Malcolm Bull
Training and Consultancy Publications

MB-Guide to

File-save and file-restore

Malcolm Bull

MB-Guide
to
File-save and file-restore

MB-Guide

to

File-save and file-restore

by

Malcolm Bull

MALCOLM BULL Training and Consultancy Services
1991

First published 1991

ISBN: 1 873283 01 6

Introduction

The *MB-Guide to file-save and file-restore* is produced for those who need a quick introduction to the features of the spooler on the Pick operating system.

This *MB-Guide* contains:

* A general introduction to the concept of file-save, account-save and archiving.

* A description of the various file-save and file-restore procedures and of the account-save and account-restore procedures.

* A description of the selective restore procedure.

* A description of the use of T-DUMP and T-LOAD for archiving smaller amounts of data.

* Suggestions about the points which might be considered when devising your own archiving system.

The material will be of interest to operations personnel and to those who work in a systems support function.

You may find the following titles in the *MB-Guide* beginner's guide series useful in conjunction with the present volume:

Using backing storage
Creating and using Procs
File design
File-save and file-restore
Files: monitoring and sizing
Group format errors
Operations and systems management
The spooler

and you may also find the following *MB-Master* self-tuition courses of interest in conjunction with the material presented in this *MB-Guide*:

PICK1: Starting Pick
PICK2: Pick systems management

A suite of supporting software called MB-FILE-SAVE is available for use in conjunction with this *MB-Guide*. This is a front-end processor allowing you to perform the fundamental save and restore operations: perform file-save / file-restore; perform account-save / account-restore; perform selective restores. There is also a facility for maintaining account-save cycles of diskettes. All *MB-Software* includes a TCL stacker utility, and many individual routines are available directly from TCL. The *MB-Software* is particularly valuable to users of native PC Pick which does not provide such utilities. Please write or call for a leaflet or to place an order.

This *MB-Guide* is not intended to present a complete

description of the subject but merely to place it in context and give the reader enough information to use the facilities and to survive.

Best use can be made of this *MB-Guide* if it is read in conjunction with the reference literature which is provided for your system. You should amend your copy of this guide so that it accurately reflects the situation and the commands which are used on the implementation which you are using. By doing this, your *MB-Guide* will become a working document that you can use in your daily work.

I hope that you enjoy reading and using this *MB-Guide* and the others in the series, and welcome your comments.

1 Why do we perform file-saves?

The data held on any computer system is under constant threat:

* it may be changed accidentally: someone may delete account number 12345 when they meant to deleted 12354.

* it may be damaged intentionally: someone inside or outside the organisation may have a grudge and delete parts of the database maliciously.

* there may be mechanical problems: a disk drive may develop a fault which destroys the data or prevents data being read from it.

* the data may be removed intentionally: someone may need to recall the payment record for a customer who has long since been deleted from the system.

Since the data which the computer holds is one of the organisation's greatest resources and most valuable assets, it is essential that - should any such threats or other losses manifest themselves - the organisation can reconstitute the data to the correct state.

For these reasons most computer installations have some means of supporting their users against such losses:

* there will be a great number of checks built into the system to prevent unauthorised access to the data and to seek confirmation and authorisation for any major changes such as record deletion and/or file deletion.

* a copy of the entire database is made at the end of each day's business. This is known as a **file-save** in the Pick environment. If there is a great deal of activity on the database, such as there might be with a banking system which handles automatic tills, it may be necessary to perform a file-save more frequently than this.

* some form of transaction logging may be implemented which records all the changes made to a file. Changes may be recorded quite simply by making a note of the changed records after each amendment.

* the copies of the database will be retained for some considerable time to enable users to go back several days (or even further) to recover old records.

In this present *MB-Guide*, we are concerned with the following procedures:

* archiving data: the periodic dumping to off-line storage - such as magnetic tape or floppy-diskette - any data which is not required to be held on-line. The data may then be removed from the live system. Such

archiving regularly takes place at the month-end or at the year-end when last year's accounts are removed from the system to make way for this year's business. This can be achieved by means of the T-DUMP facility for small amounts of data or a file-save or account-save for larger quantities.

* the regular file-save and account-save procedure to serve as a backup in case data is lost.

* the recovery of data: this may be the recovery of single records or files from a T-DUMP or from a file-save or account-save dump, or it may be the reorganisation of the entire system from a file-save dump.

1.1 Control menus

The archiving and backup of the database is an important aspect of any System Manager's work. As we shall see, there are a number of standard utilities available to carry out the work:

```
:ABS/FILES
:FILELOAD
:FILES
ACCOUNT-RESTORE
ACCOUNT-SAVE
FILE-SAVE
SEL-RESTORE
```

amongst others. Rather than have the System Manager remember all these commands and type them in correctly, most implementations offer some sort of front-end processor, such as a control menu, which allows the various tasks to be chosen and performed by selection from a series of menus options. This is much simpler for the user, and the supporting software can then guide and prompt the user through the work more easily than would be the case if the System Manager had to remember the format of each command individually and exactly.

For various reasons, these control menus differ according to the implementation which you are using. In this *MB-Guide*, we shall consider the general features of file-saves and the associated operations and we shall look at the standard (and fundamental) facilities which are available on all implementations.

You may find it useful to read this present *MB-Guide* in conjunction with the reference literature for your system and add and/or amend any features which are used on the implementation which you will be using.

1.2 Finding a TCL command

If you are using TCL commands and cannot remember what verbs are available or if you have forgotten the correct form of a particular verb, you can always use Access commands such as:

LIST ONLY MD = "[SAVE]"
to find out what entries - such as verbs - there are on the MD which contain the letters SAVE

LIST ONLY MD = "[ABS]"
to find out what entries there are which contain the letters ABS.

2 Saving / restoring

There are several ways in which data can be written to backing storage and recovered from backing storage. The diagram below summarises these. In the following sections, we shall look at each facility more closely.

	To save	To recover
The system	File-save Binary-save	File-restore from file-save Binary-restore from binary-save
An account	Account-save	Account-restore from account-save or file-save
A file	T-DUMP file	T-LOAD file from T-DUMP Selective-restore from account-save or file-save
An item	T-DUMP file 'item'	T-LOAD file from T-DUMP Selective-restore from account-save or file-save

3 T-DUMP / T-LOAD

In a later section, we shall look at the use of the file-save and account-save procedures to dump a copy of the system to disk/tape, and the file-restore, account-restore and selective restore procedures which are used to recover the entire system, a single account, a file or a part of a file from such a disk/tape.

The T-DUMP, S-DUMP and T-LOAD verbs offer a simple means of archiving data:

* T-DUMP copying data from a file and saving it on backing storage, magnetic tape or floppy disk;

```
T-DUMP STOCK
```

* S-DUMP verb performing the same operation as T-DUMP but allows you to include sort specifications to dump the data in some specific sequence;

```
S-DUMP INVOICES BY ACCOUNT.NUMBER
```

  McDonnell Douglas Reality implementations use ST-DUMP instead of S-DUMP.

* T-LOAD transferring data from backing storage back on to a file.

```
T-LOAD STOCK
```

3.1 T-DUMP

A single file, or even a single item on a file, can be dumped by means of the T-DUMP (or S-DUMP) verb. Since these are Access verbs, they may include any selection criteria and any explicit item-ids must be enclosed in apostrophes:

If required, the HEADING modifier is used to specify a disk/tape label which is to be written to the volume.

A number of options are available with this command:

H to suppress the production of a disk/tape label.

I to suppress this display of the item-ids as the items are being dumped.

Some examples of its use:

```
T-DUMP STOCK
T-DUMP DICT INVOICE
T-DUMP STOCK '1000' '2000' '3000'
T-DUMP STOCK WITH COLOUR "RED"
T-DUMP INVOICE WITH AGE > "400" HEADING "ARCHIVE INVOICE"
S-DUMP STOCK BY COLOUR
T-DUMP STOCK (H
```

The following sequence dumps data to backing storage and then re-reads one of the records to show the form in which it is dumped. Before the T-DUMP command is issued, the correct disk/tape must be mounted and on-line with WRITE enabled.

```
>T-DUMP STOCK WITH COLOUR "YELLOW" HEADING "YELLOW ONLY"
BLOCK SIZE 500

1 5500
2 2000
3 9000
4 7777
4 ITEMS DUMPED.
```

```
>T-REW
BLOCK SIZE 500
[94] END OF FILE

>T-READ (1)

BLOCK SIZE 500
L 01F4 10:30:59  29 JUL 1991 STOCK YELLOW ONLY          01

RECORD = 1

   1 :5500^TABLE, YELLOW, PINE^15^LS/15/77^5000^25^2000^:
  51 :8190^{2000^SETTEE, YELLOW, OAK^18^DN/1/69^10000^30:
 101 :^1000^8176^{9000^SIDEBOARD, YELLOW, ASH^99^MN/14/7:
 151 :0^13000^15^1000^8190^{7777^STOOL, YELLOW, LAMINATE:
 201 :^25^DN/7/23^1775^60^2000^8178^{_X{{{{{{{{{{{{{{{{:
 251 :{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{:
 301 :{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{:
 351 :{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{:
 401 :{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{:
 451 :{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{:
```

As this example illustrates, any HEADING clause in the T-DUMP sentence will be used to generate a disk/tape label. If no HEADING is specified, the label will contain the block size (01F4 in this instance), the time, the date, the file name and the volume sequence-number:

```
L 01F4 09:55:42  29 JUL 1991 STOCK                     01
```

If the H option is specified, then no disk/tape label will be created.

3.2 T-DUMP with several files

Apart from writing the disk/tape label, the T-DUMP verb does no initialisation or other checking of the disk/tape. Furthermore, it only dumps the items for a single file. If you want to use T-DUMP to archive more than one set of data on a single disk/tape, then you must do the jobs one after each other - by issuing a series of T-DUMP commands. If you want to dump one file and then - after rewinding the disk/tape dump a second file, you must take care in positioning the disk/tape for the second file, otherwise you may overwrite the first.

If you are recovering a single file from a disk/tape containing a series of T-DUMPs, then you must position the disk/tape at the correct file before issuing the T-LOAD command. This demands that the contents of the disk/tape are recorded in detail - in a record book - and that each volume is labelled correctly. The creation of a disk/tape label when dumping the file can help in locating the correct file.

3.3 T-LOAD

This Access verb is used to load an entire file or individual items from disk/tape to which they have been dumped by the S-DUMP / T-DUMP verb. It has the same format as other Access sentences.

A number of options are available with this command:

I to suppress the display of the item-ids as the items are loaded

O to overwrite any existing items on the file which have the same item-ids as those being loaded.

Some examples of its use:

```
T-LOAD STOCK
T-LOAD DICT STOCK
T-LOAD STOCK '1000' '2000' '3000'
T-LOAD STOCK WITH COLOUR "RED"
T-LOAD STOCK (O
```

In these cases, we assume that there is already a file called STOCK to receive the data which is being transferred from backing storage and that, in the final example, the DICT of the STOCK file contains an attribute definition for the field COLOUR.

Some implementations have extended the T-LOAD verb so that it may be used to recover data from backing storage files which have been produced by means of the file-save and the account-save procedures. The extended T-LOAD command has all the features of the standard T-LOAD, as described above, with the options of the SEL-RESTORE command which it replaces. The format of the backing storage volume is first scanned to check whether it is T-DUMP or file-save/account-save format; if the latter, then the procedure behaves like the SEL-RESTORE command.

3.4 The TAPE modifier for archived data

The TAPE modifier in an Access sentence will allow reports to be produced upon data which has been dumped to disk/tape by means of the T-DUMP verb. Thus, the sentence:

```
LIST STOCK DESCRIPTION TAPE
```

will look to the DICT of the STOCK file for the attribute definition DESCRIPTION, and then use this to produce a report on that field of the data records which are held on **backing storage** disk/tape. Any data records on the STOCK file will be ignored by this sentence.

The following points apply when using the TAPE modifier:

1) The data must have been previously dumped to disk/tape by means of the T-DUMP verb. Data archived by any other means - such as a file-save or account-save - cannot be

interrogated in this manner.

2) The data file - STOCK in these examples - must exist on the system and will hold the attribute definitions which are used in the sentence.

3) Selection criteria may be included in the Access sentence, but sort specifications cannot be used. The data records are processed in the order in which they are held on the backing storage.

6) The disk/tape must be loaded, on-line and positioned at the start of the T-DUMP file.

This feature is valuable when processing records which have been archived to backing storage, as it allows them to be interrogated without having to load them back on to the system.

4 Saving data

There are several methods of saving data on backing storage. In the previous section, we looked at the T-DUMP and the T-LOAD verbs. In the present section, we shall look at the file-save and the account-save operations.

4.1 File-save

The general purpose of the file-save is to save - or dump - the contents of the entire system on to backing storage, that is, floppy disk or magnetic tape.

A full file-save should be carried out at least once a week, but if it is feasible, it is advisable to perform a file-save at the end of each working day. Systems which have significant activity - where a large amount of data is changed and processed each day - might be advised to perform file-saves more frequently.

4.2 Performing the file-save

When you wish to perform a file-save, you must ensure that the correct disk/tape is mounted, on-line and WRITE enabled. The file-save procedure will be invoked with a command such as:

```
FILE-SAVE
```

The utility will normally ask you a series of questions concerning:

* whether the accounts/files are to be display or printed as they are processed,

* whether the file statistics from the STAT-FILE are to be printed at the end of the procedure,

* whether the contents of the STAT-FILE are to be copied to the end of the disk/tape after the file-save has been

performed.

* whether the disk/tape is to be verified (by conducting a dummy SEL-RESTORE, as described below).

and the utility will finally ask for a piece of text which is to be used as a part of the disk/tape label.

The file-save routine scans **all** the files on the system, including the standard files used by the operating system and the users' files. This is done in the following manner:

1) The shape (the modulo and separation) and the contents of the SYSTEM file are saved, and as this is being done, the routine generates a list of all the accounts which are to be saved on the system. This list contains the names of the accounts in the order in which they are encountered on the SYSTEM file.

   As we describe below, any accounts which have a DX-pointer or a Q-pointer as their account-definition item will be ignored completely. Any accounts which have a DY-pointer will have their modulo and separation preserved but their contents will be ignored.

2) Using this list of the accounts on the system, the routine then processes each account in turn.

   For each account on the system:

3) The shape and the contents of the MD are recorded, and as this is being done, the routine generates a list of all the files which are to be saved on that account. This list contains the names of the files in the order in which they are encountered on the MD of the account. Q-pointers, DX-pointers and DY-pointers will be handled as at step (1).

4) Using this list of files on the account, the routine then processes each file in turn.

   For each file on the account:

5) The shape and the contents of the DICT section of the file are recorded, and as this is being done, the routine generates a list of all the data-level identifiers which are to be saved on that DICT. This list contains the names of the data-level identifiers in the order in which they are encountered on the DICT of that file, on McDonnell Douglas implementations there will be only one such identifier. Q-pointers, DX-pointers and DY-pointers will be handled as at step (1).

6) Using this list of data-level identifiers on the DICT, the routine then processes each data section in turn.

   For each data section on the file DICT:

7) The shape and the contents of the data section are recorded.

   Step (7) is repeated for each data section on the file DICT.

Steps (5) to (7) are repeated for each file on the account.

Steps (3) to (7) are repeated for each account on the system.

The information about the accounts and their files is stored on the statistics file STAT-FILE, as we discuss later.

4.3 D-pointers

As the file-save (or the account-save) proceeds, each item is scanned and any D-pointer (that is, an item with a D in attribute 1) will be regarded as identifying a file which is to be eligible for dumping. Any other items (including Q-pointers) will be dumped to disk/tape without any further action being taken. A D-pointer always identifies a file, the significance of a D-pointer and the file to which it refers depends upon where the pointer is held:

* a D-pointer - an account definition item - on the SYSTEM file identifies the MD of an account,

* a D-pointer - a file-definition item - on the MD of an account identifies the DICT section of a file belonging to that account, and

* a D-pointer - a data-level identifier - on the DICT section of a file identifies a data section of that file.

If a D-pointer contains

```
DX
```

in attribute 1, then that file will be completely ignored by the file-save. If a D-pointer contains

```
DY
```

in attribute 1, then the **contents** of the file will be completely ignored by the file-save, but the modulo and separation of the (empty) file will be dumped to disk/tape. If a D-pointer contains

```
DC
```

in attribute 1, then the file and its contents will be dumped to disk/tape and an C-pointer items in the file (which point to such items as saved-lists and Basic object programs) will be followed up and the frames to which they point will also be dumped to disk/tape.

Part of a typical display produced as the file-save (and also the account-save) takes place might look like that below, showing each account and its files; others may show only the names of the accounts which are dumped.

```
*     1 SYSTEM              The SYSTEM file
1     2  SALES              MD of the SALES account
1     3   WAGES             DICT section of the WAGES file
1     4    WAGES            data section of the WAGES file
1     5   PROCFILE          DICT (only) section of the PROCFILE file
1     6   RATES             DICT section of the RATES file
1     7    RATES            data section of the RATES file
1     8    STD              data section of the RATES,STD file
1     9   INVOICE           DICT section of the INVOICE file
1    10    INVOICE          data section of the INVOICE file
```

The number in column 1 shows the reel number: 1 for the first disk/tape, 2 for the second and so on; this is 0 if the data is not being dumped to backing storage and just the file statistics are being produced. The number in column 2 shows the sequence number of that file on the disk/tape and is the item-id of the item on the STAT-FILE. The successive indentation of the file name in the third column indicates whether this is an account name, a DICT file name, or a data file name. Our italic annotation here describes each file.

If the file-save occupies more than a single disk/tape, then you will mount the further volumes - that is, the individual disk or reel of tape - when the system asks you. If this is the case, then you cannot leave a file-save operation to run overnight unattended.

If anything goes wrong during a file-save operation, it is best to start again from the beginning.

4.4 A cycle of file-save disk/tapes

It may require several floppy disks or several magnetic tapes to hold a complete file-save. For the following explanation, we shall use the term **set of volumes** to mean all the disks or all the tapes which are needed to hold each file-save dump.

It is usually not sufficient to use the same set of volumes over and over again for all your file-saves. It is too dangerous to put all your eggs (or data!) in one basket.

A simple solution is to have **several** sets of volumes and use these in a cycle. So if you use three sets and perform a file-save at the end of each day's business, you would use the sets like this:

```
Monday ..................... SET 1
Tuesday .................... SET 2
Wednesday .................. SET 3
Thursday ................... SET 1
Friday ..................... SET 2
Saturday ................... SET 3
```

If you work a five-day week, then the pattern might look like this:

```
Week 1 : Monday ..................... SET 1
Week 1 : Tuesday .................... SET 2
Week 1 : Wednesday .................. SET 3
Week 1 : Thursday ................... SET 1
Week 1 : Friday ..................... SET 2

Week 2 : Monday ..................... SET 3
Week 2 : Tuesday .................... SET 1
Week 2 : Wednesday .................. SET 2
Week 2 : Thursday ................... SET 3
Week 2 : Friday ..................... SET 1
```

and so on.

Some larger installations have many more than three sets of backup disks/tapes and even more complicated patterns.

A common practice is to make **two** copies at the end of each week and **two** at the end of each month and lodge one of these copies in a safe place (such as a bank or a distant office) so that, in the event of the computer site being damaged, the data will still be recoverable.

Whatever system you use, you should always label the volumes clearly to show

* their contents,

* the date and time when they were produced, and

* if each file-save occupies more than a single volume, the number of each volume within the set.

A similar system of cycling the disks/tapes would also be used for the account-saves for each account which is saved individually.

4.5 Partial file-save

At some installations, it is not possible to perform an entire file-save each night. This could be for a number of reasons: the file-save process may be too large and take up so many disk/tapes that it could not be left running without someone in attendance; only a few accounts may be changed whilst the major part of the system (programs and such) is static and unchanging from week to week.

If this is the case, then you can take advantage of the DX pointers which we mentioned above to indicate that specific accounts are not to be saved. We illustrate this by means of an example: let us imagine that on Monday and Wednesday, we wish to save just accounts ACCA, ACCB and ACCC, and on Tuesday and Thursday we wish to save just accounts ACCD and ACCE. Then on Monday and Wednesday we should set attribute 1 of the **account definition items** for ACCD and ACCE to DX by means of a sequence of Editor command such as

```
EDIT SYSTEM ACCD
R
DX
FI
```

thus instructing the file-save to ignore these accounts. Having done this for all the accounts which we are to ignore we would then perform a full file-save. When the file-save has been completed, we must remember to reset the DX back to D for those accounts. Similarly, on Tuesday and Thursday, we would inhibit accounts ACCA, ACCB and ACCC in the same manner.

Obviously, such a *file-save* disk/tape could not be used to perform a complete file-restore, otherwise the accounts which were ignored would not be returned to the system.

The same process can be repeated to prevent individual files - such as large program files - from being saved every night. In this case, we should change the D-pointer (or the DC pointer, in the case of a program file) of the file definition item in the MD of the owner's account to DX (or DCX).

McDonnell Douglas implementations have a further utility:

```
F-S
```

which first displays the status - D or DX - of the accounts on the system, thereby allowing the user to check the accounts which will (or will not) be saved.

4.6 Multiple account-save

Some implementations, such as McDonnell Douglas, offer a multiple account-save facility. This allows several single accounts to be saved in one operation. The utility is invoked by a command such as:

```
M-A-S
```

and proceeds, asking the user for the name of each account which is to be saved; the user may also specify the name of a saved-list which contains the names of the accounts to be saved.

There is a corresponding multiple account-restore.

4.7 Account-save

Just as the file-save dumps the contents of the entire system to disk/tape, so the account-save procedure dumps the contents of a single account to disk/tape.

When you wish to perform an account-save, you must ensure that the correct disk/tape is mounted, on-line and WRITE enabled. The account-save procedure will be invoked with a command such as:

```
ACCOUNT-SAVE
```

or by means of the SAVE verb, and dumps the contents of a single account on to disk/tape.

the standard ACCOUNT-SAVE utility asks for two pieces of information:

* a piece of text which is to be used as a disk/tape label, and

* the name of the account which is to be saved.

As mentioned above, any files which are identified by a DX-pointer instead of a D-pointer will be ignored. Those with a DY-pointer will have their contents ignored but the file itself - the modulo and separation - will be recorded on the dump.

4.8 File statistics

The file statistics which are produced by the file-save and the account-save procedures are placed on the STAT-FILE. You can interrogate the STAT-FILE by means of normal Access sentences

```
LIST STAT-FILE
SORT STAT-FILE WITH GFE
```

or your implementation may have a utility Proc such as

```
LIST-FILE-STATS
```

The utility will then ask you a series of questions about the detail and the destination of the output report.

When the file-save procedure, the account-save procedure and the SAVE verb produce the file statistics, they clear any the data from the STAT-FILE before they proceed.

4.9 The fields on the STAT-FILE

As new items are written to the STAT-FILE, they are given an item-id of the form

```
v:n
```

where v is the number of the volume (disk or tape) on which that file is dumped, and n is the sequence number of the file.

The actual datanames which are given to the dictionary definitions for the various fields the STAT-FILE may be different on different implementations. The following list shows some typical datanames and an explanation of the less obvious:

£OVF ..... *The number of overflow frames*

%UT ..... *The % of the file space used*
ACCOUNT
AV/GP ..... *Average items per group*
AV/ITM ..... *Average bytes per item*
BASE
BMS ..... *Base, modulo and separation*
FILE
FRAMES
FRMS/GP ..... *Average frames per group*
GFE ..... *Number of GFEs in file*
ITEMS.
MOD
PAD ..... *Unused space in file*
SEP
SEQN ..... *Sequence number of file during save*
SIZE ..... *Total number of bytes used*

Of course, the command

    LISTDICT STAT-FILE

will produce a list of the available datanames.

4.10 SAVE verb

The FILE-SAVE procedure and the ACCOUNT-SAVE procedure are simply Procs which offer a *user-friendly* interface to the fundamental TCL SAVE command. Because the SAVE verb can be used directly - without the software of the FILE-SAVE and ACCOUNT-SAVE Procs getting in the way - you may find it easier to apply the SAVE verb in any front-end processes which you provide for your users. The SAVE verb performs a number of tasks and has a number of facilities which are not readily available via the standard utilities:

1 Copy the contents of all or a part of the system to disk/tape. That is, a file-save, a binary-save or an account-save.

2 Produce file statistics.

3 Advanced Pick and Ultimate implementations have an incremental file-save facility which will dump only those groups which have been changed since the last complete system save (file-save).

  The items (or groups on Ultimate) which are changed by any process have a flag set to indicate this. When an incremental file-save is performed, only these flagged items will be saved. The various options may then un-flag these items, as described below.

4) Advanced Pick has a facility to save only those items whose item-ids are held in attribute 25 onwards of the item on the FILES file which relates to the file being processed.

The verb differs slightly between generic Pick [P], Advanced Pick [AP] and Ultimate [U], which use the form:

SAVE (options

and McDonnell Douglas [M], which has the form:

SAVE SYSTEM * (options

There are a number of options:

A includes the contents of the ABS frames (or bootstrap section) at the front of the disk/tape. [M]

B performs a binary-save. This assumes that all other users are logged off and performs a quick dump of the entire system, frame by frame. [P]

D processes the data files.

F displays a list of the names of the files as they are processed. Otherwise, only the account names are displayed.

G fixes any GFEs which are encountered. [U]

I processes a single account. The process will issue a conversational prompt for the name of the account to be saved.

This option is exercised by the ACCOUNT-SAVE Proc.

If this option is omitted, then the entire system will be processed.

K suppresses the output to disk/tape. This is only valid when using the S option to produce file-statistics without performing the file-save. [M]

L saves only those items whose item-ids are held in attribute 25 onwards of the appropriate item on the FILES file. If there is no such list, then the file is treated as if it were defined by a DY-pointer. Items thus saved, remain flagged for any subsequent incremental file-saves. [AP]

M is similar to the L option in that it saves only those items whose item-ids are held in attribute 25 onwards of the appropriate item on the FILES file. With the M option, however, any items thus saved, are un-flagged and ignored on any subsequent incremental file-saves. [AP]

N no overflow space is required. [P] [U]

P sends the printed output to the printer.

R produces a list of the items which are saved and retains this on the FILES file. [AP]

R resets the **group changed** flags if an incremental file-save is to be performed. [U]

S writes the file statistics to STAT-FILE.

T saves the data on the disk/tape.

This option will be **omitted** if, for example, you simply want to produce the current file statistics by using the S option. [P] [U]

T suppresses the generation of disk/tape-labels. [M]

U invokes the incremental file-save and saves only those groups which have a **group changed** flag set. [AP] [U]

V performs a file-save but does not un-flag any changed items. [AP]

Z writes the file statistics to a file specified by the user. [M]

The process will issue a conversational prompt for the name of the file.

Z creates the disk/tape in SYS-GEN sequence rather than the physical order of the system. This is useful if you are transferring the contents to another system. [U]

4.11 Checking the disk with SAVE

If you need to check the system for GFEs, you can do this by performing a dummy file-save by means of the command

```
SAVE (DF
```

This is a variant of the technique which we introduce below for producing the file-statistics and will start at the SYSTEM file and scan all the files on the disk. If any errors are encountered, these will be reported by the system.

4.12 SAVE and file statistics

As you may appreciate, it is possible to perform any of the file-save / account-save operations directly by using the SAVE verb. Some of the more useful applications include:

a) file-statistics for the entire system

You may achieve this by means of a command such as:

```
SAVE (DS
```

on Pick and Ultimate implementations, or

```
SAVE SYSTEM * (DKS
```

on McDonnell Douglas.

b) file-statistics for a single account

You may achieve this by means of a command such as:

```
SAVE (DIS
```

on Pick and Ultimate implementations, or

```
SAVE SYSTEM * (DIKS
```

on McDonnell Douglas.

4.13 Binary-save

There is also a **binary-save** facility which dumps a very quick **snapshot** of the current state of the entire disk. This dump can then be reloaded by means of the corresponding **binary-restore** facility. Unlike the normal file-save, the binary-save/binary-restore process does not re-allocate the disk space. It is provided for those situations when an instant backup is needed at short notice (as when the engineer arrives unexpectedly).

The binary-save is invoked by a TCL command such as

```
BINARY-SAVE
```

or by issuing a SAVE command with the appropriate options, as described above.

5 Recovering data

The manner in which you recover data depends upon how it was saved or archived to backing storage.

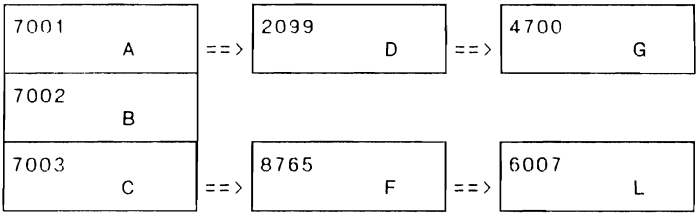
We looked at T-DUMP and T-LOAD earlier. In this section, we shall look at the following processes:

```
File-restore
Account-restore
Selective-restore
```

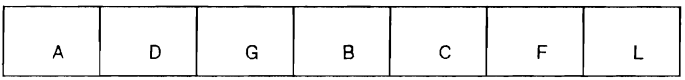
5.1 File-restore

A file-restore loads the contents of the entire system back on to the system disk (the hard disk) from a file-save disk/tape.

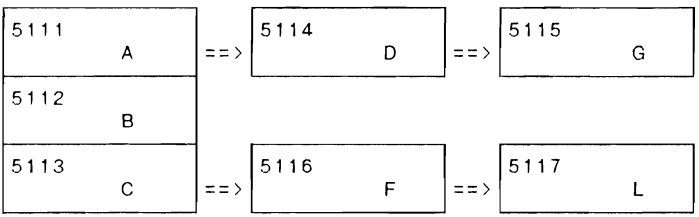
An important feature of a file-restore is that the entire contents of the system disk - the files and accounts - are reorganised as they are transferred from the disk/tape. Thus, a file which was badly fragmented before the file-save was performed will be collected together as the file-save takes places and held as a block of data on the disk/tape. When the file-restore takes place, the block of data representing the contents of the file is placed on to the hard disk as a set of contiguous frames. We can illustrate this with a simple example. A file with a modulo of 3 and a separation of 1 might expand and overflow into additional frames so that it looks like this:



where A, B, C and so on are the item-ids of the 6 (large) items on the file and the numbers indicate the FIDs of the frames which the data occupies. When a file-save is performed, all the data will be gathered into one chunk on the disk/tape, like this:



When a file-restore is performed, all the files are recreated (possibly with a new modulo and separation) and the items written back to the file in the sequence in which they are encountered on the disk/tape, so our file might reappear possibly looking like this:



In the *MB-Guide to files: monitoring* and resizing, we discuss the way in which the modulo and separation can be changed when a file-restore or an account-restore is performed.

5.2 Performing a file-restore

When you wish to perform a file-restore, you must ensure that the correct disk/tape is mounted and on-line. The file-restore procedure will be invoked with a command such as:

```
    :FILELOAD
or
    :FILES
```

As the file-restore takes place, the accounts and their files are restored in the same sequence as they are held on the file-save disk/tape, being placed sequentially on the system disk. When the file-restore is complete, the unused

disk space will be available as a large block of contiguous frames.

Once a file-restore has begun, no other users should log on until the file-restore is complete.

If the file-save occupies more than a single disk/tape, then you must be sure to mount the volumes in the correct sequence. If this is the case, then you cannot leave the file-save or the file-restore operation to run overnight unattended.

The moment when you begin to perform a file-restore is a fairly emotional time. If there are any errors on the disk/tape then your entire system may be jeopardised. For this reason, some users prefer to produce two copies (two file-save disk/tapes) and check them both before attempting to perform a file-restore with one of the copies. If anything does go wrong during a file-restore operation, it is best to start again from the beginning; your reference literature may give some guidance in the event of problems during a file-restore.

The screen display during a file-restore is similar to that of the account-restore shown below.

5.3 Account-restore

The account-restore procedure loads the files of an account from a file-save disk/tape or from an account-save disk/tape.

When you wish to perform a account-save, you must ensure that the correct disk/tape is mounted and on-line. The account-save procedure will be invoked with a command such as:

```
ACCOUNT-RESTORE newaccount
```

where **newaccount** is the name of the account as it is to be loaded on to the system. The process will then ask for the name of the account as it is held on the disk/tape.

Note that you cannot reload an account which is already on the system; the system will reject your ACCOUNT-RESTORE command if you attempt to do so. If you are in this situation - you may need to recover several files from an account-save disk/tape - then you can restore the old account to the system with a dummy name. For example:

```
ACCOUNT-RESTORE COPY.WAGES
```

Part of a typical display produced as the account-save (and also the file-save) takes place might look like that below, showing each account and its files.

```
SALES                          The account name
 SALES 30389,1,1               MD of the SALES account
  WAGES 53195,1,1              DICT section of the WAGES file
   WAGES 46527,211,1           data section of the WAGES file
  PROCFILE 57100,17,1          DICT (only) section of the PROCFILE file
  INVOICE 46424,1,1            DICT section of the INVOICE file
   INVOICE 46425,11,1          data section of the INVOICE file
```

The figures following the name of each file show the base FID (the frame identifier of the first frame in the file), the and the modulo and separation of the file as it now exists on the system.

5.4 RESTORE-ACCOUNTS

Some implementations provide a utility which will read through a file-save disk/tape and restore all those accounts which are not already on the system. This is invoked by the command

```
RESTORE-ACCOUNTS
```

If any accounts on the disk/tape already exist on the system, then they will be skipped.

5.5 Multiple account-restore

Those implementations, such as McDonnell Douglas, which have a multiple account-save facility will have a complementary multiple account-restore facility. This is invoked by a command such as

```
M-A-R
```

and offers the user an opportunity to restore all the accounts or to specify the name(s) of the individual account(s) to be restored from the multiple account-save disk/tape.

5.6 Selective restore - SEL-RESTORE

The selective-restore procedure allows you to recover one, several or all the items on a file from a file-save or an account-save disk/tape.

When you wish to perform a selective-restore, you must first logon to **the account to which the file or item is to be restored**, and then load the disk/tape on which the file is held and make sure that the device is ready and on-line.

The selective-restore procedure will be invoked with the command:

```
SEL-RESTORE file itemlist (options}
```

or

```
SEL-RESTORE file * (options}
```

where:

file is the file on the system into which items will be placed - this does not have to be the same file name as that which holds the item on the disk/tape. The file must be defined on the account from which the restore is run;

itemlist is either a list of item-ids identifying the items to be restored from the disk/tape, or an asterisk if you want to restore all the items on the file;

options are any combination of the following:

A This indicates that the disk/tape is already positioned at the correct account. Typically, this option is used when you are recovering several files, one after another, and avoids the process having to scan through the disk/tape each time to find the correct account. Obviously, when you do this, you must restore the files in the correct sequence.

C This option is used in conjunction with the N option and causes every item before the next end-of-file mark to be considered for restore. This ensures that data can be restored even though the D-pointer for the file might be damaged on the disk/tape.

F This displays the names of all the files on the disk/tape as they are encountered.

I This suppresses the display of the item-ids as they are restored to the file.

N The file is to be identified on disk/tape by its file number. We'll say more about this below;

O Overwrite any items which already exist on the file;

At the prompt:

```
Account name on tape?
```

you should reply with the name of the account under which the file was saved - this is not necessarily the same account name as that to which the items are being restored.

The next prompt will be one of the following:

```
FILE£?
```

if the N option was used. The reply will be the **file-number**. This identifies the sequence-number (or the **order**) of the file on the disk/tape, and is shown on the file statistics printout for the appropriate file-save.

```
FILE NAME?
```

if the N option was **not** used. Your reply will be one of the

following:

a) null - in which case the Master Dictionary of the account will be restored,

b) the name of the file as it is on the disk/tape - this will restore the data section of the file. For example:

    WAGES

    will restore the item(s) from the data section of the WAGES file on the disk/tape and place them into the file specified on the SEL-RESTORE command.

c) DICT file - specifying that the DICT section of the file is to be restored:

    DICT STOCK

d) file **ppppp** - where **ppppp** is a D-pointer in the dictionary for **file**. This is used when the DICT section of the file has several data sections:

    SALES,NORTHERN

The response will indicate the name of the file as it appears on the disk/tape. Remember that this is not necessarily the same file name as that to which the items are being restored.

The system will then scan through the disk/tape, displaying the names of all the accounts and the files which it encounters on the disk/tape.

When the restore is complete the following message will be displayed:

    n ITEM(S) HAVE BEEN RESTORED

or, if the end-of-file marker is found and the appropriate file or items have not been found, then the message will be:

    0 ITEMS HAVE BEEN RESTORED

The S option will suppress this message.

Incidentally, the SEL-RESTORE command is a useful means of checking a file-save or account-save disk/tape by causing the processor to scan through the entire disk/tape in response to a command such as:

    SEL-RESTORE MD *

and entering a *fictitious* account name ZZZZZZ and file name ZZZZZZ when invited. The processor will scan through the entire disk/tape and point out any parity errors or, if there are no errors on the disk/tape, it will display the message

```
    0 ITEM(S) HAVE BEEN RESTORED
```

This is the *dummy restore* which the FILE-SAVE utility offers you to check the file-save disk/tape.

Some implementations have replaced the SEL-RESTORE facility by an extended T-LOAD verb. The T-LOAD verb may be used to recover data from backing storage files which have been produced by either the T-DUMP verb or by the file-save and account-save procedures. The extended T-LOAD command has all the features of the standard T-LOAD,
such as selection criteria:

```
    T-LOAD STOCK WITH COLOUR "RED"
```

together with the options of the SEL-RESTORE command which it replaces. The format of the backing storage volume is first scanned to check whether it is T-DUMP or file-save / account-save format; if the latter, then the procedure behaves like the SEL-RESTORE command.

6 Glossary

The following terms are used in this *MB-Guide*:

Account-restore The process of recovering the contents of a specific account from a copy dumped to backing store by the account-save process.

Account-save The process of dumping the contents of a specific account to backing storage.

Archive, as a verb: to dump selected parts of the system (accounts, files or items) to backing storage for safekeeping and to allow the user to remove such dumped data from the system when it is no longer required on-line.

Binary-restore The recovery of the entire contents of the system from backing storage.

Binary-save The dumping of the entire contents of the system to backing storage.

File-restore The process of recovering the entire system (or a part of it) from a copy dumped to backing store by the file-save process. The files on the hard disk are completely reorganised as the process takes place.

File-save The process of dumping the contents of the entire system to backing storage. File-saves are performed for security purposes and allow and lost or corrupt items to be recovered to their state at the time the file-save was performed.

Incremental file-save a file-save which archives only those items which have been changed since the last file-save was performed.

Selective-restore The process of recovering specific items or files from a backing storage dump which was created by the account-save or file-save process.

In this and the other *MB-Guides*, the keyboard control keys have been represented by their name enclosed in angle brackets:

<BREAK> identifies the break key or the equivalent sequence which interrupts the current process.

<CTRL> identifies the control key.

Certain characters are entered at the keyboard as a combination of one or more of the above keys together with other keyboard characters. For example, the subvalue-mark (character 252) may be entered as:

<CTRL> \

that is, by holding down the <CTRL> key and typing the normal \ character at the same time. Similarly, the value-mark can be keyed in as the sequence <CTRL>] and the attribute-mark as the sequence <CTRL> ^

<ENTER> identifies the ENTER key which is used to transmit each piece of data to the system.

This is generally represented by the <RETURN> key in the text.

<ESC> identifies the ESCAPE key.

<LINE FEED> identifies the line-feed key. On some keyboards, this may be the down-pointing arrow. The sequence <CTRL> J is equivalent.

<RETURN> identifies the RETURN key which is used to transmit each piece of data to the system.

On some keyboards, this may be the <ENTER> key or the down-left-pointing arrow key. The sequence <CTRL> M is equivalent.

Certain features of Access and the operating system are peculiar to specific implementations. In this *MB-Guide,* we have used the following symbols to indicate where that feature is available:

[AM] ADDS Mentor implementations and the INFO/ACCESS language,

[AP] Advanced Pick and the ACCESS language,

[OA] Open Architecture and, unless stated otherwise, Advanced Pick.

[P] Generic Pick, R83 and, unless stated otherwise, Open Architecture and Advanced Pick.

[PI] Prime Information and the INFORM language,

[M] McDonnell Douglas and Reality implementations and the ENGLISH language,

[U] Ultimate implementations and the ULTIMATE language,

[UP] the Update Processor of Advanced Pick.

An entry with no annotation will normally be available on all implementations.

Index

MB-Guides

MB-Guides are designed to serve as introductory texts to a range of fundamental topics within the Pick operating system. They are also available for the following subjects:

- MB-Guide to Access conversions and correlatives
- MB-Guide to Access sentences
- MB-Guide to Basic programming
- MB-Guide to Creating and using Procs
- MB-Guide to using the Editors
- MB-Guide to File design
- MB-Guide to File-save and file-restore
- MB-Guide to Files: monitoring and sizing
- MB-Guide to Group format errors
- MB-Guide to Operations and systems management
- MB-Guide to Pick on the PC
- MB-Guide to Program design
- MB-Guide to Security
- MB-Guide to The Basic symbolic debugger
- MB-Guide to The spooler
- MB-Guide to The system debugger
- MB-Guide to Using backing storage

The format of the *MB-Guides* is such that they may be easily updated and amended to reflect the current state of the operating system. In order that this and the other Mating MB-Guides continue to meet the needs of the users, we would appreciate your comments on this guide and your suggestions for further titles in this series.

MB-Master self tuition courses are also available on a wide range of topics related to the Pick operating system:

- Access techniques
- Advancing in Basic
- Moving to Basic - a conversion course
- Pick systems management
- Programming in Basic
- Starting Access
- Starting ACCU/PLOT
- Starting CompuSheet+
- Starting Jet
- Starting Pick
- Starting Runoff
- Starting SB+
- Systems development
- Writing Procs

MB-Guides

014
002

The booklets in the *MB-Guide* series cover a range of fundamental topics of interest to users and those responsible for running Pick systems.

Each *MB-Guide* deals with a specific aspect of the operating system and the booklets represent an economical introduction to the various topics and the whole series forms an integrated presentation of the subject matter.

The booklets are intended to be a working document and, for this reason, space is provided for the user's notes, and the reader is encouraged to amend the booklet so that it applies to his/her own system.

It is anticipated that the series of *MB-Guides* will be of special interest to new users, and it should prove useful for software houses and others who are responsible for the instruction of their clients and staff in the fundamental aspects of the Pick operating system.

File-save/restore

Malcolm Bull

Training and Consultancy Publications