This manual was designed and written by Human-Computer Interface Limited, Cambridge.

First published in 1986 by Acorn Computers Limited.

Typeset by Interaction Systems Limited, Cambridge.

British Broadcasting Corporation has been abbreviated to BBC in this publication.

FIRST EDITION

ISBN 1 85250 028 X
Part no 0449,880

Published by Acorn Computers Limited, Fulbourn Road, Cherry Hinton,
Cambridge CB1 4JN, England.

View Family Filing System

Guide

ACORNSOFT
The choice of experience
in software.

ACORNSOFT

The choice of experience
in software.

Contents

Introduction

In order to get the most out of the business-productivity tools in the View family you need to understand something about how your filing system works. The aim of this guide is to help you get the best out of the particular filing system you are using. It will also help you to choose a filing system if you are undecided about which one to use.

What is a filing system?

Your filing system provides long term storage for the work you have created using the View-family programs, and that you wish to keep for use at some later time. The filing system allows you to:

- store (SAVE) information or documents in a more permanent form. Once saved, they are retained even when the computer's power is turned off.
- retrieve (LOAD) information or documents that have been saved on a previous occasion.
- access existing information (ie selectively retrieve part of the information, or alter it). This is termed random-access, and enables the View-family programs to access files containing far more information than would fit into the computer's memory at one time.

Installation

This guide describes the operation of the View-family software with the various filing systems. The software is supplied in a variety of media and formats depending partly on the model of microcomputer for which it is intended. In order to install the View-family software onto the particular microcomputer (eg Model B, Master Compact, etc) and onto your chosen filing system (eg DFS, ADFS, etc) it is necessary to refer to the instructions in the configuration leaflet supplied.

Filing systems available in your microcomputer

Depending on which model of microcomputer you own, you may have different filing systems available as shown in the table below.

	Computer				
Filing System	BBC Micro	Electron	Master 128	Master Compact	**Storage Medium**
CFS	Standard	Standard	Standard	N/A	Cassette
DFS	Standard	N/A	Standard	Optional	Disc drive
ADFS	Optional Upgrade	Optional (Plus 3)	Standard	Standard	Disc drive
ADFS	Optional	N/A	Optional	N/A	Winchester
NFS	Optional	N/A	N/A	N/A	Econet file-server
ANFS	N/A	N/A	Optional	Optional	Econet file-server

You can tell which filing systems are present in your computer by typing `*HELP`. The presence of any filing system, apart from CFS, will be indicated in the list of ROMs present in the computer.

The following sections give a brief description of the facilities provided by each filing system.

Cassette Filing System

The Cassette Filing System, or CFS, provides a slow but inexpensive way of saving and loading data using an ordinary domestic cassette tape-recorder. Files are stored on the tape in strict sequence, and the Cassette Filing System makes no attempt to maintain a record of which files are stored on a particular cassette.

To find information, the user therefore has to wind the cassette to the correct position before starting the loading process. This makes it unsuitable for large database programs, which need to be able to read and write data from different parts of a file. The CFS therefore cannot be used with ViewStore, or with ViewSpell. Both VIEW and ViewSheet can, however, be used with the CFS.

Disc Filing System

The standard Disc Filing System provides a fast and straightforward way of saving and accessing data. It supports random access, which enables information to be found very quickly wherever it is on the disc, and programs can read or update part of a file without having to load or save the whole file.

The DFS can handle up to four disc surfaces which will typically be on two double-sided discs, but imposes a limit of 31 files on each drive.

Advanced Disc Filing System

The Advanced Disc Filing System provides essentially the same facilities as the DFS, but its advanced capabilities offer a number of advantages; namely:

- a hierarchical directory structure which overcomes the limit of 31 files per surface of the DFS. The ADFS can therefore make efficient use of disc units with far greater storage capacities, such as the Winchester hard disc units.
- the ability to use both sides of a disc as a single entity.
- faster access to random-access files and a higher storage capacity on floppy discs.

As in the DFS, files are grouped into directories. However each entry in a directory can either be a simple file, or a subordinate directory, which itself may contain files and further subordinate directories.

At any time a directory is selected; this is known as the 'current directory'. When a disc is first used, the current directory is set to `$`, the 'root' directory which contains all other files and directories.

A new directory can be created within the current directory at any time with the `*CDIR` command. The directory can then be selected as the current directory with the `*DIR` command. All files are loaded from and saved to the current directory, unless a different directory or list of directories is specifically given as part of the file specification.

If you wish to work with several View-family applications on one ADFS disc, one approach, explained in greater detail in the ADFS chapter below, is to create a 'work' directory to contain all work with a specific View-family program; these could be called WordWork, SheetWork, StoreWork, SpellWork, etc. Alternatively, all files in one project could be kept in a single directory, called for example, CarWork.

Some of the View-family programs require particular types of files to be kept in specific sub-directories within the current directory. You therefore need to ensure that these sub-directories have already been created within the directory you are working in, with the `*CDIR` command. In particular, information can be transferred between ViewSpell, ViewStore and ViewPlot using linking files, which are always kept in directory `V`.

Network Filing System

The NFS allows several users to access a central computer, acting as a file server and providing filing facilities for all users. It is even possible for one network to have access to several separate file servers.

Users log onto the system by identifying themselves to the file server, and are supplied with a directory for their own use. The filing system is hierarchical, similar to the ADFS, so within the user's own directory the user may create files or further directories.

In addition to having access to their own files, users may be given access to some files created by other users, depending on the access rights set up on those files. Normally, the only access users would have to files outside their own filespace would be READ access to communal library files. It is appropriate to keep the View-family utility programs in a library, so that all users can have access to them – subject to licence restrictions.

Conventions followed in this guide

In this guide, all the commands you type yourself, and any text that you see on the screen, are shown in a computer typeface. Bold type is used to indicate a key on the keyboard. For example, to give the command

`*HELP` **RETURN**

you should type `*HELP` and then press the **RETURN** key.

Words shown in italics are not to be typed in literally; they indicate what the word represents. In the example

`*READ` *`filename`* **RETURN**

you should type the name of your file in place of *`filename`*.

1 The Cassette Filing System (CFS)

This chapter explains the use of VIEW and ViewSheet with the Cassette Filing System, or CFS. The CFS cannot be used with ViewStore or ViewSpell, since they each require random-access files.

If at all possible, use a cassette recorder with a remote control (REM) socket; otherwise, you will find yourself continually rewinding and estimating positions on the tape.

Before you can use a cassette recorder, you should select the Cassette Filing System by typing

`*TAPE` **RETURN**

Filenames on the CFS can be up to ten characters long, and may contain full stops.

VIEW

Saving text to cassette

Make sure there is a blank cassette in the recorder.

To save the whole contents of text memory to cassette, type

`SAVE` *`filename`* **RETURN**

The message appears on the screen:

```
RECORD THEN RETURN
```

Press `RECORD` on the cassette recorder and **RETURN** on the computer. When the prompt `=>` returns the file has been saved.

If your cassette recorder has no motor control (REM) socket, stop it quickly. If it has a REM connection, it will stop automatically.

The filename may be in upper or lower case, or any mixture of cases. So `TEXT`, `text` and `Text` are equally valid.

To save the file named in the heading on the command screen use the same procedure as above, except that you may miss out the filename.

Type

`SAVE` **RETURN**

To save part of the contents of text memory only, first set markers 1 and 2 at the beginning and end of the part you wish to save. Then use

`WRITE` *`filename`* `1 2` **RETURN**

Retrieving files from cassette

When using VIEW with cassettes, you cannot use the `LOAD` command. Instead, there is the `READ` command which adds new text from the file to any text currently in the memory. If you wish to wipe text from memory first type

`NEW` **RETURN**

Wind back the cassette to the appropriate point. Type

`READ` *`filename`* **RETURN**

Press `PLAY` on the cassette recorder. Depending on the cassette recorder, you may hear high- or low-pitched sounds. When the prompt `=>` returns the file has been read in.

If there is not enough memory available to read the whole of the file, VIEW gives you the message: `Not all read in.`

To read the contents of the file into a specific place in the text memory, first place marker 1 at the point where you want the text read in. Then use

`READ` *`filename`* `1` **RETURN**

ViewSheet

The ViewSheet commands `SAVE`, `LOAD`, `SW` (Save Windows), and `LW` (Load Windows) can all be used with cassette. However, linking files cannot be used with cassette since these require random access.

ViewStore and ViewSpell

Both ViewStore and ViewSpell require random-access files, and so are not suitable for use with the CFS.

2 The Disc Filing System (DFS)

This chapter describes the Disc Filing System. It provides access to up to four disc surfaces, which can be selected by the `*DRIVE` command. It is also possible to access a particular drive without changing the current drive number by including a drive specification in the filename.

There may be up to 31 files on each drive surface, and these files can be organised into groups, called directories. Each directory is specified by a single character, and the default directory is called `$`.

A full file specification might be

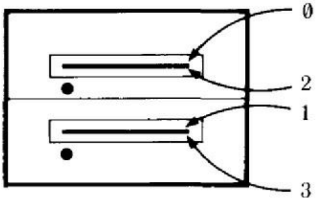
```
:2.T.bookch7
```

where `:2` is the drive number, which can be 0 to 3;

`T` is the directory, which can be any single letter or digit;

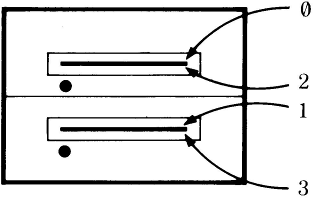
`bookch7` is the filename, of up to 7 letters and digits and some punctuation characters.

On a dual-drive system the drives are numbered as follows:



On a single-sided drive system the only available surface corresponds to drive 0.

The DFS allows 5 files to be open at any time, and this provides a limit on the number of indexes that can be updated by ViewStore.


0
2
1
3

Running out of room

Operations that cause the size of an existing file to be increased on the disc can cause filing-system errors when using the DFS. These errors are `Can't Extend` and `Disc Full`.

`Can't Extend` indicates that the expansion of the file has caused it to bump into the next file on the disc.

`Disc Full` indicates that the expansion has caused the file to bump into the end of the disc.

The ideal solution is to allocate files that may extend during their use to a separate disc surface of their own, or at least to ensure that they are the last file on each disc with the remainder of the disc space after them unallocated. Details of how to ensure this are given below for each member of the View family.

VIEW

All of the facilities of VIEW are available with the DFS.

The only commands that may cause problems with a `Can't Extend` message are the `EDIT` and `WRITE` commands, since in each case the output file expands as text is written to it. Such problems can be avoided by allocating the output file to a separate empty disc surface, or by compacting the disc with `*COMPACT` before performing the `EDIT` or `WRITE` command so that the output file is created as the last file on the disc.

ViewSheet

All ViewSheet commands can be used with DFS.

Linking files are created with the filename `V.VSn`, where `n` is the number of the file from 1-255. Up to five files may be used simultaneously with one sheet.

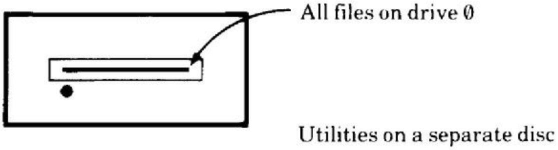
ViewStore

As records are added to the ViewStore database file, the size of the file on the disc will grow. If the file bumps into another file you will receive the `Can't Extend` message. The same problem can occur with indexes, which expand as they are updated. These problems can be largely avoided by careful organisation of the files on the available disc surfaces.

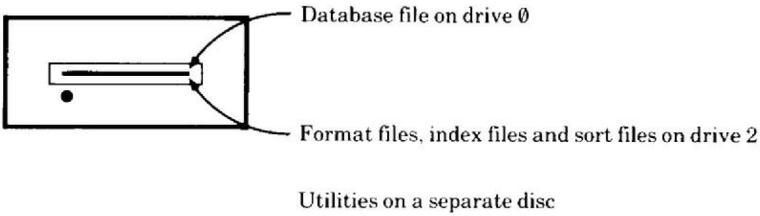
The ideal arrangement is to store the database on a surface by itself, allowing it maximum room for growth. Store the indexes on a different surface and rebuild them when necessary. On a single drive system you must store the database and indexes on the same surface. Keep the utilities on a separate disc.

The diagram below presents some recommended arrangements depending on the number of drives you have available.

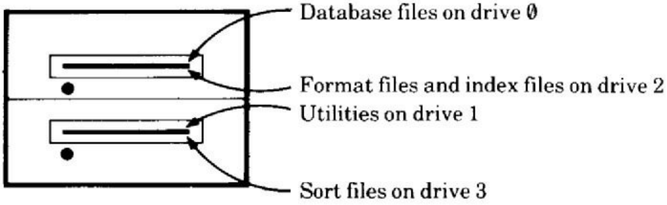
Single sided drive



Double sided drive


Database file on drive 0
Format files, index files and sort files on drive 2
Utilities on a separate disc

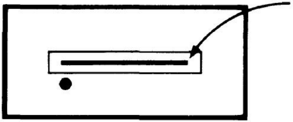
Dual disc drive


Database files on drive 0
Format files and index files on drive 2
Utilities on drive 1
Sort files on drive 3

These arrangements can be implemented with the following prefix commands for a double sided drive:

```
PREFIX D :0.
PREFIX F :2.
PREFIX I :2.
PREFIX S :2.
PREFIX U :0.
```

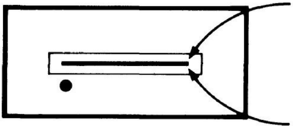
Single sided drive



All files on drive Ø

Utilities on a separate disc

Double sided drive

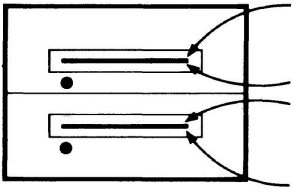


Database file on drive Ø

Format files, index files and sort files on drive 2

Utilities on a separate disc

Dual disc drive



Database files on drive Ø

Format files and index files on drive 2

Utilities on drive 1

Sort files on drive 3

And these commands for a dual drive:

```
PREFIX D :Ø.
PREFIX F :2.
PREFIX I :2.
PREFIX S :3.
PREFIX U :1.
```

If you are storing all of the files on one disc surface it is important to use the `SETUP` utility which creates the files in the best sequence on the disc, and allocates extra space on the disc for them. The files are created in the following order:

Format file, index files, database file.

The database file is the last file on the disc so that it can expand into the remaining free space.

If you are working with one single-sided or double-sided drive, sort files should be deleted when they are no longer required so that the index and database files do not bump into them when they expand.

Running out of room

The error message will indicate which type of file caused the error.

```
Can't Extend
```

The problem is that the database is prevented from being extended by another file on the disc after the database file. Place the database file at the end of the disc using the method described below.

```
Index: Can't Extend
```

Expand the index using the method described below. If you have the database on the same surface as the index replace the database at the end of the disc, after expanding the index.

```
Disc Full
```

The database has bumped up against the end of the disc. Delete as many files as you can from the disc and `*COMPACT` it. If you cannot delete any files, try deleting records from the database and use `CONVERT` to reclaim the disc space. If this does not work you will need a bigger disc system to store your database.

```
Index: Disc Full
```

Expand the index using the method described below. If you have the database on the same surface as the index replace the database at the end of the disc, after expanding the index.

To expand an index:

- `*DELETE` the index.
- `*COMPACT` the disc surface.
- Rebuild the index using the `INDEX` utility.

To place a file at the end of a disc:

- `*COPY` the file to another disc
- `*DELETE` the file
- `*COMPACT` the disc
- `*COPY` the file back to the disc.

ViewSpell

Because of the size of the dictionary, ViewSpell requires at least 200K bytes per surface; thus single-density (100K bytes) discs are not suitable.

In ViewSpell you can keep the dictionaries separate from the input text using the `PREFIX` command; the recommended values for a dual disc drive are:

```
PREFIX M :0.
PREFIX U :2.
PREFIX T :1.
```

to indicate master dictionary on drive 0, user dictionary on drive 2, and text on drive 1. ViewSpell places all dictionaries in directory `W`.

3 The Advanced Disc Filing System (ADFS)

This chapter describes the Advanced Disc Filing System, or ADFS, which, when available, is the recommended choice for use with the View-family programs. Since it is more complex, it is slightly harder to use, but provides greater flexibility.

The main directory, called the root directory, has the unalterable name `$`. A file in the ADFS can be uniquely specified by the drive identifier, followed by the path name of directories, followed by the filename, as in:

```
:1.$.WordWork.book.ch7
```

where:

`:1` is the drive number. This can be 0 for a single drive, and 0 or 1 for a dual drive. Note that the floppy drive numbers are 4 and 5 if the system is equipped with both floppy disc and Winchester drive, and the Winchester is then drive 0.

`$.WordWork.book` is the path name. `WordWork` is a directory within the root directory `$` containing directory `book`, which contains the file `ch7`.

`ch7` is the filename. Filenames can be up to 10 characters and can consist of letters, numbers, and some punctuation characters.

To avoid having to specify a complete list of pathnames for each file, a directory can be selected as the Currently Selected Directory, or CSD, by specifying, for example

```
*DIR WordWork
```

if the CSD is currently `$` or, in full

```
*DIR :1.$.WordWork
```

Initially the CSD is set to `:0.$`. A newly-formatted ADFS disc will contain only directory `$`, the root directory. Before using the View-family programs such as ViewStore, which create files in specific directories, these directories must first be set up with the `*CDIR` command, as explained below.

The method of space allocation used by the ADFS should ensure that `Can't Extend` problems will rarely occur when files are expanded.

There are two approaches to using the View family with the ADFS. One is to keep each type of application on a separate disc. Work files can then be stored in the root directory `$`, or grouped together in directories within the root directory.

Alternatively, some or all of the View-family programs can be used on one ADFS disc, such as a 640K floppy disc or a Winchester disc. In this case, applications relating to different projects are best kept in separate directories. This approach is described in the section 'Integrated View Family' below.

Changing discs

Note that, when using ADFS, you cannot change discs in the way required by DFS when using, for example, a single 100K disc drive. If, after typing a command, you see the message

```
Insert filename disc and hit a key
```

do not attempt to change the disc. Instead, press any key, then check you have spelt the command correctly.

VIEW

All VIEW facilities are available with the ADFS. Simply store the VIEW file anywhere in the directory structure, and set the CSD appropriately.

ViewSheet

ViewSheet requires the existence of a sub-directory `V` for the creation of linking files. This should therefore be created within the directory selected for the ViewSheet applications before creating linking files with the ViewSheet `CREATE` command.

On versions of ViewSheet supplied on an ADFS format disc, directory `$.V` has already been created.

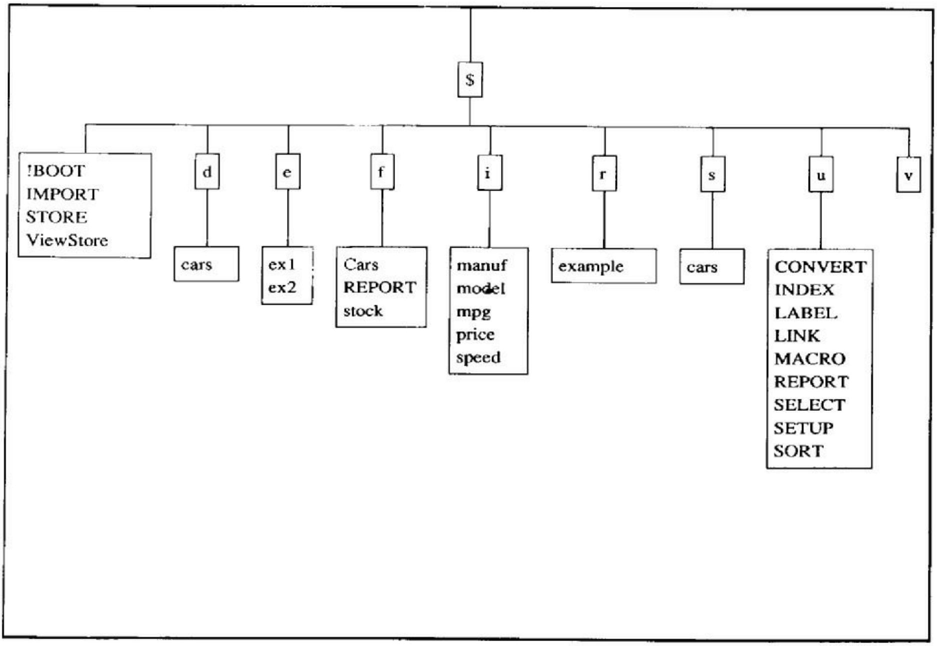
Up to five linking files can be used with one sheet.

ViewStore

If a disc is to be dedicated to database work, you can use the root directory, or create a separate directory such as `Filing`, for all related database work, and set up sub-directories `D`, `F`, `I`, `R` and `S` within this. On a dual-drive system the utilities could be stored on drive 1 in directory `Library.U`. The sub-directories `D`, `F`, `I`, `R`, `S` and `U` must be created with `*CDIR` commands and the CSD set to the chosen directory by typing, for example:

```
*DIR Filing
```

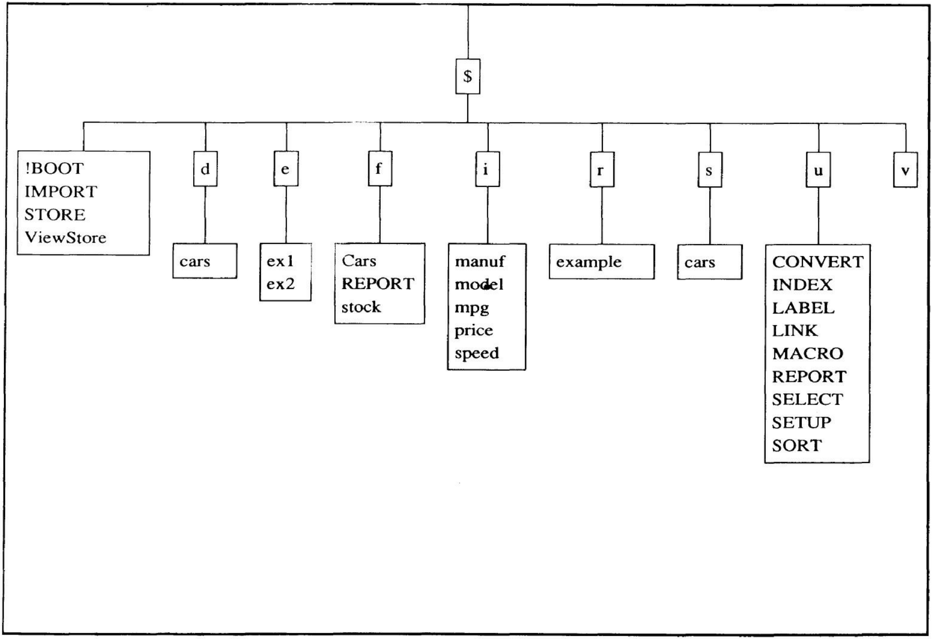
If linking files are to be used, sub-directory `V` will also need to be created. On versions of ViewStore supplied on an ADFS format disc, the sub-directories have already been set up as shown below. In this case, the CSD should be set to `$` and the default `PREFIX` settings can be used.



If a database is likely to become very large it can be put onto a different drive from the other files to give it maximum room. In this case the `PREFIX` command can be used each time ViewStore is entered to set up an appropriate pathname, as follows:

```
PREFIX D :1.
```

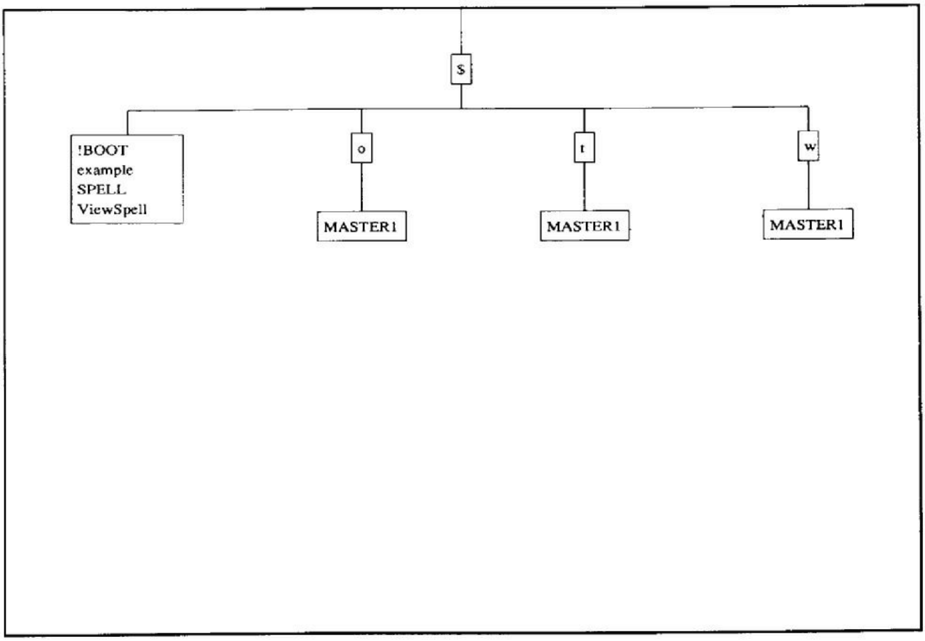
The maximum number of indexes that can be updated on the ADFS is 8, and the maximum length of prefix is 13 characters. Note that the file `F.report` should be included with user format files in the directory `F`.


$
!BOOT
IMPORT
STORE
ViewStore
d
cars
e
ex1
ex2
f
Cars
REPORT
stock
i
manuf
model
mpg
price
speed
r
example
s
cars
u
CONVERT
INDEX
LABEL
LINK
MACRO
REPORT
SELECT
SETUP
SORT
v

ViewSpell

ViewSpell stores dictionary files in sub-directory W. Before copying dictionaries to the filing system, or creating new user dictionaries with the CREATE command, a sub-directory W should be created with the *CDIR command. Sub-directories O and T are also required for the master dictionary.

On versions of ViewSpell supplied on ADFS format disc this directory has already been created, and the directory structure is shown below:

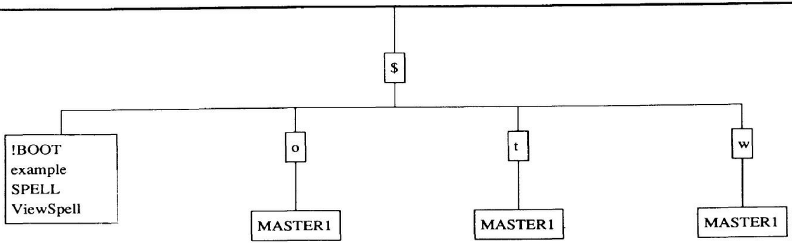


The ViewSpell PREFIX command can be used to set up pathnames for the dictionaries and text files. For example, on a dual-drive system:

```
PREFIX M $.
PREFIX U $.User.
PREFIX T :1.
```

ViewIndex

The ViewIndex manual recommends the use of directories I, M and O to contain the initial index, merged index and output index respectively. On the ADFS, these directories will need to be created before use.

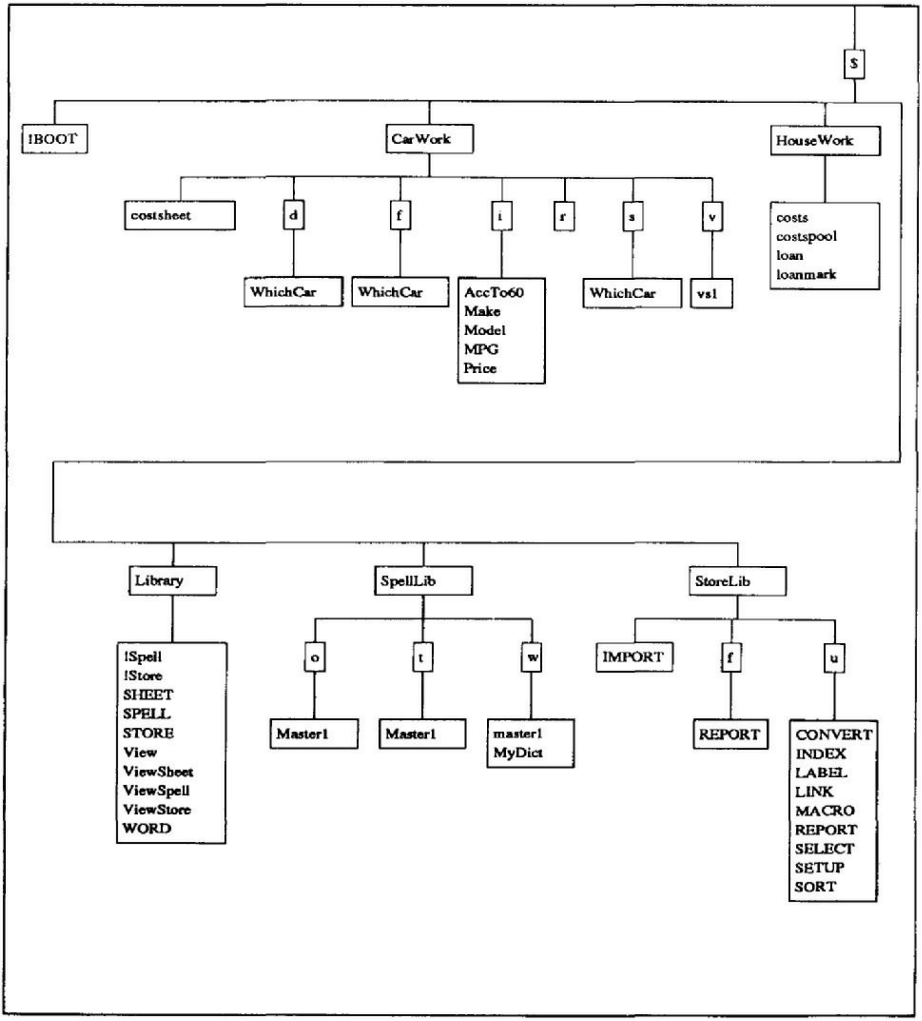

$
!BOOT
example
SPELL
ViewSpell
o
MASTER1
t
MASTER1
w
MASTER1

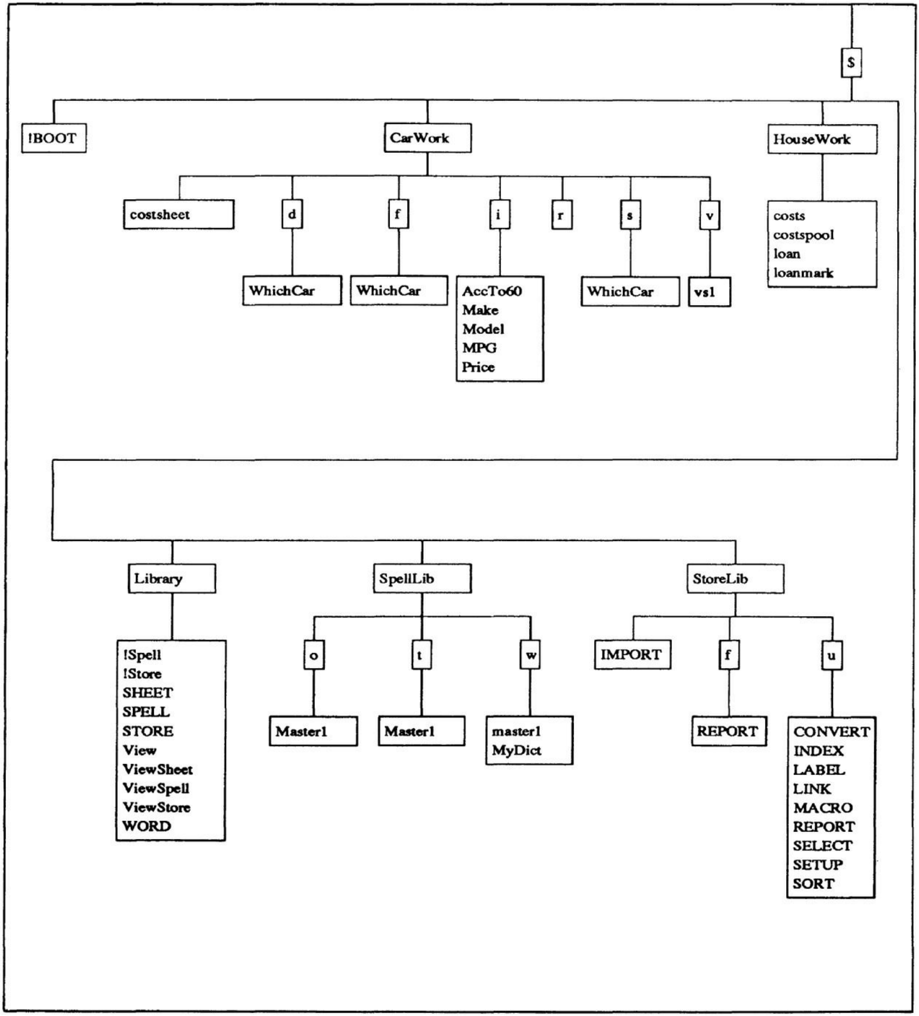
Printer Driver Generator

The Printer Driver Generator program is stored in directory `B`, and assumes that answer files are provided in directory `A`. These directories will already have been created for versions of the program supplied on ADFS format disc; otherwise they will need to be created before use.

Integrated View family

An alternative directory structure, which allows applications for several different View-family programs to be kept on one disc, is shown below.




$
!BOOT
CarWork
HouseWork
costsheet
d
f
i
r
s
v
WhichCar
WhichCar
AccTo60
Make
Model
MPG
Price
WhichCar
vs1
costs
costspool
loan
loanmark
Library
SpellLib
StoreLib
!Spell
!Store
SHEET
SPELL
STORE
View
ViewSheet
ViewSpell
ViewStore
WORD
o
t
w
Master1
Master1
master1
MyDict
IMPORT
f
u
REPORT
CONVERT
INDEX
LABEL
LINK
MACRO
REPORT
SELECT
SETUP
SORT

With this arrangement, each directory is created to hold user files for each separate 'project'. In the diagram, for example, there are 'project' directories called CarWork and HouseWork – perhaps to be used for choosing a car and obtaining an improvement loan respectively. The View-family programs are kept in library directories. With this arrangement, data may be passed easily from one View-family application to another. In the diagram, for example, a linking file `V.VSI` has been used to extract data from a ViewStore database called WhichCar into a ViewSheet file, called CostSheet. If no such transfer of information is required, it may be convenient to keep all database files in one directory, called perhaps StoreWork, all text files in WordWork, and so on. With this arrangement, the corresponding 'Work' directory is made the CSD. In this example this is done by typing

```
*DIR CarWork
```

Within each work directory, the sub-directories required by the View-family program are created, to hold the files of a particular type. Thus database data files are kept in directory `$.CarWork.D`.

The utilities, required by ViewStore and ViewSpell, are stored in Library directories called, respectively, StoreLib and SpellLib. These utilities must be accessed by specifying the complete path of directories, since they are not contained within the CSD. This can be achieved in each case by specifying the full pathname in the ViewStore and ViewSpell `PREFIX` commands.

The following two `EXEC` files, in directory 'library', can then be used to assign prefixes by typing

```
*EXEC $.Library.!Store
```

from within ViewStore, or

```
*EXEC $.Library.!Spell
```

from within ViewSpell.

With suitable configuration, these may be abbreviated to, for example

```
*!Store
```

The file `!Store` in the library directory should be created within VIEW as follows:

```
PREFIX D $.StoreWork.
PREFIX F $.StoreWork.
PREFIX I $.StoreWork.
PREFIX U $.StoreLib.
PREFIX S $.StoreWork.
```

and the file `!Spell` as follows:

```
PREFIX M $.SpellLib.
PREFIX U $.SpellLib.
```

Where the View-family programs themselves are loaded into sideways RAM (as on the Master Compact), these images VIEW, ViewSheet, etc can also be stored in the library directory `$.Library`.

Note that you may need to set the library with the command

```
*LIB Library
```

unless the `DIR` configuration option was set in the startup options.

4 Network Filing Systems (NFS/ANFS)

The Network Filing System (NFS) and Advanced Network Filing System (ANFS) are similar to the ADFS in operation, and are the filing systems available to users of a central file server on the Econet network (see Chapter 3 for more details). This chapter deals with the ANFS, which has superseded the NFS on newer models, although the two are largely similar.

The ANFS root directory contains the user directories, and each user is given a user directory when logging on. For example, after logging on with

```
*I am Jim
```

a typical pathname would be:

```
&.WordWork.book.ch7
```

where:

`&` is the user's root directory (similar to `$` on the ADFS);

`WordWork` is the user's text directory within `&`;

`book` is a directory containing the chapters of the book;

`ch7` is a file within directory `book`.

An alternative full form for the above, which another user might use to access user Jim's files, would be

```
*disc1.$.Jim.WordWork.book.Ch7
```

where:

`disc1` is the name of the disc containing the file server;

`$` is the root directory of the file server;

`Jim` is the user directory within `$`.

However, in this case, the access rights will be 'public' rather than 'owner'.

ViewSheet and ViewStore assume the existence of specific directories; these should therefore be created before using them as described below.

Use of View-family products from file servers may be subject to licensing restrictions.

VIEW

All VIEW facilities are available with the ANFS.

ViewSheet

ViewSheet requires the existence of a sub-directory `V` for the creation of linking files. This should therefore be created with the `*CDIR` command before using the ViewSheet `CREATE` command. Up to five linking files can be used with one sheet.

ViewStore

The ViewStore utilities will normally be placed in a library for all users to use. For example, directory

```
$.StoreLib.U
```

should first be created to hold them. Access to the utility files should be set to `RW/R` to allow all users to read them.

A typical set of prefixes for a user on the ANFS would then be:

```
PREFIX D StoreWork.
PREFIX F StoreWork.
PREFIX I StoreWork.
PREFIX S StoreWork.
PREFIX U $.StoreLib.
```

As an alternative, omit the first four, but set the current directory like this:

```
*DIR StoreWork
```

This has all the files connected with the database listed in sub-directories `D`, `F`, `I`, `R` and `S` of the directory `StoreWork` within the log-on directory. The utilities are in a library directory. Note that the sub-directories must be created before use with the `*CDIR` command.

ViewSpell

The ViewSpell master dictionary should be in the library for everyone to use, with a filename such as:

```
$.SpellLib.W.MASTER1
```

In addition, it requires the files

```
$.SpellLib.T.MASTER1
```

and

```
$.SpellLib.O.MASTER1
```

The access on these files should be set to WR/R so that all users on the network can read them. These files can be accessed by setting up a prefix with the ViewSpell PREFIX command:

```
PREFIX M $.SpellLib.
```

User dictionaries created with the ViewSpell CREATE command will be stored in the user's sub-directory W. Before building a dictionary the user should first create this directory with the *CDIR command. A typical set of PREFIX commands on the ANFS will then be:

```
PREFIX M $.SpellLib.
PREFIX T &.WordWork.
PREFIX U &.SpellWork.
```

ViewIndex and Printer Driver Generator

ViewIndex and Printer Driver Generator directories may need to be created before using ViewIndex or the Printer Driver Generator on the ANFS. See the ADFS section on page 12 onwards.

Integrated View family

The directory structure shown for ADFS on page 16 may also be used under ANFS; library files should be stored in directory $ and user files in the user's directory.