

**ALICE/99-09 Internal Note/Computing
IT/PDP/XXX
Version 1.00
Last modified 10/2/99**

A user guide to the ION WGS (initial release)

Editor: A. Miotto

Authors: M. Dodgson, A. Miotto, T. Cass,
R.W.L. Jones

Table of contents

| | |
|--|-----------|
| 1. Overview of IT computer services | 7 |
| 1.1 Batch and Interactive Services | 7 |
| 1.1.1 Public Login Unix Servers (PLUS) | 7 |
| 1.1.2 Work Group Servers (WGS) | 7 |
| 1.1.3 Scalable Heterogeneous Integrated Computing Facility (SHIFT) | 7 |
| 1.1.4 Central Simulation Facility (CSF) | 7 |
| 1.2 Other services | 7 |
| 2. The ION central services | 8 |
| 2.1 Work Group Server | 8 |
| 2.1.1 Hardware description | 8 |
| 2.1.2 The ISS service | 8 |
| 2.1.3 The user environment | 8 |
| 2.1.4 Mail service | 8 |
| 2.1.5 Printing | 8 |
| 2.2 SHIFTALICE | 9 |
| 2.2.1 Hardware description | 9 |
| 2.2.2 Access to batch services | 9 |
| 2.3 SHIFTNA49 | 9 |
| 2.3.1 Hardware description | 9 |
| 2.3.2 Access to batch services | 9 |
| 3. WGS User start-up | 10 |
| 3.1 Getting An Account | 10 |
| 3.2 Connecting via X | 10 |
| 3.2.1 NCD X terminals | 10 |
| 3.2.2 HP X terminals | 10 |
| 3.2.3 PCs (NICE 95) | 10 |
| 3.2.4 Logging in | 10 |
| 3.3 Connecting via telnet | 10 |
| 3.4 Changing the password | 10 |
| 3.5 Changing the shell | 11 |
| 3.6 Setting up the account | 11 |
| 3.6.1 Access to the home directory | 11 |
| 3.6.2 Access to subdirectories | 11 |
| 3.6.3 Mail | 11 |
| 3.6.4 .rhosts | 11 |
| 3.7 Terminating the session | 12 |
| 3.7.1 Telnet session | 12 |
| 3.7.2 X session | 12 |
| 4. Unix Security | 13 |
| 4.1 Passwords | 13 |
| 4.2 X Windows and granting access to the display | 13 |
| 4.3 AFS directory access | 13 |

| | | |
|-----------|--|-----------|
| 5. | AFS: Essential facts | 14 |
| 5.1 | Why AFS | 14 |
| 5.2 | Authentication | 14 |
| 5.2.1 | Tokens | 14 |
| 5.2.2 | Tickets | 14 |
| 5.3 | Changing the AFS password | 14 |
| 5.4 | /afs | 15 |
| 5.5 | File Security | 15 |
| 5.6 | Home directory structure | 16 |
| 5.7 | Quota | 16 |
| 5.8 | Backup | 17 |
| 5.9 | AFS output caching | 17 |
| 5.10 | Replacing running programs on HP-UX systems | 17 |
| 6. | Shells | 18 |
| 6.1 | Introduction to shells | 18 |
| 6.2 | User start-up scripts | 18 |
| 6.3 | HEPiX start-up scripts | 18 |
| 6.4 | Example features of zsh | 19 |
| 6.5 | Customizing the shell environment | 19 |
| 7. | The HEPiX X11 environment | 20 |
| 7.1 | The standard environment | 20 |
| 7.1.1 | Defining clients to be started at login time | 20 |
| 7.1.2 | Other tailorings | 20 |
| 7.1.3 | fvwm2 | 20 |
| 7.1.4 | The mouse | 20 |
| 7.1.5 | Using the fvwm2 pager (“virtual desktop manager”) | 20 |
| 7.2 | Introduction to fvwm2 | 21 |
| 7.2.1 | The fvwm2 Window Manager | 21 |
| 7.2.2 | The Present Default | 21 |
| 7.3 | Simple customisation of fvwm2 | 23 |
| 7.3.1 | How the Customisation Works with the HEPiX scripts | 23 |
| 7.3.2 | Menus | 23 |
| 7.3.3 | Desktops | 24 |
| 7.3.4 | Focus | 25 |
| 7.3.5 | Colours and fonts | 25 |
| 7.3.6 | Icon Boxes | 26 |
| 7.4 | Advanced customisation of the X11 environment | 26 |
| 7.4.1 | Changing Window Manager | 26 |
| 7.4.2 | Using alternate desktops | 27 |
| 7.5 | Solving problems | 27 |
| 8. | Talking to Other Users | 28 |
| 8.1 | Short Messages—Zephyr | 28 |
| 8.1.1 | Logging in to Zephyr | 28 |
| 8.1.2 | Subscribing to Zephyr message classes | 28 |
| 8.1.3 | Sending Zephyr messages | 28 |
| 8.1.4 | What to do if you get a Zephyr Message | 29 |
| 8.1.5 | Other Zephyr Facilities | 29 |
| 8.2 | Mail Messages | 29 |

| | | |
|--------------------|--|-----------|
| 8.2.1 | A Quick Introduction to some Mail terms | 29 |
| 8.2.2 | The CERN Mail Server | 30 |
| 8.2.3 | Forwarding mail | 30 |
| 8.2.4 | A Quick Introduction to PINE | 31 |
| 8.2.5 | Leaving pine | 36 |
| 9. | A quick introduction to LSF at CERN | 38 |
| 9.1 | What is LSF? | 38 |
| 9.2 | How to use LSF | 38 |
| 9.2.1 | Authentication | 38 |
| 9.2.2 | AFS and file access | 38 |
| 9.2.3 | Queues | 38 |
| 9.2.4 | Command line programs | 39 |
| 9.2.5 | X11 programs | 42 |
| 9.2.6 | LSF documentation | 43 |
| 9.3 | The CERN job starter | 43 |
| 9.3.1 | Working directory | 43 |
| 9.3.2 | Environment | 43 |
| 9.3.3 | Output file | 43 |
| 9.3.4 | Other files | 44 |
| 9.4 | Fault tolerance | 44 |
| 10. | Access to Magnetic Tape Data | 45 |
| 10.1 | Tape Staging | 45 |
| 10.2 | Stage Commands | 45 |
| 10.3 | Stage Pools | 45 |
| 10.4 | Examples | 45 |
| 10.5 | Tape problems | 46 |
| Appendix A. | Bibliography | 47 |
| A.1 | CERN documents | 47 |
| A.1.1 | Recommended manuals | 47 |
| A.1.2 | Other reading | 47 |
| A.1.3 | WWW pages | 47 |

1. Overview of IT computer services

This chapter contains a short overview of Central Computing Services provided by the IT division.

1.1 Batch and Interactive Services

1.1.1 Public Login Unix Servers (PLUS)

PLUS are public machines provided by IT for interactive work. CERN provides PLUS for some of the major UNIX flavours. Services currently available are RSPLUS (IBM/AIX), HPPLUS (HP-UX) and DXPLUS (Digital Unix). All machines are clients of the Andrew File System (AFS¹) and thus share both user file-space and application directories with each other and with other machines connected to AFS.

1.1.2 Work Group Servers (WGS)

WGS are machines based on PLUS but dedicated to specific experiments. They are provided for interactive as opposed to batch work.

1.1.3 Scalable Heterogeneous Integrated Computing Facility (SHIFT)

SHIFT is a scalable batch processing facility for physics data analysis. SHIFT machines are grouped in clusters dedicated to given experiments, and they are all AFS clients.

1.1.4 Central Simulation Facility (CSF)

CSF is a MonteCarlo production facility which is not intended for general use. CSF machines are AFS clients and they are thus integrated with the other services.

1.2 Other services

The infrastructure is held together by other services including X terminal support [11] (boot and font servers), mail and news services [6] (electronic mail directory and mail servers), AFS file servers, print servers [15], network services [20] (nameservers and time daemons) and WWW servers [19].

1. The Andrew File System (AFS) is a distributed file system. Files are stored on servers and are accessed, via the network, by client machines. A small (but inadequate) chapter of this guide concerns AFS - interested users are recommended to read [2].

2. The ION central services

2.1 Work Group Server

2.1.1 Hardware description

The Work Group Server for the ION experiment is based on HP hardware running HP-UX and PCs running Linux. There are currently 3HP D270 workstations and 3 450MHz Pentium II PCs in the cluster.

N.B. As all machines are identically configured, these names shall not be used to refer to individual nodes, and the ISS name should be used instead (see “The ISS service” on page 8).

The number and type of machines available in each service, as well as the version of the OS, can change at any time. Any upgrades are usually notified in the ION news.

2.1.2 The ISS service

ISS (Interactive Session Support) is a load balancing service that works by changing the address associated with the cluster name to route users to one of the machines in the cluster. Machines under system maintenance are automatically excluded by ISS.

- From within CERN ISS provides a way to route incoming logins to the least loaded machine in a group.
- From outside CERN users should be routed to a machine chosen at “random”.

The ION WGS HP nodes can be accessed as **ion** (inside CERN only), **ion.cern.ch** or the full ISS name **ion.wgs.cern.ch**. The ION WGS Linux nodes can be accessed as **ionlx** (inside CERN only), **ionlx.cern.ch** or the full ISS name **ionlx.wgs.cern.ch**.

2.1.3 The user environment

In order to help users in preserving their environment while working with different shells, on different UNIX platforms or even at different HEP sites, the HEPiX project provides a set of scripts which are executed at login. HEPiX is also offering a standard graphical (X Windows) environment, HEPiX X11.

The HEPiX environment is quite transparent to the user, who normally would only notice its absence on machines where it is not installed when some commands do not work because some variable are not set-up correctly. See [9] for a complete HEPiX user’s guide.

2.1.4 Mail service

WGS services do not provide local mailboxes, as mail messages would be spread into the different machines with the result that some messages could possibly remain unnoticed to the user. All incoming mail is routed to the CERN mail server (`mail.cern.ch`). All sender addresses in outgoing mail are changed so that mail appears to be sent from the CERN mail server.

This guide contains a section covering one of the recommended mail clients. Additional information about other mail clients and the mail service in general can be found in [6].

2.1.5 Printing

From ION nodes users can access any printer on the CERN network. However, in order to give this versatility the UNIX `lp` and `lpr` commands do not have any effect, since individual printers are not known to the system, and the command `xprint` must be used instead.

2.2 SHIFTALICE

2.2.1 Hardware description

The SHIFTALICE service is currently based on an HP machine running HP-UX, but Linux batch capacity will also arrive soon.

2.2.2 Access to batch services

Submission of jobs to SHIFTALICE is possible from the WGS using LSF (Load Sharing Facility) – see the chapter “A quick introduction to LSF at CERN” on page 38.

Submission to CSF is currently possible only from the HPPLUS service.

2.3 SHIFTNA49

2.3.1 Hardware description

The SHIFTNA49 service is based on an PCs running Linux. There are currently 29 PCs, 16 300MHz, 8 400MHz and 5 450MHz machines.

2.3.2 Access to batch services

Submission of jobs to SHIFTNA49 is only possible from the master node na49b01 using LSF (Load Sharing Facility) – see the chapter “A quick introduction to LSF at CERN” on page 38.

Submission to CSF is currently possible only from the HPPLUS service.

3. WGS User start-up

3.1 Getting An Account

People wanting an account on the ION Work Group Server should contact one of their group administrators, or the UCO¹ in case they are unreachable. For further details see [16].

3.2 Connecting via X

Some X terminals might already be set-up to present a login window with the message “Welcome to X11”. If the login window is not present apply one of the following procedures.

3.2.1 NCD X terminals

Choose the “Login New X Session...” item in the “Login” menu of the Console window, type `ion` in the “Host:” dialogue and press `<return>`.

If the login window does not appear, the security settings of the X terminal might be incorrect. The default configuration file loaded at boot time normally has correct security settings.

3.2.2 HP X terminals

Setup or ask your X terminal administrator to setup your HP X terminal to open a direct XDM login session to the `ion` service.

3.2.3 PCs (NICE 95)

From the **Start** button select the **More Applications**, then the **Exceed X-Terminal** and **Predefined Connections** menu. Double-click the ION icon..

3.2.4 Logging in

Type your username, press `<return>` (remember UNIX is case sensitive!), then type your password, `<return>`.

N.B. In case of problems you can still login in “fail-safe mode” by typing your username and password and then pressing the F1 key instead of `<return>`.

3.3 Connecting via telnet

Type `telnet ion`, type your username, press `<return>` (remember UNIX is case sensitive!), then type your password, `<return>`.

3.4 Changing the password

Type `passwd`²: you will be asked to enter your current password and then the new one. The procedure can be aborted by entering an empty line.

1. User Consultancy Office – `User.Support@cern.ch`.

3.5 Changing the shell

The user default shell is set when the account is created. If for any reason you want to use a different default shell you can use the command `chsh`. This change must then be propagated to all the nodes of the ION, which is done at regular time intervals, but may mean a small delay before all nodes register the change (a few minutes).

The login shell can also be changed on all central services at once using a graphical interface, invoked with the command `userinfo`

IT recommends you use either `zsh` or `tcsh` as default shell. On some Work Group Servers the experiment/group enforces a standardisation on a particular shell, which is defined later (for additional information see the chapter “Shells” on page 18).

3.6 Setting up the account

Users’ home directories are on AFS, a network-distributed file system (see “AFS: Essential facts” on page 14), and are shared by most central services where the user has an account, quite likely on machines with different architectures. For these and other reasons the home directory setup follows particular rules.

If your AFS account has just been created it should be properly set-up already. If it is an old account some settings might be different from current ones. You can run the command `uco -n` to verify some of these settings. Re-running the command without the `-n` option will apply the suggested modifications.

The `uco` function can be used to create standard setup files in the home directory or to reset them to default. Some cases will be presented further on in this guide. Type `man uco` for more information about this command.

3.6.1 Access to the home directory

AFS access control is performed at the directory level and not by looking at the permission for individual files. For this reason, and because the home directory contains some files which are important for security, the home directory access allows users other than the owner to read the contents of the home directory but not the contents of files in the directory (**do not change this default**).

3.6.2 Access to subdirectories

When the user account is created two general purpose subdirectories are also created by default: the first, `~/public`, has a default access allowing everybody to read the files in it; the second, `~/private`, does not allow even access to the directory contents.

All subdirectory created by the user inherit the default access of the parent directory. To view access rights for directory `dir` use the command:

```
fs listacl dir
```

to change them use the command:

```
fs setacl dir username rights
```

3.6.3 Mail

The section “Mail Messages” on page 29 contains the recipe for setting up a forward for mail directed to central services. This section also contains a description of the CERN mail server and a quick introduction to the recommended mail reader.

3.6.4 .rhosts

With AFS we recommend¹ that each user has a `.rhosts` file. With a `.rhosts` file you can use the following

-
2. Users’ password are not stored in the `/etc/passwd` file but on AFS, so the UNIX command `/bin/passwd` coming with the system does not work. On PLUS and WGS services a smart `passwd` command is installed in order to invoke the proper one (named `kpasswd`).

commands:

- **xrsh *hostname*** to open a new X session on the target machine. This command correctly transmits the DISPLAY variable used by X11 applications¹, and also sets the X11 permissions avoiding the need to use the xhost command.
- **rlogin *hostname*** to login to another AFS host without needing to type your password (but this does not set the DISPLAY variable nor handle X11 permissions).

N.B. Because you need a token to use AFS you cannot use the above commands from a non-AFS machine to an AFS machine.

The `.rhost` file format is "machine username" (you can have several lines with one entry on each). With AFS home directories the entry "+ *your_username*" allows you to connect from all machines where you have the same username and a valid AFS token.

3.7 Terminating the session

3.7.1 Telnet session

Type `logout` or `exit` to terminate your session and disconnect.

3.7.2 X session

Within an X session, either on an X terminal or on a PC, press the right menu button when the pointer is on the "root window" (the background of your screen) and choose the item "**Logout -> Yes,Really Quit**" (if you use a window manager other than `fwm` the name might be rather "Quit" or "Logout" instead).

- N.B. Typing `logout` or `exit` at the shell prompt when you have an X session does not terminate it! Anybody can use still your account and start another shell or other programs using the mouse buttons.**
- N.B. Avoid switching off the X terminal to terminate the session, as this might not kill the processes that have been started.**

1. Since your home directory is protected by AFS the `.rhosts` file can only be read if you already have an AFS token greatly increasing the security of the `.rhosts` file. This is not the case for non AFS home directories where the use of a + presents a big security risk.

1. This is one of the reason why `xrsh` is preferred to `rsh` and `rlogin` in an X environment: there is no easy way with `rlogin` to set automatically a correct DISPLAY variable in all situations, e.g. when having an `xdm` session on an X-terminal.

4. Unix Security

Machine security is a problem and some of the WGS/PLUS machines are open to the outside world so please don't take security lightly - read reference [5]!

4.1 Passwords

Please change your password as soon as possible after your account is created. It is important to choose a good password - at least 6 characters long and with mixed case, numbers and/or other characters (such as `!_+=.,;` but not special characters such as `@#\~ BackSpace` or `Delete`¹). It should also be hard to guess (e.g. not your name) and should not be in standard dictionaries of any language!

The rules enforced at CERN in order to reduce the number of guessable passwords can be found in [13].

4.2 X Windows and granting access to the display

X Windows security is a problem — if you are not careful people can spy on your screen and watch you entering sensitive info like passwords. In particular be careful when using the command `xhost`:

- **never** open your display to everybody in the world by typing “`xhost +`”
- **avoid** opening your screen to a public machine by typing “`xhost public_machine`” as this allows anyone on that machine to access your screen.

When connecting from AFS to other machines you can setup your `.rhosts` file (see page 11) and try to use

```
xrsh hostname
```

Unfortunately `xrsh` does not work in many cases — such as connecting from outside CERN or from non AFS machines to AFS machines. We are looking at new tools which avoid these restrictions but for the moment in case of problems we recommend the following procedure:

- start the `mxconns` program as explained in reference [5] to create a virtual X server
- use “`telnet hostname`” to login to the remote host, and set the `DISPLAY` variable to the display created by `mxconns`

Each time a process opens a connection to your display, you will be prompted and asked whether you accept or deny the connection.

4.3 AFS directory access

Some files stored in the home directory contain information which allow the reader to access the user's display, or his account. As on AFS there is no way to provide access restrictions for individual files, the home directory contents (i.e. the names of files stored in the directory) are readable by everybody, while the file themselves are not.

Please do not change the access control list in your home directory. If you want to share files you can store them in the `~/public` subdirectory, or create other subdirectories accessible by everybody.

1. Such characters are allowed in some systems but are guaranteed not to work on others. As the AFS password is the same on all systems, the safe choice is not to use them at all.

5. AFS: Essential facts

AFS is the distributed file system which has been chosen by IT. This section will try to provide an introduction to AFS and a few of its commands. This introduction may be insufficient, users are recommended to read [2].

5.1 Why AFS

Some of the reasons that CERN has chosen AFS are that:

- It is relatively secure - user passwords are only stored on the AFS servers; communications with this server are encrypted; when a user logs on he acquires a software ticket which is then a guarantee of his identity.
- File management is centralised. Client machines just connect to AFS and then can access the whole file system. "Pure AFS machines" only use local disks for system software - all home and application directories are stored in AFS and thus are available to all AFS clients.
- It is cached. Files are copied to a disk cache on the local file system. this gives fast access to files/programs which are frequently used.

5.2 Authentication

5.2.1 Tokens

User identification to AFS is done using "tokens" which last 25 hours. You can check that you have a token by typing the command `tokens`. This displays your token and its expiry time.

You can get a new token when you:

- Login to the system.
- Type "`klog username`"
- Lock and then unlock your screen using the AFS version of `xlock`

N.B. This means that if you leave yourself logged on for 25 hours without issuing one of the above commands you will lose the rights to access your files. This situation can be solved by performing one of the above actions.

Notice that `xlock` will refresh the tokens of applications running on the local machine only. If you leave an editing session on a remote machine overnight you might be forced to save your work in the `/tmp` directory and quit the program.

5.2.2 Tickets

The same procedures that provide a token give also a "Kerberos Ticket Granting Ticket", with the same expiration time of 25 hours. A valid TGT might be needed to authenticate users to some services (e.g. `pubarch`, `LSF`, `ssh`), without the need of typing a password.

There are important differences between an AFS token and a Kerberos TGT: the former is used by AFS for access control only, the latter is a general purpose authentication mechanism, which can also be used to generate a valid AFS token. In addition, tokens conserve their validity when they are passed from one host to another, while it is necessary to enter the password in order to generate a new TGT on a different host.

5.3 Changing the AFS password

Users do not have a local password on the WGS and PLUS machines: it is stored remotely on an AFS server and

valid on all AFS machines. Users of central services can use the `passwd` command to change the AFS password¹.

5.4 /afs

AFS is a world wide file system with a world wide file space. This gives big advantages when sharing files with other labs, but **do not** look into the directory `/afs` by, for example, typing “`ls -l /afs`” nor “`find / . . .`”: you’ll hang your terminal whilst it sends queries around the world!

CERN path names are long

`/afs/cern.ch/`
the root directory for CERN files

`/afs/cern.ch/user/`
user directories

`/afs/cern.ch/exp/`
experiment private file space, where `exp` is `atlas`, `chorus`, `cms`, `na48`, `na49`, ...

`/afs/cern.ch/group/gg/`
This is the directory pointed to by the `$GROUP_DIR` variable and is used to store group start-up files and templates.

`/afs/cern.ch/asis/`
the CERN software repository

5.5 File Security

In normal Unix you can define access protection on individual files giving read, write and/or execute permission to the owner, group and or any other user (see any Unix manual for details). **In AFS the *group* and *other* access permissions are ignored!** However, an additional protection is available - Access Control Lists (ACLs):

Each AFS **directory** has an associated access control list defining an additional access permission for all files in the directory. This access permission is “ANDed” with the owner part of the file access permission to decide if a given user has the correct rights to access the file (e.g. only if the access list on the **directory** gives the user `dodgson` read access to the file AND the file is defined readable to the owner by the Unix protection can the user `dodgson` read the file.

The `afs` access control lists can be inspected by typing:

```
fs listacl directory_name
```

and set by typing

```
fs setacl directory_name ...+ args
```

e.g. to list the access rights to the current directory type

```
fs listacl .
```

which will return an output like

```
Normal rights:  
system:anyuser l  
dodgson rlidwka
```

where:

r=read, l=lookup (i.e see which files are in a directory), i=create , d=delete, w=write, k=lock (a file) and a=change the access control list.

To change the list to allow users belonging to group `gg` to browse the directory and read files in it, type:

```
fs setacl . cern:gg rl
```

1. This is not the UNIX command but a wrapper around the AFS `kpasswd` command.

Or to remove lookup rights for `system:anyuser` type:

```
fs setacl . system:anyuser none
```

Several other groups are useful:

| | |
|------------------------------|--|
| <code>system:anyuser</code> | any one who has access to AFS anywhere from anywhere |
| <code>system:authuser</code> | any user of AFS with a valid CERN token. |
| <code>cern:nodes</code> | any one on a machine at CERN |
| <code>cern:gg</code> | any user belonging to group <code>gg</code> |

ACLs of new directories are inherited from the parent directory.

5.6 Home directory structure

When a new user is created a home directory (referred to by the `~` sign) containing the following sub directories are created:

- `~/mail`
a directory for storing your mail folders
- `~/public`
a directory which is readable by everybody. **You should not change its ACL.**
- `~/private`
a directory which is readable/searchable only by you. **You should not change its ACL.**
- `~/.@sys` directories
`@sys` is a special AFS variable which has a value dependent on the architecture of the machine you are working on - its value can be printed with the command "**fs sysname**", Examples are `hp_ux102` (HP), `rs_aix42` (IBM), `sun4x_56` (Sun). These directories can be used to store platform dependent binary files¹.
- `~/bin`
a symbolic link to `~/.@sys/bin`
- `~/scripts`
a directory where architecture independent scripts can be stored
- `~/ .forward`
a symbolic link to `~/public/ .forward`. The Unix mail system redirects your mail based on information in `~/ .forward` (which normally contains the information `user@address`), however, since home directories are protected, even system programs cannot read it. The link to `~/public/ .forward` is thus created in order to let the mail agent read the forward address. **You should not change this link!**
- `~/ .hepix`
a directory containing files used to customise the X session and the window manager.

The home directory structure can be checked or reset with the command `uco -home`.

5.7 Quota

To find out the quota you have available on your home directory (in Kbyte units) use the command:

```
fs listquota $HOME
```

AFS quota for individual users is assigned by CERN group administrators, so contact them in the case you would need to increase it.

1. Please note that these directories are set when the user account is created, and **not** added automatically when new AFS architectures are introduced at CERN!

5.8 Backup

A backup of all home directories from the evening before can be found under the directory:

```
/afs/cern.ch/ubackup/u/username/
```

where *u* is the first letter of your username. For the recovery of older files please send a mail to **afs.support@cern.ch** with the name of the file, its directory and the date you want the file recovered from.

5.9 AFS output caching

AFS caches both input and output files. When you change a file your change is only distributed to other machines after you close the file (e.g. if you modify a file using an editor and **do not release** that file it is possible that your local machine can open the file and obtain the new version of the file BUT another AFS machine still sees the old version of the file!).

5.10 Replacing running programs on HP-UX systems

When you use a binary program or shared library on the local disk the system locks this binary so that it cannot be replaced. On remote file systems, such as AFS and NFS, the system cannot lock the binary. As a result strange behaviour may occur if a binary is changed while the old copy is running, for example:

- the new binary may not run (giving a strange error message) until the old programs been killed
- running the new binary may give you the old version of the binary until the old program has been killed

Because of this it is recommended not to overwrite new binaries but to remove them and then recreate the new copy — this changes the binaries inode and avoids the above effects.

6. Shells

6.1 Introduction to shells

A “shell” is a program which interprets user commands - it is the text base user interface of the machine. The original Unix shell `sh` was a poor interactive shell and thus a more user friendly `csh` was developed. There are now six major shells: two shells using a “csh” like syntax (`csh` and `tcsh`) and four shells using a “sh” like syntax (`sh`, `ksh`, `bash` and `zsh`). The later generations of shells (`ksh`, `zsh`, `tcsh`, `bash`) introduce features such as command line editing, the ability to scroll back through previous commands (the history mechanism) and command/filename completion.

IT recommends that you use either `zsh` or `tcsh`, and this chapter gives a short introduction to their features (you can read [3] for more details). On the ION service there is a standardisation on `zsh`, so you will find only the information for the relevant shell.

6.2 User start-up scripts

Start-up scripts are scripts (list of shell commands) stored in the users home directory and read by the shell. They allow the users to tailor their environment, for instance setting up aliases for common commands and defining which editor to use by default.

`zsh`

`zsh` looks for the following files in the users home directory:

- at login time it calls (in order) `.zshenv` `.zprofile` `.zshrc` `.zlogin`
- When opening an interactive shell it calls (in order) `.zshenv` `.zshrc`
- When running a non interactive shells it calls `.zshenv`
- When logging out it calls `.zlogout`.

and thus it is recommended that:

- `.zprofile` and/or `.zlogin` should be used for commands that only need to be executed at login time.
- `.zshenv` should contain commands to set the search path and other important environment variables. It should not include commands which output or try to setup the terminal (e.g. `tset` or `stty`).
- `.zshrc` should be used to setup the interactive environment (aliases, functions, key bindings etc.)
- `.zlogout` should be used to close down clients when you logout.

Most of these files should already exist in your home directory. If not, you can create an empty template using the command “`ucc -shell`” and choosing the option **Reset the environment of your default shell**. This command will also save any existing users files by renaming them. (type “`man ucc`” for details).

6.3 HEPiX start-up scripts

Independent of the scripts in your home directory, standard HEPiX start-up scripts are read. This means that you should be careful shipping scripts from machines which do not run the HEPiX scripts - they may spoil the environment rather than improve it! (for details see [9])

6.4 Example features of zsh

Some of the nicest shell features are listed below:

- Editing commands - you can use the up and down arrows to scroll through previous commands and then the left and right arrow keys to edit them
- Command completion - you can type the first letters of the command or file and then press the `TAB` key - the shell will either (1) interpret the `TAB` as a space if the filename is complete (2) complete the filename if it is identified uniquely (3) display all the possible ambiguities
- Command history search - you can type the beginning of a previously issued command and then `ESC-p` to complete it. Repeating `ESC-p` will search through the history for other occurrences of the same pattern
- Spelling correction - `ESC-s` will try to correct the word immediately before the cursor, `ESC-$` the whole command line

6.5 Customizing the shell environment

Most of the variables you might need are already defined by HEPiX scripts and by the group scripts. You might anyway want to add new variables or aliases, or change existing defaults (e.g. the default editor command, printer or prompt).

New users or users who have executed the command `uco -shell` should already have default shell start-up files which contain several examples of how to do such customisations. If this is not the case, it is advisable to execute the above commands and then edit the start-up files to introduce your own changes.

7. The HEPiX X11 environment

7.1 The standard environment

This section describes the environment you should get if you start with a new account or with an account where you do not override the defaults (defaults can be overridden by your group's defaults or by you having one of the following files: `~/.xsession`, `~/.xprofile` `~/.fvwm2rc` or `~/.hepixonprofile`).

7.1.1 Defining clients to be started at login time

When entering the Xsession, two clients are started by default: `xterm` (with login shell) and `xclock`. If you want to have different start-up applications then create a file named `xclients` in the `~/.hepixon` directory, and add a line for each application desired, for example:

```
(sleep 3; xterm -geometry 80x24+10+10 -n login -ls) &
(sleep 3; xclock) &
(sleep 3; ical) &
```

The sleep 3 tries to make sure that the client starts after the window manager has started (otherwise you may have problems to localise the window) and do not forget to end each line with an ampersand '&' character to run the client as a background process.

7.1.2 Other tailorings

You can define your own application defaults in your `~/.Xresources` file. For example the following would make `xterm` windows bigger:

```
XTerm*vt100*geometry: 80x50
```

You can define your root window colour in your `~/.hepixon/xprofile` file by having a line:

```
HX_ROOT_WINDOW="xsetroot -solid grey"
```

or you can define a picture as your root menu using:

```
HX_ROOT_WINDOW="xv -root -quit -rm 5 -rbg grey file.gif &"
```

7.1.3 fvwm2

This is the HEPiX X11 group's recommended window manager for public machines. It offers advanced features such as multiple workspaces whilst being light on both the system and network. For details on `fvwm2` please read [17].

7.1.4 The mouse

If you click on the root window (i.e. the main screen) you will see that the three mouse buttons have been defined as:

| | |
|----------------------|---|
| Left button | The user menu. This can be tailored by you to launch the applications you prefer. |
| Middle Button | The group menu. This is the recommended button for groups to tailor. |
| Right button | The system menu. This is setup to launch recommended applications and allows you to logout of the X session. It will be maintained and updated centrally and should not be changed. |

7.1.5 Using the fvwm2 pager ("virtual desktop manager")

At the bottom right of the screen (by default) you will see the **fvwm pager**. It shows a view of each of your four virtual desktops. The current desktop (the one currently occupying your screen) is shown highlighted and the currently active window is also highlighted. You may move to a new desktop by selecting it with the left mouse but-

ton and you can move windows between desktops by dragging the pager image with the middle mouse button.

If you want a window to appear on all desktops you should use the (un)stick command from the window menu (using the left mouse button, click the small icon on the upper left of the window).

7.2 Introduction to fvwm2

7.2.1 The fvwm2 Window Manager

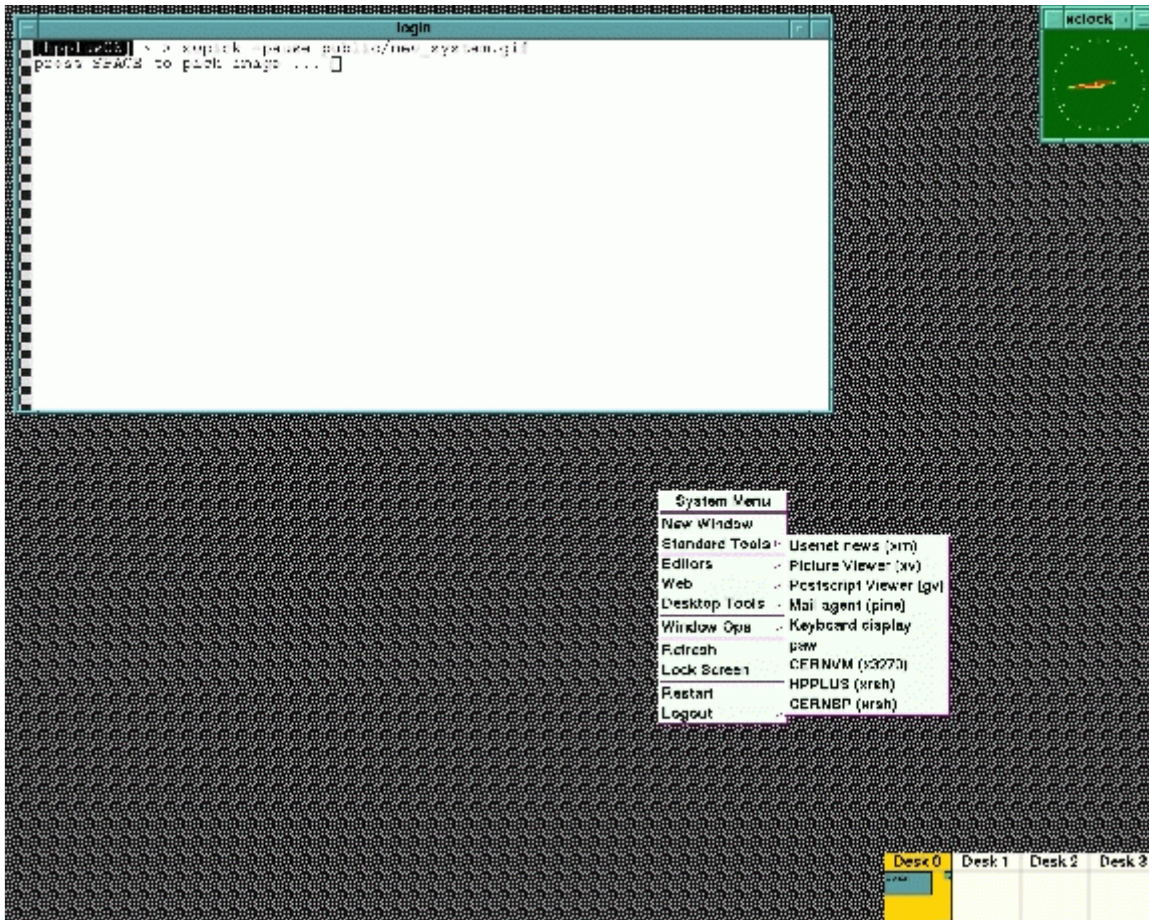
The default window manager for the HEPiX X11 scripts, and used on the CERN central UNIX platforms, is fvwm2. The default configuration is very simple, and the window manager makes few demands on the operating system. However, simple tailoring of the window manager is to be expected at the user and group level. Indeed, we have provided a hook for a 'Group' menu to be found when the middle mouse button is clicked over the root window ('the backdrop'), and we already provide a 'User' menu when the left mouse button is clicked in the root window.

This chapter indicates how to make simple customisations of the menus etc. More complex customisation is also possible (indeed, we are not aware of any extra limitations on the customisation imposed by the HEPiX scheme); a guide to the full setup is provided in [17]. Before any such major change, we strongly recommend you read the fvwm2 documentation.

7.2.2 The Present Default

7.2.2.1 At startup

The client programs (such as clocks, desktop managers etc) and root window ('backdrop') colours and patterns are not controlled by the window manager, but are controllable in the HEPiX X11 scheme. Indeed, other window managers can be requested, though they are not necessarily supported, and window managers that make a heavy demand on system resources are strongly discouraged on multi-user machines. Such things are not within the scope of this article, and you are referred to the relevant documentation.



When you begin a default HEPiX X11 session, you will be presented with a clock, an xterminal labelled `login' and a window in the bottom right-hand corner of the screen called the `pager'. This pager is subdivided into four areas labelled `Desk 0', `Desk 1' *etc.*, each of which corresponds to a virtual desktop. By default, you start on Desk 0, and so you will see small representations of the login window, the clock and the pager in the `Desk 0' section of the pager. You can move to another virtual desktop by clicking with the left or middle mouse button in the appropriate area of the pager. Unless they have been declared `sticky' (which the clock and the pager have, by default), then your windows will not change desktop as you do. You can move windows between desktops by clicking with the middle mouse button on their representation in the pager, and dragging them to the representation of the desktop you want in the pager.

7.2.2.2 Menus

A `System' menu is defined which contains many of the commands you will commonly want to use. It may be obtained by clicking with the right mouse button while over the root menu (the `backdrop'). Some of the commands may be found in submenus (indicated by arrowhead symbols on the main menu; keeping the mouse-button depressed, just move down the menu to the submenu name to obtain the submenu. At present, we provide submenus for editors, web browsers, standard tools, common window management tasks (such as making windows `sticky' or killing them) etc.

A `User' menu is provided by default; it is a copy of the System menu at the time that the user's account is created. This can be tailored by the user, as described below. A hook for a `Group' menu is also provided associated with the middle mouse button, but is left empty by default. Experiments and other groups should use this hook to provide common tools that their users might want.

7.2.2.3 Interacting with X-windows

Windows may be enlarged by clicking on the border with the left or middle mouse buttons and with the button still depressed dragging the window border to the desired size. Clicking on the border with the right mouse button produces a menu of commands that act on windows. Clicking and releasing on the window border or title bar

`raises' the window above any others. The window may be brought into `focus' (which is jargon for `becomes the active window') merely by moving the cursor into the window area. Thus, you can type into a window while it is still partially obscured by another, which can often be useful. If you prefer to just click in the window to raise it and bring it into the focus, instructions for this customisation are given below.

7.3 Simple customisation of fvwm2

7.3.1 How the Customisation Works with the HEPiX scripts

Fvwm2 may be configured by means of a startup file named `fvwm2_configuration` that should be put in the directory `${HOME}/.hepix/wm`. The sessions for a whole group may be changed by placing a file with the same name in the `${GROUP_DIR}/wm` directory. A user can always override what has been provided at the group level, and a group can also override actions defined at a higher (HEPiX, site or cluster) level.

The files at the user, group and higher levels are combined (whenever present) into the file `${HOME}/.hepix/wm/generated_fvwmrc2` when the window manager is started. This file should **not** be used to customise the window manager, as it will be overwritten when fvwm2 is next started.

7.3.2 Menus

Perhaps the most common things that will be customised are the menus obtained when clicking on the mouse buttons when the cursor is over the root window. By default, the right button (button 3) is reserved for a system menu (defined at site or cluster level). The left mouse button (button 1) is provided for the user menu, and this is initially created as a copy of the system menu. The default looks as follows:

```
# This menu will fire up some very common utilities
AddToMenu UserMenu
+      "User Menu"                Title
+      "New Window"              Exec xterm &
+      "Standard Tools"          Popup StdTools
+      "Editors"                  Popup Editors
+      "Web"                       Popup Web
+      "X Tools"                   Popup XTools
+      ""                          Nop
+      "Window Ops"               Popup WindowOps
+      ""                          Nop
+      "Refresh"                   Refresh
+      "Lock Screen"              Exec xlock -remote -mode blank &
+      ""                          Nop
+ "Regenerate fvwm2rc"            Exec /usr/local/lib/hepix/X11/tools/prepare-
wmrc fvwm2
+      "Restart"                   Restart fvwm2
+      "Logout"                     Popup QuitVerify
```

Similarly, the middle mouse button (button 2) is intended for group-defined menus. By default this menu is empty except for a title; the default looks as follows:

```
AddToMenu GroupMenu
+      "Group Menu"                Title
```

In order to customise the user and/or group menus the following line can be included in the file `fvwm2_configuration`:

```
DestroyMenu "<Menu>"
```

where `<Menu>` is either `UserMenu` or `GroupMenu`. This line removes any previous declaration of `<Menu>`. If it is missing, lines are added at the bottom of what was previously defined for that menu. The line

```
AddtoMenu "<Menu>"
```

declares the beginning of a section of lines, each starting with the character '+', containing the menu labels and

their associated definitions.

Strings can be quoted or not, but they must be quoted when they include blanks, e.g.: 'DestroyMenu "User-Menu"' is the same as 'DestroyMenu UserMenu', but you must quote 'DestroyMenu "User Menu"'.

The typical elements that define a menu are:

```
"<string>" Title
    makes <string> the title (first, inactive entry) in the menu. It can safely be omitted if you do not like titles

"<string>" Exec <command>
    makes an entry <string> in the menu that executes <command> when selected. The command is a Bourne-shell command.

"<Identifying string>" Popup <popup_name>
    creates a menu entry sub-menu entry "<Identifying string>" which when selected pops-up the sub-menu named <popup_name>.

" " Nop
    creates a dividing line to separate menu options.
```

The order of menu/sub-menu definitions is not important.

7.3.3 Desktops

By default, fvwm2 gives you four virtual desktops, each the size of your physical screen. This may be too few for some (it is unlikely to be too many, and the 'extra' desktops will not inconvenience users if not used). Extra desktops may be requested by adding the following lines to fvwm2_configuration in either \${HOME}/.hewix/wm for user specification and \${GROUP_DIR}/wm for group specification.:

```
# InitFunction:
# -----
# Executed when FVWM is started for the first time in an X session.
DestroyFunc InitFunction
AddToFunc InitFunction
+ "I"      Module FvwmPager 0 3
+ "I"      Desk 0 0

# RestartFunction:
# -----
# Executed each time FVWM is restarted.
DestroyFunc RestartFunction
AddToFunc RestartFunction
+ "I"      Module FvwmPager 0 3
```

The line

```
DestroyFunc <FunctionName>
```

removes any previous declaration of <FunctionName>. If the line is missing, everything following the line

```
AddToFunc <FunctionName>
```

is executed in addition to what was previously declared.

To request six desktops (labelled 0-5), one should change the lines

```
+ "I"      Module FvwmPager 0 3
```

to read

```
+ "I"      Module FvwmPager 0 5
```

We strongly recommend that you always have a desktop zero defined, as this is the default desktop on which processes start if the 'current desktop' information is unavailable for any reason. We also strongly recommend that the first '0' in the lines;

```
+ "I"      Desk 0 0
```


should **not** be changed, as this can crash your session if you restart fvwm several times. (The second zero instructs the session to start or restart on desk zero; this can be altered without bad effects.)

Users may also wish to have desktops that are several times the physical screen size. To do this, include the instruction

```
DeskTopSize 2x3
```

would define the desktop to be two physical screens tall and three wide.

The command:

```
EdgeScroll 60 40
```

would tell fvwm to scroll by 60% of a physical screen when the cursor reaches the top or bottom of the physical screen, and 40% of a physical screen width when it reaches either side of the screen. This scrolling may be delayed with the command:

```
EdgeResistance 250 50
```

where the first number is the time in milliseconds to delay before scrolling, and the second is the number of pixel *beyond* the edge of the screen the cursor must move before scrolling occurs.

The desktops are controlled through a window representing them (and the windows on them) called the pager. This is by default in the bottom right hand corner of the screen. It may be made larger by decreasing the scale factor between the real desktop and its representation, *i.e.* the smaller the factor given, the larger the pager size:

```
*FvwmPagerDeskTopScale 20
```

Finally, if you wish to assign different labels to the desktops in the pager, you need to add lines of the form

```
*FvwmPagerLabel <desk> <label>
```

Here, *desk* is the desktop number (as quoted in the desktop label in the default configuration) and *label* is the desired label for that desktop.

7.3.4 Focus

When a window becomes `active' and you can interact with it, it is said to come into focus. The default style for changing the focus is called `SloppyFocus'. This means that a window comes into focus when the cursor is within it, but the focus is only lost when the cursor enters another window **other than the root window**. This is set into the default configuration with the line:

```
Style "*" SloppyFocus
```

The "*" in the Style line applies the option to all windows. Any style can be applied to a set of named windows, e.g. "xterm" or "*term*".

If you prefer the focus only to change when you **click** in a window, include the following line into `fvwm2_configuration`:

```
Style "*" ClickToFocus
```

Similarly, some people like the active window to be raised in front of any other window. This can be enabled by including the following lines:

```
AddToFunc InitFunction
+          "I"          Module FvwmAuto 750
```

```
AddToFunc RestartFunction
+          "I"          Module FvwmAuto 750
```

A positive argument to the FvwmAuto module specifies the number of milliseconds after the window comes into focus before it is raised.

A combination of `ClickToFocus' with the FvwmAuto module gives a behaviour where the window is in focus and raised after clicking within the body of the window.

7.3.5 Colours and fonts

The default colours and fonts associated with windows **other than the pager** are the following:

```

# Set the fore and back border colors for the window that has focus
HighlightColor    Black CadetBlue

# Set fore/back border colors for all other windows
Style    "*"      Color Black/LightGrey

# Set colors/font for pop-up menus
# Syntax: MenuStyle forecolor backcolor shadecolor font style(fvwm/mwm)
MenuStyle        Black Honeydew SlateGrey variable mwm

# Set fonts to use on title bar and icon label
WindowFont       variable
IconFont         variable

```

The colours may be given either

- by name as given in the `/usr/local/lib/X11/rgb.txt` file.
- or by their RGB value. Colours specified in this way must begin by the ``#'` symbol followed by 3,6,9 or 12 hexadecimal digits. Valid formats are: `#RGB`, `#RRGGBB`, `#RRRGGG` and `#RRRRGGGG`. Each format has its range scaled to [0..255].

The pager is in fact a object known as a module, and its default colours and fonts are defined as:

```

# FvwmPager:
#
*FvwmPagerFore      Black
*FvwmPagerBack      Ivory
*FvwmPagerHighlight Gold
*FvwmPagerFont       variable
*FvwmPagerGeometry -1-1
*FvwmPagerSmallFont 5x8

```

Again, new colours may be given either

- by name as given in the `/usr/local/lib/X11/rgb.txt` file.
- or by their RGB value.

7.3.6 Icon Boxes

By default, when windows are iconized, icons are placed in one line at the bottom of the screen defined by the command:

```

# Auto Place Icons is a nice feature (Left Top Right Bottom)
Style    "*"      IconBox 5 -85 920 -1

```

It is possible to change the position of the default box, or to define different icon boxes for different window names.

7.4 Advanced customisation of the X11 environment

7.4.1 Changing Window Manager

The default window manager started by `xsession` is the `fvwm2` window manager. This behaviour can be changed by defining the variable `HX_WM` in the file `/${HOME}/.hepix/xprofile`.

```

HX_WM=local
    tells xsession not to start a window manager, but rather leave the window control to the local win-

```

down manager running on the X terminal or the PC.

```
HX_WM=/usr/bin/X11/mwm
```

will start the motif window manager. This is not recommended nor supported.

7.4.2 Using alternate desktops

With HEPiX X11 it is also possible to select a complete desktop if available, for example DCE. This is not recommended as it uses an unreasonable fraction of system resources for a public machine.

7.5 Solving problems

The file `$HOME/.hepixon/xsession.log` contains a log of the Xsession and should be used to track down problems.

8. Talking to Other Users

8.1 Short Messages—Zephyr

The Zephyr messaging facility is a very convenient way to send messages to somebody on a unix system at CERN. People to whom you want to send a message must be “logged in” to Zephyr but you do not need to know which system they are using.

The sections below give only a brief introduction to the Zephyr system; for more details you should see the UCO flyer “The Zephyr Message Service for Unix Users” or [10], also available from the UCO. Note that although the examples below all assume that you are using an X terminal or a workstation, Zephyr can also be used if you have a dumb terminal; the “Zephyr at CERN” manual has more details about using Zephyr on such terminals.

8.1.1 Logging in to Zephyr

Zephyr is installed on all WGS and PLUS clusters and users logging in with XDM should find that they are automatically logged in to Zephyr. You can check this with the `zlocate` command (replace “dougal” with your loginid):

```
zlocate dougal
hpplus01.cern.ch          ues1:3  Tue Jul 11 11:12:24 1995
```

A reply like this means that you are logged into Zephyr. If you see a reply like:

```
zlocate dougal
Hidden or not logged-in
```

It means that you are not logged into Zephyr and you need to use the command

```
zwgc -nofork &
```

before you can receive Zephyr messages.

8.1.2 Subscribing to Zephyr message classes

As well as messages from one person to another Zephyr can be used to send messages to a group of people. The operators use Zephyr in this way to send important messages to all users and also to send out messages about a given service, for example RSPLUS. WGS and PLUS users will normally be subscribed to the appropriate Zephyr “class” so that they receive these messages, but you may wish to use the `zctl` command to make sure you receive these messages if you are logged into a different system. To receive the messages for the ION service use for example:

```
zctl subscribe ***** \* \*
```

(You need to use “*” and not just “*” to stop your shell changing the asterisk into a list of all the files in your current directory.)

There are Zephyr messages classes for all the PLUS systems and also for some experiments; if you are a member of the CHORUS collaboration, for example, you should subscribe to the `chorus` Zephyr class.

8.1.3 Sending Zephyr messages

The command `zwrite` is used to send Zephyr messages:

```
zwrite dougal
Type your message now.  End with control-D or a dot on a line by itself.
Hello There!
.
dougal: Message sent
```

You can also send a short message on one line:

```
zwrite dougal -m Hello
```

8.1.4 What to do if you get a Zephyr Message

If someone sends you a Zephyr message you should see a window on your X terminal looking like:



The window shows who the message is from and the text of the message. To make the window disappear just click in it somewhere using the left mouse button. If you want to copy the text of the message then hold down the <shift> key while dragging across the text with the mouse — this will stop the window disappearing when you press the mouse button at the start of the text. Once you have selected the text it can be pasted into another window in the usual way (by pressing the middle mouse button).

8.1.5 Other Zephyr Facilities

As well as just sending messages between people, Zephyr can be used to

- send messages to a group of users,
- send replies when you are away — similar to the CERNVM utility **gone**
- send yourself reminders.

For more details about using Zephyr in this way see the manual “Zephyr at CERN”, available from the UCO.

8.2 Mail Messages

There are many different “Mail User Agents” (programs which allow you to send and read electronic mail) for Unix. However, at CERN we currently recommend the **pine** and **netscape** programs: both are compatible with the mail server and exist in UNIX and PC versions. **pine** is usable by people with “dumb” or “ascii” terminals (such as FALCOs) as well as those with X terminals, while **netscape** has an X interface only. For a quick introduction to **pine** turn straight to “A Quick Introduction to PINE” on page 31. There is no information about **netscape** in this guide, but most of the information contained in [7] applies to the UNIX version of **netscape** as well.

“A Quick Introduction to some Mail terms” on page 29 and “The CERN Mail Server” on page 30 give some useful background about mail at CERN, but this information is not essential.

8.2.1 A Quick Introduction to some Mail terms

You will probably meet many of the terms in this list when people talk about sending and reading electronic mail—some of the terms are even used in later sections of this guide. Mostly, though, you don’t need to know the details, just that the WGS and PLUS are set up so that you can use the recommended mail programs to read and send mail.

attachment

a MIME (see below) attachment is a binary or text file added to the mail “in parallel” with the main text in the mail body. A Mail User Agent that understands MIME is needed to include or read this attachment.

body

the part of the mail containing the main (or only) message text.

folder

a file in which a number of mails are stored

header

the part of the mail that shows who the mail is to, who sent it and the subject.

IMAP

The Internet Mail Access Protocol. Pine and the CERN Mail Server talk to each other using IMAP—this allows you to read mails from your inbox even though they are stored on a remote system. With IMAP mail items stay on the mail server until you delete them or file them away somewhere else.

inbox

a name for the folder into which mails sent to you are placed; see spoolfile.

mail server

a special machine used for storing incoming mail so that mails can be read from many client machines.

Mail Transfer Agent

a program to send mail between workstations; users do not normally need to worry about these.

Mail User Agent

a program allowing you to send and read mail such as PINE.

MIME

MIME stands for Multipurpose Internet Mail Extensions and is a way of sending such things as pictures, audio messages or formatted documents in a mail message as well as plain text. The plain text part of a message is sent as the message body as normal; any other special parts of the message are sent as attachments.

POP

for Post Office Protocol. This is another protocol used between mail agents and mail servers. Unlike IMAP the whole contents of your inbox are transferred from the remote server to a local folder. This is **not** a CERN recommended protocol.

sendmail

the Mail Transfer Agent most commonly used on unix workstations; you don't normally need to worry about this but if you have your own workstation you should be sure you have the correct sendmail configuration file.

spoolfile

a file used to store mails that have been sent to you that you haven't deleted or stored in another folder. Your mail spoolfile is also called an inbox.

8.2.2 The CERN Mail Server

On a normal Unix workstation mail for a user would be written to a local spoolfile, typically with a name like `/usr/spool/mail/natalie`. This behaviour isn't very good when a cluster of workstations is being used, though, since it means either that your mail is spread about across the cluster depending on when and how it was delivered or that you must log in to one machine to read your mail. Also, at CERN you might want to read and send mail from, say, the CHORUS Work Group Server and the RSPLUS service; having a mail spoolfile on each cluster would not be very helpful.

For all of these reasons we have introduced a special mail server at CERN which keeps mail sent to you in one central place and which allows you to read your mail from any workstation at CERN. (If you use a mail agent other than the recommended ones, you will need to check that it is compatible with the mail server, though.)

8.2.3 Forwarding mail

Mail sent to any of the WGS or PLUS clusters will be forwarded automatically to the Mail Server as we do not allow local mail spoolfiles on these systems. If you have your own workstation, however, you may wish to create a `.forward` file in your home directory so that any mail delivered to your machine will be sent on to the mail server. There is a simple command called `mailfwd` that will automatically setup your `.forward` file for you so you can just use the command

```
mailfwd <your_userid>@mail.cern.ch
```

A correct forward setup is also necessary if you want to forward mail outside CERN. In this case if you have an account on the CERN mail server you must set at the same time the forward on the server, so use instead

`mailfwd -global your-preferred-address`

When you want to stop forwarding mail you can use the command “`mailfwd -reset`”, or respectively “`mailfwd -global -reset`”.

Of course, if you have your own workstation you can also choose have all your mail delivered to a spoolfile on that workstation. However, remember that if you do this

- you will not be able to read your mail from anywhere else,
- your workstation must be kept on to receive mail all the time, and that
- you must backup your own mail spoolfile; mail stored on the mail server is backed up by us.

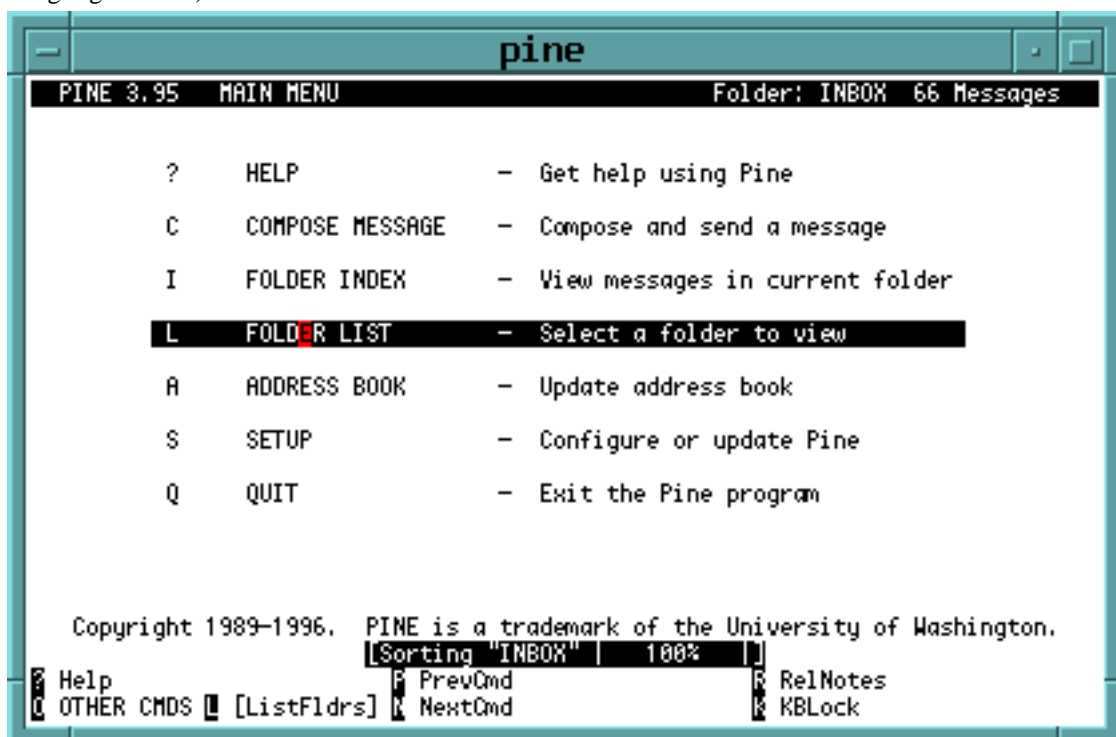
N.B. If you set the mail server forward to the generic address `name.surname@cern.ch` be sure that your preferred e-mail address (PEM) does not point to the mail server itself!

8.2.4 A Quick Introduction to PINE

Pine can be started in a number of ways.

1. All users can type the command `pine` at a shell prompt.
2. On an X display use the leftmost button of the mouse and choose “Mail agent” from the “Standard Tools” submenu.
3. If you use ADAM you can start pine by double clicking on the “Mail” icon in the “Standard tools” tray.

In all cases you should end up with a screen looking like the MAIN MENU figure below (look for the title in the upper highlighted line).

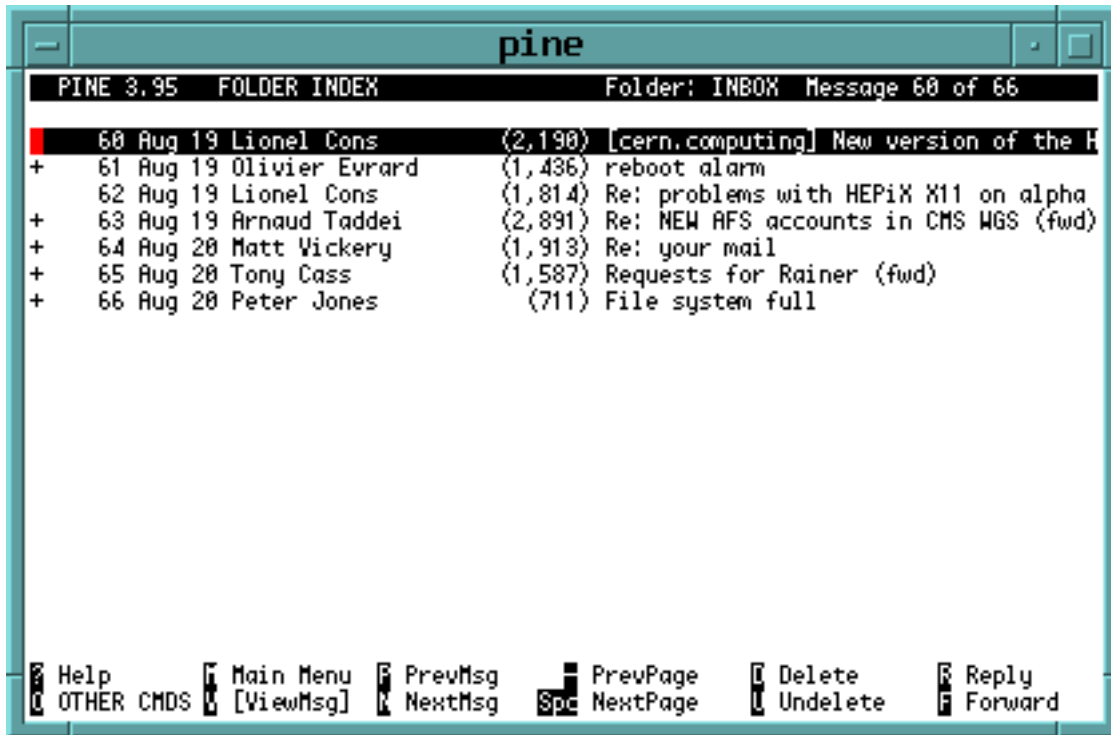


Note the two lines at the bottom of the screen show other commands available by hitting some key — entering `?`, for example, will show some help information. All pine pages show important commands on the bottom two lines of the screen. If there are more commands available than will fit on the two lines then pressing `O` (for Other commands) will show another subset—the screen above shows this option.

N.B. Commands are case insensitive, though we use uppercase letters in the rest of this introduction.

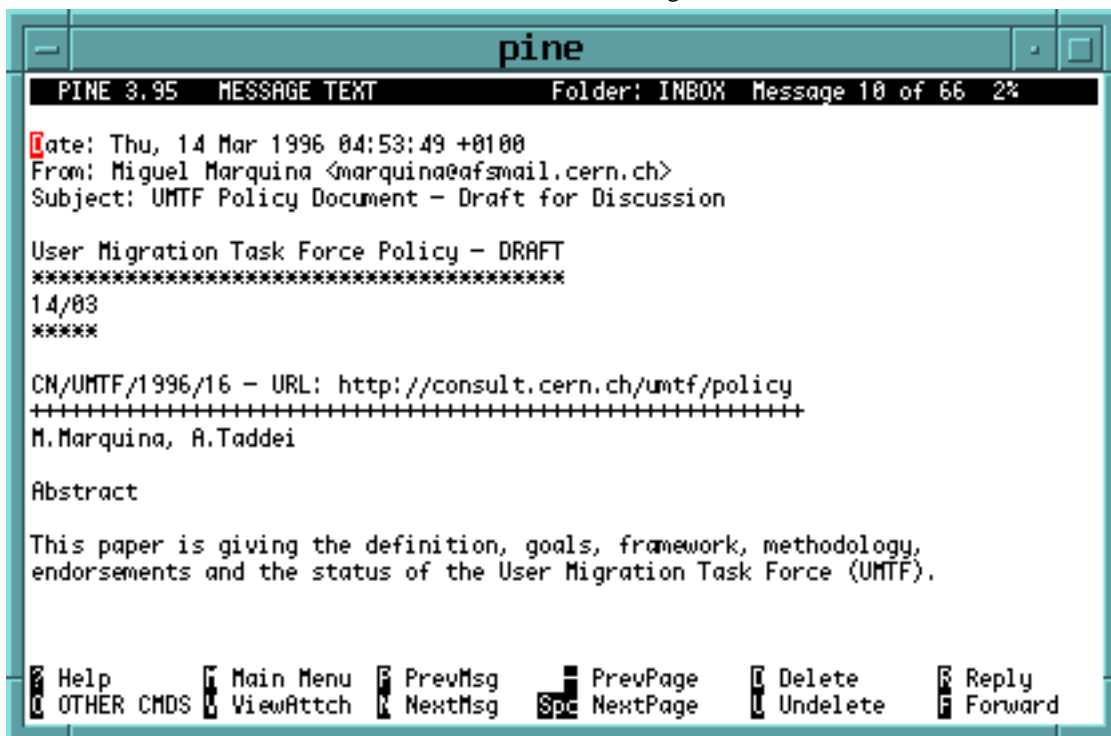
Reading Mail

To read incoming mail select the “View messages in current folder” option either by typing `I` or by moving the highlight line with the arrow keys and then pressing `<return>` (or `<enter>` depending on the keyboard type — we always use `<return>` in the rest of the chapter). You should end up with a screen like the FOLDER INDEX in the next figure.



(You can also go straight to this display by starting pine with the command `pine -i` from the command line. I find this most useful; you can read, send and save mail all from this screen without ever needing the main menu.)

This menu shows a list of your incoming mails¹ with a flag down the left hand side. The flag column is blank if you have just read a mail, otherwise you will see **N** for new, **A** for answered or **D** for deleted. This status also appears in the top right hand corner of the screen for the currently selected mail item, but as **NEW**, **ANS** or **DEL**. To read a mail just move the highlight line with the arrow keys to the mail you want to read and then press `<return>`. You should see a screen like the MESSAGE TEXT figure below.



1. Actually, it shows the index of the current mail folder. This is the inbox when pine is started but you can change the current folder so that you can read mails you have saved. See the section on changing mail folders below.

You can scroll up and down the mail with the `-` and `<space>` keys—note the list on the bottom two lines of the screen.

Replying to mail

You can reply to a mail by hitting the `R` key. (You don't have to be reading it first; you can also highlight the mail in the list using the arrow keys and hit `R` for reply rather than `<return>` to read.) You will be asked if you wish to include the original text in the reply and, if there were several recipients of the mail, whether everyone should see your reply. After you have answered these questions you see a screen like that when sending mail—see the section below.

Forwarding a mail

You can forward a mail to someone else by using the `F` key. Just as for reply, you can forward a mail when you are reading it or after you have selected it from the index by moving the highlight bar with the arrow keys. When you forward a mail you will immediately see a screen like that for sending mail and be positioned so that you can enter the address for the person to whom you want the mail to be forwarded.

Saving mail

You can save a mail to some folder using the `S` key. Again, this works from the mail reading or the mail index display. Pine will display a little prompt at the bottom of the screen asking for the name of a folder in which the mail should be saved. After you save a mail you will still see it in the mail index but it will be flagged with a `D`, just as if you had deleted it.

Deleting mail

To delete a mail use the `D` key. There are no prizes for guessing that this will work either when reading the mail or after selecting the mail with the highlight bar. If you are reading the mail pine will automatically show you the next mail in the folder if there is one; if you are in the mail index the highlight bar will move down to the next mail item if possible.

Notice that deleted mails still stay in the inbox, they are just flagged as having been deleted. This means that if you change your mind you can either undelete them with the `U` key (which works when reading mail or in the mail index) or save them if you haven't already done this. (Remember that saved mails are automatically flagged as deleted.)

If you really want to remove mails from your inbox (and you should; mail response will be poor if you keep hundreds of mails in the inbox) you need to eXpunge them from the inbox by using the `X` key. This works **only** from the index. (All those that thought it would also work from the mail reading screen should write out "I must not jump to conclusions." 100 times.)

Printing mail

To print a mail item you need to use the `Y` key (`prYnt`, `P` already being taken for Previous mail...). Like reply, delete and save this works just as well from the index as from the mail reading screen.

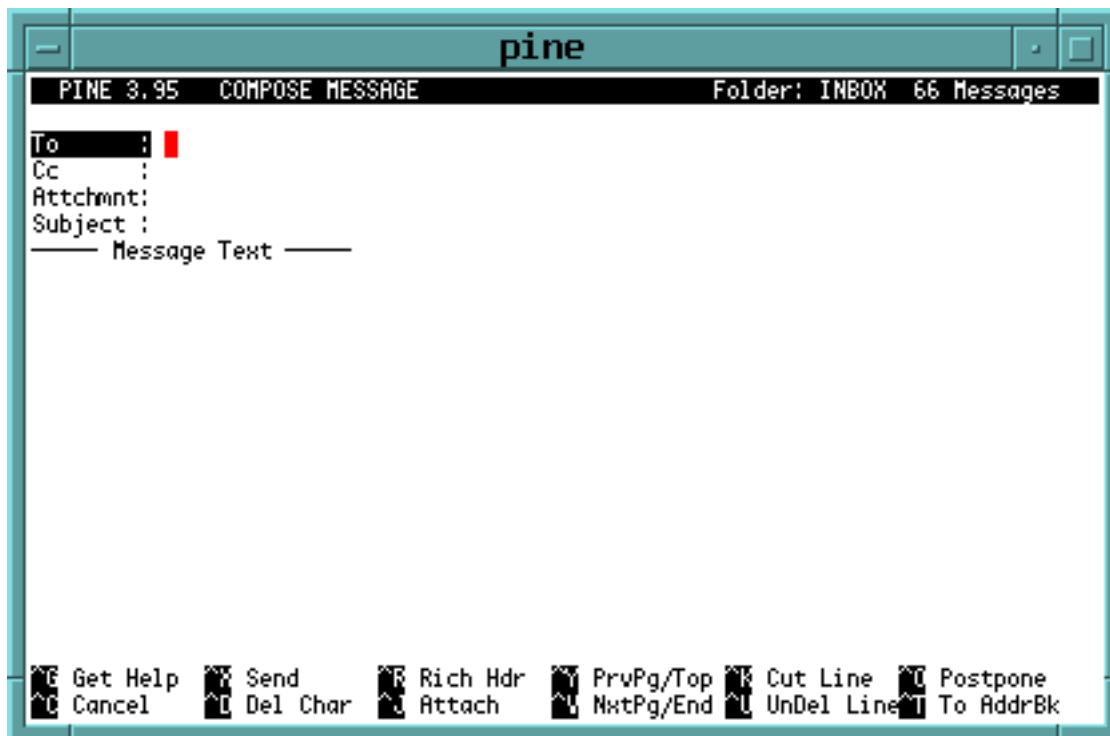
The default print command is set to `xprint` in the pine configuration file for the CERN site (on ASIS) but this can be changed, either permanently in the user's configuration or at printing time by using the `C` key (Custom print). The default printer, specified by the environment variable `$PRINTER`, must be set by the user in the shell startup file otherwise the print will fail unless the option `-P<printername>` is specified explicitly.

8.2.4.1 Sending Mail

To send a mail you use the `C` key for Compose. You can choose compose from the main menu (where you can also select the Compose Message line with the highlight bar and then press `<return>`), from the mail index page or when reading a mail. However you do it you should see the COMPOSE MESSAGE screen shown in the next figure.

You can see that there is a **mail header**, a separator line (`----- Message Text -----`) and then space for you to type your mail message.

Filling in the header is straightforward—after you fill in each section (or leave it blank) just hit `<return>` to move to the next line. After you fill in the `Subject :` line pressing `<return>` will move you to the start of the



mail body. The meanings of the fields in the mail header are as for any other mailer:

- To
the address(es) of the person or people to whom you are sending the mail.
- Cc
the address(es) of the person or people who should see a copy of the mail.
- Attchmnt
the name(s) of file(s) to send with the mail.
- Subject
the subject of the mail.

If you have multiple addresses for the `To` or `Cc` fields, or multiple files to list as attachments use a comma to separate the different addresses or files.

You must give either full mail addresses (like `user@host`) or use nicknames for the `To` and `Cc` fields. See the section below for information about creating pine nicknames.

Once you have filled in the mail header, typing the mail message is easy. You just need to type away in the message body area beneath the `----- Message Text -----` line. The editor used by pine, pico (for **pine composer**) will automatically wrap round to the next line when necessary. The bottom two lines of the screen show the key combinations you can use to move about in the mail, delete lines or use the spell checker. (The little `^` character means that you have to hold down the `<control>` key as well, so press `<control>` and `K` to delete a line of text.)

If you want to include text from another file in your mail message¹ then you can use `^R` (`<control>` and `R`) to read a file—pine will ask you for the name of the file to read.

Sending the mail message

When you have finished composing your mail use `^X` (`<control>` and `X`) to send the mail. Pine will ask you if you are sure (with the prompt `Send message?`); hit `Y` to send or `N` if you are not sure — typing `N` will not lose the message, just put you back in compose mode so you can make any changes or delete the message as you like.

1. Note that this is different from attaching a file. If you read a file in then it appears in your mail just as if you had typed it. An attached file is sent as a completely separate part of your mail message.

Canceling the mail message

If you decide you don't want to send the message then use **^C** (<control> and **C**) to cancel it. Pine will prompt to make sure you really want to cancel the message

```
Cancel message (answering "Yes" will abandon your mail message) ?
```

Hit **Y** or **N** as appropriate.

Postponing a mail message

If you want to postpone composing a mail message so that you can do something else but then continue later, use the **^O** command (<control> and **O**). Pine will save the message for you and then, each time you choose compose, ask if you want to continue with the postponed message. You don't have to — pine will let you compose a new mail without destroying the postponed message — but just choose compose when you decide you want to continue and then carry on as normal.

8.2.4.2 Changing mail folders

Pine starts with the current mail folder set to your inbox. This is convenient since you will normally want to see your incoming mail. To see mail that you have saved in other folders you need to use the **Goto Folder** or the **List Folders** command. Both commands can be selected using the appropriate letter (**G** or **L**) from the main menu, the folder index or when reading a mail. List Folders can also be selected from the main menu page by moving the highlight bar with the arrow keys.

With the Goto Folder command you should see a dialog like

```
GOTO Folder in <Incoming Message Folders>:
```

or

```
GOTO Folder in <mail/[ ]> [sent-mail]:
```

The name within <> is the name of the folder collection¹. Type the name of the folder you want and hit **<return>** and pine will show you the index page for this new folder. If the folder is in a different collection from the one shown, you have to use **^P** (<control> and **P**) or **^N** (<control> and **N**) to select the Previous or Next Collection before hitting the **<return>** key. To go back to your inbox just use the Goto command again.

If you have forgotten the name or the collection of a folder you can use the List folders command or **^T** (for "To folders") after the Goto folders command. Either way, pine should show the GOTO: SELECT FOLDER screen shown in the next figure.

Use the arrow keys to select the folder you want with the highlight bar and then the **<return>** key to select the folder. If you used the List folders command you will see the folder index straight away; with the Goto folder command the name of the folder is entered on the dialog line and you need to hit **<return>** again to view the folder index.

8.2.4.3 Defining nicknames with pine

Pine allows you to have an addressbook to store nicknames for people to whom you send mail often (or whose Email addresses you can never remember). Using nicknames is easy; you can either

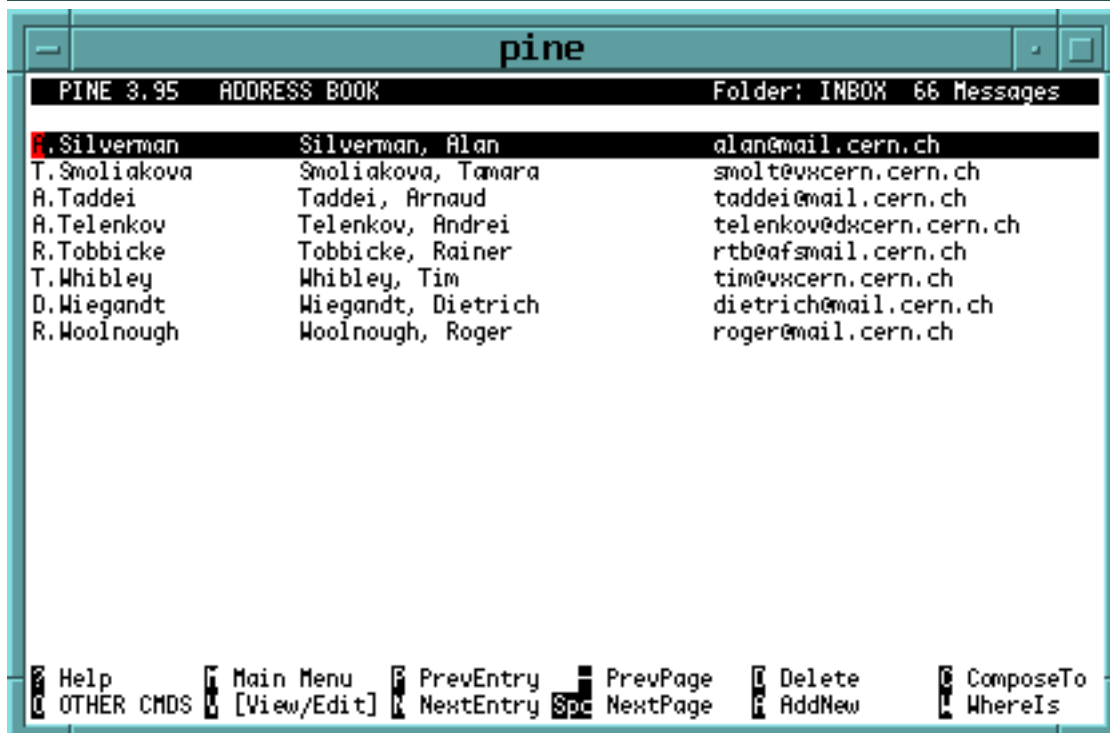
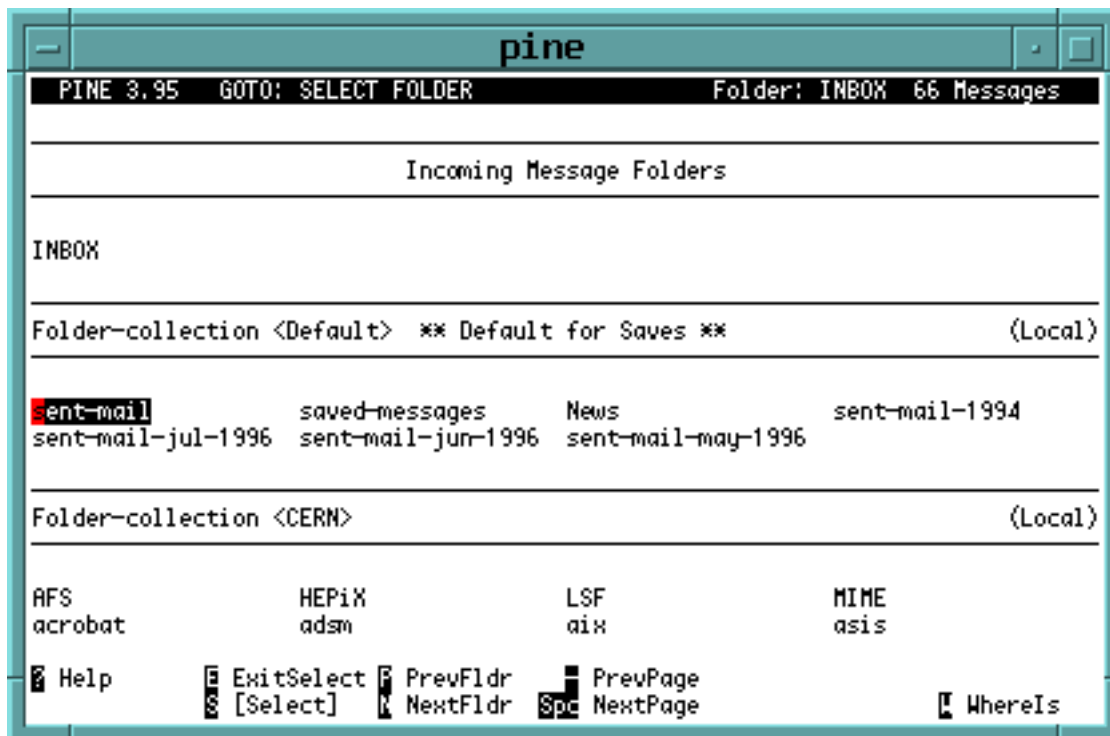
1. simply type in the nickname in the **To** or **Cc** field of the mail header when sending mail, or
2. use the **^T** command (To addressbook) when the **To** or **Cc** field is highlighted and then select a nickname using the arrow keys.

To create nicknames you need to choose the "Update address book" option from the main menu. Use the **M** key if necessary to return to the main menu and then select the address book option using either the **A** key or by moving the highlight bar with the arrow keys and then pressing **<return>**. You should see a screen something like the ADDRESS BOOK figure below — but only if you have already some entries in your addressbook.

As usual the bottom two lines of the screen show the available commands. Use the **A** key to add a new nickname. The ADDRESS BOOK (Edit) screen below, with embedded help text, will appear.

When this is finished the entry will be added to the list (which is ordered alphabetically) using the last name, first

1. It is possible to have several folders with the same name, one in each collection. By default two collection are specified: <Incoming Message Folders> (containing the INBOX) and <mail/[]> (default for Saves).



name field.

When you have finished use the **M** key to return to the main menu; you can select Compose directly and pine will automatically use the highlighted nickname (usually the new nickname you have added) for the To field. However, when you finish sending your mail you will be returned to the addressbook listing.

8.2.5 Leaving pine

When you have finished with pine use the **Q** key to quit. If you have any deleted mail then pine will ask you if it should be expunged before pine exits (See “Deleting mail” on page 33.)



9. A quick introduction to LSF at CERN

This chapter provides a short tutorial for users of the Load Sharing Facility (LSF) software. Users should be familiar with executing commands in a UNIX and/or X11 environment. A complete set of user and administrator guides can be found at [18].

9.1 What is LSF?

LSF is a product that runs batch or interactive jobs, selecting execution hosts based on current load conditions and the resource requirements of the application. Interactive jobs start running as soon as a command is entered. Batch jobs are kept in queues until the appropriate resources are available.

Hosts are divided in clusters. LSF provides features to make transparent access to resources in separate clusters, e.g. PLUS/WGS and SHIFT.

9.2 How to use LSF

9.2.1 Authentication

In order to submit jobs the user must be authenticated on the execution node. The authentication is based on Kerberos: a Kerberos Ticket Granting Ticket (TGT) is created whenever a user makes a login via xdm (on a X-Terminal) or via telnet to any LSF host, and has the same lifetime as an AFS token. On the contrary, logging via rsh/rlogin does not automatically provide a TGT.

The output

```
Failed in an LSF library call: External authentication failed
```

means that the user does not have a valid TGT: this can be verified with the command `tokens`, which should show a line like:

```
User xxx's tokens for krbtgt.CERN.CH@cern.ch [Expires ...]
```

If the line does not appear, or if it appears but ending with “[>> Expired <<]”, the command `klog` can be used to create the TGT.

N.B. The TGT is not replicated in the execution node, so commands that require Kerberos authentication cannot be executed via LSF.

9.2.2 AFS and file access

In order to access files in AFS directories, a valid token must be held at the time of the submission. The token is transmitted and replicated in the execution node, and its lifetime extended for the duration of the job.

9.2.3 Queues

The basic settings of LSF queues was chosen in order to

- ease maintenance of batch services
- have compatible time limits on machines with different CPU power (measured in CERN Units - defined as the equivalent of one IBM 370/168 unit)
- give to users in different experiments a common way to define where to send their jobs.

CPU limits are defined in terms of NCU (New CERN Units = 100 CERN Units). The CPU power of typical machines in use now ranges between 0.30 NCU (HP D250 PA7100@125MHz) and 1.37 (Digital Ultimate WS Alpha@533MHz).

The following set of queues is now defined (the equivalence in terms of old names is indicative because it depends on the service/experiment):

1nm - 1 NCU minute

The equivalent of 100 minutes on a IBM 370/168, or about 3 minutes on a HP 735. A typical 'express' queue.

8nm - 8 NCU minutes

About 1/2 hour on a HP 735. In the range of old 'express' to 'short' queues.

1nh - 60 NCU minutes

1 NCU hour, or about 3 hours on a HP 735. A 'short' or 'medium' queue.

8nh - 480 NCU minutes

8 NCU hours, or about 1 day on a HP 735. A 'medium' queue.

1nd - 1440 NCU minutes

1 NCU day, or about 1 day on a IBM model 43P140@333MHz. In the 'long' range.

1nw - 10080 NCU minutes

1 NCU week, about 6 days on a Digital PSW500au. Enough for a 'verylong' queue.

4nw - 40320 NCU minutes

4 NCU weeks, for special engineering applications.

Each experiment can choose a subset of the above queue above for their services. In general, this subset includes queue that have a CPU limit greater or equal than old NQS queues.

On interactive WGS the following queues are defined:

bg - 480 NCU minutes

Background queue. Active 24h a day with low priority.

np - 480 NCU minutes

Non prime queue, active during nights and week-ends

9.2.4 Command line programs

A short help about single commands can be obtained adding the "-h" option. Most commands that display information have a "-l" (long output) option. Some commands accept a *cluster* name as first parameter. The full manual pages are available online.

9.2.4.1 Submitting interactive jobs

Interactive jobs (either scripts or programs) can be submitted to a remote host with the command `lsrun`:

```
lsrun dvips file.dvi
```

N.B. Interactive jobs on WGS are subject to CPU restrictions, and they can be killed if certain thresholds (typically 1 CPU hour) are exceeded.

9.2.4.2 Submitting batch jobs

Executable scripts or binaries can be submitted with the command `bsub`. A few useful options are:

-q *queue*

specify queue

-c *cpu*

specify CPU limit

-w *dependencies*

condition for job dependency

-r

automatic resubmission in case of host crash

-o outfile

specify output file

-R resource

specify a resource on the execution host, e.g. "type==any" to run on any machine in a multi-architecture cluster.

```
[rsplus] ~ > bsub /tmp/testjob
Job <924> is submitted to default queue <8nm>.
```

Users that are submitting scripts used with other batch systems should verify whether they use any special feature not supported by LSF.

N.B. Jobs submitted with a request to execute on hosts/clusters where the specified resource/queue is not defined will stay pending forever. Unless specified otherwise (by the user or by the queue resources) a job will be executed on a host with the same architecture of the submission host.

9.2.4.3 Finding information about batch jobs

The status of submitted jobs can be displayed by the command `bjobs`. In case jobs are pending, the "-p" option will also display the reason. The "-l" option shows all the information about the job.

```
[rsplus] ~ > bjobs -l

Job Id <924>, User <alemi>, Project <default>, Status <PEND>, Queue <8nm>,
      Command </tmp/testjob>
Tue Mar  4 16:53:45: Submitted from host <rsplus01>, CWD <${HOME}>;

PENDING REASONS:
The CPU utilization (ut) is beyond threshold: 6 hosts;

SCHEDULING PARAMETERS:
      r15s  r1m  r15m  ut      pg   io   ls   it   tmp  swp  mem
loadSched -   -   1.0  0.9   -   -   -   -   -   -   -
loadStop  -   -   2.5  -     -   -   -   -   -   -   -
```

The job output can be displayed by the command

bpeek job_id

The "-f" option makes the program wait for further output rather than exiting after reaching the end of file.

9.2.4.4 Job output

By default LSF returns the job output by mail after the execution is completed. At CERN the output is moved to a subdirectory created in the submission directory, and the mail just returns to the user the full path of this subdirectory. Users can anyway use the "-o" option of `bsub` to define their own output file.

9.2.4.5 Finding equivalent clusters

The command `lsclusters` shows information about LSF clusters known by this host.

```
[rsplus01] ~ > lsclusters
CLUSTER_NAME  STATUS  MASTER_HOST  ADMIN  HOSTS  SERVERS
rsplus        ok      rsplus01     lsfadmin  15     15
parc          ok      parc01       lsfadmin  10     10
rsbatch       ok      rsbat004     lsfadmin  33     33
```

9.2.4.6 Batch host information

The command `bhosts` displays the number and the status of batch jobs in each host of the local cluster.

```
[rsplus01] ~ > bhosts
HOST_NAME  STATUS  JL/U  MAX  NJOBS  RUN  SSUSP  USUSP  RSV
rsplus01  ok      -    -    0      0     0      0      0
rsplus02  ok      -    -    0      0     0      0      0
rsplus03  ok      -    -    0      0     0      0      0
[...]
```



```
[rsplus01] ~ > bhosts rsbatch
HOST_NAME      STATUS      JL/U  MAX  NJOBS  RUN  SSUSP  USUSP  RSV
rsbat004      closed      -     2    2     1    0     0     1
rsbat005      closed      -     2    2     1    0     0     1
rsbat006      closed      -     2    2     1    0     0     1
[...]
```

9.2.4.7 Batch queues

The command `bqueues` displays a list of batch queues available from the local system. The option “`-m remote_cluster`” shows details about non-local queues.

```
[rsplus] ~ > bqueues
QUEUE_NAME     PRIO     STATUS      MAX  JL/U  JL/P  JL/H  NJOBS  PEND  RUN  SUSP
8nm            60      Open:Active  -    -    -    -     0     0     0     0
e_8nm         60      Open:Active  -    -    -    -     0     0     0     0
1nh           50      Open:Active  -    -    -    -     6     5     1     0
e_1nh        50      Open:Active  -    -    -    -     0     0     0     0
8nh           40      Open:Active  -    -    -    -     9     2     7     0
1nd           30      Open:Active  -    -    -    -    13     5     8     0
e_1nd        30      Open:Active  -    -    -    -     3     0     3     0
e_1nw        20      Open:Active  -    -    -    -     3     0     3     0
[rsplus] ~ > bqueues -m rsbatch
QUEUE_NAME     PRIO     STATUS      MAX  JL/U  JL/P  JL/H  NJOBS  PEND  RUN  SUSP
8nm            60      Open:Active  -    5    1    -     1     0     1     0
1nh           50      Open:Active  -    5    1    -    29    23     6     0
8nh           40      Open:Active  -    -    1    -     9     2     7     0
1nd           30      Open:Active  -    -    2    -    14     1     9     0
```

The command `bparams` displays the name of the default queue for submissions when `bsub` is invoked with no options.

```
[rsplus] ~ > bparams
Default Queues: 8nm 1nh 8nh 1nd
Default Host Specification: NCU
Job Dispatch Interval: 60 seconds
Job Checking Interval: 30 seconds
Job Accepting Interval: 60 seconds
```

9.2.4.8 Batch clusters and remote queues

The command `bclusters` displays a list of local queues together with their relationship with queues in remote equivalent clusters.

```
[rsplus] ~ > bclusters
LOCAL_QUEUE    JOB_FLOW  REMOTE      CLUSTER     STATUS
1nd            send      1nd         rsbatch     ok
8nh            send      8nh         rsbatch     ok
1nh            send      1nh         rsbatch     ok
8nm            send      8nm         rsbatch     ok
e_1nw          send      1nw         parc        ok
e_1nd          send      1nd         parc        ok
e_1nh          send      1nh         parc        ok
e_8nm          send      8nm         parc        ok
```

9.2.4.9 Displaying host information

The command `lshosts` shows information about all hosts in equivalent clusters.

```
[rsplus] ~ > lshosts
HOST_NAME      type     model      cpuf  ncpus  maxmem  maxswp  server  RESOURCES
rsplus01      IBMAIX4  M43P240_  0.7   2     511M   1024M   Yes    (ppc aix unix plus)
rsplus02      IBMAIX4  M43P240_  0.7   1     511M   1024M   Yes    (ppc aix unix plus)
rsplus03      IBMAIX4  M43P240_  0.7   2     511M   1024M   Yes    (ppc aix unix plus)
[...]
```

9.2.4.10 Displaying load information

The command `lsload` shows information about the load and other system indexes (e.g. paging, swap, memory) of hosts in the cluster.

```
[rsplus] ~ > lsload
HOST_NAME      status  r15s  r1m  r15m  ut    pg  ls    it    tmp    swp    mem
rsplus14      ok     0.1   0.2   0.2   6%    2.5  6    109   182M   981M  1430M
rsplus12      ok     0.2   0.3   0.5   16%   4.2  32    0    174M   770M  257M
rsplus04      ok     0.4   0.6   0.9   26%   4.2  31    0    170M   779M  325M
rsplus11      ok     0.5   0.7   0.9   39%  26.0  29    0    162M   810M  267M
[...]
[rsplus] ~ > lsload rsbatch
HOST_NAME      status  r15s  r1m  r15m  ut    pg  ls    it    tmp    swp    mem
rsbat018      ok     0.0   0.0   0.0   1%    0.4  0   1309   52M   734M  867M
rsbat015      ok     0.1   0.2   0.4   13%   6.0  0    16    54M   726M  76M
rsbat013      ok     0.1   0.3   0.5   16%   6.8  0  15400   54M   723M  65M
rsbat010      ok     0.1   0.2   0.2   15%   5.3  1  3272   52M   724M  68M
[...]
```

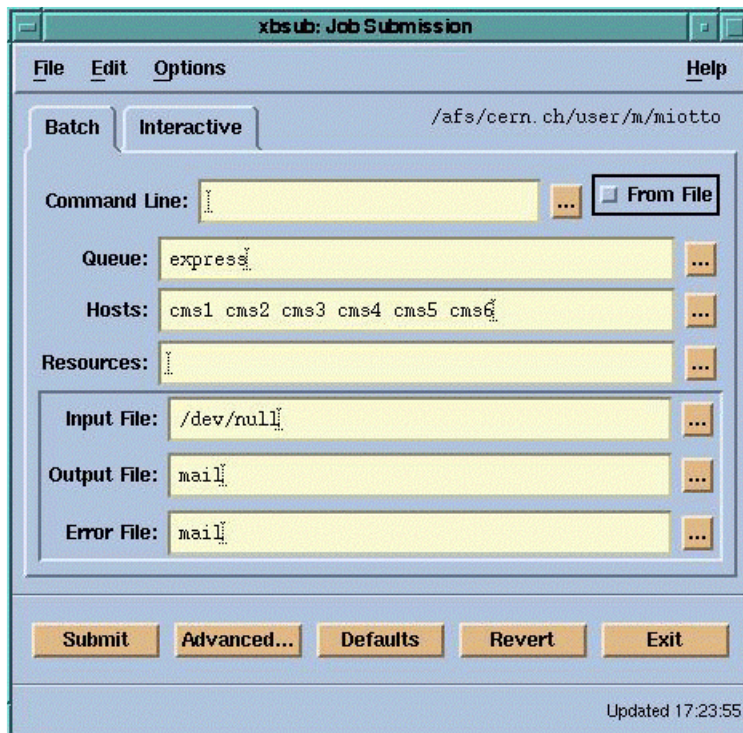
9.2.5 X11 programs

The above commands can be more easily executed via a graphic interface, where options and parameters are replaced by menu items and form entries. The screen shots below give an idea of their possibilities. For further details refer to the full LSF documentation or to the online help (rightmost menu).

These programs do not give, however, full access to information in remote clusters, and command line programs still have to be used for this purpose.

9.2.5.1 Hosts and queues information: `xlsbatch`

9.2.5.2 Job submission and handling:xbsub



9.2.6 LSF documentation

The full LSF documentation can be found at [18]. A list of known bugs is also kept up to date (whenever possible).

9.3 The CERN job starter

At CERN a job starter script, executed before users' jobs, modifies the default LSF behaviour.

9.3.1 Working directory

The working directory is automatically set up by the CERN job starter to be in a local disk to the execution host, rather than in the submission directory as per LSF default.

9.3.2 Environment

To take care of the possible different architectures of the submission and execution hosts, the environment is recreated before the job execution by re-sourcing the HEPiX scripts

9.3.3 Output file

The output file is not returned by mail.

If the user does not specify it explicitly with the “-o” option, the directory LSFJOB_\$(LSB_JOBID) (or LSFJOB_\$(LSB_JOBID)_\$(LSB_REMOTEJID) in case of jobs executed in different clusters) is created in the submission directory, and the standard output copied there to a file named STDOUT. The job starter automatically trims this file if its size is too big (beginning and end of the standard output are saved).

9.3.4 Other files

Files can be automatically returned to the directory where the output file is stored by listing them – one per line – in a file named `RETURN` in the working directory itself. Shell expansion is performed, so wildcards `*`, `?` and `[. . .]` can be used.

All files in the working directory, except the standard output, are automatically deleted when the job terminates. For debugging purposes it is possible to save some files from being deleted by listing them – one per line – in a file named `NOCLEAN` in the working directory. The same file expansion is performed. The working directory is deleted after 24/48 hours, depending on the service and the space available on the local disk.

9.4 Fault tolerance

No LSF operation is possible when the master host in the cluster is unavailable. Possible causes can be:

- The cluster has been reconfigured: this operation will be completed in a few minutes.
- The master host just crashed: another host will immediately take the mastership. When a execution host crashes all jobs running on it are lost, but no other pending or running jobs are affected. Jobs can be submitted to LSF with an option to automatically restart if the job is lost because of a host failure.

In addition, no new job submission is possible when one of the following conditions applies:

- All licenses are in use or the license server is unavailable.
- The file system with the shared cluster information is unavailable.

10. Access to Magnetic Tape Data

10.1 Tape Staging

Tape access on ION is provided for reading and writing using the CERN Program Library RFIO package as used by ZEBRA and EPIO with the C I/O options. If files in other formats are required they can be copied between local disks (eg the \$WORKDIR directory) using the `rfcp` command (this has the same syntax as the Unix `cp` command). This is necessary for example to read tar files or those to be read with FORTRAN input/output. Such files can, however, be staged out from \$WORK (or interactively) using the `stagewrt` command rather than using `rfcp`.

10.2 Stage Commands

Tape files are moved to and from disk by the following commands:

| | |
|-----------------------|--|
| <code>stagein</code> | Allocate and link space in a stage in pool and read tape data into the linked file. |
| <code>stageout</code> | Allocate and link space in a stage out pool. It is recommended to use the <code>sp2_out</code> pool (see below). |
| <code>stageput</code> | Write the stageout file to tape. |
| <code>stagewrt</code> | Write an existing file to tape |
| <code>stageqry</code> | Query the status of staging requests. |

For more information, type `man one-of-the-above-commands`.

10.3 Stage Pools

With the current software, stage pools are associated with particular server nodes. The ION default stage pool is called `public` and it is enough to refer to `-p public` in the stage commands. To access other pools users must define an environment variable called `$STAGE_HOST` before using stage commands referring to the matching pool.

10.4 Examples

- For example to create a file linked to `fort.10` for output staging:

```
setenv STAGE_HOST sp018 (in tcsh), or,
export STAGE_HOST=sp018 (in zsh) followed by
stageout -v 123456 -U 10 -p sp2_out
```

Stream 10 could then be opened with a `CFOPEN` call and written to with the Zebra FZ package. After the writing job step finishes, to trigger writing to tape requires:

```
stageput -U 10
```

- To extract a tar file from file 1 of tape 123456, written with a blocksize of 10240:

```
stagein -v 123456 -q 1 -F F -b 10240 myfile
this creates a local file myfile which is in fact a link into the public staging pool. Only rfio can directly read such a file.
rfcp myfile local.tar
this copies from the staging pool into the current directory
tar -xf local.tar
to unpack the tar file into the current directory.
```

- A loadleveler batch job to read an FZ file written with the small blocksize of 3600:

```

#!/usr/bin/csh -x
# @ job_cpu_limit = 1000
# @ queue
set CERNLIB=`cernlib`
cat > temp.f <<EOF
COMMON/MYSTOR/IFENCE(10),LINKS(10),LINKR(10),ISTORE(4000)
COMMON/QUEST/IQUEST(100)
DIMENSION LQ(999),IQ(999),Q(999),MMBK(5),iuhead(10)
EQUIVALENCE (LINKS(9),LQ(9),IQ(1),Q(1))
DATA MMBK/4HTEST,1,1,1000,0/
CALL MZEBRA(-1)
CALL MZSTOR(IXSTOR,'MYSTOR',' ',IFENCE,LINKS(1),LINKR(1)
+,ISTORE(1),ISTORE(2000),ISTORE(4000) )
CALL CFOPEN(IQUEST(1),0,900,'R',0,'fort.10',istat)
write(6,('( ' cfopen status=' ',i3)') istat
CALL FZFILE(10,900,'LXR')
IQUEST(1)= -1
nuh= 10
10 CONTINUE
CALL FZIN(10,IXSTOR,LADD,2,' ',NUH,IUHEAD)
WRITE (6,'((15I5))')(IQUEST(I),I=1,15)
WRITE(6,'( ' LWORD 1 ' ',10I8)') (LQ(LADD+I),I=0,9)
WRITE(6,'( ' IWORD 1 ' ',7I8)') LADD,(IQ(LADD+I),I=0,5)
IF(IQUEST(1).NE.0) GO TO 20
CALL MZDROP(IXSTOR,LADD,' ')
GO TO 10
20 CONTINUE
END
EOF
hepf77 temp.f $CERNLIB
stagein -v I29012 -U 10 -F F -b 3600 -L 3600
a.out

```

10.5 Tape problems

In case of problems with tapes, please contact Tape.Support@cern.ch.

Appendix A. Bibliography

A.1 CERN documents

A.1.1 Recommended manuals

- [1] The CERN UNIX guide
(CERN-UCO/94/164 <http://consult.cern.ch/writeup/unixguide>)
- [2] The AFS user's guide
This contains a lot of important information on AFS.
(CERN-UCO/94/167, <http://consult.cern.ch/writeup/afsguide>)
- [3] "Shell Support - tcsh and zsh for pedestrians"
(CERN-UCO/94/163, <http://consult.cern.ch/writeup/shellsintro>)
- [4] Guide for the Usage of X Window at CERN (X Window System and X Terminals)
(CERN-UCO/95/173, <http://consult.cern.ch/writeup/xusage>)

A.1.2 Other reading

- [5] The CERN security Handbook
This explains how and why you should keep your account secure.
(CERN-UCO/93/161, <http://wsspinfo.cern.ch/file/security>)
- [6] Electronic Mail Services at CERN
(CERN-UCO/96/6, <http://consult.cern.ch/writeup/mailguide>)
- [7] The CERN Electronic Mail Guide for PC users
(CERN-UCO/1997/198, <http://consult.cern.ch/writeup/pcmailguide/>)
- [8] Introduction to Pine
(CERN-UCO/1995/1223, <http://consult.cern.ch/writeup/refcard23/>)
- [9] The HEPiX shells and X11 Login Scripts Project - Implementation at CERN - User Guide
(CERN-UCO/95/184, <http://consult.cern.ch/writeup/hepixuser>).
- [10] The CERN Zephyr manual.
A system to send interactive messages to other users.
(CERN-UCO/94/165, <http://consult.cern.ch/writeup/zephyr>)
- [11] CERN XTerminal Guide
(CERN-UCO/93/158, <http://consult.cern.ch/writeup/xterminal>)
- [12] M. Goossens, A.Samarin, "TeX at CERN - Local Guide"
(CERN/US/90/136)

A.1.3 WWW pages

- [13] Computer Security at CERN
(<http://wwwinfo.cern.ch/dis/security/>)
- [14] Pine Information Center.
(<http://www.washington.edu/pine>)

- [15] Printer Support WWW page
(<http://wwwcio.cern.ch/Others/printerspage.html>)
- [16] User Registration
(<http://consult.cern.ch/uco/registration.html>)
- [17] FVWM2 Documentation
(http://wwwinfo.cern.ch/umtf/working-groups/X11/window_managers/fvwm/fvwm2.html)
- [18] LSF - Load Sharing Facility
(<http://wwwinfo.cern.ch/pdp/lsf/>)
- [19] Web Office
(<http://www.cern.ch/WebOffice/>)
- [20] Communication Systems Group
(<http://wwwcs.cern.ch/>)

Symbols

.forward..... 11, 16, 30
 .rhosts 11
 .Xresources 20

A

AFS..... 14
 access control lists (ACL)..... 15
 CERN path names..... 15
 file security 15
 home directory structure 16
 tokens 14

B

batch 38
 job starter 43
 queues 38, 41
 submission 39

C

Central Simulation Facility..... 7
 changing
 login shell..... 11
 password 10, 14
 chsh..... 11
 connecting
 via dumb terminal 10
 via PC..... 10
 via X Terminal 10
 CSF..... 7
 customisation
 X11 environment 26
 customization
 shell environment..... 19, 20
 X11 environment 20

F

forwarding mail 11, 30
 fvwm..... 20

H

HEPiX scripts..... 18

I

inbox..... 30
 ISS 8

K

Kerberos TGT 38
 klog 14, 38
 kpasswd..... 10

L

logging into other services 11
 LSF..... 38

M

mail reader..... 31
 Mail server..... 30
 mailbox..... 30
 messages
 initialising 28
 locating users 28
 sending to other users 28
 subscribing..... 28
 MIME 29, 30

P

password
 changing 14
 choosing..... 13
 pine 31
 PLUS 7
 Public Login Unix Servers 7

S

shells..... 8, 18

| | |
|----------------|----|
| zsh..... | 18 |
| SHIFT | 7 |
| spoolfile..... | 30 |
| staging | 45 |

T

| | |
|--------------|----|
| tapes | 45 |
| tcsh..... | 8 |
| tokens | 14 |

U

| | |
|--------------|----|
| uco | |
| -home..... | 16 |
| -shell | 18 |
| uco | 19 |

W

| | |
|--------------------------|---|
| WGS..... | 7 |
| Work Group Servers | 7 |

X

| | |
|------------|----|
| xlock..... | 14 |
| xrsh..... | 11 |

Z

| | |
|--------------|----|
| zctl..... | 28 |
| Zephyr..... | 28 |
| zlocate..... | 28 |
| zsh | 18 |
| zwgc..... | 28 |
| zwrite..... | 28 |