Chapter 5:
Setting up users

5.1 Overview

Your system will have a number of different users to whom you will want to be able to give facilitites to create files for themselves, to read certain communal files (for example library programs) and to have selective access to other users' files.

The list of authorised users in a SJ Research File Server is kept in a file called the *password file*. This file can be read and saved only by someone with *system privilege*; normally only the system manager himself and only when the front panel key-switch is turned to the SYST position. The password file contains information about each user: his password, any accounts he has access to, and administrative information concerning start-up (boot) options, library directories and user root directories.

If someone logs-on to the system, and his name does not appear in the password file, then he will will be logged on as the *default user*, if one has been set up by the system manager using EDITPASS (see Section 5.3). If no default user has been set by the system manager, the user will receive the message **User not known.**

When a user listed in the password file logs-on, any password he quotes will be checked against the one in the password file, before the log-on is allowed to proceed. He will then be given any rights and privileges listed against his name in the password file. The system will then search the disc on which the user's password file entry was found, for the *User Root Directory* specified for that user in the password file, which by default has *that user's name*, and will set this to be the currently selected directory (see Section 3.3 under *I AM for details). If no appropriately named directory is found, the disc root directory will be selected.

As described more fully in Section 3.3 (under *ACCESS and *ACCOUNT), the *account(s)* to which a user is given access, control two things:

First, every file (or directory) has an account number, and *if a user has access to this account, then he is an owner of that file (or directory).* Only an owner may create files in a directory, and only an owner may delete a file or change its access letters (see Section 3.3 under *ACCESS command). Note that there can be *more than one owner* of a file (or directory), simply by allocating access to its account to more than one user - this can be useful for communal files in a project.

Second, each account has a *credit balance* of storage space, and an attempt to create a file which would cause that balance to become less than zero will be prevented, and cause the error **Account bankrupt**

5.1.1 Keeping a List of Users

It is wise to plan your list of users, and the accounts for them, on paper and keep it up to date. There is no security required for account numbers and users' names, and even a moderately sized system can have more users and accounts than can be displayed on a screen.

User names may have up to ten characters, which may include letters, numbers and dashes, and must start with a letter. Normally the user's name would be his own surname or initials. However, user names must be unique in the system, so you may wish to add figures to the end of a name.

Account numbers range between 0 and 3FF (hexadecimal), but you may of course ignore the hexadecimal part and just use numbers up to 399. Allocating account number 0 gives ownership of the system root directory, so *account 0 should be allocated only to system privileged users.*

5.1.2 Entering Users on to the System

Switch on your File Server unit and log-on as a system privileged user. If you have just taken delivery of the system, use the name SYST, with the password SYST. In other words, at a BBC Microcomputer on the network, type:

```
*I AM SYST SYST
```

You will then need to edit the password file to enter your list of users and their accounts, then create *user root directories* for each of them, and then set the *account number* of each user root directory to the same as that allocated to the user, so that the user has owner access to this directory. These three operations are described below.

5.2 Listusers

The system manager should keep a list of all the users and their accounts as they are created. When the system manager loses track of the account structure then **Listusers** can be run to display all the users and their account numbers. As with Editpass, this program manipulates the password file so the usual precaution of *PROT and turning off the BBC microcomputer, after running the program, should be observed.

To send the ouput to the printer press **CTRL B** before running the program. For more information see the system managers utility programs, section 4.3.

*DIR SYSPROGS

5.3 Editpass

To edit the password the key must be in the SYST position, you must have system privilege and have access to account 0. The Editpass program is also detailed in section 4.3.

5.3.1 Using Editpass

Please see Section 5.3.3 if you are likely to require password files on more than one disc (for example - if you wish some users to have access to the system only when a particular floppy disc is present).

A BASIC program called EDITPASS (fully described in Section 4.3 under EDITPASS) is provided in the library of the system: to use it, check that the front panel key-switch is in the SYST position and type:

```
*PROT        (PROT prevents other users peeking your machine)
CHAIN "EDITPASS"
```

The program will ask whether you want to edit the password file from disc (option **E**), or to start a new password file (option **N**). As delivered, the system disc contains a password file with at least the three users SYST, FRED and BOOT, so you would normally edit this one - it is usually preferable to edit the existing file (unless you have had to delete the password file, in which case use **N**).

The contents of the existing file will now appear, along with a menu of commands. Use command **A** to add new users. To save time, enter all the foreseeable users this stage, ending each one by typing **<Return>**, and end the list by typing **<Return>** on its own.

To allocate accounts, select users one by one, using the brown cursor control keys to move the list past the cursor. After selecting a user, type **X** to enter the *expanded mode* of EDITPASS. The screen will now display the accounts allocated to that user, and will also display other information relating to that user. Press key **A** to enter the account(s) to be allocated to the user. A list separated by spaces, or a series with a dash between numbers, may be entered. To remove accounts, prefix the number(s) with a dash (as minus sign). Press **<Return>** on its own to return to the main list of users, and repeat the account setting for other users.

Before finishing with EDITPASS, check these points:

a) There must be at least one system privileged user on at least one disc (see Section 5.3.3), otherwise it will not be possible to read or save the password file in the future. If you do not wish to use the name SYST, then enter a suitable name for yourself. Type **X**, give yourself system privilege (type **S**), and allocate yourself all accounts (type **A 0-3FF**). Give yourself a sensible password that no-one is likely to guess.

b) There must be a user on at least one disc (see Section 5.3.3) with access to account 0, otherwise it will not be possible to create user directories or edit the password file . Ensure that you have given yourself (and any other system privileged users) access to account 0. If you omit to do so, you may have to delete the password file and start again - a tiresome procedure.

   The EDITPASS program will display a warning if either (a) or (b) above are not complied with. Note however that you do not need a system user on every disc - in fact it can be a useful security aid to lock away floppy disc(s) with a system user when one is not required. See Section 5.3.3.

c) It is wise for the system manager to have a different name for day to day use from the one he uses for system privileged operations. This means that someone watching him log on will probably not see the important password.

d) If you are going to use the name SYST for the system privileged user, then **change the password**, otherwise a casual reader of this manual will be able to log on with system privilege.

e) If you want to have a *default user* on the system, set this up with option **D** in the expanded mode for the appropriate user. If there is no default user, unrecognised user identifiers will get error **User not known**; if there is, they will be logged on as the default. It is normally recommended that you do not allocate any accounts to the default user, so that someone logging on as an unrecognised user will not have owner access to anything.

f) If you want anonymous printing enabled then there must a user **ANONPRINT** or a default user. The corresponding logical printer should have the **anonymous printing allowed** option set to yes (see section 4.3 under EDITPRINT and section 6.3).

When you have completed the editing of the password file, return to the list of users, and press **S** to re-save the file. The program will display the name of the default user (if any) and a warning if there exists no user with system privilege or access to account 0, then will prompt **Save? (Y/N)** - check that the user list is correct, and that the default user and system privileged user(s) are correct before pressing **Y**.

*After using EDITPASS, type ***BYE** and switch off the BBC Microcomputer. The password file remains in memory after EDITPASS, and could be looked at by an unauthorised user otherwise.*

5.3.2 Keeping a Security Copy of the Password File

It is wise to keep a copy of the password file elsewhere, so that the main file can be restored quickly if an accident does happen. An easy way is to use the BASIC program COPIER (described fully in Section 3.3), to copy the file either to a floppy disc on a suitably equipped BBC Microcomputer, or to a file in the File Server. In the latter case, set the access status (see *ACCESS in Section 3.3) of the copy to **PWR/**, and give it an account number that *no-one else has access to* (e.g. Account 0).

The file itself is called %PASSWORDS, and is not in any particular directory. If a mishap does occur, use COPIER again, copying from the backup copy, back to %PASSWORDS.

After using COPIER for this purpose, type ***BYE**, and switch off your computer, since the file will be left in the computer's memory.

5.3.3 Password Files on more than one Disc

On a Modular Disc File Server, the system will search through the password file on every disc, starting with the hard disc drives (if fitted) and then the floppy drives, until a match is found with the user name. It is possible to place users on specific floppy discs (using the *SDISC command before or during running EDITPASS), so that they cannot log on unless that disc is in the File Server. In addition, it may be desirable to place the system privileged user(s) and/or all users with access to Account 0, on one particular floppy disc - which can be physically locked away when system privileged operations are not required.

When a user logs on, the system will search for his name in the password file on each disc, starting from the first disc alphabetically. When it finds this user identifier, it will search the same disc (and no others) for a user root directory. Hence a user's entry in the password file and his root directory should appear on the same disc, otherwise this automatic selection will not take place.

See Section 3.3 under *I AM command for full details of the action of logging on. Note in particular that it is not desirable for a particular user to appear in the password file of more than one disc, since the stored password and account information may be different on the various discs, and the version used by the system will be the one in the lowest numbered drive - this could change at random for the user depending on the drives in which floppy discs had been inserted, and information on the second hard disc would never be used.

If any disc in the system does not have a password file on it, the File Server *will give system privilege to anyone attempting to log on.* The disc formatter in Utility Mode will automatically create a null password file on every disc, so that this cannot happen accidentally.If the password file has to be deleted (this is done from Utility Mode - see Section 7.3.8), ensure that the system is left with at least a null password file %PASSWORDS on every disc (with, of course, a 'real' password file on one or more discs). The null file can be saved with the program EDITPASS - after selecting the appropriate disc. Log on as a system privileged user (if one exists in any password file at this stage) with the front panel key-switch in the SYST position, type **CHAIN "EDITPASS"**, then **N** for a new file, then **S** to save an empty file.

5.4 Finishing the process

Now that the password file has been defined it is necessary to create the directory structure and then set the account numbers to coincide with the *user root directories*.

5.4.1 Creating User Root Directory

You need to have access to account 0 to create directories in the root, as described in Section 5.3.1 above. In addition, if you are going to allocate accounts to users, you will need access to all other accounts. Check that you are in the disc root directory (that is the directory in which all the user root directories appear) on the desired disc, by typing:

```
*DIR $  or
*DIR $<disc name>
```

Then create a directory for each user, using the CDIR command:

```
*CDIR ALLEN
*CDIR BURTON
```

and so on. At the end, check that the directories exist by typing *CAT, which will give a list on the screen of all the directories in the root.

5.4.2 Setting the Account Numbers of User Root Directories

You have already allocated one or more accounts to each user whilst editing the password file. In order for the users to have owner access to their own user root directories, you need to give the corresponding account number to their root directory. The File Server will then automatically take care of accounting, by the rule that any file (or sub-directory) created will have the same account number as that of the directory that it is in. A user may change the account number of a file, but only if he has access to another account to change it to.

Before setting account numbers, ensure that each account that you are going to allocate has some credit. See Section 3.3 for information about *STATEMENT and Section 4.3 on*CREDIT.

The root directory $ on each disc always has account number 0. By the above rule, all the directories will initially have account number 0, and hence only users having access to account 0 will own them, allowing them to create files in them.

To change the account numbers of these directories, use the *ACCOUNT command. Suppose that you had allocated account 22 to user ALLEN, and account 23 to user BURTON. Then type:

```
*ACCOUNT ALLEN 22
*ACCOUNT BURTON 23
```

and so on. You need to have access to all accounts to be able to do these allocations - how to do this is explained in Section 5.3.1, particularly in paragraph (a).

You may get the error message **Account Bankrupt** whilst running the *ACCOUNT command. It will be necessary to credit the account using the *CREDIT command (see below).

If you have set a default user in EDITPASS (sections 5.3.1 & 4.3), you will probably want to set up a user root directory for the default, and put a !BOOT file in it (see Section 3.3 under *OPT4 for details). In this case, it is recommended that you leave this directory with account number 0, so that only you can change it.

Alternatively, you may wish to allocate some 'scratch-pad' filing space to anonymous users, in which case both the default user and the corresponding user root directory should be given an account (with the balance suitably set using *CREDIT) to permit this.

5.4.3 Distinction between Users' Names and User Root Directories

There is the possibility of confusion between a user called FRED (for example) and a user root directory (or URD) called FRED. The concept of a user exists *only in the password file*, and is the means of identification for the purpose of allocating accounts, privileges and other options.

At log-on, the system will search the disc on which the user was found in the password file for a URD, as a convenience to that user and others. (Note that this is different from Acorn systems, which search *all* discs for a URD). This root directory will have the same name as that selected by the URD option in the password file. If no particular selection has been made, the "normal" entry in the password gives a URD with the same name as the user: for example, user MARY is assumed to have URD MARY unless another URD name has been entered into the password file. Others can gain public access to MARY's files (assuming that she has allowed public access) with commands like

```
LOAD "$.MARY.PROGRAMS.EXAMPLE1"
```

Note however that there is actually no need to allocate a separate URD to every user. The allocation of accounts is the *only* thing that determines whether a user has owner access to a file.

For example, for a sixth form project with three people working on it, it may be useful to have a root directory called PROJECT. Suppose the three users had user names JOHN, KATHY and MARY, you might allocate them all account 98. Then set the account number of PROJECT to 98, and all three users have owner access to directory PROJECT, and can all create files in it.

At the same time, any or all those users can be given owner access to other files or directories, by allocating them other accounts; so that you could also allocate account 43 to user MARY, and then set up a directory called MARY, with account number 43 - only MARY would have owner access to this latter directory. The URD option in the password file is used to specify which of the directories that MARY has owner access to is her URD.

If there is no directory on the disc to match the URD specified in the password file for a particular user, say JOHN, then when JOHN logs on, the File Server will select the root directory $. Note that only the first disc on which the user was found in the password file will be searched. JOHN will need to select directory PROJECT explicitly befre he can begin work, by typing:

```
*DIR $.PROJECT
```

Alternatively, the system manager can set the user root directories of JOHN, MARY and KATHY to be $.PROJECT by pressing U at the expanded information stage of Editpass.

5.4.4 Crediting or Debiting an Account

When the system manager formats a new disc, the formatting program will initially set account 0 with credit equal to the disc size, and all the other accounts with zero credit. See Chapter 7 for a full description of the Utility Mode commands and how to use them. To find out what the credit balances are, log-on as a user with access to all accounts, and type:

```
*STATEMENT

Disc            0
Account         Balance
50              5K

Disc            1
Account         Balance
50              10100K
```

This will produce for each disc a list of all the accounts to which a user has access, with the associated credit balances. For the system manager it would be wise to send this information to a printer, so type:

```
*PS              this selects the network printer
<Ctrl-B>*STATEMENT  <Ctrl-B> means press B while holding down the CTRL key
<Ctrl-C>
```

The network printer will then produce a copy of the screen. To change the balance of any account, use the DEBIT and CREDIT commands. For example:

```
*CREDIT 43 500
```

will add another 500 kilobytes of space for use by files with account number 43. Similarly *DEBIT will subtract space from that allocated. These two commands require system privilege and the front panel key-switch must be in the SYST position.

These commands only change the outstanding credit balance of available space. Since the credit balance cannot exceed 65535 kilobytes, the command:

```
*DEBIT 44 65535
```

is guaranteed to wipe out any credit balance left to account 44.

When a file is created, the account corresponding to its account number is debited by an amount equal to the size of the file (but not its extent - see Section 3.3 under OPENOUT). If there is insufficient balance to allow saving of a file, the error message **Account bankrupt** will be produced, and the user must either delete something, or move some files to another account - assuming that be has access to another. No account can be debited below zero credit, so deleting files will always give some positive balance.

On a system which has more than one disc in use, the STATEMENT utility will print a list of the accounts to which the user has access, separately for each disc. The user can however only save files on discs which contain one or more directories with account number(s) to which he has access.

5.5 Password File Management System

5.5.1 Overview

Batch mode Editor Documentation:-

5.5.1.1 Memory Requirements
5.5.2 *CONVERT
5.5.3 *MERGE
5.5.3.1 `!makdir` and `!remdir`
5.5.4 *GENERATE
5.5.5 Keywords
5.5.6 Mod-file Examples
5.5.7 Errors
5.5.8 Formal File Definition
5.5.9 Known Problems

5.5.1 Overview

The password file management system software consists of the following programs, all of which are found in the directory `$.SYSPROGS` of the release disc:-

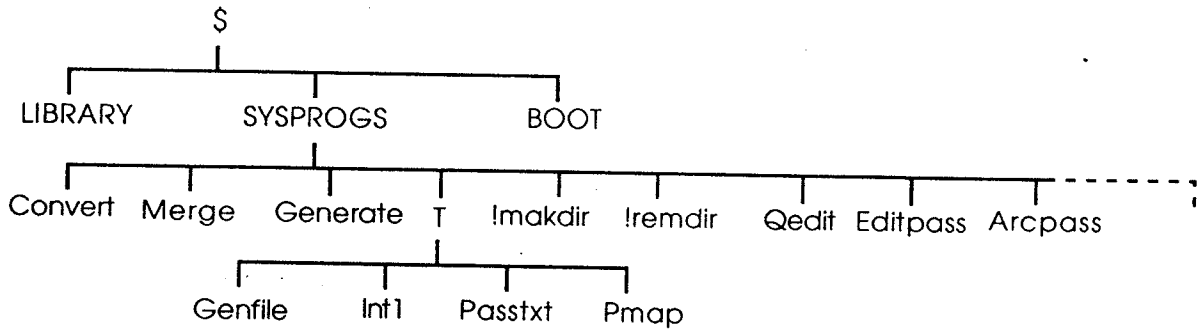
a) The batch mode editing suite:

CONVERT (Machine code program)
MERGE (Machine code program)
GENERATE (Machine code program)

b) The interactive editors:

QEDIT (BASIC program)
EDITPASS (BASIC program)
ARCPASS (Archimedes BASIC program)

The directory stucture is shown thus:-



The existing EDITPASS program restricts the size of the password file to the size of the memory in the local computer, and this typically allows around 200 users. There is now a version of version of this program (called ARCPASS) for an Archimedes allowing about 7000 users. The batch mode editor and QEDIT are a means of editing large password files on standard BBC microcomputers.

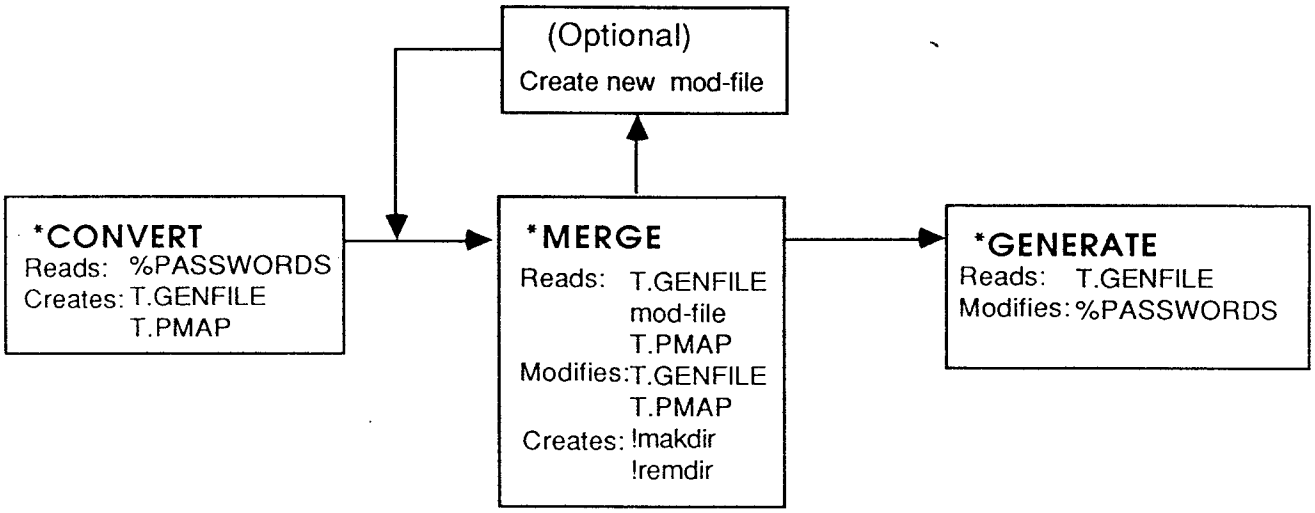
QEDIT is a version of EDITPASS which allows the password file to be edited on a user-by-user basis. The password file is not held in the local computer; individual user entries are modified and then written back to the password file directly, so the restriction on file size is removed. However, QEDIT does not allow you to insert or remove users, or to change the URD or LIB strings.

With the batch mode editor, the system manager prepares a text file (the *mod-file*) containing instructions for modifying the password file. The commands available can be very powerful; for instance, the system can automatically allocate a spare account number, create the appropriate user directory, set its account number

and credit that account. The same process can be repeated automatically, so with little more than a list of names, an entire class can be entered onto the system in a matter of minutes.

The batch mode editor uses a three-stage process: *CONVERT converts the (machine-readable) password file %PASSWORDS into a (human-readable) text file. *MERGE combines this with the mod-file to produce another text file. *GENERATE converts this text file back into a machine-readable format file, which it then installs on the appropriate disc.

The process is shown in the diagram below:-



Important

Since all of the passwords are stored in the text files, it is very important that only the system manager has access to them, and they should be treated with as much respect as %PASSWORDS itself. Each of the programs protects the machine from remote network operations to stop unauthorised people being able to read the files, but security is only as good as the system manager makes it. The T directory should be set to Private (`*ACCESS T +P`). **Only `%PASSWORDS` is protected by the key: the other files are only protected by the main file access controls.**

Off-line / Off-site Operation

An advantage of the batch mode editor is that it can be run off-site using a local disc filing system (DFS or ADFS), thus reducing the risks of security breaches. *CONVERT is run (on the network), `T.GENFILE` and `T.PMAP` are copied onto local disc: the network copies should then be deleted. All the edits (i.e. preparing mod-files and running *MERGE) can then be done whilst the computer is disconnected from the network. `T.GENFILE`, `!makdir` and `!remdir` are copied back onto the fileserver, and *GENERATE is run. `T.GENFILE` should then be deleted from the fileserver.

General Suggestions

If the password file is fairly small then EDITPASS can be used. If an Archimedes is available then ARCPASS can be used (on virtually all sizes of password file). If the file is too big for EDITPASS then QEDIT can be used, subject to the limitations of QEDIT itself.

If a large number of users are being added or modified, then, whether the password file is large or small, we recommend that you use the batch mode editor. For extra security the batch mode system should be used off-site.

Temporary files created by the batch mode editor

There are a number of temporary files used by the batch mode editor which are all held in the directory T. T.GENFILE and T.INT1 should be deleted (for security reasons) after a session has finished. The files are:-

```
T.GENFILE    T.PMAP    T.INT1    T.PASSTXT
```

T.INT1 and T.PASSTXT are temporary files created and used only by *MERGE. The latter is the updated version of T.GENFILE and is normally *RENAMED as this before MERGE exits. However, if MERGE fails it is possible that both T.GENFILE and T.PASSTXT will remain. Thus T.PASSTXT may be deleted at any time (except while MERGE is actually running).

There also are two files created by *MERGE that will require be to *EXECed by the user, which are:-

```
!makdir      !remdir
```

5.5.1.1 Memory Requirements

*MERGE requires HIMEM at &7C00 or greater. On a BBC microcomputer without shadow RAM, MODE 7 is required (and will automatically be selected is this is not already the case). If HIMEM is less than &7C00 and you have shadow RAM, MODE 131 will automatically be selected.

N.B. If HIMEM is less than &7C00 and you load *MERGE, you will see the program being loaded into the screen. Normally, this does not metter because the first thing that the program does is to change to a different mode. However, if you in addition have *OPT 1 1 set, there is a fair chance that the text printed by the OS will actually overwrite the loaded program, which will then crash.

You are therefore warned against using *OPT 1,1.

If you are using a RISC OS computer then the *CONVERT, *MERGE,*GENERATE suite of programs can be run using the 65Tube module.

5.5.2 *CONVERT: Converting the existing password file

Syntax: `*CONVERT [<Disc name>]`
System Privilege Required.

Using the program CONVERT, a (human-readable) text file `T.GENFILE` is created from the current password file, `%PASSWORDS`. The discname is recorded in this file. In addition the program will create the file `T.PMAP` which contains a *bit-map* of all the personal account numbers currently allocated in the password file.

Note that users are allowed to modify some aspects of the password file themselves (by using *PASS or *OPT 4,n), so you should not use the old copy of T.GENFILE but create an up-to-date copy (you *cannot* use the 'last update date/time' to see whether the file has been directly modified in this way). However, if you wish to use a sequence of *MERGE operations you must only run CONVERT once (if you run CONVERT immediately after MERGE, the file T.GENFILE will be overwritten and any modifications will be lost).

Having typed `*CONVERT` the software will respond with :-

```
*Convert
Version 1.12, (C) SJ Research
Converts %PASSWORDS into text file T.GenFile
 also makes T.pmap
```

For every ten users processed a dot will be printed.

If you want to look at the resulting file, type *TYPE T.GENFILE. To convert a password file other than the one on your currently selected disc specify the disc name after the *CONVERT command.

e.g. *CONVERT MAIN

Corrupt %PASSWORDS files

CONVERT will give a warning when it finds corrupt URD/LIB pointers (i.e. those that point off the end of the file; pointers that point to other random places in the file could produce warnings of the URD/LIB text exceeding 80 chars).

5.5.3 *MERGE and the mod-file

Syntax: *MERGE [<mod-file name>]
System Privilege Not Required.

<mod-file name>, if not specified, defaults to MODF.

Changes to the password file are made by creating a *mod-file*. This file should contain a *mode* keyword, telling MERGE whether to add new users, modify existing users or remove existing users. There is then a section defining attributes that should apply to the new users: these are called *global* assignments. When removing users, this section is obviously not required. Then follows the list of usernames on which we wish to act. Each username is followed by a section (enclosed in curly brackets { }) that defines actions to be done to that user only (these are called *local* assignments).

Creating a Mod-file

A standard ASCII text editor is required. We suggest using Acorn EDIT (supplied with a BBC Master microcomputer), or WORDWISE on a BBC micro. VIEW can be used, but the format and justify options should be turned off and you should do not create any new rulers nor enter any formatting commands (shift-f8). EDWORD files are not suitable, but spooled output from this editor is acceptable. For very short mod-files it would be possible to use *BUILD.

The file may have any name as MERGE takes the filename as a parameter. Typically you might have a mod-file for each class held permanently on the fileserver, so that you can make changes to an entire class (e.g. remove them when they leave) very easily.

There may only be one occurence of any given username in the mod-file.

Global keywords are specified outside a user definition and take effect for all the following users up to the next mode keyword. There may be many global assignments following each mode keyword; the assignments to a particular keyword are not cumulative (e.g. ACC="1"; followed later by ACC="2" is not the same as putting ACC="1 2";). All global keywords are reset by each mode keyword.

Local keywords are specified after a username within curly brackets { } and only affect that username. Local keywords take precedence over global keywords and again are not cumulative (e.g. global Flag="Pw"; followed by local Flag="Ns"; does not give Flag="PwNs").

Comments can be specified by inserting an "&" as the first character of a line; the rest of the line up to a CR (CHR$13) or a LF (CHR$10) is then ignored.

There is one important restriction on the size of the Mod-file, that is that it cannot contain more than 256 users. However, this should not present a problem as MERGE can be run as many times as necessary on different mod-files, wthout having to re-run CONVERT or GENERATE.

The general form of a mod-file is shown below :-

.mode.

Global assignments
Username { local assignments }
Username { local assignments }
Global assignments
Username
.
.
.mode.
Global assignments
Username { local assignments }
.END.

Running *MERGE

```
*MERGE
*Merge [<mod-file>]
Version 1.16, (C) SJ Research
Parses mod-file and produces T.Int1,
 then Merges T.Int1 with T.Genfile
Also produces !makdir & !remdir

Parsing Mod-file....
Warning - !makdir already exists.
Warning - !remdir already exists.
(O)verwrite, (A)ppend or (Quit)
O/A/Q ?O
Merging T.INT1 & T.GenFile.
****** Error : at line 1
TONY {}
------
SYST - User not found
...
Errors during Merge - aborted
```

Errors

During the *parse* stage, the line number of the line in error and a relevant portion of the mod-file will be printed. During the merge stage, the line number and relevant portion of `T.GENFILE` will be printed. Lines are numbered starting from 1.

The following keywords may be defined either globally or locally :-
`ACC, BASE, BOOT, CREDIT, FLAG, LIB, PACC, PASS, URD.`

The keyword `DEFAULT` may only be defined locally.

In addition there are the following modes.
`.ADD. , .REMOVE. , .MODIFY. , .END.`

Please note that changing mode sets all global assignments to their defaults.

5.5.3.1 !makdir and !remdir

N.B. To use either of these files, the user requires system privilege, key in the SYSTEM position, and ownership of all accounts.

MERGE always creates, in the currently seleted directory, the files `!makdir` and `!remdir`. These files contain a sequence of commands which will make/remove directory structures and also to credit/debit accounts, for any users added or removed during the merge process. In .ADD. mode, commands to create

the relevant URD directory and to credit the relevant account are added to !makdir, in .REMOVE. mode commands to delete the users' URD and all its contents and to debit the account are added to !remdir. In .MODIFY. mode, no commands are added to either file. To use the files, type

`*EXEC !makdir` or `*EXEC !remdir`.

N.B. Once either file has been *EXEC'ed, it should be deleted by using **`*DELETE !makdir`** or **`*DELETE !remdir`**.

MERGE can also append new information to the end of an existing file, so that a sequence of MERGEs can be done, followed by a single *EXEC command.

The following !makdir file was created as a result of the mod-file in §5.5.6. The disc name is `Work`. Bold type indicates user input.

```
*EXEC !makdir
*DIR :Work
*CDIR CLASS87
*DIR CLASS87
*CDIR FRED
*DEBIT 145 65535
*CREDIT 145 100
*ACCOUNT FRED 145
*DIR :Work
*CDIR TONY
*DEBIT 10B 65535
*CREDIT 10B 50
*ACCOUNT TONY 10B
*DELETE !makdir
```

The *CDIR commands are inserted unless the URD keyword was defined to be the root, i.e. "$" or ":<discname>. If URD=""; the directory $.<username> will be created. The *DEBIT, *CREDIT and *ACCOUNT commands are included whenever the user has a Personal account number (i.e. PACC<>"").

An example of the contents of a `!remdir` file:-

```
*BASIC
LOAD"ERAQ"
RUN
:Work.CLASS87.FRED
N
*DEBIT 145 65535
```

For !remdir, URD and PACC definitions are taken from `T.GENFILE`. The "ERAQ" sequence of commands (lines 3 to 5) is present unless the URD is defined to be the root of any disc, the *DEBIT is inserted when the user has a PACC. If some of the commands in !remdir are undesirable, they may be removed/modified with a text editor.

5.5.4 *GENERATE

Syntax: `*GENERATE`
System Privilege Required.

The key need only be in the SYST position during the actual installation of `%PASSWORDS`, i.e. only for the very last phase of the program. Ownership of the root on the relevant disc is also required (usually account 0).

The GENERATE program creates a new fileserver-format password file `PASSWDS` from `T.GENFILE`.

GENERATE then installs this as %PASSWORDS and the user should then delete T.GENFILE and turn the key back into the SECURE position.

The file PASSWDS is always created in the root of the disc onto which the new password file needs to be written. This is because the file PASSWDS is transferred to %PASSWORDS using a *RENAME command. This has the advantage of being an 'atomic' operation, i.e. no operations from other users are allowed while it is happening (especially logging-on). It also automatically deletes the original file. [This is a special case whereby *RENAME is able to rename a file 'on top of' another already existing file; this does not work in reverse, and you cannot *RENAME the password-file back out again.]

Note: Whenever the message Output now in file $<discname>.PASSWDS occurs, this file will exist (with access "PWR/") and contain sensitive information until it is successfully installed as %PASSWORDS. Therefore, if the file is not installed, you should delete it. The file will also remain if you abort the installation.

GENERATE will only ask you whether you wish to install the new file when no serious errors were deteced, and that the contents of the file are only guaranteed to be valid when there are no errors/warnings given. When GENERATE prompts the user, pressing any key other than 'Y' or 'y' will abort. Note that aborting in this way will leave the file in the root.

Running *GENERATE:-

```
*GENERATE
Password file editing system - GENERATE
Reads file T.GENFILE, writes file $.PASSWDS
Version 1.08

.END. found


Number of users =&0022


Finished :
00 warning(s)
00 serious error(s)

Output now in file :SMALL1.PASSWDS

Install new file as %PASSWORDS ? (Y/N)Y

Installed.
```

5.5.5 Keywords

All keyword assignments have the form:

<Keyword>="<value>";

Note the double-quotes and semicolon, which must always be present.

ACC — Defines the group account numbers given to a user (see also PACC). Each account number is a number (in hexadecimal i.e. 0 to 9 then A,B,C,D,E,F) in the range 0 to 7FF. Multiple account numbers are separated by a space. To specify a range of accounts the first account number is the lower range followed by the '>' character followed by the upper range of the account number.

Group account numbers in the range 100..7FF are actually allocated in blocks of 40(hex). That is, if any account from 100 to 13F is specified, all accounts 100..13F will be owned by the user.

In .MODIFY. mode, the characters '+' and '-' may be used in the definition. All account numbers following such a character add or remove accounts as appropriate. A single definition may include both characters, and they will be evaluated on a left-to-right basis. E.g. ACC="0>7F -10>6F + 30 32" would give be equivalent to ACC="0>0F 30 32 70>7F". A whole account block can be removed by specifying a single account; for instance, ACC="-100" would remove accounts 100..13F.

When used in .ADD. mode, '+' or '-' will cause the error `Bad character`. In .REMOVE. mode, whilst local assignments are parsed, no error will be caused.

For example
```
ACC="10 2C 30>FF 140";
```
will assign account numbers 10, 2C, 30 through FF as well as group accounts 140..17F.

ACC defaults to "" i.e. no accounts.

BASE This defines the base user root directory name to which the username is added. It will generally be defined in a Global assignment, although it can be defined in a local assignment. No "$." prefix is required - see the URD keyword.

For example:
```
.ADD.
BASE="CLASS87";
TONY {}
```
will give TONY the user root directory "$.CLASS87.TONY"

BASE will override a URD assignment if it is defined at a later stage. Default for BASE is "$" (and overrides until a URD is defined).

BOOT This defines the boot option for a user. The number ranges from 0 to 3 and has the following meaning on the BBC computer :-

```
0 - No action
1 - *LOAD !boot
2 - *RUN  !boot
3 - *EXEC !boot
```

The default value of BOOT is 0.

CREDIT In .ADD. mode (and no other), this will set the amount of space, in K, that is CREDITed to the personal account number. The password file itself will not be affected by the value of this keyword, only the !makdir file is affected.

The number is in decimal and in the range 0-65535. The default value of CREDIT is 100.

DEFAULT This takes either 0 or 1 as a parameter. "1" sets this user to be the default user, "0" means that the user should no longer be so. MERGE never produces any warnings about 'silly' uses of DEFAULT (e.g. using DEFAULT="0"; on a user who wasn't the default user anyway).

May only be defined as a local keyword, and defaults to 0.

DISCNAME This keyword is always present in `T.GENFILE` and never anywhere else. It gives the name of the disc from which the password file came, and is put there by *CONVERT.

FLAG For this keyword the assignment of data is in the form of two letter combinations which are

as follows.

Pw	password locked
Sy	system privilege
Ns	No short saves
En	Permanent *ENABLE
Nl	Library only used for *RUN commands
Ro	'*RUN only' user

A user with this option enabled may use *RUN and certain other *commands. Also permitted are FScall #14 (Read disc info), FScall #16 (Read date/time), FScall #25 (Read FS version number), and FScall #65 (Read/Write misc info). All other commands will give the error message `Who are you?`

Al	Auxiliary account locked

When this is set, the user is not allowed to change the auxiliary account of any file or directory under any circumstances.

X2	Reserved

See Editpass for more information (section 4.3, page 4-8)

In addition to this, in .MODIFY. mode either '+' or '-' may precede any flags to either add or remove options.

For example:

```
.MODIFY.
TONY {FLAG="+SyEn-Pw"; }
```

will take the existing flag options set for TONY and add the system privilege and permanent *ENABLE and remove the option for password locked. If neither '+' or '-' are used in .MODIFY. mode then the new assignment will override the old definition for the flag.

Default is Flag=""; i.e. Password/*OPT4 not locked, Not System Privileged, Short Saves allowed, *ENABLE required (for wild-card *DELETE), Library used for all operations, Not 'Run Only', Auxiliary Account not locked.

LIB

This sets the initial library directory for the user. Default is "", which means that the fileserver will select $.LIBRARY on the lowest numbered disc (a hard disc drive, if you have one).

PACC

This keyword defines a personal account number and is a hexadecimal number in the range of 1 to 7FF. If set to "" it means that no personal account number is allocated. When used in a local assignment that particularly personal account number is given to the user. If the personal account number has already been allocated to another user (as a PACC) then a warning will be given.

In .ADD. mode, when PACC is used in a global assignment it assigns the next free account number greater than or equal to the one specified. That account is then marked as allocated so that the next user will get a different account number. The file T.Pmap contains a map of the currently allocated PACCs, and this file is read and updated when using this feature. To disable this feature, set PACC="" in another global assignment; in a local assignment you would PACC="344" to assign a specific account number.

For example:

```
.ADD.
FRED { PACC="500"; }
STU {}
```

```
TONY {}
.END.
```

will allocate personal account 500 to FRED, personal account 100 to STU (assuming it has not already been allocated to some other user) and personal account 101 to TONY. If personal accounts 100, 101 and 103 have been allocated to some other users then STU would be assigned personal account 102 and TONY would be allocated personal account 104.

Default is "100", i.e. allocation will start from 100 (.ADD. mode only).

PASS This sets a user's password, the default password is "".

URD This sets the user's root directory, and overrides any BASE definition. The fileserver selects the URD relative to the root of the disc on which %PASSWORDS is. Therefore you do not need to prefix it by "$." unless the directory required is on a different disc (when you should use :<discname>). By default, BASE is set to "$." and the URD is undefined (that is, it is not referenced). If URD is set to "", the URD becomes the default, i.e. $.<username>. To set the URD to null, use URD="$".

Mode Keywords

.ADD. In this mode the user entries are taken as new users. If a user of this name already exists an error is generated. A set of commands are placed in the file "!makdir" to create the appropriate URD and credit the appropriate personal account (the system manager will *EXEC !makdir at a later stage). If MERGE is used repeatedly, new commands will be appended to the existing !makdir file, and a warning will be given. Therefore, once !makdir has been *EXECed it should be deleted.

If a user has PACC set to "" then the account number of the URD created will be the account number of the parent directory (i.e. no *ACCOUNT command will be placed in the !makdir file). In this case it is possible that the user will not have owner access to his URD.

.REMOVE. The specified users are removed from the password file. Obviously no global assignments or local assignments are needed, however it is not an error for these to exist. This makes it possible to remove blocks of users and later restore them just by changing the mode keyword to .REMOVE. The curly brackets {} must be present after each user name, although there needn't be any text within them. However, the text inside curly brackets is parsed, so don't put garbage in there!

A set of instructions is placed in the file "!remdir" to delete the appropriate URD and its contents and also to debit all the space allocated to that account.

.MODIFY. The data in the user entries is used to modify the data already held in an existing entry. It is an error for the user not to already exist. To add or remove accounts or flags from a user entry the characters '+' or '-' may be used.

.END. This signifies the end of data in both the mod-file and the gen-file. The use of .END. is optional, but *CONVERT always puts a .END. at the the end of the gen-file.

5.5.6 Mod-file Examples

Consider the following mod-file :-

```
.ADD.
```

```
ACC="1 23"; BASE="CLASS87";
FRED {PACC="145";}
TONY { URD=""; CREDIT="50";}
.END.
```

The mode keyword .ADD. specifies that the users are to be added to the password file.

The next line is a global assignment: the keyword ACC is assigned the values 1 and 23 and the keyword BASE is assigned the name CLASS87; as these appear outside a username definition they are global assignments.

The username FRED has a local assignment defining his Personal Account Number as 145, so he will have access to Accounts 1 & 23 (from the global assignment) and Account 145, and his default directory after logon will be `$.class87`. TONY has the keyword URD defined locally which overrides the global BASE assignment.

User TONY will have a URD of $.TONY, and will have a personal account number allocated (the lowest free one above 100). He will also have access to accounts 2 and 23, but will only have 50k of disc space allocated to his personal account.

The following mod-file has exactly the same effect as the previous example.

```
.ADD.
TONY { URD=""; ACC="1 23"; }
FRED { ACC="1 23"; URD="CLASS87.FRED"; PACC="145"; }
.END.
```

A typical application of the batch mode editor would be to add a new year's entry to the system.

Suppose we have a mod-file called `CLASS4A` thus:-

```
.ADD.

PACC="200";
BASE="Class4a";
CREDIT="50";
FLAG="PwNsAl";

ArdleighW {}
BassetF {}
MunroeM {}
BunterW {PACC="";}
WilliamJ {}
BottVE {}
KermitF {}
BigglesDSO {}

.END.
```

By typing ***CONVERT , *MERGE CLASS4A , *GENERATE** you will now have an updated password file installed. To create the necessary directories, type ***EXEC !makdir**. That's all there is to it.

You would normally keep the file `CLASS4A` around; in order to delete the users, change the .ADD. keyword to .REMOVE. and repeat the process, only this time finish off with ***EXEC !remdir.**

5.5.7 Warnings and Errors

Warnings and errors are accompanied by a portion of the relevant file and two inverse exclamation marks

(*CONVERT/*MERGE) or a left-pointing arrow (*GENERATE) to indicate the approximate location of the error. N.B. in mode 7, these symbols appear as white squares (character 255).

***CONVERT**

URD/LIB pointer corrupt for user - <username>
EOF (no terminating user entry, or password file corrupt)
Bad disc name
%PASSWORDS not found
Directory called T not found
--*-* System Error : <OS error message>

***MERGE - Parse stage**

>256 users in mod-file
.END. is a global keyword
Bad character
Can't find mod-file
Can't find T.pmap
Default is only valid as a local keyword
End of file inside quoted string (or missing ")
Expecting a "
Expecting a }
Expecting a number
Flag not known
Keyword/userid too long
Mode keywords must be specified globally
Mode not specified, using .modify. by default
Need an = to assign value
No more personal account numbers!
Number too large
Parameter too long
Personal account number already allocated
T.pmap has not been generated by *CONVERT
Text found after .END.
Unknown keyword
User already exists (two users of same name in the mod-file)

***MERGE - Merge stage**

Bad keyword in Genfile
Flag not recognised
No users found in Mod-file!
Second number in range smaller than first
T.Genfile not found
<Username> - User already exists in password file (in .add. mode)
<Username> - User not found (in .modify. or .remove. modes)
Warning - !makdir already exists
Warning - !remdir already exists

*GENERATE

Fatal errors:

File not found - T.GENFILE

Errors:

Discname too long
Mismatched {} brackets
Larger number of users in pass 2 - output file useless
Two users with same name, or not in alphabetical order
Userid is missing/zero length
--*-* System error : <OS error text>

Warnings:

Bad number
Bad operator - expecting "=" or "{"
Bad range (second number after range indicator wasn't a number)
URD text exceeds 80 characters
BASE text exceeds 80 characters
Boot option >3
Constructed URD exceeds 80 chars
DEFAULT cannot be used as a global keyword
DISCNAME must be global keyword
Keyword/userid too long
Missing " in assignment to keyword
More than one default user - this one ignored
Odd number of characters in FLAG text
Password exceeds 10 characters
Significant text after .END. - ignored
Smaller number of users in pass 2
Start of range bigger than end
Unrecognised flag name
Unrecognised keyword

5.5.8 File Specification

All characters ASCII 0 through ASCII 31 are considered as a SPACE. Top bits are stripped. There is no case sensitivity, as every alpha-numeric is taken as upper case.

<file> ::= <gen-file> | <mod-file>

<gen-file> ::= DISCNAME="<discname>"; <pw data> .END.

<mod-file> ::= .<mode>. <pw data> [mod-file] [.End.]

<pw data> ::= [<userdata> | <global assignment> | <comment>] [<pw data>]

<mode> ::= Add | Modify | Remove

<comment> ::= & <text> <line terminator>

<line terminator> ::= <CHR$13> | <CHR$10>

<global assignment> ::= <global keyword> = "<keyword value>";

<userdata> ::= <UserID> { [<local assigment>] } [<userdata>]

<local assignment> ::= <local keyword> = "<keyword value>"; [<local assignment>]

<UserID> ::= [<alphanum>]

<keyword> ::= ACC | BASE | BOOT | CREDIT | FLAG | LIB | PACC | PASS | URD

<global keyword> ::= <keyword> | DISCNAME

<local keyword> ::= <keyword> | DEFAULT

<keyword value> ::= <acc> | <lib> | <pass> | <boot> | <urd> | <flag> | <pacc> | <default> | <base>
| ""

<acc> ::= <hex> | <hex> > <hex> | -<acc> | +<acc>

<lib> ::= <path>

<pass> ::= <alphanum>

<boot> ::= 0 | 1 | 2 | 3

<urd> ::= <path>

<pacc> ::= <hex>

<default> ::= 0 | 1

<base> ::= <path>

<flag> ::= [<flagsymbol> | +<flagsymbol> | -<flagsymbol>]

<flagsymbol> ::= Pw | Sy | Ns | En | Nl | Ro | Al | X2

<path> ::= <name>[.<path>] | :<discname>[.<path>]

<discname> ::= <alphanum>

<name> ::= <alphanum>

<hex> ::= <hexit> | <hexit><hexit> | <hexit><hexit><hexit>

<hexit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F

5.5.9 Known Problems

General:-

Whilst all programs set protection against remote network operations (*VIEW etc) during operation, the current versions of CONVERT/MERGE do not clear RAM before exiting (GENERATE does), although they do leave the computer protected after exiting. Therefore, you should always logoff and then switch the computer off (the order is important) after you have finished using these programs.

***CONVERT, version 1.12:-**

A corrupt %PASSWORDS file (with no terminating user entry) will give an EOF error. The resulting T.GENFILE will contain useful information, but T.PMAP will not have been saved. You should *TYPE T.GENFILE to find out whether most of the users have been included; there may be some corrupt users at the end of the file - these should not matter. Then, using *GENERATE (which does not require T.PMAP) you can create a repaired password file, which can then be re-CONVERTed correctly.

Some fatal errors (e.g. Account Bankrupt, Disc full, and Network errors) cause CONVERT to abort without closing files.

If the DEFAULT USER pointer is corrupt, CONVERT will not produce a warning; no user will have DEFAULT="1" in the gen-file.

***MERGE, version 1.16:-**

*MERGE does not give a warning when more than one user has been assigned DEFAULT="1". GENERATE will however give a warning, and will ignore subsequent assignments. In the case where DEFAULT="0" is specified when the particular user was not already the default user, again no warning will be given.

High-numbered group accounts do not act as blocks e.g. if you have ACC="100>17F" in T.GENFILE, the do ACC="-140", the result will be ACC="100>13F 141>200". GENERATE however does treat them in blocks, so will give access to A/cs 100>200 as before. Actually this is not normally a problem, since CONVERT does not produce ranges for high numbered accounts (ranges are only produced by MERGE), it merely specifies the base account number. From a password file it would have given ACC="100 140" whereupon removing account 140 using MERGE would have had the desired effect.

***GENERATE, version 1.08:-**

Does not give a warning if there are no system privileged users with access to account 0.