may also be used to repeat previous command (see Figure 26 on page 54).

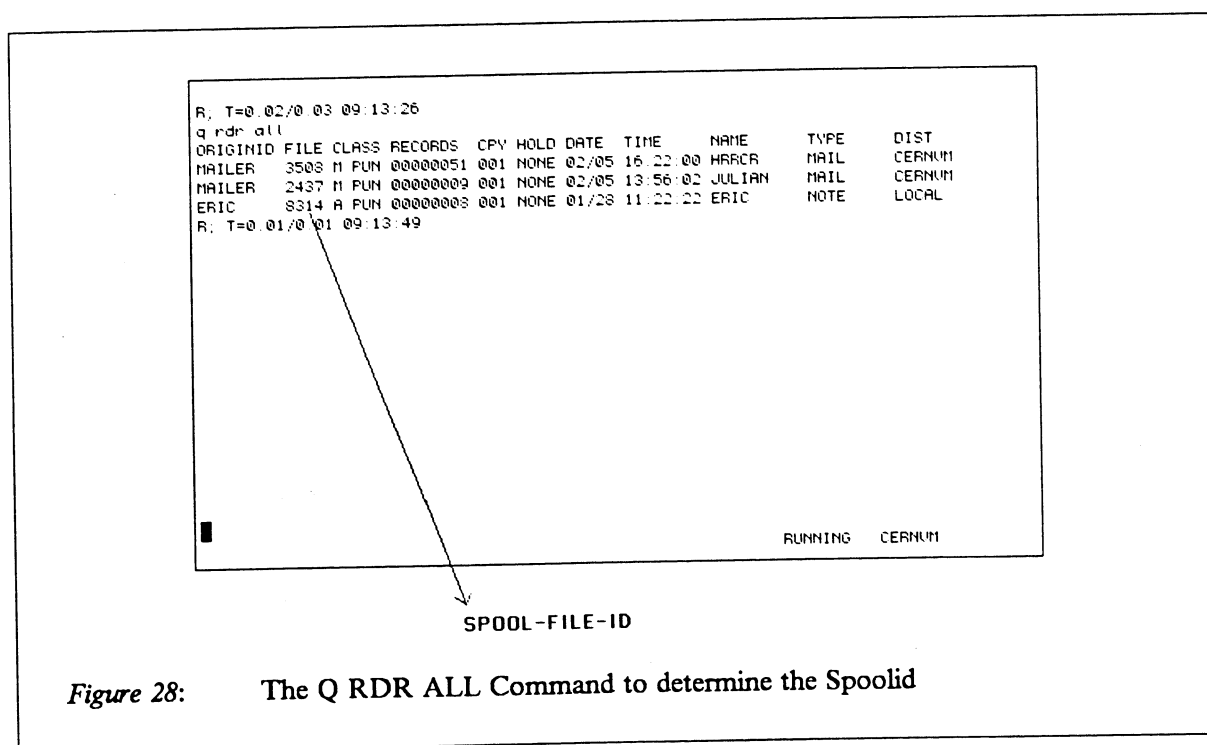
## 5.6 Spool Files

Most of the communication between two virtual machines proceeds via the **spool**. The virtual reader, virtual printer and virtual punch all use the spool. It is a temporary area which can be compared to the **FETCH QUEUE** on Wylbur (MVS) except that it has many more possibilities. The spool is like an in/out tray (see Figure 23 on page 47); to get files from the spool (tray) onto your disk (desk), you have to give a specific command called **receive** (see next page). All mail and output from the BATCH machine will be directed to you reader, as well as anything you may send there yourself. For example, the **sendfile** command, e.g.

```
SENDFILE fn ft fm [TO] userid
```

will result in a copy of the file **fn ft fm** being sent to the virtual reader of **userid** (which could be your own). All messages sent by **note** and **mail** also end up there (see sect. 8.1.3 on page 80).

Each file in the reader, punch or printer has a **spoolid** which identifies the file for later use; it is the equivalent of the **fileid** for files. The spoolid is a 4 figure number and can be found using the commands **Query RDR ALL**, **Query PRT ALL** or **Query PUN ALL**. The result of the command **Query RDR ALL** is shown in Figure 28 on page 56. The spoolid is the number displayed in the second column.



To look at what files are in your reader, use the command

### RDRLIST

(Figure 29). RDRLIST displays the list of files in the reader. Each is similar to an entry in **filelist**, with the following exceptions:

```

DUPDOM  RDRLIST  A0  U 108  Trunc=108  Size=3  Line=1  Col=1  Att=0
Cmd  Filename  Filetype  Class  User  at  Node  Hold  Records  Date  Time
ERIC  NOTE  PUN  A  ERIC  CERNUN  NONE  8  01/28  11:22:22
JULIAM  MAIL  PUN  H  JULIAM  CERNUN  NONE  9  02/05  13:56:02
HRRCR  MAIL  PUN  H  HRRCR  CERNUN  NONE  51  02/05  16:22:00

1= Help      2= Refresh  3= Quit      4= Sort(type) 5= Sort(date) 6= Sort(user)
7= Backward  8= Forward  9= Receive   10= Browse   11= Peek     12= Cursor

====>
XEDIT 1 File

```

Figure 29: The RDRLIST display

1. **Type of the file.** Normally PUN (i.e. the file originated from a virtual punch of another machine) or PRT (i.e. the file originated from a virtual printer of another machine, e.g. files that are sent to you from Wylbur go via the virtual printer of RSCS).
2. **Class of the file.** Normally A. Should be compatible with to the class of your virtual reader. If the reader has class "\*" all classes can be read (see Figure 22 on page 46). Some common classes are A (files which have been sendfiled), M (mail from Wylbur or the mailer), Y (output from the batch machine).
3. **User.** The userid from which the file originated.
4. **At node.** The node where the file originated from.
5. **HOLD.** Normally spool files will not be held, which is seen from the status NONE. When a file is in a hold status, the file remains in the reader until the hold status is removed and the file is read, or until the file has been discarded.
6. **Records.** The number of records in the file. Files in the reader with a length greater than 1000 records are purged by the system after 96 hours; those with a length less than 1000 are purged after 30 days. All files in the printer or punch are purged after 96 hours.

Similar to the filelist environment, the rdrlist environment allows a list of convenient commands to be executed within it (if there is not enough space in the area below CMD, the text of the line may be typed over):

**RECEIVE fn ft fm** (receive the spool file on disk as fn ft fm, remove it from the reader)

**DISCARD** (discard the spoolfile from the reader)

**PEEK** (look at the spoolfile); if the file is too large use:

**BROWSE .** (does not reformat; result may look strange); or:

**SCAN .**

**XPRINT . /** (print the spoolfile and remove it from the reader)

Please note the "." following BROWSE, SCAN and XPRINT. This is a special syntax to tell these commands that they are working on spoolfiles rather than disk files. There are blanks between the commandname, the., the slash and the rest of the command. The blank acts as a separator for CMS; if it was given, what follows it would be assumed to be part of the name of the command.

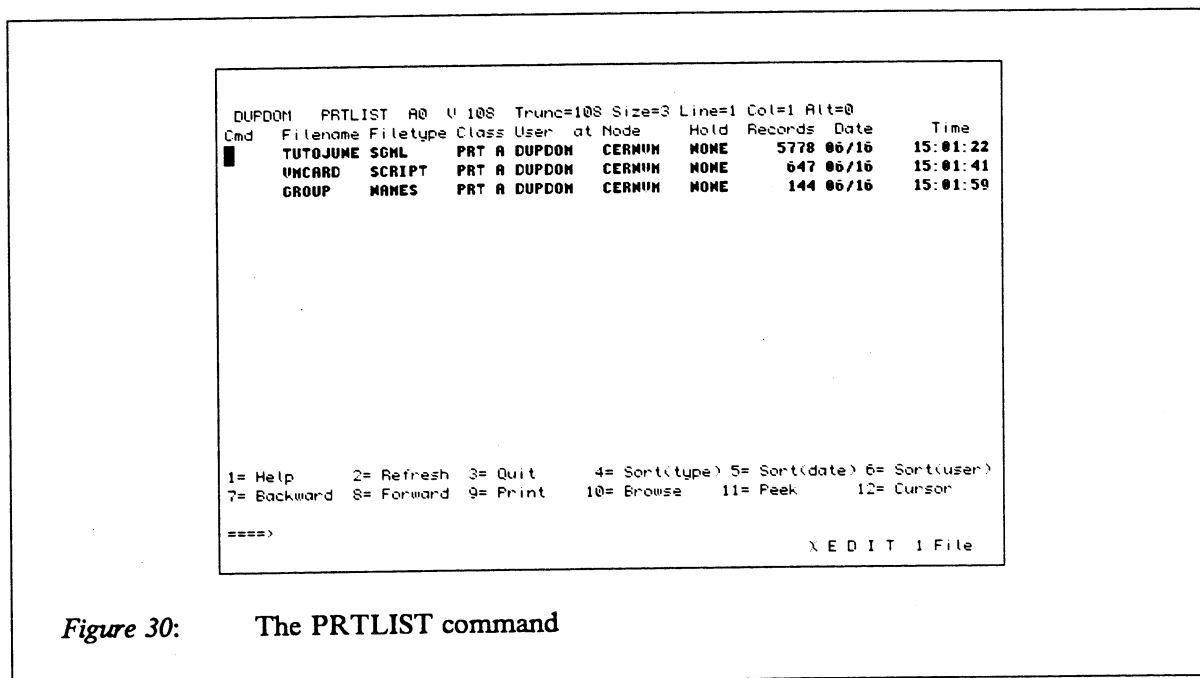
So far we have discussed the virtual reader only. To find out what is in the virtual printer and the virtual punch respectively the commands

### QUERY PRT ALL or QUERY PUN ALL

should be used. However, since we have no real printers or punches connected to CMS at CERN, you will seldom have to use your printer or punch. If there is something in it, transfer it to your reader with the CP command. (Please note it is impossible to transfer prt files with a width > 80 chars to the PUNCH):

### CP TRANSFER PRINTER|PUNCH spoolid TO \* RDR

To look at files in your virtual reader, the file PRLIST may also be used. This command produces a display which is similar to that of RDRLIST. (see Figure 30 on page 58.)



### 5.6.1 Spooling of virtual devices

The virtual spooling devices may be directed elsewhere with the **SPOOL** command. For details the CP Command Reference Manual should be consulted [7], but we note one useful application here. The **SPOOL** command may also be used to start and stop the spooling of virtual console input/output. Using this facility a console-log may be obtained, containing a record of all terminal transactions. This is very useful when a problem in the system occurs. To achieve the spooling of the virtual console, the following line should be added to the PROFILE EXEC (or executed on the commandline):

**CP SPOOL CONSOLE START TO \***

Here \* means to the machine itself; however one may equally substitute the name of any other virtual machine. The command

**CP SPOOL CONSOLE CLOSE STOP**

will halt the spooling and send a file to the machine's reader containing the console-log (see Figure 31 on page 59).

```

R; T=0.17/0.33 08:05:31
CP SPOOL CONSOLE START TO *
R; T=0.01/0.01 08:06:01
Q DISK
LAEEL CUU M STAT CYL TYPE BLKSIZE FILES BLKS USED-<X> BLKS LEFT BLK TOTAL
DOM191 191 A R/W 10 3380 2048 104 2285-85 415 2700
DOM193 193 B R/W 15 3380 4096 208 1985-88 265 2250
DOM192 192 D R/W 8 3380 2048 112 1487-69 673 2160
PTOOLS 197 G R/O 15 3380 2048 149 2863-71 1187 4050
T-SOFT 1A6 D R/O 50 3380 4096 134 5698-76 1802 7500
/P/CAN 199 P/P R/O 400 3380 4096 393 19540-33 40460 60000
SP4SYS 190 S R/O 28 3 10 4096 199 2955-70 1245 4200
SP4HLP 190 X/X R/O 27 3380 4096 82 3419-84 631 4050
CHSEXT 19E V/S R/O 55 3380 4096 228 7448-90 802 8250
R; T=0.01/0.01 08:06:10
CP SPOOL CONSOLE CLOSE
CON FILE 0488 TO DUPDOM COPY 001 NOHOLD
R; T=0.01/0.01 08:06:39

```

RUNNING CERNUM

*Figure 31:* Issuing the command **CP SPOOL CONSOLE CLOSE STOP**

This file may be viewed with **PEEK** (but better with **BROWSE**), or if requested by someone (e.g. the **UCO**), transferred to their reader by the command:

**CP TRANSFER READER spoolid TO userid RDR**

A typical console log is displayed in Figure 32.

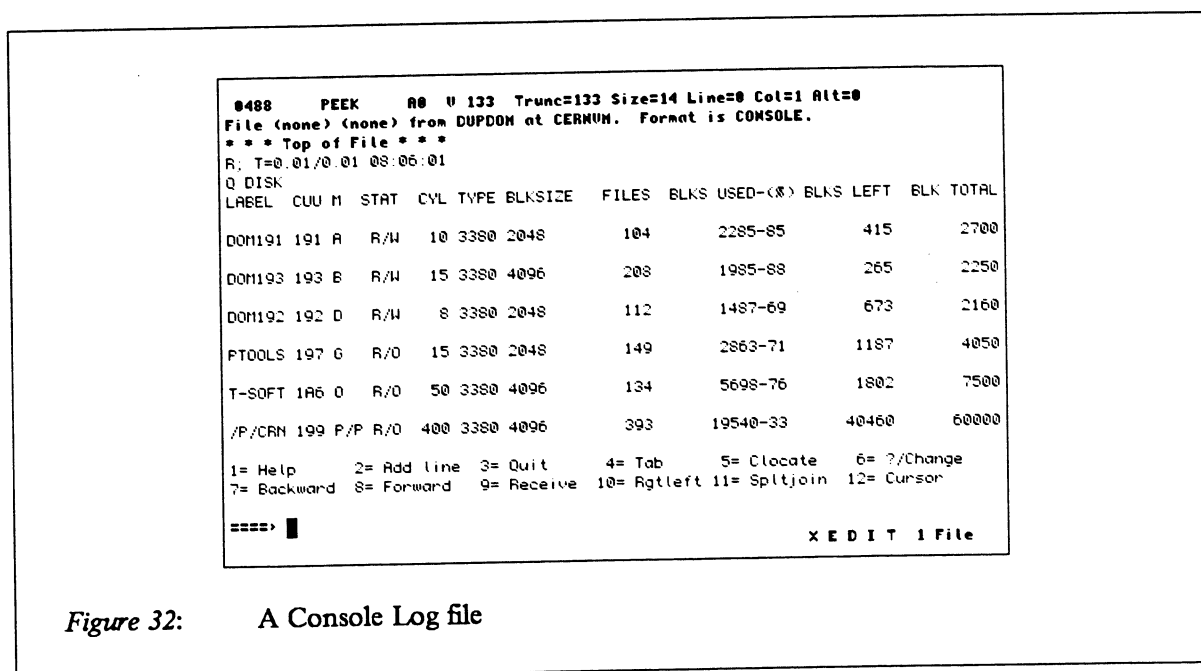


Figure 32: A Console Log file

## 5.7 Exercises for Chapter 5

The first exercise is designed to teach you how to determine various properties of your virtual machine. There is no interactive tutorial for this Chapter, simply try to find the commands and execute them to complete these exercises.

1. About your Virtual Machine.
  - a. How much Virtual Storage do you have ? Try editing the file TEST FORTRAN on the disk PUBCP 201. Notice that the BROWSE command will let you view the file although it is too big for your virtual storage.
  - b. Increase the Virtual Storage of your machine to the maximum the system will allow. Now try editing the file TEST FORTRAN again.
  - c. What class is associated with your virtual punch, your virtual reader and your virtual printer ?
  - d. How many minidisks are defined in your machine ?
2. About your files.
  - a. How many CMS files are accessed in your machine (in total)?
  - b. What is the fileid of the largest file?
  - c. How much space is there left on your A-disk? Is there enough space to copy the file TEST FORTRAN on PUBCP 201 ? If not use the GIME command to obtain a temporary disk of 5 cylinders and copy TEST FORTRAN onto this disk.



- d. How many EXEC files are there on the P-disk ?
  - e. How large are the blocks on your A-disk and on your P-disk?
  - f. Are there any files on the P-disk with mode number 1 ?
  - g. How many files are there on the "X" disk ?
3. Some exercises on manipulating files.
- a. How many files of filetype SCRIPT are there on the P-disk?
  - b. Type the first 5 lines of CRNBASIC SCRIPT P with the TYPE command. Use the CMS HELP command to discover the details of the TYPE command.
  - c. Copy the file CRNLET SCRIPT P to your temporary disk. Repeat FILELIST to confirm that it is on that disk.
  - d. Rename the file CRNLET SCRIPT to MYLET SCRIPT. Now extend this file by using GET to append the file CRNMEMO SCRIPT P to it.
  - e. Execute the command FILELIST \*LET SCRIPT \* . Notice the respective size of CRNLET SCRIPT P and MYLET SCRIPT.
  - f. Sorting a file can be done by using the CMS SORT command or the Xedit SORT command. Type FIND SORT to study both commands.
  - g. Sort the file CRNLET SCRIPT on the first 2 columns using the CMS SORT command. To be able to do this, CRNLET SCRIPT must be changed into a RECFM F file. Study the COPYFILE command to determine how this may be achieved. Do the same for MYLET SCRIPT using the Xedit SORT command (it is not necessary to change the record format).
  - h. Discover how to send output from XCOMPARE to a file on your A disk or your temporary disk rather than the terminal. Compare CRNLET SCRIPT and MYLET SCRIPT using this option. If you direct output to your terminal and you wish to stop it try the command HT. Tip: you may find the command SWAPMODE A T useful.
  - i. Finally execute FILELIST again. Look at the files on your temporary disk. Erase CRNLET SCRIPT, MYLET SCRIPT and any other file you may have created during this exercise. If you no longer need the temporary disk, use the DROP command to hand it back to the system.
4. Some exercises using your virtual reader.
- a. Send the file PROFILE EXEC on your A-disk to yourself by using the SENDFILE command.
  - b. Use the RDRLIST command to see what is in your reader.
  - c. Try PEEK or BROWSE . to look at PROFILE EXEC (notice the '.').
  - d. Now type DISCARD to get rid of the file in your reader.

## 6. HELP

### Summary of Chapter 6.

At the end of this chapter, you should have learnt how to obtain HELP.

- An extensive on-line HELP facility forms part of the VM System. Each CP, CMS, Xedit or REXX command has a "HELP" file associated with it which may be viewed by the HELP command. The HELP file will contain the same description of the command as may be found in the IBM manuals. An example of a HELP file is shown in Figure 33 on page 62.

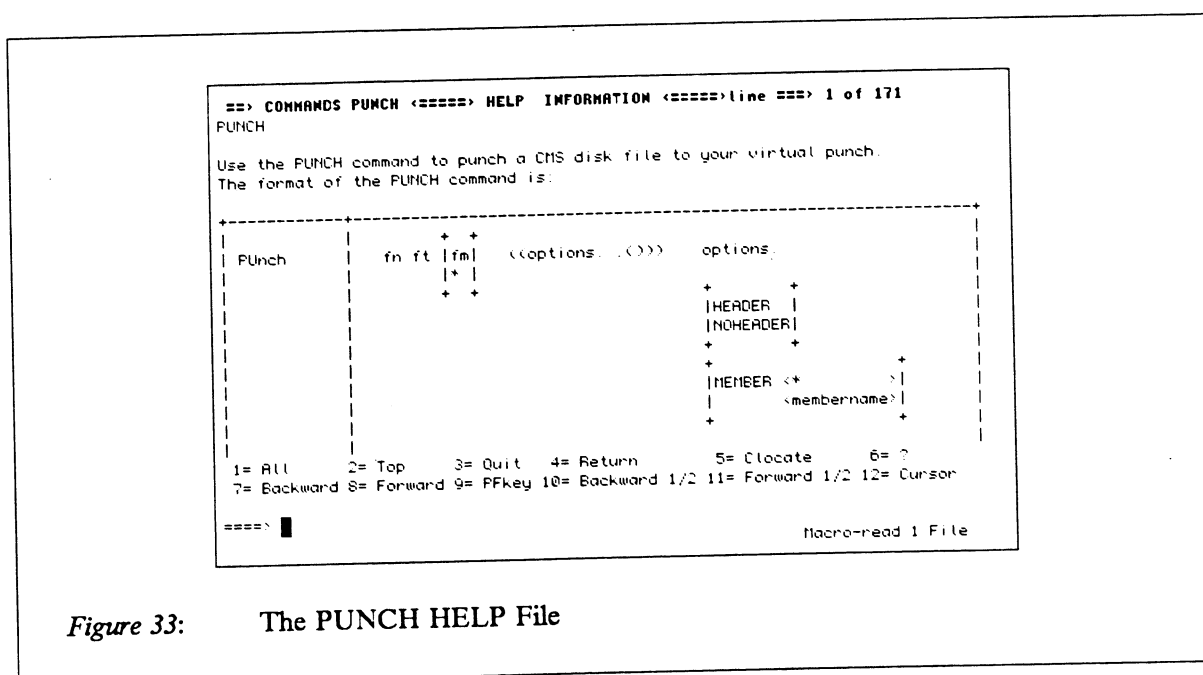


Figure 33: The PUNCH HELP File

Unfortunately the syntax description of the various commands is not always consistent. However, the following points may be noted:

1. Optional arguments are given between symmetric pairs of brackets: "(" and ")", "[" and "]", "|" and "|" (with '+'s above and below), "<" and ">".
2. The "|" is sometimes used as "or".
3. Options are given following a parenthesis "(". The closing parenthesis may be omitted.

## 6.1 FIND

Unfortunately the HELP files are organised in such a way that it is difficult to find HELP on a topic if the name of the command or the VM component it belongs to (CP, CMS, REXX, Xedit, etc.) is not known. For this reason a facility called **FIND** was developed locally at CERN. The FIND command works entirely via keywords and searches horizontally through all components and tasks, displaying a menu of all instances where the given keyword was found:

**FIND KEYWORD1 KEYWORD2 KEYWORD3 .....**

For example, the command

**FIND RDR**

results in the display shown in Figure 34 on page 63. Naturally, the more keywords are given, the narrower the search will be, resulting in less irrelevant information being displayed.

```

====> PUBLIC disks: 15 items

----- CHS = CHS commands, execs and modules -----
EXECIO  reads/writes from disk files, program stack or spool devices
PEEK    displays virtual reader files without writing them to disk
RDR     determines characteristics of next file in your reader
RECEIVE moves a spool file in your virtual reader to your disk
----- CP = CP commands -----
CHange  alters external attributes of closed spool file(s)
DEFine  alters your virtual machine configuration
INDicate displays use of system resources
ORDer   changes order of closed spool files in your spool queues
PURge   removes your own closed spool files from the spool queues
Query   to query system status and machine configuration
SPool   controls virtual console, readers, printers or punches
TRANsfer transfers your closed spool files to specified user or queue
----- REXX = REXX commands and functions -----
CPOP    restores previously saved status
CPUSH   saves status for future restore
QSPool  returns various information on a spool file

Press P18 if you want to search the USER disk.
Pf1=Display 3=Quit 7=      8=Forward 11=Display 12=Print etc.

```

Figure 34: The result of FIND RDR

Please note that inside the Xedit environment you should prefix your FIND command with CMS, to avoid confusion with the Xedit FIND command. FIND is the recommended method to obtain HELP if you are trying to obtain information about something you don't know. If you know the name of the command and the component it belongs to, the HELP command itself may be quicker (see the following subsections).

## 6.2 HELP MENUS

Help is ordered in a many level tree-like structure. At the top level, one has the component name and at the lower, one has the individual items. Information can be found on any of the VM components CP, CMS, XEDIT, REXX etc. For a list of all these components or categories, type

### HELP MENU

(see Figure 35 on page 64).

```

==> MENUS MENU <===== HELP INFORMATION <=====> line ==> 1 of 11

A file may be selected for viewing by placing the cursor under any character
of the file wanted and pressing the ENTER key or the PF 1 key. A MENU file
is indicated when a name is preceded by an asterisk (*). For a description
of the operands and options type HELP HELP.

*BATCH *COMMANDS *EDIT *GROUP *PREFIX *RSCSV2 *UMARCH
*BROWSE *CP *EXEC *IPCS *REXX *SET *MVL
*CMS *DEBUG *EXEC2 *MAILBOOK *REXXLOC *SPACE *XEDIT
*COMMAND *DUMPSCAN *GCS
* * * End of File * * *

1= Help 2= Top 3= Quit 4= Return 5= Clocate 6= ?
7= Backward 8= Forward 9= PFkey 10= Backward 1/2 11= Forward 1/2 12= Cursor

====> █ Macro-read 1 File

```

Figure 35: The list of categories for HELP

Information on all commands belonging to any of these components can then be found by typing

### HELP component MENU

For example, HELP CMS MENU displays a menu as shown in Figure 36. Help on any topic can subsequently be found directly via the command

### HELP componentname itemname

This approach is fine, as long as one has enough time to search through the components to find the relevant item, or if one knows the component beforehand.

```

===> CMS MENU <===== HELP INFORMATION <=====> line ===> 1 of 43

A file may be selected for viewing by placing the cursor under any character
of the file wanted and pressing the ENTER key or the PF 1 key. A MENU file
is indicated when a name is preceded by an asterisk (*). For a description
of the operands and options type HELP HELP.

*DEBUG  DEMSE  FIMIS  INSTFPP2  OSCOPY  SENITYPE  TS
*EDIT  DESBUF  FLIST  IUCUTRAP  OSLIST  SENDfile  TUSCREEN
*EXEC  DIRhaint  FORMAT  Labeldef  OSPREFIX  SENTRIES  TXTlib
*EXEC2  DISK  FORTUS  LFILE  OSRUN  SET  Type
*GROUP  DISKID  FTP  LISTDS  OSVOL  SETPRT  Update
*REXX  DISKLIST  FFTP  Listfile  PAD  SETSTOR  UPNEWS
*SPACE  DISKMAP  FTPQ  LISTIO  PEEK  SETUP  USE
*UMARCH  DISKTAPE  GENDIRT  LKED  PREPFIND  SHOWCAT  USERS
*XEDIT  DLBL  GENERATE  LOAD  PRINT  SHOHIME  UFORT
ACcess  DOCFIND  Genmod  LOADLIB  PROFILE  SNTMAP  UMARCH
ACE  DOCSEND  GETFILE  LOADMod  PSERV  SO  UMBLIST
1= Help 2= Top 3= Quit 4= Return 5= Clocate 6= ?
7= Backward 8= Forward 9= PFkey 10= Backward 1/2 11= Forward 1/2 12= Cursor

===== > █
Macro-read 1 File

```

Figure 36: The list of HELP-items in component CMS

## 6.3 HELP TASKS

Fortunately, the HELP files are also organised in a more "horizontal" fashion, according to tasks. To see which tasks are available, use the command

### HELP TASKS

Please notice that this command gives a different response depending on your environment. See Figure 37 for HELP TASKS issued in the CMS and the Xedit environment.

Selecting one of the topics displayed will result in a HELP file being shown or a HELP component MENU.

## 6.4 Error Messages

HELP may also give you more detailed explanations of system errors that sometimes occur. For example, if you obtain the message

```
DMSSVT120S INPUT ERROR 001 ON FT11F001
```

more information can be obtained by typing

### HELP DMSSVT120S

You will be presented with a helpfile containing more information on this error-code. Notice that the letters 'SVT' may be omitted when giving the HELP command.

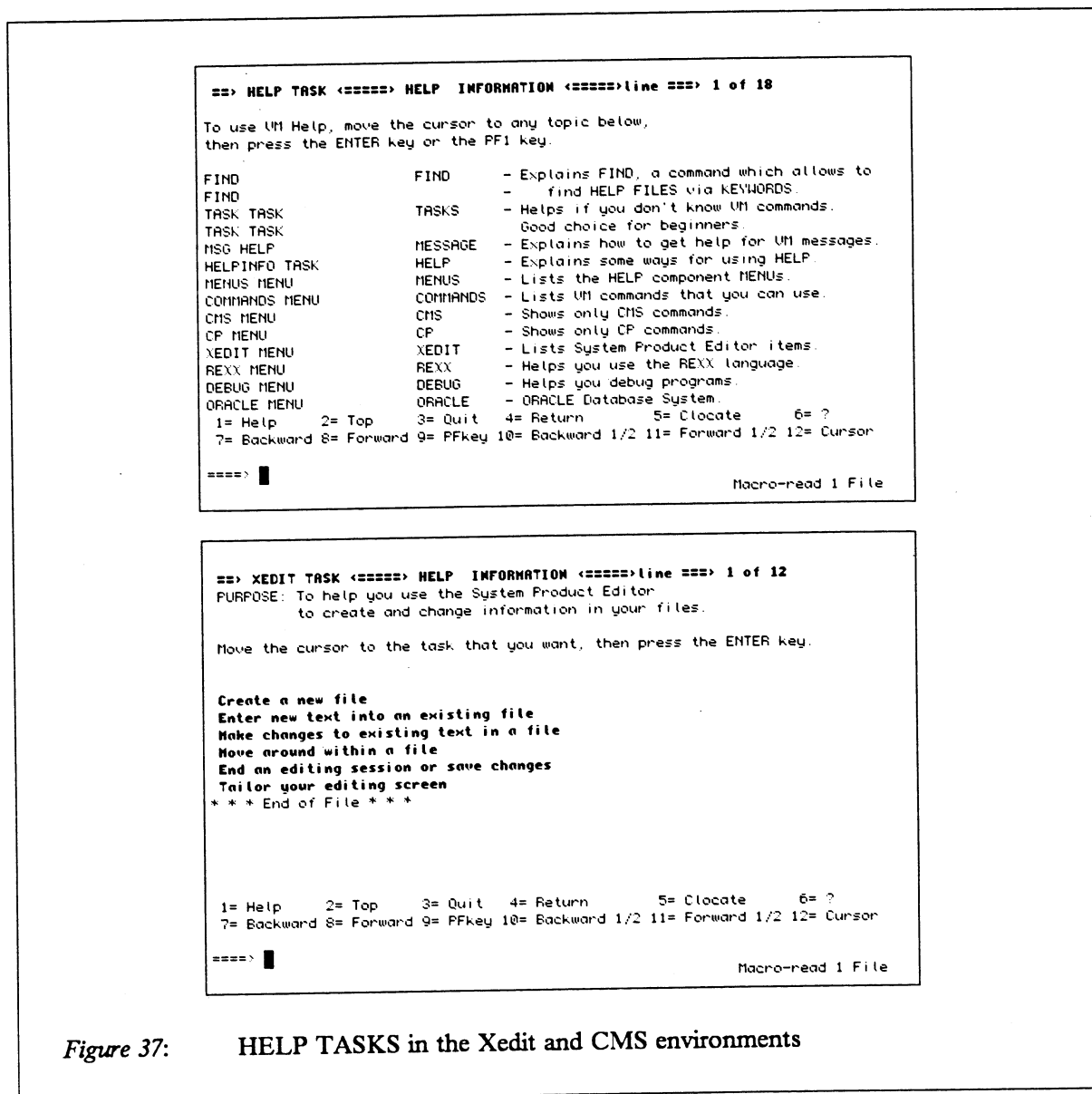


Figure 37: HELP TASKS in the Xedit and CMS environments

## 6.5 Exercises for Chapter 6

1. What are the options on the XPRINT command ?
2. Is there a DATA option on the FILELIST command ?
3. Can COPYFILE deal with more than one input file ?
4. Can you get help for the LOGON command ? Which component does it belong to ?
5. What must you type to get help with this error message.

DMSERD107S DISK 'A(0191)' IF FULL

6. What response do you get when you try it?
7. What is the command called with which one can change the logical linend character "#"?

## 7. Adapting CMS to Your Taste

### Summary of Chapter 7.

At the end of this chapter, you should have learnt how to;

- Set and use synonyms for CMS commands.
- Set and use PF keys.
- Change some general CMS settings.
- Understand and create EXEC files; in particular the PROFILE EXEC.
- Set and change defaults for your environment.

### 7.1 Synonyms for CMS commands

It is possible to define a synonym for most CMS commands. This is particularly useful for creating a 'personalised' environment. Synonyms and abbreviations can be added to the CMS table of synonyms in the same way as this can be done inside Xedit. They can then be used with, or in place of, the standard CMS commands and abbreviations.

Adding synonyms to the CMS synonyms is done by placing them in a file, which must have file-type **SYNONYM**. The filename can be any valid filename, for example, **MYSYN**. Each record in this file must contain the following information:

**original-name your-synonym length-of-minimal-abbreviation**

The number of blanks between these fields is immaterial. For example, to define a synonym called **GRABDISK** for the command **GIME** which should have a minimal abbreviation **GRA**, a line

**GIME GRABDISK 3**

should be added to the file **MYSYN SYNONYM**. Please note everything should be in upper case (watch a possible case **M I** setting in your **PROFILE XEDIT** !). To make this synonym accessible to CMS, use the **SYNONYM** command:

**SYNONYM fn [SYNONYM [fm|A]] [ (options )**

In the above example, **fn** should be replaced with **MYSYN**. When defining synonyms, one should be aware of the following:

1. Synonyms cannot be defined for CP commands.
2. Synonyms can be created for EXEC files. However, if a synonym is created for a CMS Command which has the same name as an EXEC file the exec file is used in preference to the synonymed CMS command.
3. The command



### SET ABBREV OFF

stops the system using standard system abbreviations (including your own which may have been defined by the SYNONYM command). You must then specify the full name of the command.

4. To determine which synonyms and abbreviations are in force at any given moment, use the

### Q SYNONYM ALL

command.

## 7.2 Programmable Function Keys (PF-keys)

To program the function keys for use inside the CMS environment use the CP command:

**SET PF<sub>n</sub> [IMMED|DELAYED] string**

where;

- *n* is the number of the function key (1-24; normally keys 13-24 are set equal to 1-12).
- **IMMED** specifies the function is to be executed immediately after pressing the key.
- **DELAYED** (the default) specifies that the execution is delayed. The function is displayed on the command line and only executed after **RETURN** was pressed.
- *String* is any valid CP or CMS command or set of commands.

To determine how the function keys are set, issue the command

### Query PF ALL

The PF keys are set for the CMS environment by the PROFILE EXEC (see 7.4.1 on page 73 ;CP itself does not have any default settings). At CERN, the settings are:

1. **IMMED HELP CMS MENU**. Displays a menu of available CMS commands.
2. **IMMED BROWSE**. Go into **BROWSE**.
3. **IMMED CP QUERY READER ALL \***. Displays a list of spoolfiles currently in the virtual reader together with their attributes.
4. **DELAY XEDIT**. Puts the **Xedit** command on the commandline.
5. **IMMED RDRLIST**. Executes the **rdrlist** command.
6. **RETRIEVE**. Redisplays the most current input line from the input area, allowing it to be modified or repeated. Pressing **PF** again displays the previous input line and so on, until the oldest input line has been reached, in which case it returns to the most current line. The number of lines kept depends on their length, but at least one line is kept. The buffer will be cleared after **DIS-Connect**, **GONE**, or **LOGging off**.

- 
7. IMMED QUERY DISK. Displays which disks are configured in the virtual machine.
  8. IMMED QUERY SEARCH. Displays the CMS search order for commands in the machine.
  9. IMMED CP QUERY PF. Displays a list of the current function key settings.
  10. DELAY FILELIST \* \* A. Places FILELIST \* \* A on the commandline, allowing change of mode or execution.
  11. DELAY XEDIT. Identical to PF4.
  12. IMMED FILELIST. Execute the FILELIST command.

Many environments have their own function key settings, such as CMS, Xedit, Filelist, Readerlist etc. The keys will be set to functions which are useful inside these environments. Whenever an environment is left, the function keys will be reset to whatever they were before it was entered (see Figure 38). Notice that PF1 (HELP), PF3 (QUIT), PF7 (BACKWARD) and PF8 (FORWARD) are the same for Xedit, Browse, Scan, Filelist and Rdrlist.

### 7.3 Some general CMS settings

A number of general CMS settings is available to tailor the response of various commands and to change your screen to your taste. Some of them are:

1. *SET RDYMSG:*

You may shorten the CMS ready-message from

```
R; T=0.12/0.15 16:27:10
```

to

```
R;
```

with the command

```
SET RDYMSG SMSG
```

Use LMSG to obtain the long ready-message.

2. *EMSG:*

The display of error messages can be controlled by the command

```
CP SET EMSG [TEXT|ON|OFF|CODE]
```

where;

TEXT	means display only the message text
ON	means display the text and the code
OFF	means display nothing

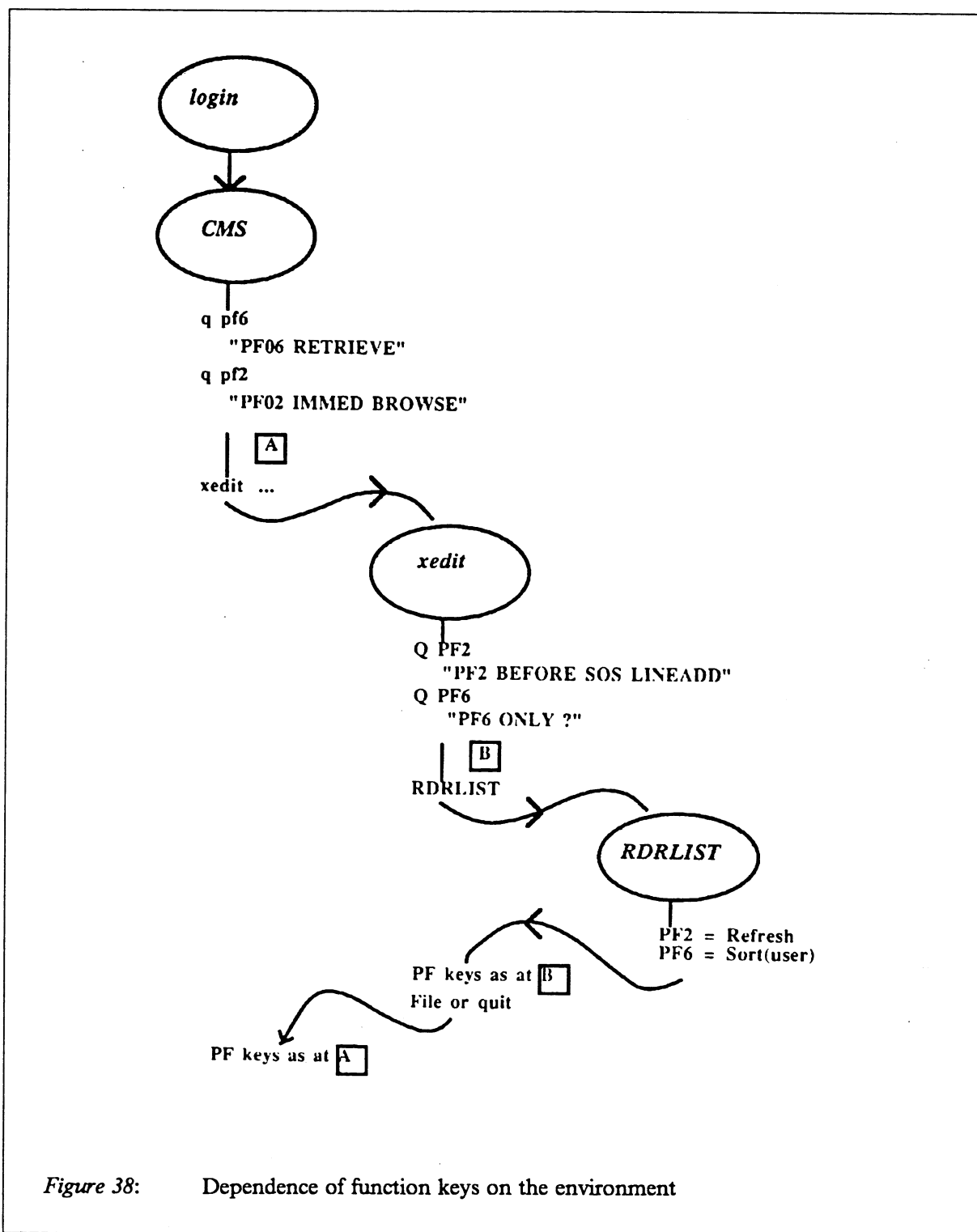


Figure 38: Dependence of function keys on the environment

CODE means display only the code of the message

### 3. *SET RUN:*

The command **CP SET RUN ON** causes automatic return from CP to CMS if PA1 was pressed. Useful if this is done accidentally; however, sometimes it is necessary to remain in PA1, e.g. when debugging a program. **SET RUN OFF** is normally used for CMS. Also useful for running programs in a disconnected machine.

### 4. *SET BLIP:*

The command **SET BLIP myblip** defines the blip character as myblip. The blip character is sent to the screen after each 2 second period of processor time used by the virtual machine. Used to send reassuring messages when a long program is being executed.

### 5. Various settings are available for colour terminal (CP SCREEN).

The settings defined by these commands can all be queried by the

#### Query SET

command.

## 7.4 Creating and Using EXEC Procedures (Files)

An EXEC procedure or file is a file with a filetype of EXEC. It is a collection of commands and operands which are executed when the filename of the exec is entered on the commandline. The exec is interpreted by the System Product Interpreter, more commonly known as REXX. Exec files are very useful to add commands to the system which perform functions that are not foreseen by CMS; or to simplify a collection of frequently used commands, thus hiding complicated features. The REXX language contains many features. Three simple rules are:

1. For REXX to recognise an exec file, the first line must be within comments ( /\* and \*/ ).
2. To avoid REXX interpreting CMS commands and their arguments as variables, they should be given in quotes.
3. Anything given within quotes is taken literally, i.e. no translation to uppercase takes place. CMS commands within quotes must therefore be given in uppercase.

For example, it may be useful to extend the disk configuration of a virtual machine at each logon time by a temporary disk and a link and access of a friend's disk. This could be done by a simple GETDISKS EXEC:

```
/* Get some disks */  
"GIME 10 K"  
"GIME PUBCP 201"
```

Typing GETDISK would have the same effect as typing the two individual GIME commands. More possibilities of exec files will be explored later.

### 7.4.1 The PROFILE EXEC

A special file called **PROFILE EXEC** can be used to set up personal default values for a CMS session (such as a particular disk setup). When CMS is IPL-ed, which happens automatically upon logging in to VM, CMS searches for a file called **PROFILE EXEC** on the A disk. The commands contained in this file will be executed automatically following the first **RETURN** entered, unless the first command given is

#### **ACCESS ( NOPROF**

The CERN standard **PROFILE EXEC** (a copy of which is on the P disk) then executes another file called **STANPROF EXEC** which is also on the P disk. This **EXEC** defines suitable defaults for the CERN system, and among other things executes the command.

#### **NEWS ( UNSEEN**

which will display the news items of the last 14 days that have not yet been read.

Individual news items may be viewed by entering their number on the commandline. To return to the main menu, hit PF3 or type **Return**. To quit news, hit PF3 or type **Exit**. Once a news item has been viewed, it is marked as "seen" and will not be displayed again. All news items from a certain date on (even seen ones) may be viewed using the **FROM** date option.

To tailor your own profile to your personal needs you should activate the relevant commands in your copy of **PROFILE EXEC** (which you obtained with your account) after the **EXEC STANPROF** statement.

Mode 1 is recommended for the **PROFILE EXEC** to allow execution in batch. You must not have your own version of 'STANPROF EXEC' or other system profiles.

## 7.5 Setting Defaults

Defaults for your virtual machine can be set up using the **DIRM** or **DEFAULTS SET** commands. The **DIRM** command allows you to set device related defaults, such as your logon password, automatic links to other minidisks, changing disk passwords etc. Type **DIRM** to see which defaults can be set. The **DEFAULTS** command allows defaults to be set for various commands and locally written execs. These can be very useful in the case of complicated execs such as **XPRINT** or **XSCRIPT**. To set defaults for a given command, use

#### **DEFAULTS SET commandname**

To list the defaults for a given command, use

#### **DEFAULTS LIST commandname**

Defaults can be set for the following commands and execfiles:

**FILELIST**  
**NOTE**  
**PEEK**  
**DRLIST**  
**RECEIVE**

SENDFILE  
TELL  
HELP

Most locally written execs such as XPRINT, XSCRIPT, VFORT, WHOIS

To determine whether a command will allow defaults to be set, try the command

**DEFAULTS LIST** commandname.

## 7.6 Halting output to the terminal

The output of programs to the terminal may be halted with the command

**HT**

(Halt Typing). To stop execution of a program, use the command

**HX**

(Halt Execution). To terminate a REXX exec file, use

**HI**

(Halt Interpretation). For REXX execs this should be used instead of HX as many exec files intercept this command and will leave you machine in a "clean" state. Typing hx will often result in the harmless message

```
DMSABN150W nn (HEX xx) DOUBLEWORDS OF SYSTEM STORAGE WERE NOT RECOVERED
```

This message may look alarming but has no consequences on the results of programs run after this message was displayed. However it does imply that the virtual storage was decreased by the amount stated in the message and consequently eventually a re-IPL of CMS may be necessary (or a redefinition of the size of storage).

## 7.7 Exercises for Chapter 7

1. Create a SYNONYM file SYNONYMS SYNONYM on your A-disk.
2. Type Q SYNONYM ALL and note which synonyms are set.
3. Now modify your PROFILE EXEC to contain the line: SYNONYM SYNONYMS.
4. First type the command PROFILE, then type Q SYNONYM ALL. See if the result is different to that obtained above.
5. Try DEFAULTS SET XPRINT to set some defaults for the XPRINT exec. Try DEFAULTS LIST XPRINT when you have set some defaults.

6. Add the line **CP SPOOL CONSOLE START TO \*** to your PROFILE EXEC. What does it do? Modify your PROFILE EXEC to contain any other things you found useful during this chapter. The following questions are optional.
7. Try **FIND LFILE** to understand the purpose of the EXEC option.
8. Type **LFILE \* EXEC A ( EXEC** and look at the resulting file **CMS EXEC A**.
9. Add a line: **&TRACE ALL** to the beginning of **CMS EXEC A**.
10. Try **EXEC CMS PUNCH**, and **Q PUN ALL**. You have discovered a handy way to perform a single operation (**PUNCH**, but it could have been **XPRINT**) on a large set of files.
11. Now **ERASE CMS EXEC** to avoid problems later.

## 8. Communicating with other Computer Users and Transferring Files

### Summary of Chapter 8.

At the end of this chapter, you should have learnt how to;

- Use the NAMES file.
- Send messages and mail to other computer users.
- Send files to other computer users.
- Transfer files from Wylbur and VMS to CMS and vice versa.

### 8.1 The Names File

To communicate with other users, one needs to know which computer they are working on (the node name, e.g. CERNVM or GEN), and the userid. Use the

#### Identify

command to find the userid and node name of the logged on virtual machine. It is useful to keep this information together with names and addresses, telephone numbers and other things. This data can be conveniently kept in the NAMES file, which is like an electronic address book. Each entry is defined by a nickname which can be used to refer to the entry. Communication may occur using the nickname instead of the userid and the node name. A nickname can be defined for a group of userid's, making it easy to communicate with many users simultaneously. A NAMES file

```
userid NAMES A0
```

is created on userid's A-disk with the command

#### NAMES

This results in a screen as shown in Figure 39.  
The fields on the screen are explained as follows.

**Nickname:** The nickname may be any name between 1 and 8 characters. It can be used to refer to entries with the various commands that use the NAMES file. At least a nickname and a userid (or a list of names) must be supplied to form a valid entry, i.e. a nickname and a userid define an entry.

**Notebook:** Specifies the name of the notebook file you wish all electronic mail originating from this userid to be received into. By default, all mail is received into the file ALL NOTEBOOK A0.

**userid:** The userid belonging to the nickname. At CERN, the NAMES panel has been modified to allow this field to contain userid's that are longer than 8 characters for certain remote users.



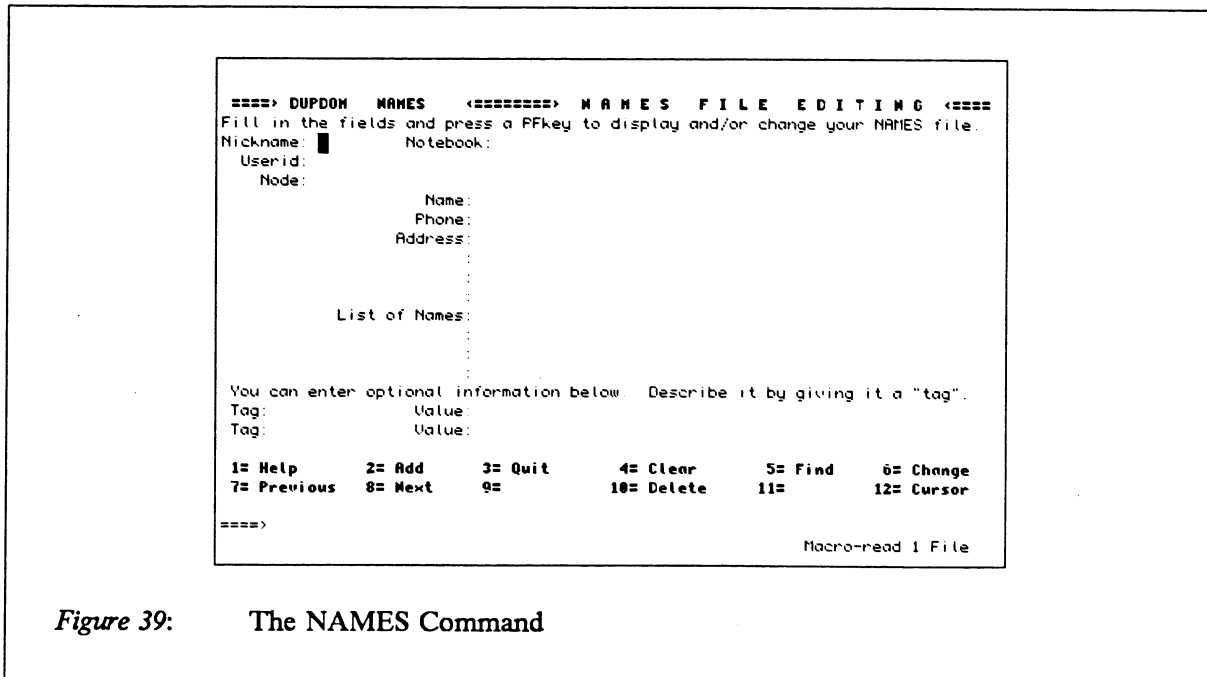


Figure 39: The NAMES Command

**Node:** The name which identifies the computer on the network. This does not necessarily have to be an EARN/BITNET node name. By default, this is assumed to be CERNVM. For nickname entries corresponding to Wylbur users, use GEN; for PRIAM users, use CERNVAX. To find a list of current EARN/BITNET nodes, type

```

GIME EARN
X BITEARN NODES

```

A local CERN modification allows for node names longer than eight characters.

**Name:** The name of the person owning the userid.

**Phone:** His/her telephone number<sup>8</sup>.

**Address:** His/her address.

**List of names:** The distribution list identified by the nickname. This is a list of nicknames or userid's separated by blanks (not comma's). If a list of names is given, none of the other fields are required.

**Tag: Value:** The tag and value fields are for optionally entering other data you may need to store. Tag: defines the name (nickname:, userid:, node:, etc. are all tags), and Value: is the value you wish this tag to have. Useful for using with the NAMEFIND command.

To modify or add entries in the NAMES file, hit the function keys as required.

---

<sup>8</sup> Due to a problem with the NAMEFIND command, an address should not contain - (minus) signs.

The main commands that use the NAMES file are: NAMEFIND, TELL, NOTE, MAIL, SENDFILE, WHOIS.

### 8.1.1 Interrogating the Names File using Namefind and Whois

To display the information in a NAMES file one may use the

```
NAMES [nickname]
```

command and use the function keys to scroll through the various entries. Another method is to use the

```
NAMEFIND :tag value [:tag [value]] ....
```

command. For example, given the fact that the entry in a NAMES file defined by nickname 'ERIC' contains the tag :phone with value 5010, the command

```
NAMEFIND :nick ERIC :phone
```

would have give the response

```
5010
```

For more possibilities of the NAMEFIND command (which is particularly useful in EXEC files) type **HELP CMS NAMEFIND** or consult [5].

The **WHOIS** command also interfaces with the names file in a similar way, but is much easier to use. To use the above example, the command

```
WHOIS ERIC [( Phone]
```

would give the following response (note the phone option may be omitted):

```
Name:          Eric van Herwijnen
Userid:        ERIC AT CERNVM
Phone_No(s):   5010
Address:       etc.
```

Whois displays the information found in the names file. However, if there is no entry corresponding to the userid or nickname the VM Userlist is searched for any matching name. If a match is found, the VM-userid's belonging to this person are displayed in alphabetical order. For example,

```
WHOIS VAN HERWIJNEN
```

would give:

Name	Forename(s)	VM Id
VAN HERWIJNEN	ERIC	CERNDOC
VAN HERWIJNEN	ERIC	ERIC
VAN HERWIJNEN	ERIC	FREEDA
VAN HERWIJNEN	ERIC	MATHS
.... etc		

To just find the name of the person owning a given VM-userid, use the option VMID:

**WHOIS userid ( VMID**

The option WYLBUR or WYLID can be used to find Wylbur userid's. See HELP WHOIS for more details.

### 8.1.2 The Tell Command

Using the command

**TELL [nickname][userid [AT node name]] message**

it is possible to send userid (corresponding to nickname if this was given instead) a one line message. It is an interactive message, i.e. the person you are sending the message to will immediately have it displayed on his/her screen provided they are logged on. To send a message to the operators, use the command

**TELL OPSCT text-of-the-message**

To find out whether someone is logged on, use the command

**Query userid**

For EARN/BITNET users, use the command

**QBIT [nickname][userid [AT node name]]**

Do not confuse the tell command with the CP command

**CP MSG userid [AT node name] message**

which does NOT use the NAMES file and does not understand nicknames. To prevent being disturbed by these interactive messages enter the command

**SET MSG OFF**

From this moment onwards, no messages will be displayed on the screen. However, beware as the response from some commands will now also be lost as well as error messages which could be important. It is better to run with the default setting (SET MSG ON).

### 8.1.3 Notes

A **note** is a non-interactive message which usually spans more than one line. The user does not have to be logged on to receive a note; however, if they are, a notification line will be displayed. The note is stored in the reader until the person is ready to read it. Note allows you to send to any node connected to EARN/BITNET. For a more general facility, use **MAIL** (see 8.1.5 on page 82). The command

**NOTE [nickname][userid [AT node name]]**

places the user into Xedit and produces a screen as shown in Figure 40 on page 80.

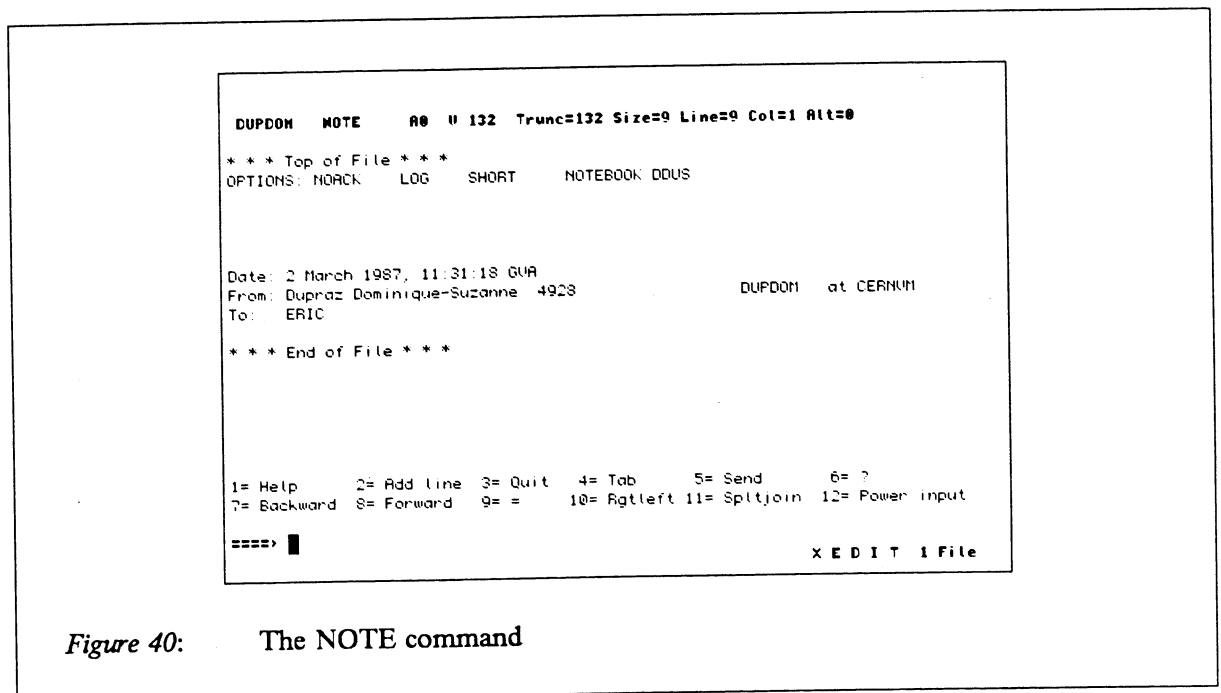


Figure 40: The NOTE command

To write something in the note, it is most convenient to go into input mode by typing **Input** on the command line. To send the note, hit PF5. To file the note and send it later, type **FILE**. This saves a copy of the note with fileid **userid NOTE A0**. Return to the note at a later time by simply typing **NOTE**.

The **NOTE** command has several options for example (**ACK** if an acknowledgment of receipt of the note is required. Defaults for options such as these (type **HELP NOTE** for a full description) may be set via the **defaults set note** command.

Notes sent to other VM users are stored in their virtual reader. A copy is kept in the sender's file **ALL NOTEBOOK**, or if an entry exists in the sender's **NAMES** file specifying a different **NOTEBOOK** it will be kept in there. To read a **NOTE** which has been sent to a given reader, use the commands

**RDRLIST**  
**PEEK** (in CMD area)

or

### PEEK spoolid

To find the spoolid of a file in the reader, use Q RDR (see 5.6 on page 57). A note may be received in a notebook by the RECEIVE command, i.e. the note is copied from the spool onto disk. By default a note is placed in the file ALL NOTEBOOK A0. However, if an entry exists in the file userid NAMES A0 specifying the name of a different notebook, the note will be placed in there. In this way, mail from different users can be kept in separate files. To delete the note from the reader, use the

### DISCard [spoolid]

command. This command can be issued from the CMD area in RDRLIST, on the commandline in CMS (with argument spoolid), or from PEEK (without argument).

### 8.1.4 The Sendfile Command

To send someone else a file, use the command

```
SendFile [fn ft fm [nickname]][[userid [AT node name]]]
```

If sendfile is called without arguments, a panel will be displayed (see Figure 41 on page 81). For beginners, this is the simplest way to use it. The panel is also very convenient if you have to send more than one file, as it allows you to select from list by placing a "\*" in any of the fileid fields. Inside RDRLIST, fn ft fm may be replaced by a "/" (slash).

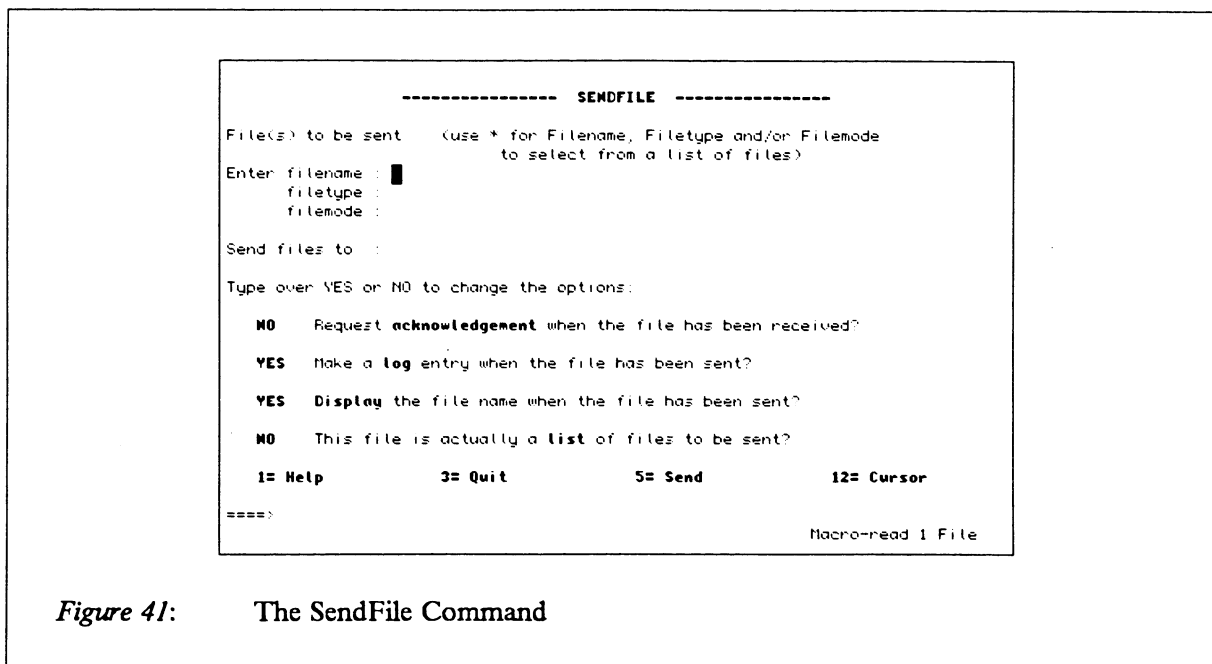


Figure 41: The SendFile Command

The processing of spoolfiles, which are the result of the sendfile command, is identical to those of the note command, except the RECEIVE command places the files on disk with their original names (or a

different one if required). For example, from inside RDRLIST one could issue the following command (use the REPLACE option if newfn newft newfm already exists):

**RECEIVE / newfn newft newfm ( REPLACE**

### 8.1.5 Mail

The NOTE command is fine for sending mail to EARN/BITNET sites; however it does not allow you to send mail to other places. For this reason, the **Columbia Mailer** was installed at CERN. It is used by the **mail** command. (See [10]). The mail command allows you to prepare, send, read and log (store), reply to and forward electronic mail. The mail command transfers your incoming mail from the reader into UNREAD NOTEBOOK A0. When a mail is logged, it is transferred into the NOTEBOOK defined by your NAMES file ( ALL NOTEBOOK by default). To speed up MAIL logging/discarding should be done regularly. The mail command has the following syntax (for sending mail):

**MAIL nickname | userid [ ( fileid ]**

where;

**nickname** is an entry in the NAMES file corresponding to a userid or a list of userids.

**userid** is a userid, followed by possibly a network address, for example:

JUDY	CERNVM
JAR\$CP@GEN	Wylbur
DB@DLGM.DL.AC.UK	Janet
BEBO@SLACVM	Earn/bitnet(SLAC)

For more details on network addresses, please consult the User Guide to Electronic Mail Services at CERN [9].

**fileid** The fileid of the file containing the text of the mail.

For example

**MAIL ERIC**

will give a screen similar to that produced by the NOTE command (see Figure 42).

To send the mail simply hit PF5, or type SEND on the command line. A copy will be kept in the notebook defined by your NAMES file (ALL NOTEBOOK by default). Sending mail to a list of people may be done by giving a nickname as argument. Interrupting a mail may be done by giving the command SUSPEND. This will create a file userid MAIL A0 where userid is the first recipient of the mail. To continue at later time, type

**MAIL userid ( RESUME**

To read your mail, use the command

**MAIL or MAILBOOK**

```

>>> MAIL 86.359 <<< Sending Mail: Enter your message below      Line 5 of 18
=====
* * * Top of File * * *
Date:      Tue, 19 May 87 11:20:19 GUA
From:      Dupraz Dominique-Suzanne <DUPDOM@CERN.CH>
To:        Van Herwijnen Eric <ERIC@CERN.CH>
=====

[ ]

* * * End of File * * *
PF1=Help   PF2=Addline  PF3=Quit   PF4 =Page  PF5 =Send   PF6 =Suspend
PF7=ScrollUp PF8=ScrollDown PF9=Delline PF10=Switch PF11=SpltJoin PF12=Cursor
    
```

Figure 42: Sending a MAIL

with no arguments (see Figure 43) When reading mail commands are available for reading (READ or PF2), replying (REPLY or PF5), discarding (DISCARD or PF9) or logging mail (LOG). To change the function keys and other defaults, copy the file MAILUSER XEDIT P from the P disk to your A-disk and make the required changes. To read your mail, use the command

**MAIL or MAILBOOK**

with no arguments (see Figure 43)

```

>>> MAIL 86.359 <<< Review Incoming Mail
=====
From:      [ ]
To:        [ ]
Date:      [ ]
Subject:   [ ]
Van Herwijnen Eric Dupraz Dominique-Su>10/09/86 No Subject
Van Herwijnen Eric Dupraz Dominique-Su>10/09/86 No Subject
Renshall Harry Robe Dupraz Dominique-Su>10/07/86 Section Leaders Meeting
Van Herwijnen Eric Dupraz Dominique-Su>10/07/86 No Subject
Koppanyi Attila Dupraz Dominique-Su>12/01/86 Christmas dinner reminder
Goossens Michel Dupraz Dominique-Su>12/03/86 Network tutorials
Van Herwijnen Eric Dupraz Dominique-Su>11/24/86 No Subject
FAILURE@CERN.CH Dupraz Dominique-Su>12/03/86 No Subject
Renshall Harry Robe Dupraz Dominique-Su>12/02/86 Section Leaders Meeting 2 Dece
Koppanyi Attila Michael Metcalf 11/27/86 Christmas Dinner
CERNDOC@CERN.CH Dupraz Dominique-Su> 3/25/87 Project UNICOURSE
CERNDOC@CERN.CH DUPDOM@CERN.CH 3/25/87 Project UNICOURSE
* * * End of File * * *

PF1=Help   PF2=Read   PF3=Quit   PF4 =XPrint PF5 =Reply  PF6 =Forward
PF7=ScrollUp PF8=ScrollDown PF9=Discard PF10=SortUp PF11=SortDown PF12=Select
    
```

Figure 43: The MAILBOOK command

Mail has various other useful options. For more details, type **HELP MAIL**, **HELP MAILBOOK**, or consult [9].

## 8.2 EMDIR

EMDIR is a distributed electronic mail directory service, running on VM/CMS, VMS and UNIX at CERN.

To query someone's electronic mail address, type

**EMDIR last name firstname division**

where \*'s may be used as wild characters. For example

**EMDIR VAN\*HERWIJNEN ERIC DD**

will give:

van HERWIJNEN ERIC/DD	(ext.5010)
CERN	as of 25-May-87
MAIL:ERIC@CERNVM.CERN	
ALSO:EVH@SLACVM.BITNET.	VM/CMS-NEWS

To add your preferred electronic mail address to the data base send a request to:

**CERNMAIL AT VXCRNA**

Type **HELP EMDIR MENU** for more details.

## 8.3 Automatic forwarding of MAIL

Automatic forwarding of mail may be achieved by using the **LISTSERV** facility:

**TELL LISTSERV /FORWARD**

to cancel autoforwarding

**TELL LISTSERV /FORWARD address-of-forwarding**

to define the address where your mail (coming to your VM userid) should be forwarded to. Very useful if you normally don't use VM or if you are leaving for a different place temporarily.



## 8.4 Transferring Files from Wylbur to VM and vice versa

Various commands are available to do this. We should like to warn you however, that users should not use VM as an editor or front-end for work on MVS, as one day the MVS/Wylbur system may disappear (end'88, see CNL 186). The tools that are described in this section are to facilitate the migration to VM, and it is the user's interest to convert as completely as is possible. If facilities are required that are available on MVS but not on VM, please contact anyone in the User Support Group or the UCO (Tel. 4952, or via NOTE UCO).

It is possible to login to Wylbur while remaining inside a VM session by using the Wylbur bridge (see appendix C on page 112).

### 8.4.1 Use

The command

**USE gg.uuu.xxxx [(TRUNC trunc)]**

where gg.uuu.xxx is the fully qualified MVS dataset name (which may be preceded by a "\$"). USE will find the MVS disk on which the dataset resides and link to that disk. At the end of the exec the user will be left in XEDIT with the WYLBUR dataset as the "active file". If the file is to be kept permanently on a minidisk, the user must SAVE or FILE it himself. The filename and filetype may need to be changed (see Figure 44 on page 85).

```

LOGON   LIB      A1  U 80  Trunc=80  Size=46  Line=0  Col=1  Alt=0

===== * * * Top of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== ; EXAMPLE FOR LOGON EXEC FILE ..... 78/06/06 ..... B FOLLERMANN (PEO)
=====
; !!!!!!! ---> DO NOT CHANGE LINES 1/9 <--- !!!!!!!
; ----- DO CHANGE LINE 10 -----
=====
===== set esc : ; set the escape character
===== ON ATT EXEC 2 ; if BREAK occurs, jump to line 2
===== if (BULLNUM ge SYSBUL) exe 7 ; skip news if user has read it
===== set BULLNUM=(BULLNUM+1) ; update the users BULLNUM
=====> █

XEDIT 1 File

```

*Figure 44:* The Use Command

The following points should be noted:

1. USE makes use of the MVS audit dataset. This means that datasets created since the last audit cannot be USED. (The audit is normally done around 2 a.m.).
2. Datasets with a record format of VBS and MSS datasets cannot be USED. Such datasets can be accessed either using GETFILE or IMPORT (access via a job submitted to MVS). OSCOPY can also use VBS format files.
3. Use truncates the file at 80 characters. Add the TRUNC option if more characters are required.

#### 8.4.2 Getfile

Use the GETFILE command to copy WYLBUR datasets to CMS. The format of the GETFILE command is:

**GETFILE gg.uuu.xxx fn ft [fm] ( REPLACE**

where gg.uuu.xxx is the fully qualified MVS datasetname of the file to be copied, optionally preceded by a '\$'. The REPLACE option is required if the CMS file already exists. It gives permission to overwrite the existing file. If the file exists and REPLACE is not specified, GETFILE will terminate abnormally. Please note the following points:

1. Like USE, GETFILE makes use of the MVS audit.
2. GETFILE will copy RECFM=VBS datasets (e.g. MVS PAM files). On CMS PAM files and OLDLIBS copied using GETFILE can be used with all the PATCHY and HISTORIAN utilities.
3. MSS datasets cannot (at present) be copied with GETFILE.

#### 8.4.3 Import

The IMPORT command is used to copy a dataset from MVS to CMS. It is a non-interactive command which works by submitting batch job to MVS. If IMPORT is called without arguments, a panel will be displayed in which the necessary parameters can be supplied:

**IMPORT gg.uuu.xxx ( options**

Where gg.uuu.xxx is the fully qualified data set name of the MVS file to be IMPORTed. This can be either:

- A sequential dataset such as **gg.uuu.filename**
- A library such as **gg.uuu.library** (using the PDS option)
- A member of a library such as **gg.uuu.library(member)**

If the PDS option is given, "dsn" is assumed to be a PDS. At present only "source" libraries (i.e. those in WYLBUR or CARD format) are supported. The whole PDS is made into a sequential file and sent to your virtual reader. The "MAKEPDS" exec should be used to reconstitute each individual member of the PDS as separate CMS file. Please note the following points:

1. Both WYLBUR (EDIT) and CARD format datasets are handled by IMPORT.
2. IMPORT sends a batch job to MVS. This job sends the WYLBUR file as a CLASS B punch file to your virtual reader. To move the file from your reader to a mini-disk, use the CMS command RECEIVE.

#### 8.4.4 Export

The EXPORT command is used to copy a dataset from CMS to MVS. It is a non-interactive command which works by submitting a job to MVS. If EXPORT is called without arguments a panel will be displayed in which the necessary parameters may be supplied:

**EXPORT [fn [ft [fm]] [gg.uuu.xxx ( options**

where gg.uuu.xxx is the fully qualified data set name of the MVS file to which the CMS file is to be EXPORTed. Use the REPLACE option to replace the MVS file "gg.uuu.xxx" if it already exists. The default is APPEND; i.e. if the MVS file "gg.uuu.xxx" already exists, the file being EXPORTed will be APPENDED to "gg.uuu.xxx".

#### 8.5 Transferring Files from the VAX to VM and vice versa

To transfer files between CMS and the vax use the NFT command on CMS, and INTERLINK on the vax. To be able to do this, an ASCII terminal (e.g. a DMT2200 or a Pericom) is required. For example to send the file PROFILE EXEC A to VMS use the command:

**NFT SEND VXCRNA"your\_VMS\_ID password"::[your\_VMS\_id]profile.exec profile exec a**

or to receive a file x.y from VMS as the file fn ft fm on CMS, use the command

**NFT RECEIVE VXCRNA"your\_VMS\_ID password"::[your\_VMS\_id]x.y fn ft fm**

or

**NFT RECEIVE VXCRNA"your\_VMS\_ID password"::DISKSGG:[your\_VMS\_id]x.y fn ft fm**

For other options and possibilities see HELP CMS NFT.

To use interlink on the Vax, the copy command is convenient. For example

**COPY PROFILE.EXEC CERNVM"your\_VM\_ID password"::"[your\_VM\_id.addr] PROFILE.EXEC"**

will copy profile.exec from VMS onto id's minidisk at addr. This disk must **not** be linked R/W by your VM id when this command is issued.

To copy in the other direction i.e. from CMS to VMS, simply invert the arguments:

**COPY CERNVM::"[your\_VM\_id.191]PROFILE.EXEC" PROFILE.EXEC**

will copy CERNVM id's profile exec a onto profile.exec. For more information, see the Interlink User's Guide (DD/US/107).

---

## 8.6 Submitting Jobs on Networked Machines

To submit either a CMS file, or the current XEDIT file, for execution at one of the OS batch systems at CERN or linked to CERN (e.g. at RAL or DESY), the command

```
SUBMIT [fn [ft [fm]]] [(options )
```

should be used. The fileid specifies the file containing the OS jobs to be sent to the batch system. It is compulsory when the SUBMIT command is issued from CMS. When issued from XEDIT, no fileid may be given. The current XEDIT file is then submitted.

By default, the job will be submitted to the MVS system at CERN. However, by using the TO option, the machine may be specified where the job should be submitted to. This may be any machine accessible via the RSCS network or the userid of another CMS machine at CERNVM. To send a job over the RSCS network, the "TO" parameter must be of the format LOCATION.ID (e.g. GEN.JOB for MVS at CERN or UKACRL.CRAY for CRAY at RAL).

By default the output of these jobs will be sent back to the submitter's reader. For more details, see HELP SUBMIT.

To locate a job running on the MVS system at CERN, the

```
LOCATE jobid
```

command should be used. Jobid is the jobname or the jobnumber to be located. To purge a job on the MVS system, use the command

```
MVSPURGE jobid
```

This only allows jobs to be purged that were submitted by the virtual machine which issues the purge request.

## 8.7 Exercises for Chapter 8

1. Create a NAMES file on your A-disk by adding your own userid and possibly other entries.
2. Try the NAMEFIND exec to display the entries in your names file.
3. Try the WHOIS exec to determine the Wylbur identifier of the author of this tutorial.
4. Send a note to yourself using the NOTE command. View it with PEEK and DISCARD it after you have read it.
5. Try the commands USE gg.uuu.xxxx (use CP.EVH.LIB(LOGON) if you don't have a Wylbur account) and IMPORT gg.uuu.xxxx and note the difference between these two commands.

## 9. Some Local CERN Facilities

### Summary of Chapter 9.

At the end of this chapter, you should have learnt how to;

- Use the printing facility XPRINT.
- Use the textprocessor Waterloo SCRIPT via XSCRIPT.
- Recuperate backup files with VMBLIST, and archiving using VMARCH.
- Copy files from tapes to disk and vice versa.
- Use some of the functions of the SPACE machine.

Many additional commands and facilities have been added to the native VM system at CERN. This chapter is intended as an introduction to the more important ones.

### 9.1 How to print files

Printing a CMS file is complicated due to the fact that at CERN there are no printers directly attached to CP (with the exception of some printers driven by service machines that are only available for text-processing). All printers are driven by JES, which implies that the files must be transferred from CERNVM to GEN (the MVS system) before they can be printed. This is achieved by the

#### XPRINT

command. The great multitude and diversity of accessible printers explains why the panel (see Figure 45) looks rather complicated at the first sight.

However, many features are built-in to facilitate its use:

1. If the meaning of an input field is not clear, type ? or hit PF1 to obtain HELP about this particular option or parameter.
2. Do not try to fill in the entire panel. Simply give the building number where the printer is located (type ? if this is not known), and type \* if the printertype is unknown. Xprint will offer a list of choices (see Figure 46).
3. Files with filetype LISTING are always printed with carriage control (using the CC option).
4. To print many files as one large print file, use \* for the filename or filetype and use the CONT option.
5. To display all the options which Xprint has chosen, hit PF2 before printing. If any option needs to be changed, the entire panel must be refilled (it is often easier to quit Xprint and try again).
6. Xprint uses the current file, i.e. it assumes the file which should be printed is the last edited one.

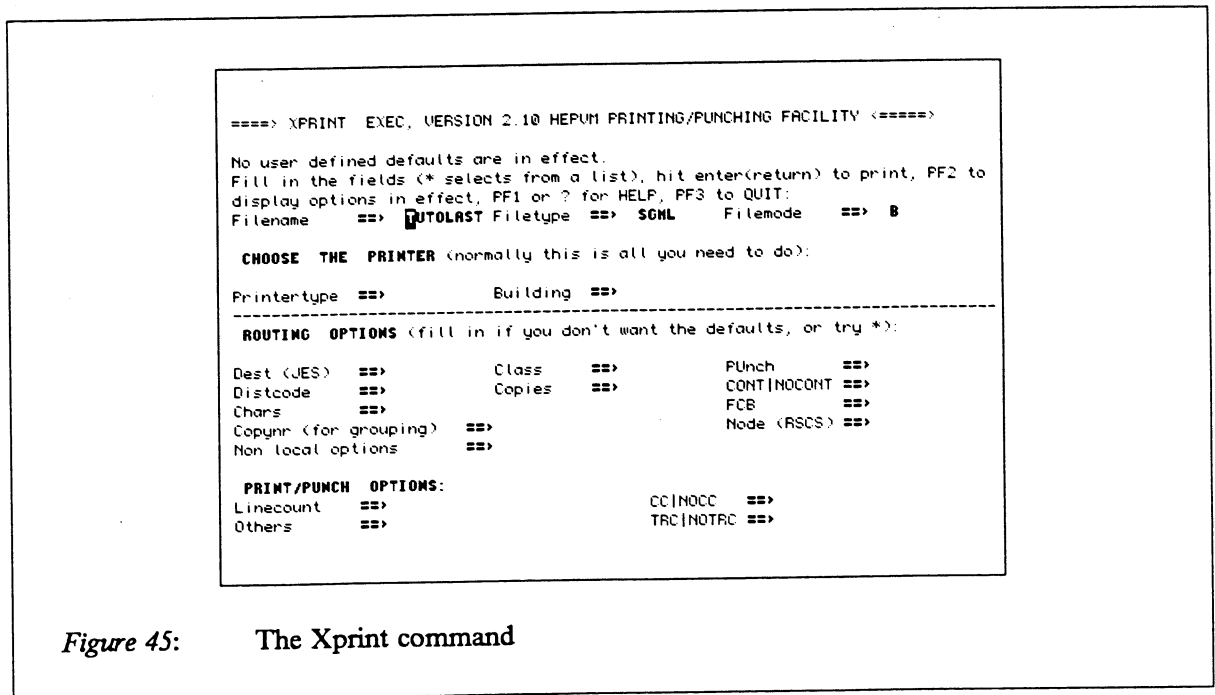


Figure 45: The Xprint command

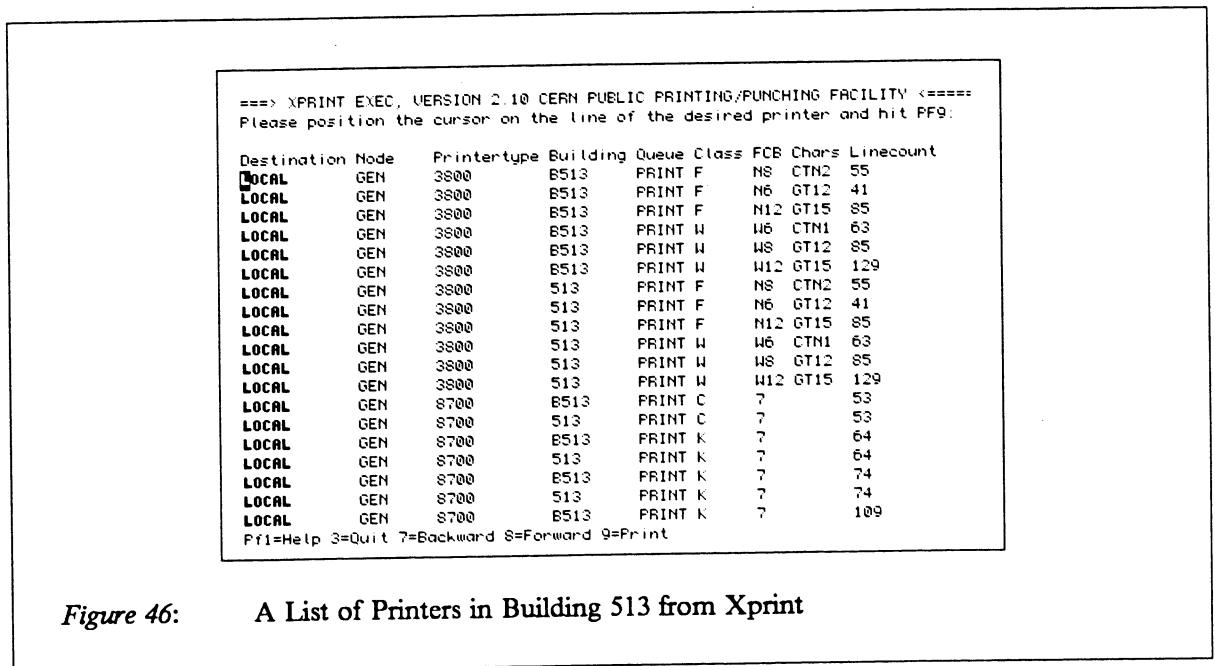


Figure 46: A List of Printers in Building 513 from Xprint

7. If Xprint finds enough arguments on the commandline, the panel is not displayed and the file is sent directly to the printer Xprint may always be forced to display the panel by giving the PANEL option. When executed in linemode or in the batch machine Xprint must always be called by giving all options on the commandline.
8. Defaults may be set for all options. For example, a **distcode** (distribution/delivery code or JES room number) may be set by the command

**DEFAULTS SET XPRINT DIST xx**

Valid distribution codes are:

Distcode	Building	Rios name
LOCAL	513	
X0	582	WH
X1	2	OC
X2	4	SB
X3	13	T2,TC
X4	112	IR
X5	865	SP
X6	892	NA
X7	162	TL
X8	22	NP
X9	54	
XA	5	
XB	57	
XC	6	PS
XN	Annecy	

However, if a distcode was set with the

**DIRM DIST xx**

command, XPRINT will use this. If only the central printers are used, the directory distcode should be set to LOCAL, otherwise problems may arise when sending mail to TSO/E sites.

If both a directory distcode and an Xprint default distcode are set, the Xprint default will be used.

9. Secondary defaults for different output classes may be set by replying "YES" to the question "DEFAULTS REQUIRED FOR OTHER CLASSES?". For example, it may be convenient to have defaults CHARS PHY2 FCB W6 for class W and CHARS EURO FCB N8 for class F.
10. Spool files may be printed by typing

**XPRINT .**

Notice the space between Xprint and the ".". The same defaults apply as used when printing disk files.

11. When inside Xedit, Xprint will behave as an ordinary Xedit command, i.e. it will print starting at the current line ending at the target. To print the entire Xedit "active" file, type

**TOP**  
**XPRINT \***

12. No panel is displayed inside Xedit. However, to print disk files from inside Xedit (with a panel), type

## CMS XPRINT ( PANEL

### 9.2 Text processing facilities available from VM

The text processor available on VM is Waterloo SCRIPT. The macro package CERNPAPER is well documented [8] and supported. This should be used in preference to pure SCRIPT. Please remember to delete any MVS JCL cards, i.e. a CERNPAPER job should start with a line

```
.im crnbasic
```

To run the SCRIPT processor on a SCRIPT input file, the

#### XSCRIPT

command should be used (see Figure 47 on page 92). Please note the following:

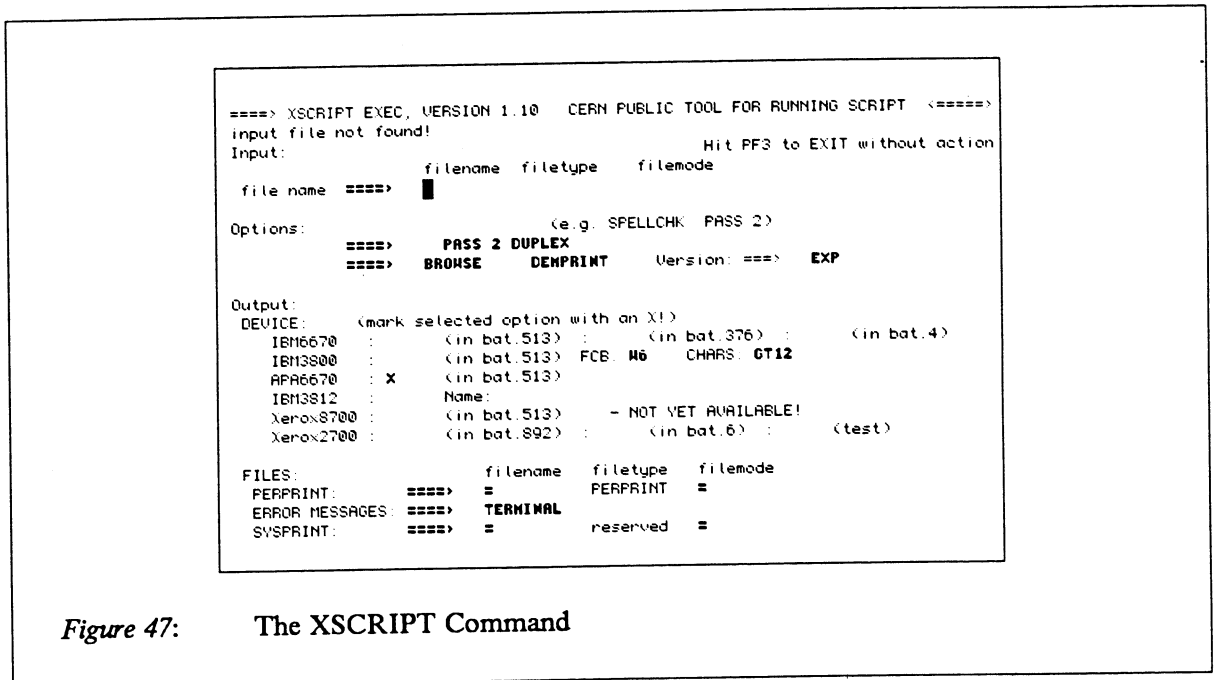


Figure 47: The XSCRIPT Command

1. All SCRIPT input files should have filetype SCRIPT.
2. The source input disk must be accessed R/W, as SCRIPT places the formatted output there.
3. Defaults may be set for all fields on the panel to make future use easier.
4. Arguments may be given on the command line to allow execution in batch and linemode, i.e. without displaying the panel.

To print a part of the document, the FROM and TO parameters should be used:



## FROM \* TO 5

would print the document starting at the title page and finishing with page 5. The Cernpaper System is now frozen and it is recommended to use SGML instead [13]. SGML marked up files should have filetype SGML and are processed with the SGML command.

The advantages of SGML are

1. Documents thus marked up conform to ISO standard 8879.
2. There is enhanced support for text and graphics (GKS metafiles, MacIntosh, Apollo bitmaps) and mathematical equations.

The SGML command is similar to XSCRIPT command see (Figure 48 on page 93).

```

====> SGML EXEC, VERS. 1.00 CERN PUBLIC TOOL FOR FORMATTING SGML DOCUMENTS <===
Input:                                     Hit PF3 to EXIT without action
      filename filetype  filemode
file name  <====>  AUTOLAST  SGML      B

Options:                                     (e.g. SPELLCHK, PASS 2)
      <====>  PASS 2 DUPLEX FROM
      <====>  BROWSE      DEMPRINT      Version: <====>  EXP

Output:
DEVICE: (mark selected option with an X!)
IBM6670 : (in bat. 513) : (in bat. 376) : (in bat. 4)
IBM3800 : (in bat. 513) FCB: M6 CHARS: GT12
APR6670 : (in bat. 513)
IBM3812 : X Name: DD31A
Xerox8700 : (in bat. 513) - NOT YET AVAILABLE!
Xerox2700 : (in bat. 892) : (in bat. 6) : (in bat. 2)

FILES:
PERPRINT: <====> = PERPRINT =
ERROR MESSAGES: <====> TERMINAL
SVSPRINT: <====> = reserved =

```

Figure 48: The SGML command

### 9.3 Recovering Backup Files and Archiving

It may occur that a file has been lost or accidentally overwritten. In such a case the **VMBLIST** utility may be used to retrieve information concerning CMS files or minidisks dumped by the backup system **VMBACKUP**. Every night this system dumps all files that have been changed during the preceding 24 hours to tape. Individual files or entire minidisks may be restored from tape. The **VMBLIST** command generates the restore job and sends the job to the **VMBACKUP** virtual machine for processing. To restore a file, use the command

```
VMBLIST fn ft [( RESTORE FROM month/day/87 ALL
```

The **RESTORE** option requests that the information on dumped files be displayed on your terminal. This option can also be used for selecting files to be restored and for generating a restore job stream. The **FROM** option considerably speeds up the restore process. As date a day a few days before today

should be given. A few minutes after issuing this command a panel containing information on the backed up files is displayed (see Figure 49 on page 94).

```

05MAR87 Release 04.01      U M B L I S T      (c) 1986, UM Software Inc.

      CMS Files Previously Backed-up For: DUPDOM
Cmd  Filename Filetype Cuv File Date/Time  Recs  Catalog  Dump Date/Time
█   PROFILE EXEC    191 11/25/86  17:57   83   FEB    02/28/87 03:24

Valid Cnds: Use R to restore a file, RM to restore a minidisk.
             1=Help    2=Refresh  3=Quit    4=Stype  5=Sdate  6=Sname
             7=Backward 8=Forward  9=        10=Restore 11=      12=Cursor
=====>

```

*Figure 49:* The Result of VMBLIST PROFILE EXEC ( RESTORE

To restore a file type R in front and hit PF10. After doing this, the system prompts for the userid where the file should be returned to. Reply by hitting RETURN. Subsequently the user is offered the choice to have the file returned to the reader or directly on to disk. Here the reply RDR must be given (see Figure 50); does not work at CERN).

These actions complete the creation of a restore job, which is sent to the VMBACKUP service machine. The output of the restore job is sent to the reader, as well as the restored file(s).

Automatic archiving does not take place on VM (unlike migration on the MVS systems). When a file is no longer needed for immediate use, but might be needed sometime in the future, it may be archived. To archive and recall files, the VMARCHIVE facility should be used. This facility is entirely under the user's control. Type **HELP VMARCH MENU** for a list of all commands of this package. The most useful ones are

**VMARCH ARCHIVE fn ft fm**

to archive the file fn, ft, fm; and

**VMARCH RECALL fn ft fm**

to recall file fn ft fm.

```

UMBLST121R ENTER THE USERID TO RECEIVE RESTORE(S)
UMBLST122R JUST PRESS ENTER TO USE DUPDOM

UMBLST125R DO YOU WANT FILE(S) PUT IN DUPDOM'S READER OR WRITTEN DIRECTLY TO DIS
K?
UMBLST126R PLEASE ANSWER 'ADR', 'DISK', OR 'QUIT' (TO EXIT)
adr
FUN FILE 3409 TO UMBACKUP COPY 001 NOHOLD
  08:29:27 MSG FROM UMBACKUP:
  UMBMSG102I UMBACKUP JOB RESTORE STARTED
R; T=7.70/8.90 08:29:27

RUNNING CERNVM

```

Figure 50: Restoring PROFILE EXEC with VMBLIST

## 9.4 Use of magnetic tapes

Use of tapes is **only** possible in the batch machine. To understand tape handling in full detail, the VM User's Guide [2] should be consulted. Here, the use of some simple copy facilities will be discussed. These facilities prepare and submit batch jobs. For most users this will suffice. Three useful tools exist: **DUMPTAPE** for analysing the contents of a tape, **DISKTAPE** for copying data from a CMS minidisk to tape, **TAPEDISK** for copying data from tape to a CMS minidisk or to the reader. All these tools create a job containing the correct sequence of CMS commands and submit this job to the batch machine.

### 9.4.1 Dumptape

The **dumptape** command should be used to analyse the label and the contents of a tape. This should be done before copying the tape to disk to determine which parameters need to be supplied to the **tapedisk** facility. The output is returned to the submitters virtual reader.

### 9.4.2 Tapedisk

The **tapedisk** command should be used to copy the contents of a tape to disk (see Figure 51). Tapedisk is capable of copying three types of files:

1. CMS files which were dumped onto tape in the CMS internal format using the command

#### TAPE DUMP

To load all files dumped onto the tape, enter \*'s for the fileid fields and indicate the format CMS (which is the default).

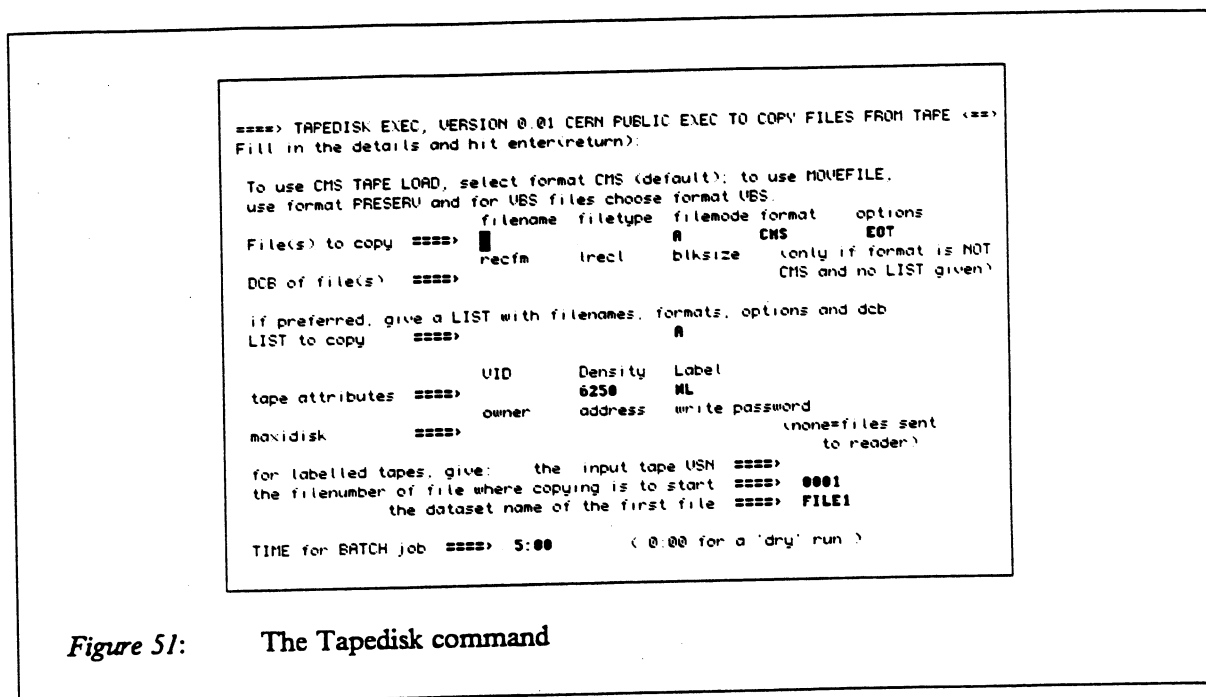


Figure 51: The Tapedisk command

2. Files which were copied onto tape preserving their format (i.e. which have record format F or V, or any OS files except those with format VBS). These may be copied with the PRESERV format. In this case, the record format and record length must also be supplied (these can be found by executing DUMPTAPE first).
3. OS files which have record format VBS. These may be copied with the VBS format.

In cases where there are many files on the tape, separate tapedisk jobs can be run to copy each individual file, or all the relevant information can be supplied in a LIST file. Mixed format tapes may be copied in this way.

Normally, the files will be copied onto the batch workers A-disk and sent to the submitters reader. However, if the files are large, this may fill up the spool. In such a case, it is better to give the name of a disk (with write password) to which the batch worker can link and write directly onto this disk without going through the spool. After execution of the TAPEDISK exec, the file TAPEDISK EXEC (which is the one submitted to the batch machine) is left behind on the A-disk, for modification and resubmission if necessary.

### 9.4.3 Disktape

The disktape command does almost exactly the inverse of tapedisk (see Figure 52). As before, to avoid filling the spool, the name of the minidisk should be given to allow the batch machine to directly copy from the disk.

```

====> DISKTAPE EXEC. (VERSION 0.01 = CERN PUBLIC EXEC TO COPY FILES TO TAPE <==>
Fill in the details (type over the defaults if necessary):

To use CMS TAPE DUMP, select format CMS (default); to use MOVEFILE,
use format PRESERV and for UBS files choose format UBS
filename filetype filemode format options
Files> to copy  <====>  |          A      CMS      WTH
Record format (OS simulated files only)  <====>

If preferred, give a LIST with filenames, formats and options:
LIST to copy  <====>  |          A

copy from disk  <====>  owner:   address   (none=files sent
with BATCH SUBMIT)

tape attributes <====>  UID      Density  Label
for labelled tapes, give:
                                the output tape USN <====>
the filename of file where copying is to start <====> 0001
                                the dataset name of the first file <====> FILE1

TIME for BATCH job <====> 5:00      ( @:00 for a 'dry' run )

```

Figure 52: The Disktape command

## 9.5 The Space Machine

The directory entry defining the disk configuration of a virtual machine may be changed with the DIRM command. However, certain changes are privileged and may not be done directly by general users. For this reason the space machine was developed (at SLAC). Requests may be sent in a specific format to this service machine which subsequently does some checking and passes on the request to the DIRM machine.

At CERN the user community is divided into groups. Each group has a space administrator who is supposed to organise the resources in a way which suits the group best. A certain amount of the functions offered by the space machine are therefore only available to the administrators. The most useful commands are (type **HELP SPACE MENU** for a menu of all the available commands):

**SPACE ADDDisk userid his\_address ncyls**

This administrator command may be used to add a permanent minidisk of ncyls cylinders to userid's directory entry at address his\_address. Userid should wait until the message FROM DIRMAINT: DVHMCB009I DIRECTORY UPDATE ONLINE ... appears. Before the disk may be used, it must now be FORMATTed with the command **GIME userid his\_address ( FORMAT**

**SPACE ADDTemp my\_address ncyls**

This general user command may be used to add a temporary disk of size ncyls (not more than 10) at address my\_address. The disk will remain part of the virtual disk configuration even after logging off, until 2.00 a.m. on the following Monday morning. See the RETURN parameter how one may recuperate it at that time. The disk is formatted automatically by the space machine.

**SPACE ADMIN Query**

This general user command may be used to determine who the administrator is in a given group. For administrators the ADD operand may be used instead of Query to create additional administrators in the group.

**SPACE CHNGSize** userid his\_address ncyls pw

This administrator command may be used to change the size of **userid's** minidisk at address **his\_address** to a new size of **ncyls** cylinders. **pw** is the read password of this disk (if it has one).

**SPACE DROPDisk** userid his\_address

This administrator command may be used to delete **userid's** minidisk at address **his\_address** from his machine's directory entry.

**SPACE DROPTemp** address

This general user command may be used to delete the temporary disk at **address** from the disk configuration.

**SPACE LIMIT** Query [gg]

This general user command may be used to list the disk space used for group **gg** (default is the users own group).

**SPACE RETurn** spacemachine\_address user\_address

This general user command may be used to reallocate a temporary disk after being reclaimed by the space machine on Monday morning. Use the **TEMPDISK** operand to determine which disks are still intact and what the space machine's address of the required disk is.

**SPACE SETpass** userid newpassword

This administrator command may be used to set **userid's** new logon password to **newpassword**. Very useful for users who forget their logon password.

**SPACE SPLITDsk** address cyls naddr1 ncyls1 naddr2 ncyls2 ...

This general user command may be used to reconfigure his minidisks without the intervention of the administrator. Minidisk at **address** is to be given a newsize of **cyls** cylinders. This must be large enough to contain all the material that is already on this minidisk. The remaining free space is then split into new minidisks at addresses **naddr1**, **naddr2**, ... with sizes of **ncyls1**, **ncyls2**,... The total size of all these minidisks must not be greater than the original minidisk.

**SPACE TEMPdisk** Query RETurn|RECLAIM

This general user command may be used to list temporary disks that have been reclaimed by the space machine. For a general list, omit the **RETurn** or **RECLAIM** operand.

Please note that some of these operations require R/W access of the space machine to a minidisk. This disk must therefore be dropped before issuing the command.

## 9.6 Exercises for Chapter 9

1. Set a delivery code for Xprint and Xscript via the command

**DIRM DIST delivery\_code**

The delivery code should be set to LOCAL if no delivery is required.

2. Try the

**DEFAULTS SET XSCRIPT**

command to change your default output device for SCRIPT to e.g. the APA6670.

3. Try the command

**VMBLIST PROFILE XEDIT ( RESTORE**

to restore your file PROFILE XEDIT.

4. Find out who the space administrator is in your group.

## 10. Creating and running programs

### Summary of Chapter 10.

At the end of this chapter, you should have learnt how to;

- Use the VFORT exec to invoke the VS-FORTRAN compiler.
- Run interactive FORTRAN programs.
- Use the FILEDEF command.
- Use the BATCH machine.

The programming languages that are available on VM include VS FORTRAN, VS PASCAL, C (Waterloo), COBOL, H ASSEMBLER, VS PROLOG. Since Fortran is the most commonly used language on the central computers, the examples and exercises in this chapter will mainly refer to Fortran. However, it is not necessary to have any prior knowledge. The concepts explained here are basic to all languages.

The VM User's Guide [2] contains an excellent chapter on how to run Fortran programs. The serious Fortran programmer should read the relevant chapter after working through the material presented here.

### 10.1 Invoking the Fortran compiler using the VFORT command

When creating a program written in a given language, the filetype is used to indicate the name of the language. For example, a program written in Fortran would have a filetype of FORTRAN. Enter the editor and create a simple Fortran program called HICERN FORTRAN A, the code of which is listed in Figure 53 on page 100.

```
PROGRAM HICERN
CHARACTER*40 NAME
1 FORMAT(A)
WRITE(6,*) 'WHAT IS YOUR NAME '
READ(5,1) NAME
WRITE(6,*) 'HI, CERN! FROM',NAME
STOP
END
```

Figure 53: A simple Fortran Program



This program may mean something to those who know Fortran language. However, it means nothing at all to the computer, who only understands **machine language**. To translate Fortran source into machine language, a special program called a **compiler** has to be run. The Fortran compiler on VM is the IBM VS Fortran compiler. To invoke it, i.e. translate Fortran source into machine or **object code**, use the command

### VFORT HICERN

This will produce a response as shown in Figure 54 on page 101.

```
R; T=0.06/0.09 09:38:34
vfort hicern
CRNUFT000I File 'HICERN FORTRAN R1' will be processed.
US FORTRAN COMPILER ENTERED. 09:38:46

**HICERN** END OF COMPILATION 1 *****
US FORTRAN COMPILER EXITED. 09:38:46

R; T=0.03/0.07 09:38:46

RUNNING CERNUM
```

*Figure 54:* Response from the VFORT command

If typing or other errors were detected, they would be displayed. Type **FILEL HICERN \* A** to discover that the Fortran compiler produced two output files:

1. The file **HICERN TEXT** containing the object code generated from the source program **HICERN FORTRAN**.
2. The file **HICERN LISTING** containing the compiler output listing. This contains a listing of the source, a map and error messages.

Default options can be set using the **DEFAULTS SET VFORT** command. Compiler options may be specified on the **VFORT** command:

```
VFORT HICERN ( NOMAP
VFORT HICERN ( SDUMP
```

**SDUMP** requests a symbolic dump after a program has aborted.

## 10.2 Running a compiled Program

Before the compiled program may be executed, the system has to be informed about various libraries (so called **TXTLIB's**) containing other compiled programs which are necessary during the execution, such as the CERN program and VS Fortran libraries. This is done using the

### CERNLIB

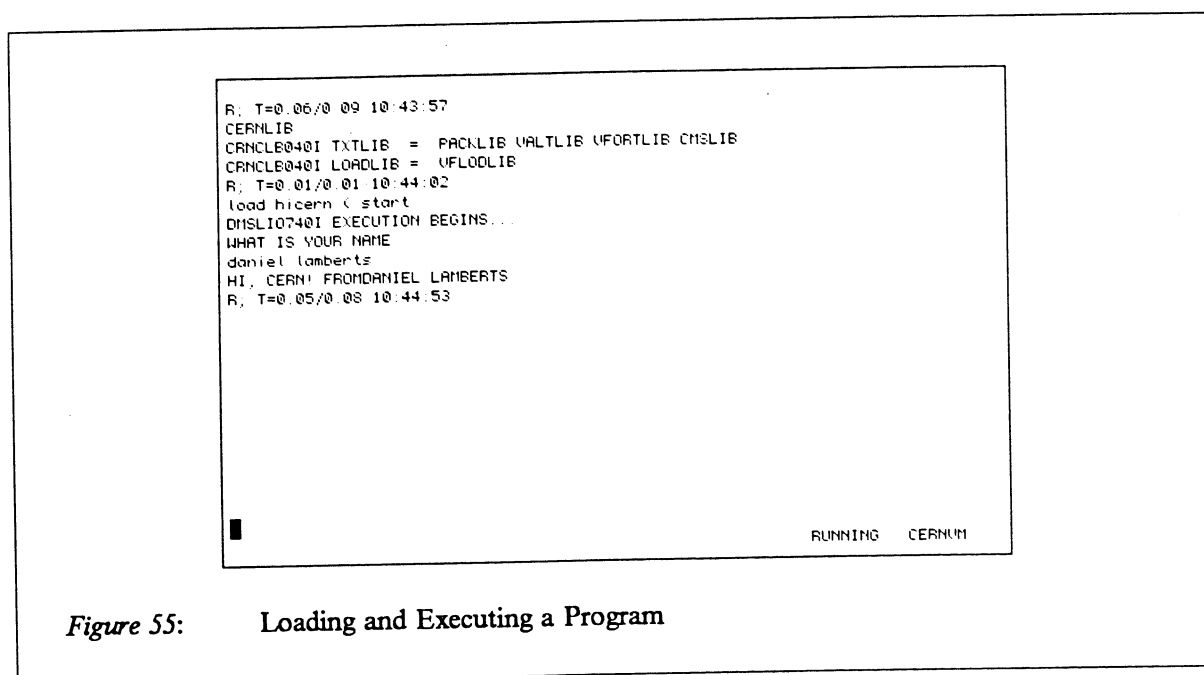
command. Any user defined **TXTLIB** should be given as argument to the **CERNLIB** command, for example

### CERNLIB GENLIB NAGLIB GROUPLIB

To load the compiled program into storage and execute it, use the **load** command with the **start** option:

### LOAD HICERN ( START

The system will respond as shown in Figure 55 on page 102.



```
R: T=0.06/0.09 10:43:57
CERNLIB
CANCLE0401 TXTLIB = PACCLIB VALLIB VFORTLIB CHSLIB
CANCLE0401 LOADLIB = VFLODLIB
R: T=0.01/0.01 10:44:02
load hicern ( start
DMSL107401 EXECUTION BEGINS...
WHAT IS YOUR NAME
daniel lamberts
HI, CERN! FROM DANIEL LAMBERTS
R: T=0.05/0.08 10:44:53
```

RUNNING CERNUM

*Figure 55:* Loading and Executing a Program

Please note that the program changes the status message to **VM READ**, indicating that a reply is expected from the terminal.

Hit return (which corresponds to an end of file).

### 10.3 Defining Input/Output Files for Programs

By default units 5 and 6 are defined to be the terminal:

```
READ          unit 5
PRINT/WRITE   unit 6
```

All other units are disk files. A program will fail if no input file is defined. If no output file is defined, the system will create a disk file with

```
filename = FILE
filetype = FT16F001 (if there was a WRITE(16 ... )
```

Files maybe defined with the command

**Filedef**

VS Fortran issues the commands

```
Filedef 5 TERMINAL ( BLKSIZE 80
Filedef 6 TERMINAL ( BLKSIZE 133
```

thus reading input from the terminal and directing output back to the terminal. To read the file HICERN DATA from FORTRAN unit 9, a command

```
Filedef FT09F001 DISK HICERN DATA A ( PERM
```

would be required (PERM is necessary because VS Fortran will reset it otherwise). To write a file, define it in the say way before creating it:

```
Filedef 6 DISK HICERN OUTPUT A ( PERM
```

### 10.4 The Batch Machine

When executing programs interactively, your machine is unable to be used for anything else. For long jobs, it is convenient to send them to the Batch machine for execution, so that you may continue using your own machine. If you are using magnetic tapes, you will **have** to use the batch machine, as no other machine is allowed access to them.

There is no difference between a job running in your machine or the batch machine, except the Batch machine is not able to read data from a terminal. It is therefore possible to debug a job interactively and then run a production version in the batch machine.

The VM User's Guide [2] contains a good description of the batch machine. We therefore only mention here the command

```
BATCH SUBMIT execname
```

which sends a job to the batch machine. An example of a complete job (HICERN EXEC) is:

```
/* Simple job */
"VFORT HICERN"
"CERNLIB"
"FILEDEF 5 DISK HICERN DATA * ( PERM"
"FILEDEF 6 DISK HICERN OUTPUT A ( PERM"
"LOAD HICERN ( START"
```

To execute this file, enter the command HICERN, to run it in batch use the command

```
BATCH SUBMIT HICERN
```

See sect. 7.4 on page 73 for a short summary about exec files.

## 10.5 Exercises for Chapter 10

### 1. Interactive VS Fortran compile, load and execute:

- a. Create a simple fortran program called TEST1 FORTRAN on your A-disk (statements start in column 7, continuation symbols are put in column 6 and labels may be in columns 1 to 5):

```
PROGRAM TEST1
  READ 100,I
  PRINT *,I
  STOP
100 FORMAT(I8)
END
```

Compile this program; use CERNLIB to declare the necessary libraries and execute it.

- b. Edit the file TEST1 FORTRAN and add the following prompt after the PROGRAM statement:

```
PRINT 102
102 FORMAT(' ENTER VALUE IN I8 FORMAT')
```

- c. Recompile and execute TEST1 FORTRAN again.

### 2. Load a text file & execute it with input card image data on disk:

- a. Create a data file TEST1 DATA on your A-disk containing one line with an integer.
- b. Now give a FILEDEF command allowing TEST1 to read data from this file rather than from the terminal:

```
FILEDEF 5 DISK TEST1 DATA A
```

Type Query FILEDEF to see if you did it correctly.

- c. Reload and re-execute TEST1. Note TEST1 does not have to be recompiled.
3. VFORT, LOAD & EXECUTE. Writing unformatted data on disk with no filedef:
- a. Create a file TEST2 FORTRAN containing the following lines:

```

PROGRAM TEST2
DIMENSION X(100)
DO 100 I=1,10
DO 50 J=1,100
50 X(J)=FLOAT(I*J)
WRITE(20) X
100 CONTINUE
WRITE(6,110)
110 FORMAT(' 10 RECORDS OF 100 NUMBERS ARE NOW ON DISK')
STOP
END

```

Compile this job using VFORT; load and execute it.

- b. Now type FILELIST and look at the result of the WRITE(20) statement in TEST2. Type:

```

X FILE FT20F001
SET HEX ON
:1 HEXTYPE

```

This allows you to look at hexadecimal numbers in the datafile.

4. VFORT, LOAD & EXECUTE. Reading unformatted data on disk with no filedef:
- a. Now type in the program TEST3 FORTRAN to read the above data file:

```

PROGRAM TEST3
DIMENSION X(100)
NREC=0
10 READ(20,END=40) X
NREC=NREC+1
GOTO 10
40 PRINT 41,NREC
41 FORMAT(' END OF FILE DETECTED ON FILE 20'/
X ' NUMBER OF RECORDS READ=',I5)
STOP
END

```

- b. Compile, load and execute this program.

5. Load text & execute. Writing and reading unformatted data on disk with filedef:

- a. Issue the following filedef command to define a disk file for unit 20 for program TEST2:  
**FILEDEF 20 DISK JOB36 DATA A4 (RECFM VBS LRECL 404 BLOCK 408**  
Load TEST2 and execute it.
- b. Xedit the file TEST2 DATA as before with FILE FT20F001.
- c. Issue the command Q FILEDEF to see if the filedef is still in effect. Load TEST3 and execute it.

## B.2 Minidisk passwords

The machine owning a disk may set read and write passwords on this disk to restrict the R/O and R/W link requests from other machines. By default all minidisks have a read password ALL, meaning anyone may establish a R/O link to the disk. However, by default all minidisks have no write passwords, implying that only the disk's owner may establish a R/W link to it. A disk containing confidential material should therefore have its read password removed. Please note that the Batch machine may establish a link to a minidisk as if it were the owner, i.e. without giving passwords (although R/W links will only be established after the real owner has dropped the disk).

## B.3 Setting default links and passwords using DIRM

CP establishes default R/W links at logon time to all minidisks defined in the directory of a machine. These are normally only the minidisks which are owned by that machine. To establish a default link to someone else's minidisk with a given link access mode a logon time (i.e. add a link to the directory), the

### DIRM LINK

command should be used. This produces a panel as shown in Figure 56 on page 109.

```

Enter LINK Parameters. Required items are BRIGHT
Press -->| (left side, above LOCK key) to skip from field to field.
If INPUT INHIBIT you have typed outside the field- press RESET. To cancel this
transaction, press CLEAR. Fill all applicable fields before pressing ENTER.

General information about linking is available by pressing PF10.
You will have to specify a LINK-MODE. For information push PF12.
(Do the above before entering data - any items entered will be lost.)

owner userid:      ===> █
owner minidisk address: ===>
your link address:  ===>
your link access mode: ===> RR      (or DELETE to delete existing link)
owner's access password ===>

Access password is not required for DELETE,
or if it is ALL (owner has made it public).

```

Figure 56: Setting a default link to a minidisk

The DIRM LINK command would normally be used for links to other user's minidisks. Please remember one needs to access this disk in PROFILE EXEC after the link has been made:

**ACCESS your-address your-letter**

(see 7.4.1 on page 74). The DIRM LINK command would also be used to establish a link to a mini-disk owned by machine, if this link should not be R/W but R/O. This is sometimes desirable if the disk contains important material and the user wishes to protect him/herself from accidentally overwriting it.

To find out which links are defined in your CP directory, use the

### DIRM REVIEW

command. You will be prompted with your CP password, and be sent a file containing a copy of your directory entry to your reader (see Figure 57 on page 110).

```

2104 PEEK  A0 U 80 Trunc=80 Size=11 Line=0 Col=1 Alt=0
File DUPDOM DIRECT from DIRMAINT at CERNVM. Format is PUNCH.
*** Top of File ***
USER DUPDOM XXXXXXX 06144K 6M G                                01150903
ACCOUNT DOM$CP LOCAL                                          10270948
IPL CMS PARM AUTOCR                                          10270948
CONSOLE 009 0270                                             10270948
LINK CERNSYS 19A 19A RR                                       10270948
MDISK 193 3380 69 10 UNUS45 MR XXXXXXX                      02121139
MDISK 191 3380 674 5 UNUS06 MR XXXXXXX                      02101627

SETOPTN  SETTINGS :
LOGGING  OFF
1= Help      2= Add line  3= Quit    4= Tab     5= Clocate  6= ?/Change
7= Backward  8= Forward  9= Receive 10= Rgtleft 11= Spltjoin 12= Cursor

====> █                                                    X E D I T  1 File

```

Figure 57: The CP Directory Entry with Default Links to Minidisks

To set a read or write password on a disk, the

### DIRM MDISK

command should be used (see Figure 57).

Please note that at CERN the system has been modified to disable MULTI passwords being set, although the panel indicates otherwise. The most useful setting (to allow others to write to a disk as well) is accessmode MR with Read password ALL and a WRITE password of your choice.

To find out the passwords on any minidisk owned by your virtual machine, issue the command

### DIRM MDPW



Enter **MDISK** Parameters. **Required** items are BRIGHT  
 Press -->] (left side, above LOCK key) to skip from field to field.  
 If INPUT INHIBIT you have typed outside the field- press RESET. To cancel this  
 transaction, press CLERR. **Fill all applicable fields before pressing ENTER.**

For information on: (Before making entries) Press:	Mode	Passwords
	PF10	PF12

Minidisk Address: ===> █

The remaining fields are preset to = (skip those to be left alone)

Access mode : ===> = To change, enter: R RR W WR M MR MW  
 (add U to specify virtual RESERVE/RELEASE)

To modify passwords, enter new value. To remove existing passwords,  
 type a blank as the first character of the password to be  
 deleted. REMOVING A PASSWORD ALSO REMOVES ANY BELOW IT.

Read password : ===>

Write password : ===> (READ password required)

Multi password : ===> (READ & WRITE passwords required)

*Figure 58:*      **Setting a Read or Write Password on a Minidisk**

---

## Appendix C

### The Wylbur Bridge

Although a Wylbur editor is available on VM, only a limited number of commands is available. For example, the `exec` command is not available. REXX is a much better language to write execfiles in anyway. If you wish to use your Wylbur account from VM, you may do so (for short period) with the

#### BRIDGE

command. The system will respond with

#### INTERFACE TASK:

and offer you the default reply `MILTEN` on the command line. You reply by hitting RETURN. The system responds with

#### HOST:

an offer you the default reply `IBMA` on the command line. You reply by hitting RETURN and the system will offer you the usual Wylbur login prompts.

Once logged in to Wylbur, hitting PA1 twice acts as BREAK key. Remember you may address the screen and use function keys as if you were using CMS; however, this will be of limited use. To use the settings of functions set by Wylbur, use the sequence `:Fn` where `'F'` is the Wylbur escape character and `n` is the number of the function.

If CERNET is down, you will obtain messages such as

```
:NET
```

To return to CMS in this case type

```
EXIT
```

key	IBM Function
ENTER	Execute the commands on the screen
↑	Cursor up
↓	Cursor down
←	Cursor left
⇒	Cursor right
→	Move cursor forward to first position of next input field, (TAB). (On the left of the keyboard for all terminals except PA.)
←	Move cursor back to first position of current input field or preceding input field, (backwards TAB).
INSERT MODE	Is a flip/flop. Hit it once to get into insert mode, hit it + once more to get out. On most CERN terminals there is no 'INSERT MODE' indicator as on a real IBM 3270. The terminal will beep at you if you are still in 'INSERT MODE' when you try to type in certain special areas of the screen, eg prefix area. Press the 'INSERT MODE' key a second time to get out of insert mode.
ERASE EOF	Erase end-of-field. Erases all characters from the + current position of the cursor to the end of the field, normally the end of the line.
CLEAR SCRN	Refreshes the screen, normally when MORE... or HOLDING appears in the bottom right hand corner of the screen
DEL CHAR	Delete the character under the cursor
LINE FEED	Move the cursor to the beginning of the next line
CTRL-G	Serves as a general clear function if your terminal is + blocked - It may start beeping at you or the cursor may get stuck in the bottom + right-hand corner of the screen. A common cause of this, especially for Wylbur users, is hitting the BREAK key.

**E.2.1 D2 terminals, right hand end of the keyboard**

→|(TAB) is on the extreme left of the keyboard. PF keys 13-24 are assigned to keys F6-F14 and F17-F19 along the top of the keyboard.

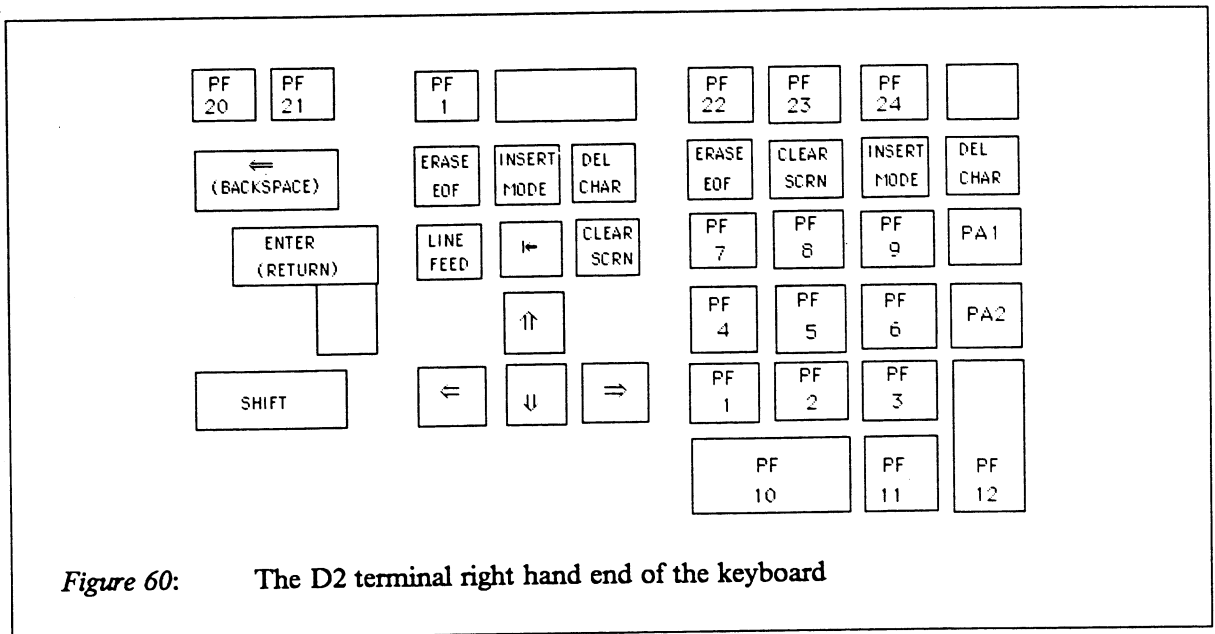


Figure 60: The D2 terminal right hand end of the keyboard

**E.2.2 PA terminals, right hand end of the keyboard**

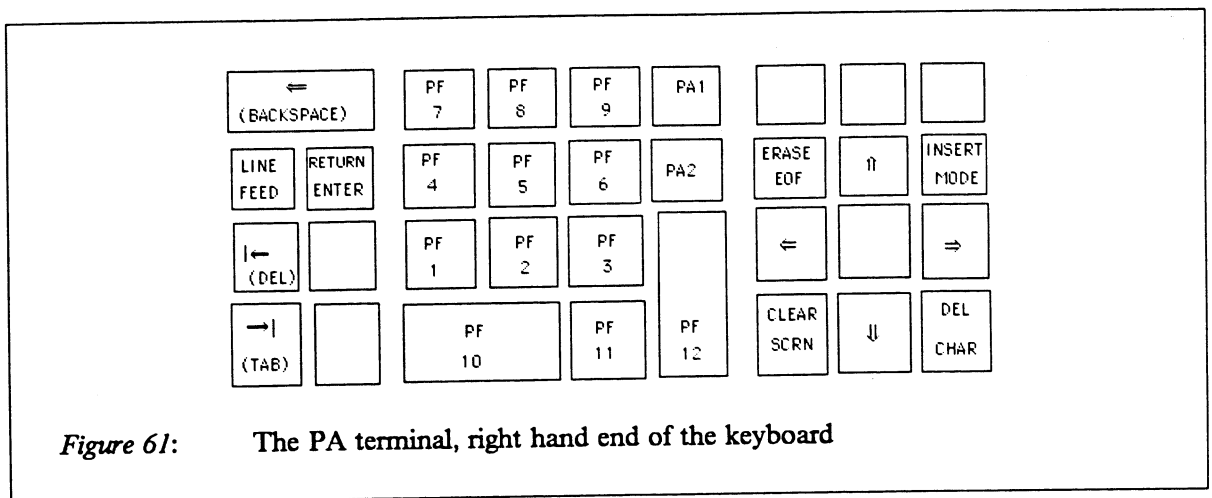
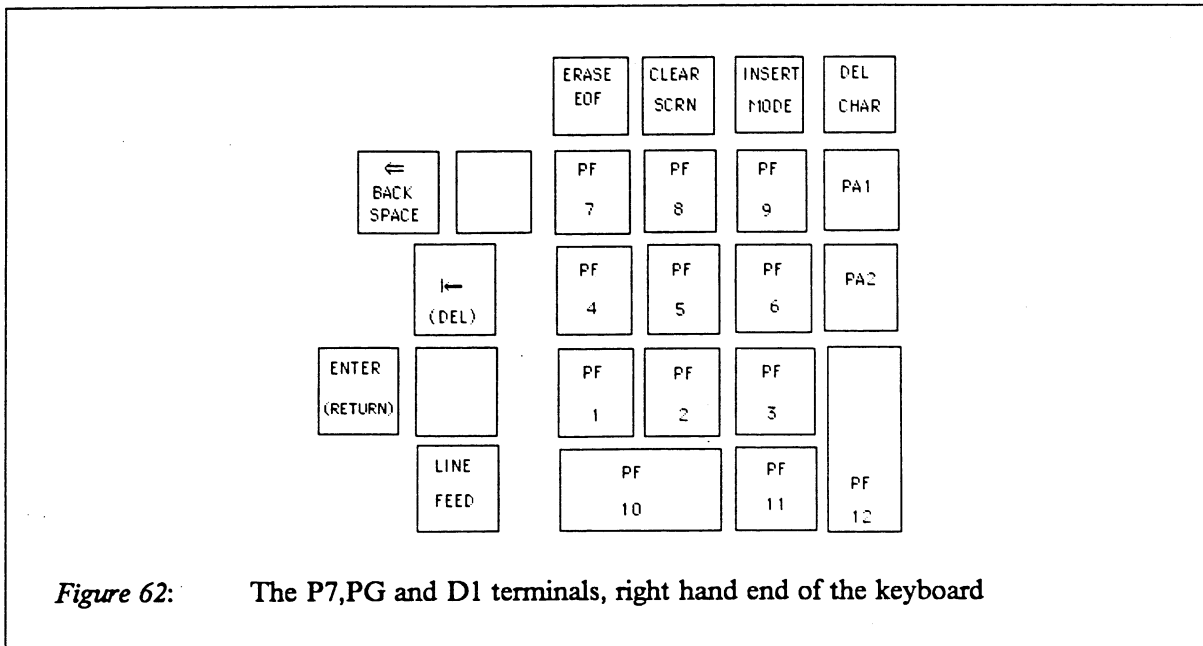


Figure 61: The PA terminal, right hand end of the keyboard

**E.2.3 P7,PG and D1 terminals, right hand end of the keyboard**

→|(TAB) is on the extreme left of the keyboard



### E.3 Locally added support for line mode terminals (TTY)

1. SCROLL mode is supported for any line mode terminal. For example

#### CP TERMINAL SCROLL 23

will make the terminal stop listing and display 'MORE' after 23 lines. For more information see HELP CP TERMINAL.

2. Function keys can be simulated by entering the logical linend character followed by a number from 1 to 24 and then pressing <enter>. The logical linend character is by default # (number or hash sign) but can be changed by the CP TERMINAL command.

### E.4 Use of Pericomms as VM Terminals

In the normal case it should not be necessary to take any special actions in order to use a Pericom as a VM terminal. However the newer Pericom terminals, marked "GRAPH", "ALPHA" or "GRAPH PAC" do have a SETUP function which must be correctly set for VM but can be easily changed by a user. If your terminal is behaving in a strange way check that it correctly SETUP as indicated below.

On PG and P7 terminals, the row of grey function keys along the top of the terminal CANNOT be used.

On some PA terminals with 24 PF keys along the top of the keyboard these PF keys can be used instead of the numeric pad. However this requires that the firmware in the terminal matches the keyboard and it is a matter of luck whether this is the case. By default PF keys 13-24 are set to the same thing as PF keys 1-12, but you can redefine them yourself if you wish.

#### E.4.1 Pericom 6803 – standard 'old' Pericom (PA)

- Terminal type: PA or PA3270
- Switch settings at the back: 11 0 6 5.

Note that this is the normal setting so it should normally not be required to change it. The settings are explained on the back of the terminal and in the Pericom Ansi User's Guide pp. 3–5.

#### E.4.2 Pericom "ALPHA"

- Terminal type: P7 or P73270
- To alter the contrast: to increase, press SETUP and then ↑ repeatedly; to decrease press SETUP and then ↓ repeatedly.
- A reasonable SETUP for the Pericom graphics terminal "ALPHA" is:

SETUP followed by B for the different options.  
 SETUP B  
 1: 0101 2: 0011 3: 0100 4: 0011 5: 0001 6: 0100

To keep a changed setup type <SHIFT> <S>.  
 Press SETUP to return to your session.

Note that this is the normal setting so it should normally not be required to change it. The SETUP options are described in more detail in section 2.3 of the Pericom manual.

#### E.4.3 Pericom "GRAPH"

Note that the setups for the Pericom GRAPH and GRAPH PAC terminals are slightly different.

- Terminal type: PG or PG3270
- To alter the contrast: to increase, press SETUP and then ↑ repeatedly; to decrease press SETUP and then ↓ repeatedly.
- A reasonable SETUP for the Pericom graphics terminal "GRAPH" is:

SETUP followed by B or G for the different options.  
 SETUP B  
 1: 0101 2: 0011 3: 0100 4: 0011 5: 0001 6: 0100  
 SETUP G  
 p: 0111 q: 0001 r: 1000 s: 0000 t: 0010 u: 0000  
 SETUPs C, D, E and F should be all zero

To keep a changed setup type <SHIFT> <S> .  
Press SETUP to return to your session.

Note that this is the normal setting so it should normally not be required to change it. The SETUP options are described in more detail in section 2.3 of the Pericom manual.

- If the terminal is in graphics mode it can be reset by the (ALPHA) MODE / GRAPH key.

#### E.4.4 Pericom "GRAPH PAC"

Note that the setups for the Pericom GRAPH and GRAPH PAC terminals are slightly different.

- Terminal type: PG or PG3270
- To alter the contrast: to increase, press SETUP and then ↑ repeatedly; to decrease press SETUP and then ↓ repeatedly.
- A reasonable SETUP for the Pericom graphics terminal "GRAPH PAC" is:

SETUP followed by B or G for the different options.

SETUP B

1: 0101 2: 0011 3: 0100 4: 0011 5: 0001 6: 0100

SETUP G

q: 0111 r: 0011 s: 1100 t: 1110 u: 0000 v: 0000

SETUPs C, D, E and F should be all zero.

To keep a changed setup type <SHIFT> <S> .  
Press SETUP to return to your session.

Note that this is the normal setting so it should normally not be required to change it. The SETUP options are described in more detail in section 2.3 of the Pericom manual.

- If the terminal is in graphics mode it can be reset by the VDU and GRAPH keys. To clear the graphics screen type <CTRL> <CLR> .

#### E.5 Use of the DMT2200 as a VM terminal

- Terminal type: D2 or D23270

For the SETUP see Table 1.

## E.6 Use of the Tandberg as a VM terminal

- Terminal type: TDV
- A reasonable SETUP for the Tandberg terminal is:
 

parity:	none
code length:	8 bits
stop bits:	2
- The keyboard layout is as similar as possible to other CERN terminals.
 

ERASE TO END OF FIELD	DELETE (G47) or F5 (E51)
CLEAR SCREEN	CANCEL (D48) or F6 (E52)
INSERT MODE	APPND/EXPND (D99) or F7 (E53)
DELETE CHAR	a (E14) or F8 (E54)
PA1	(D54)
PA2	– (C54)
PFK1-9	Numeric pad 1-9 (D51-D53,C51-C53,B51-B53)
PFK10	Numeric pad 0 (A51)
PFK11	Numeric pad . (A53)
PFK12	Numeric pad ENTER (B54)
Cursor movement:	
backward tab	(A47)
forward tab	(A49)
cursor left	(B47)
cursor right	(B49)
cursor up	(C48)
cursor down	(A48)
cursor home	(B48)
LINE FEED	(D13)
ENTER (CR)	(C13)

## E.7 Use of the Macintosh Plus as a VM terminal

It is recommended to use VersaTerm PRO as the terminal emulator for the Macintosh. It provides VT100 emulation, allowing the Macintosh to be used as terminal type D1 and contains Kermit, the CERN standard protocol for file transfer. VersaTerm PRO can be bought by contacting F. Saldana / EP, phone 6256 or electronic mail SALDANA@VXCRNA.

### E.7.1 VersaTerm Settings

The following settings have been used successfully to access CERN machines.

File:	Kermit
Baud:	4800
Settings:	Map BS&DARRIDEL
Parity:	None
Char size:	8 Bits



Stop bits: 1.0  
Emulation: DEC VT100

### E.7.2 Keyboard layout

Using VersaTerm and VT100 emulation the terminal type should be D1. The special IBM functions are assigned to the numeric keyboard pad with the same layout as for D1 terminals. **LINE FEED** is obtained by typing **RETURN** and the **CONTROL KEY** (just to the left of the space bar) simultaneously.

### E.7.3 File Transfer

Kermit should be used for file transfer. It must be activated on both the Macintosh and the IBM. To activate Kermit on the IBM side enter the command

**KERMIT**

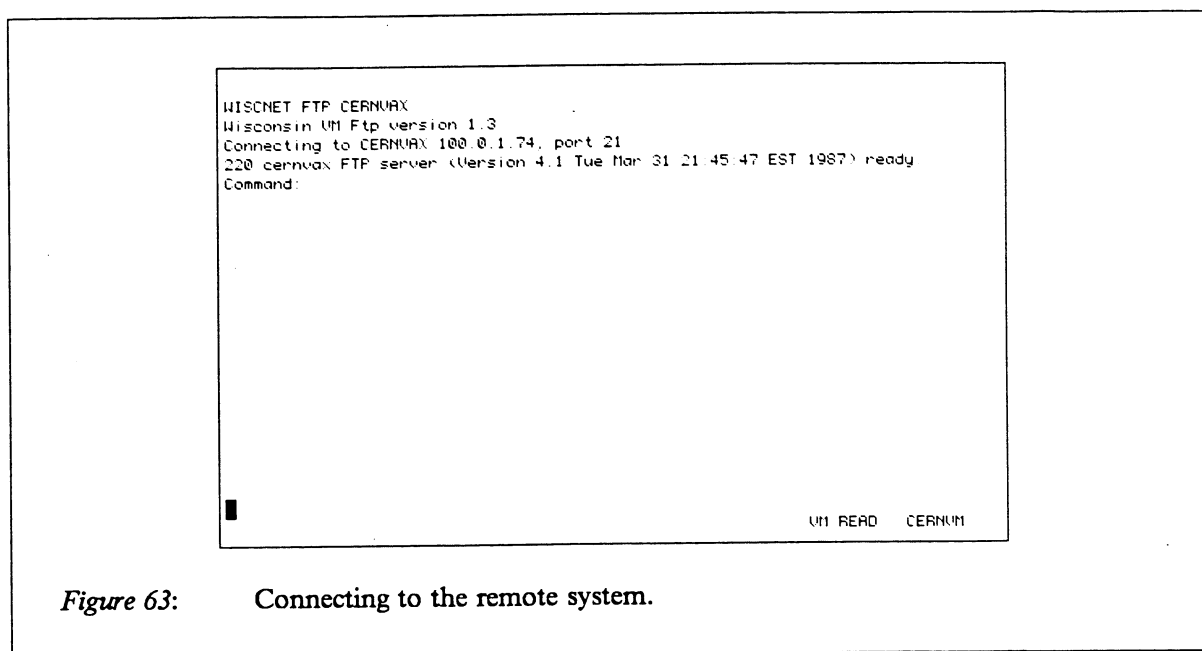
More information on the IBM side of Kermit can be obtained via the command **HELP KERMIT**.

More information on using the Macintosh can be found in the document 'Graphics Applications on the Macintosh for SGML documents', by F.G. de Bilio, DD/US/105, available from the self-service area of the UCO.

## Appendix F

### FILE TRANSFER USING WISCNET

This is how to use ftp for transferring a file to a remote system when you are logged on to the VM/CMS system:



Once connected, ftp enters command mode awaiting further instructions. Typing help in command mode gives a list of WISCNET ftp commands. To log in to the remote system see Figure 64. Then use one of the following commands to transfer files:

```
GET remotefile localfile
SEND localfile remotefile
```

Where localfile is of the form fn.ft.fm For example:

```
GET ufile vmfile.exec.a
SEND vmfile.exec.a ufile
```

Once all transfers are complete, quit is used to exit from ftp. For more details see [14].

```
LOGIN lex
>>>USER lex
331 Password required for lex.
Password:
>>>PASS *****
230 User lex logged in.
Command:

UM READ CERNUM
```

*Figure 64:* Supplying login information.

---

## Appendix G

### References

1. CMS Primer. IBM SC24-5236-2
2. VM – CMS at CERN, Introductory Users Guide
3. Systemes VM et PC XT/370, M. Eisinger, Editests 1985.
4. System Product Editor Command and Macro Reference Manual. IBM SC2 4-5221-3
5. CMS Command and Macro Reference Manual. IBM SC19-6209&hype n.3
6. CMS User's Guide. IBM SC19-6210-3
7. CP Command Reference for General Users. IBM SC19-6211-3
8. Cernpaper User's Guide, User Support Group, 3rd reprint January 1986, DD/US/50.
9. A User Guide to Electronic Mail Services at CERN, CS Group, May 1986, DD/US/6.
10. MAIL/MAILBOOK version, G. R. Schafer, 86 – 359, Rice University, 1986.
11. Terminal User's Guide, M. Brodmann, April 1987, DD/US/106
12. Interlink User's Guide, J. Bunn, DD/US/107.
13. SGML User's Guide, Anders Berglund, October 1986.
14. A Guide to TCP/IP Networking at CERN, May 1987, DD/US/108.

## Appendix H

### Index

- < (prefix) ... 28
- / (prefix) ... 28
- > (prefix) ... 28
- ^ (prefix) ... 28
- = sign inside commands ... 56
- access ( noprof (CMS) ... 74
- access (CMS) ... 52
- accessmode ... 52
- ace (cms) ... 114
- add (prefix) ... 28
- add (Xedit) ... 23
- adddisk (space) ... 98
- addtemp (space) ... 98
- admin (space) ... 98
- align (prefix) ... 28
- all (Xedit) ... 33
- all notebook ... 82
- alteration count ... 21
- arbchar (Xedit) ... 35
- arbitrary character (Xedit) ... 35
- archiving ... 94
- assembler ... 101
- autosave (Xedit) ... 20
- back (Xedit) ... 26
- backing up files ... 94
- batch mode ... 4
- begin ... 10
- Bitearn nodes ... 78
- blank separator for CMS ... 59
- block commands (prefix) ... 29
- blocksize ... 48
- bottom (Xedit) ... 26
- bridge (Wylbur) ... 113
- browse (CMS) ... 56, 58
- c language ... 101
- calculator ... 114
- cancel (Xedit) ... 39
- card punch/reader ... 4
- case (Xedit) ... 34
- category (help) ... 66
- centre (prefix) ... 28
- cernlib (CMS) ... 103
- cernpaper ... 93
- changing data (Xedit) ... 37
- changing modes ... 53
- clocate (Xedit) ... 30
- cobol ... 101
- command line ... 22
- compare (CMS) ... 56
- component (help) ... 66
- console ... 4
- console log ... 60
- copy (prefix) ... 28
- copyfile (CMS) ... 56
- crmpdisk owners ... 50
- current files (Xedit) ... 42
- current line ... 22
- defaults (CMS) ... 74
- define storage (CP) ... 48
- delete (prefix) ... 28
- delete (Xedit) ... 24
- deleting columns ... 38
- delivery code ... 91
- dial lterm ... 7
- directory ... 11
- dirm ... 74
- dirm (CMS) ... 11
- dirm dist (CP) ... 92
- dirm link (CMS) ... 110
- dirm mdisk (CMS) ... 110
- dirm mdpw (CMS) ... 110–111
- DIRM REVIEW ... 111
- dirm storage ... 48
- discard (CMS) ... 56, 58
- disconnect ... 10
- disconnected ... 4
- disks ... 4
- disktape (CMS) ... 96–97
- distribution code ... 91
- DMT2200 terminals ... 115, 120
- down (Xedit) ... 26
- drop (CMS) ... 51
- dropdisk (space) ... 99
- droptemp (space) ... 99
- dumptape (CMS) ... 96
- duplicate (prefix) ... 28

- D1 terminals ... 115, 117
- D2 terminals ... 115–116
  
- EARN/BITNET nodes ... 78
- editor ... 19
- EMDIR ... 85
- emsg (CP) ... 71
- equals sign inside commands ... 56
- erase (CMS) ... 56
- eval (cms) ... 114
- exclude (prefix) ... 28
- executing programs ... 103
- export (CMS) ... 88
- extend (prefix) ... 28
  
- file ... 18
- file (Xedit) ... 19
- file transfer wylbur ... 87
- fileid ... 18
- filelist (CMS) ... 53
- filemode ... 18, 55
- filename ... 18
- files ... 18
- filetype ... 18
- find (CMS) ... 64
- following (prefix) ... 28
- forgotten CP password ... 11, 99
- format (cms) ... 49
- fortran ... 101
- forward (Xedit) ... 26
- function keys (CP) ... 70
- function keys (Xedit) ... 41
  
- get (prefix) ... 28
- getfile (cms) ... 87
- gime (CMS) ... 50
- gone ... 11
- graphics terminals ... 115
  
- halt execution ... 75
- halt typing to terminal ... 75
- HELP ... 63
- help file ... 66
- HOLDING (status message) ... 15
- HT (CMS) ... 75
- HX (CMS) ... 75
  
- IBM PC as a terminal ... 115
- IBM 3101 terminals ... 115
- identify (CMS) ... 77
- impcp (CMS) ... 16
- import (CMS) ... 87
- insert (prefix) ... 28
  
- item (help) ... 66
  
- jobs locating ... 89
- justify (prefix) ... 28
  
- left (Xedit) ... 26
- lfile (CMS) ... 53
- libraries importing ... 87
- limit (space) ... 99
- line mode ... 118
- linend character (Xedit) ... 22
- link (CP) ... 52
- linking disks in batch\* ... 110
- listfile (CMS) ... 53
- LISTSERV ... 85
- load (CMS) ... 103
- local distcode ... 92
- locate (CMS) ... 89
- locate (Xedit) ... 30
- logged on users ... 80
- logging off ... 10
- logging on ... 6
- logmsg ... 8
- logon password ... 11, 99
- lowercase (prefix) ... 28
- lowercase (Xedit) ... 34
- LTERM ... 4, 7
  
- Macintosh terminal ... 121
- macros (XEDIT) ... 41
- magnetic tapes ... 96
- mail ... 83
- mail (CMS) ... 57
- mail to TSO/E sites ... 92
- menu (help) ... 66
- message line ... 21
- messages to other users ... 80
- minidisk passwords ... 110
- minidisks ... 4, 48
- mode (file) ... 54
- modes swapping ... 53
- MORE... (status message) ... 15
- move (prefix) ... 28
- moving inside xedit ... 26
- msg (CP) ... 80
- mvs locating jobs ... 89
- mvspurge (CMS) ... 89
  
- names (CMS) ... 77
- names file ... 77
- news (CMS) ... 74
- next (Xedit) ... 26
- NFT command ... 88

- nickname ... 77
- node name ... 77
- NOT ACCEPTED (status message) ... 15
- note (CMS) ... 57, 81
- notebooks ... 82
- numbers "in prefix area" ... 30
  
- Olivetti PC as a terminal ... 115
- operators ... 80
  
- PA terminals ... 115, 117, 119
- parentheses and CMS commands" ... 42
- pascal ... 101
- password ... 11
- passwords minidisk ... 110
- pds importing ... 87
- peek (CMS) ... 58, 81
- PENDING (status message) ... 30
- Pericom Alpha terminals ... 115
- Pericom terminals ... 115, 117
- pf Keys (CMS) ... 70
- pf keys (XEDIT) ... 41
- PG terminals ... 115, 117, 119–120
- porting vbs files to cms ... 87
- powerinput (Xedit) ... 24
- preceding (prefix) ... 28
- prefix area ... 21
- prefix commands ... 28
- print (CMS) ... 56
- printer virtual ... 59
- printers ... 4
- printing ... 90
- printing spool files ... 92
- profile exec ... 74
- profile xedit ... 42
- programmable function keys (CP) ... 70
- PRTLST ... 59
- punch ... 4
- punch virtual ... 59
- purging jobs ... 89
- put (prefix) ... 28
- put delete (prefix) ... 28
- P7 terminals ... 115, 117, 119
  
- qbit (CMS) ... 80
- qq (Xedit) ... 19
- query (Xedit) ... 37
- query cmslevel (CP) ... 48
- query disk (cms) ... 49
- query names (CP) ... 48
- query pf (CP) ... 70
- query prt (CMS) ... 58
- query pun (CMS) ... 58
  
- query rdr (CMS) ... 58
- query ring (Xedit) ... 38
- query set (CMS) ... 73
- query synonym all (CMS) ... 70
- query time (CP) ... 48
- query" "all (CP) ... 46
- query" "case (Xedit) ... 43
- query" "search (CMS) ... 55
- query" "storage (CP) ... 48
- query" "synonym (Xedit) ... 41
- query" "userid (CP) ... 80
- quitting Xedit ... 19
  
- R/O link to minidisk ... 109
- R/W link to minidisk ... 109
- rdrlist (CMS) ... 58
- reader ... 4
- reader virtual ... 59
- ready-message ... 71
- recalling archived files ... 94
- receive (CMS) ... 58, 83
- record ... 18
- record format ... 21
- records ... 21
- recover (Xedit) ... 24
- recovering backed up files ... 94
- removing minidisk passwords ... 110
- rename (CMS) ... 56
- reset (Xedit) ... 30
- return (space) ... 99
- REXX ... 73
- rexxmath ... 114
- right (Xedit) ... 26
- right adjust (prefix) ... 28
- ring (Xedit) ... 38
- room number ... 91
- RUNNING (status message) ... 15
  
- save (Xedit) ... 19
- scale (prefix) ... 29
- scale line ... 22
- scan (CMS) ... 58
- schange (Xedit) ... 37, 40
- screen (Xedit) ... 39
- script ... 93
- search order (CMS) ... 54–55
- selective changes (Xedit) ... 40
- sendfile (CMS) ... 56–57
- service machines ... 4
- set arbchar (Xedit) ... 35
- set case (Xedit) ... 34
- set impcp (CMS) ... 16
- set number (Xedit) ... 30

- set pf (CP) ... 70
- set shadow (Xedit) ... 33
- set span (Xedit) ... 36
- set varblank (Xedit) ... 35
- setpass (space) ... 99
- setting function keys (CP) ... 70
- setting tabs (in xedit) ... 40
- SGML ... 94
- shadow (Xedit) ... 33
- show (prefix) ... 28
- space administrator ... 48, 98
- space limit ... 99
- space machine ... 98
- space tempdisk ... 99
- span (Xedit) ... 36
- splitdisk (space) ... 99
- spool ... 57
- spool (CP) ... 60
- spool files printing ... 92
- spoolid ... 57
- stanprof exec ... 74
- stanprof xedit ... 42
- status area ... 22
- storage ... 48
- submit (CMS) ... 89
- swapmode (cms) ... 53
- synonyms (CMS) ... 69
- synonyms (XEDIT) ... 41
  
- tabl (prefix) ... 29
- tabs (xedit) ... 40
- Tandberg Terminal ... 115
- Tandberg terminals ... 121
- tapedisk (CMS) ... 96
- tapes ... 96
- targets (Xedit) ... 31
- tasks (help) ... 66
- Tektronix 4107 ... 115
- tell (CMS) ... 80
- tempdisk (space) ... 99
- temporary disks ... 50
- temporary minidisks ... 98
- terminals ... 4
- text processing ... 93
- top (Xedit) ... 26
- transfer (CP) ... 59
- truncation ... 21
- TSO/E mail to ... 92
- tvscreen ... 6
  
- unit record device ... 4
- up (Xedit) ... 26
- uppercase (prefix) ... 28
  
- uppercase (Xedit) ... 34
- use (CMS) ... 86
- userid ... 77
  
- varblank (Xedit) ... 35
- vax file transfer ... 88
- VersaTerm PRO ... 121
- veto-ing changes (Xedit) ... 40
- vfort (CMS) ... 102
- virtual storage ... 48
- vmarch (CMS) ... 94
- vmbackup ... 94
- vmblst (CMS) ... 94
- vs fortran ... 101
- vs pascal ... 101
- VT100 terminals ... 115
- VT200 terminals ... 115-116
  
- Waterloo Script ... 93
- whois (CMS) ... 79
- wylbur" "bridge ... 113
  
- xcompare (CMS) ... 56
- xedit ... 19
- xprint (CMS) ... 56, 90
- xscript (CMS) ... 93
  
- zone (Xedit) ... 36