DFU V3.2

Title: DFU Disk & File Utilities for OpenVMS

Revision Information: This manual supersedes the DFU V3.1 manual

Date: October 2006

Operating System: OpenVMS Alpha 7.3-1 and IA64 8.2 or higher

Preface

Intended Audience

This document is intended for system managers, and experienced file system users.

Document structure

This manual consists of the following chapters:

- Chapter 1 is an introduction on DFU and the DFU commands, features and restrictions.
- Chapter 2 describes the output format from DFU and the usage of Screen Management features.
- Chapter 3 gives an overview and introduction to all of the DFU commands.
- Chapter 4 describes the support for Extended File Specifications and ODS5.
- Chapter 5 describes the individual commands and command qualifiers in detail.

Related documents

For related information on the ODS2/ODS5 file system, and OpenVMS features used within this manual, see the following documents:

- *DCL Dictionary*
- *Guide to Extended File Specifications*

Chapter 1
Introduction

The Disk / File utilities is a tool developed to help finding and solving disk, directory and file problems. The usage of low-level VMS-I/O features have resulted in a powerful and high-performance utility. Also, DFU provides many features which are not available with standard DCL commands.

DFU supports all types of disk sets which comply to the OpenVMS ODS-2 and ODS-5 standard, such as volume sets, stripe sets, shadow sets and RAID sets, and combinations of these.

1.1 Command summary

DFU provides the following functions (in alphabetical order) :

- DEFRAG : This function allows simple defragmentation of a file or a list of files.
- DELETE : This function allows either a delete of a single by file-id, or a fast delete of a complete directory or directory tree with all its subdirectories.
- DIRECTORY : This function has 7 options :
    1. COMPRESS a directory
    2. DUMP a directory block by block
    3. REBUILD_MFD rebuilds the Master File directory (000000.DIR)
    4. RECOVER a corrupted directory
    5. Search all directories on the disk for files which have multiple versions
    6. Search all directories on the disk for alias file entries
    7. Search all directories on the disk for empty directories
- INDEXF : This function can analyze, defragment ,extend and truncate INDEXF.SYS. Requires VOLPRO privilege.
- REPORT : Generates a file and free space report for a disk. Also a disk space usage report, based on UIC or Identifier, can be generated, even on a disk which has Disk Quota disabled.
- SEARCH : Look up files on a disk by specific file attributes (eg. files sizes, dates, ownership, and so on)
- SET : This functions allows setting of virtually all possible file attributes.
- UNDELETE : A safe file recovery function.
- VERIFY : This function performs a fast disk structure verification and can optionally repair certain disk structure errors (comparable with ANALYZE/DISK/REPAIR).

Each function is described in detail in the command descriptions chapter.

DFU V3.2 is supported on OpenVMS Alpha V7.3-1 and higher, and OpenVMS Itanium IA64 V8.2 and higher. Starting with DFU V3.0 there is no longer support for the VAX platform.

1.2 Bugfixes in V3.2

This version of DFU offers only bugfixes. The list below gives the changes since DFU version V3.1-1 :

- Workaround for ACCVIO in REPORT and VERIFY.
  Under rare circumstances, accessing the BITMAP.SYS file may cause an ACCVIO. This is a compiler issue (only on Alpha), but DFU now has a workaround which solves it.
- SEARCH .../SINCE and /BEFORE does not correctly report files.
  The way DFU did file date compares was not accurate and could be off by a few minutes. E.g. SEARCH/CREATED=SINCE=09:00 could report files with a creation date of 08:58. This problem is now fixed.
- A wrong fileheader (file name was empty) caused a crash. Routine get_name returns a -1 length, but the file dsc$w_length is unsigned so the comparison with a -1 did not work. Changed it into 0xFFFF.
- Fixed serious issue with VERIFY/REBUILD
  Under rare circumstances we could get disk corruption.
- Fix issues when logical volume size is smaller than device size (maxblocks)

1.3 Support for ODS5 file system features

DFU fully supports the ODS5 file system (also known as the *Extended File Specifications*). The next paragraphs describes the details of using ODS5 features for DFU.

1.3.1 Extended file names

DFU supports the extended (ISO-Latin-1) file naming conventions for ODS5. To work properly, however, the following process setting must be performed prior to using DFU:

```
$ SET PROCESS/PARSE=EXTENDED
```

Failing to do so may result in errors, such as '%CLI-W-PARMDEL, invalid parameter delimiter'.

1.3.2 Using case-sensitive file names

OpenVMS V7.3-1 and higher supports treating of filenames as being case-sensitive. However, DFU does NOT support case sensitive file names. All file names used, or provided on the command line, are treated as case-blind. In fact, DFU will temporary set the process property to case-blind; it will be set back to the permanent process setting on image run-down. For more information on case-sensitive file names see the *OpenVMS Alpha Version 7.3-1 New Features and Documentation Overview*.

1.3.3 Using hardlinks

DFU V3.2 fully supports ODS5 disks with hardlinks enabled. The only restriction is that the command DELETE/DIR/TREE requires SYSPRV privilege on a disk with hardlinks enabled. The command VERIFY/DIR will correctly report linkcount mismatches.

1.4 Restrictions and Parameters

DFU is supported on OpenVMS V7.3-1 and higher on Alpha systems, and OpenVMS 8.1 and higher on Itanium systems. DFU V3.0 and higher is not supported on VAX systems.

The DFU program has a few restrictions :

- Most DFU commands require READ and/or WRITE access to INDEXF.SYS. It is therefore recommended to run DFU commands with BYPASS privilege turned on.
- DFU can not handle volume sets with more than 32 members.
- The INDEXF /DEFRAG, /TRUNCATE and /EXTEND commands can only be performed on offline (dismounted) disks and thus cannot be executed on the system or quorum disk. These commands may require VOLPRO privilege.
- The command DELETE/DIR/TREE requires SYSPRV privilege on a ODS5 disk with hardlinks enabled.

The following minimum process quotas are recommended for DFU usage:

- WSQUOTA : 1500
- WSEXTENT : 3000
- DIOLM : 40
- ASTLM : 40
- FILLM : 40
- BYTLM : 30000
- PGFLQUOTA : 30000

When using DFU on volume sets with a large number of members, it is recommended to increase the SYSGEN parameter CHANNELCNT.

If the DFU directory command is used on very large directory files (> 1000 blocks) the following process parameters may need to be increased:

- WSQUOTA at least 1000 + size-of-largest directory file
- WSEXTENT : WSQUOTA + at least 1000
- CTLPAGES (SYSGEN parameter) : at least 100
- WSMAX (SYSGEN parameter) : at least WSEXTENT

There is no need to change any parameters unless DFU reports errors such as EXCEEDED QUOTA or INSUFFICIENT WORKING SET LIMITS.

1.5 Using an indirect file list

There are 3 commands, DIRECTORY, DEFRAG and SET, which accept an indirect file as input parameter. This is achieved by using the '@' sign. An indirect file is a file containing a list of valid filenames (each line in such a file must contain exactly one filename). Such indirect files can be the output file of the DFU Search command eg.:

```
$ DFU
DFU> SEARCH <disk>/CHAR=DIRECTORY/OUTPUT=x.x
DFU> DIRECTORY/COMPRESS @x.x
```

Chapter 2
DFU output format

DFU uses OpenVMS SMG (Screen Management library routines) to perform output for an interactive DFU session. This chapter describes the layout of the DFU screen, the special commands used only in the SMG output and other items related to SMG.

2.1 Enabling/Disabling SMG

DFU automatically selects SMG output when performed on an interactive DFU session on a terminal or workstation which supports SMG. In a batch job or a non-DEC terminal DFU will use normal line oriented output. SMG output can be explicitly disabled by defining the logical DFU$NOSMG to any value:

```
$ Define Dfu$Nosmg 1
```

2.2 Automatic switching between SMG and Line mode

To perform interactive DFU sessions in SMG mode, and single DFU commands in normal line mode the following command procedure can be used:

```
$! Procedure to switch beteen SMG and line mode
$ DFU = "$DFU"
$ if p1 .eqs. ""
$ then
$!Interactive session
$ define/user sys$input sys$command
$ dfu
$ else
$!Single command so disable SMG
$ define/user dfu$nosmg 1
$ dfu 'p1' 'p2' 'p3' 'p4' 'p5' 'p6' 'p7' 'p8'
$ endif
$ exit
```

2.3 Layout of the SMG screen

After starting DFU the screen should look as follows :

```
+---------------------------------< DFU V3.2 >-------------------------
--+
|
|
|     Disk and File Utilities for OpenVMS DFU V3.2
|
|
|
|     DFU functions are :
|
|
|
|     DEFRAGMENT : Defragment files
|
|     DELETE     : Delete files by File-ID; delete directory (trees)
|
|     DIRECTORY  : Manipulate directories
|
|     INDEXF     : Modify /View INDEXF.SYS
|
|     REPORT     : Generate a complete disk report
|
|     SEARCH     : Fast file search
```

```
|
|     SET         : Modify file attributes
|
|     UNDELETE    : Recover deleted files
|
|     VERIFY      : Check and repair disk structure
|
|
|
|
|
+---------------------------------Statistics-------------------------
--+
|
|
|
|
+-----------------------------------------------------------------------
--+
 DFU>
```

This screen is divided into 3 sections:

- Main Screen : the top part of the screen is used for the output from DFU commands. Also broadcast messages are send to this window.
- Statistics : the lower part contains statistics from the current command. During most of the commands this screen will contain a progress indicator and a status for the current command.
- DFU prompt : the bottom line is used as the input command line.

2.4 Special Screen commands

In SMG mode the following keys perform special functions :

- `[HELP]` or `[PF2]` (PC key `[ / ]`): invokes on-line help
- `[CTRL/W]` : redraws the screen
- `[PF4]` (PC key `[ - ]`) : creates a screen dump into the file DFU_SCREEN.TXT
- `[PREV SCREEN]` (PC key `[Page Up]`) : Scrolls back the output screen.
- `[NEXT SCREEN]` (PC key `[Page Down]`) : Scrolls forward the output screen
- `[INSERT HERE]` (PC key `[Insert]`) : Scrolls back one line
- `[REMOVE]` (PC key `[Delete]`) : Scrolls forward one line
- `[SELECT]` (PC key `[End]`) : Toggles between 80 and 132 columns.

Although only some 20 lines are visible in the output (main) screen, DFU in fact remembers 500 output lines. To scroll back and forward through one screen, use the `[NEXT SCREEN]` and `[PREV SCREEN]` keys, to scroll one line use the `[REMOVE]` and `[INSERT HERE]` keys. At the beginning of each new DFU command these lines are cleared.

DFU commands which produces a lot of output can paginate the output with the /PAGE qualifier.

On a workstation with DECwindows/Motif or a virtual terminal emulator (e.g. PowerTerm) the output size can be changed to be more than the standard 24 lines. DFU's SMG screen will automatically be adjusted to the screen size of such a terminal.

2.5 CTRL/C Handling

There are a few side effects when using SMG mode. CTRL/C and CTRL/Y are catched but not echoed. Also all broadcast messages are trapped and will be displayed in the main screen at the end of the current DFU command. Sometimes this may slightly disrupt the main screen, which can be repaired with the `[CTRL/W]` key sequence.

Further, during the execution of a DFU command the cursor will disappear. After the command finishes the cursor will reappear at the DFU> prompt.

Chapter 3
Overview of DFU functions

This chapter gives a brief overview of the commands which can be used with DFU, and how they can provide the system manager with the necessary information.

One of most common actions by a system manager is trying to locate specific files, eg. files with a certain file size. Although a lot can be done with the VMS DIRECTORY command this can be a time consuming process, especially if the complete disk must be scanned. The SEARCH command of DFU is exactly meant for this situation. It gives a very quick list of specific files, with their size, and if needed the number of file fragments. Some useful SEARCH commands are :

```
DFU> SEARCH disk/SIZE=MINIMUM=1000 ! (files > 1000 blocks)
DFU> SEARCH disk/IDENT=SYSTEM ! (files owned by SYSTEM)
DFU> SEARCH disk/FRAGMENT=MINIMUM=10 ! (files with at least 10
fragments)
DFU> SEARCH disk/CHAR=MARKED ! (all files marked for delete)
DFU> SEARCH disk/CREATED=SINCE=YESTERDAY ! (all files created since
YESTERDAY)
```

Of course these qualifiers can be combined to narrow the search. As the SEARCH command scans INDEXF.SYS, not directories, the output generated by SEARCH may

look a bit random (but it is just the order in which the file headers are in the INDEXF.SYS file). To sort the output the /SORT qualifier should be used.

Another important action for a system manager is to gather some overall disk information, such as number of files, free space statistics and fragmentation information. The REPORT command will generate such a disk report. To get individual disk space usage information the /USAGE qualifier can be used. This will work even if there is no disk quota enabled on the disk.

At regular intervals a system manager may need to check a disk with ANALYZE/DISK. The DFU VERIFY command provides more or less the same functionality but many times faster. The /FIX qualifier provides some REPAIR options which (unlike ANALYZE/DISK/REPAIR) do NOT lock the disk, and thus do not interfere with other disk operations. However, the disk can be rebuilt completely by using the /REBUILD qualifier.

Occasionaly a user will delete the wrong file, leaving the system manager or operator with the tedious task of restoring the file from a backup saveset. The UNDELETE command may be able to recover the file if it has not yet been overwritten with a new one. Much care has been taken to make this command as safe as possible; therefore no disk corruptions will be the result of an UNDELETE. If DFU encounters a problem with a deleted file it will refuse to UNDELETE it.

Another type of typical system manager problems deals with directories. The DFU command DIRECTORY can handle such problems. DIRECTORY can generate reports about empty directories (/EMPTY qualifier) or directories which contain files with a certain number of versions (/VERSION=n qualifier). Also, directories may become internally fragmented. This can be quickly solved with the /COMPRESS qualifier.

A second problem with directories is in deleting a directory with many files. This may take a long time when using the DCL DELETE *.* command. This problem is solved by using DFU's DELETE/DIRECTORY command, which can delete large directories some 10 times faster than the normal DELETE command does. Also, using DELETE/DIR/TREE will delete a complete directory tree with just one command.

The disk's INDEXF.SYS file can sometimes pose special problems. The file system limits the number of fragments for this file. Once this limit is reached any attempt to create new files may result in a fatal SYSTEM-F-HEADERFULL error. The INDEXF/ANALYZE can analyze and report the current state of INDEXF.SYS. If needed the INDEXF/DEFRAG command can defragment INDEXF.SYS whereas INDEXF/EXTEND allows preextending the INDEXF.SYS with a required number of file headers in just one new fragment. Normally a complete BACKUP/RESTORE or a re-INIT of the disk would be necessary to perform these functions.

The last 2 commands deal with individual files. SET allows settings of some special file attributes. Since OpenVMS V6.0 this functionality is also provided by the DCL SET FILE/ATTRIBUTE command, but DFU's SET offers some more options.

The DEFRAG command is an interface to the MOVEFILE function offered since OpenVMS 5.5 and higher. This allows a simple defragmentation of individual files.

The next chapters gives a detailed description of each individual DFU command.

Chapter 4
Support for Extended File Specifications

OpenVMS 7.2 introduces an extension on the current ODS-2 file system, called the Extended File Specifications, or ODS-5. This feature is available only on Alpha and Itanium systems. Also, OpenVMS 7.2 adds support for 255 levels of subdirectories. This is often called deep-directory support. Both features are fully supported by DFU V3.2. There is no need to perform any special action to execute DFU commands on ODS-5 disks. Nevertheless the following points are worth to be considered before using DFU on an ODS-5 disk:

1. It is recommended to add the following command to LOGIN.COM :

   ```
   $ SET PROCESS/PARSE=EXTENDED
   ```

2. This allows easy usage of the new file naming scheme (such as lower-case character, multiple dots in a file name, and so on).
3. File names can become large, either trough the file name, or because there can be 255 levels of subdirectories. A filename longer than 255 characters will be abbreviated into a so-called DID-ed name.

A DID-ed name looks like 'device:[x,y,z]file.ext'. As can be seen the directory string has been replaced by the file id (x,y,z) of the parent directory. For more information on file names under ODS5, see the *OpenVMS Guide to Extended File Specifications* manual.

Chapter 5
Command Descriptions

This chapter describes all DFU commands in detail.

5.1 DEFRAG

This chapter describes the DEFRAG command with the related parameters and qualifiers.

5.1.1 Description

The DEFRAG command can be used to defragment a single file or a group of files. An indirect command file containing a list of files can also be used. With this feature one can create an output file with SEARCH and use that as an input to DEFRAG.

DEFRAG makes use of the file-primitive MOVEFILE function. Therefore files which are open, or which are marked NOMOVE will not be DEFRAGmented.

5.1.2 Error messages

The most common errors returned by DEFRAG are :

- ACCONFLICT: File is open by another user
- DEVICEFULL: Device has not enough contiguous free space to move the entire file
- FILNOTACC: File has been set to NOMOVE.
- FILENUMCHK: Illegal attempt to move a reserved file (eg INDEXF.SYS).
- RMS Errors: Almost always caused by an invalid or wrong file specifcation.

DEFRAG

Defragment a file or a list of files on the disk.

Format

DEFRAG file1,file2,@file...

Parameters

file1,file2,@file...

The file(s) to be defragmented. Also an indirect command file (@file) can be used, which contains a list of files to be defragmented. This may be an output file generated with the SEARCH/OUTPUT command.

Qualifiers

/BESTTRY

Normally DEFRAG tries to create a contiguous file. If there is not enough contiguous free space the operation will fail. With the /BESTTRY qualifier DEFRAG tries to move the file with the contiguous-best-try method. This means a maximum of 3 fragments. This qualifier could be used if a normal DEFRAG command fails. If the DEFRAG/BESTTRY also fails, this is an indication that the disk's free space is too fragmented. Run a REPORT to see what the largest contiguous free space is, and check the free space fragmentation index.

/DEVICE

The device on which the files reside. The device name is added to the file name. When a file list was generated with the DFU SEARCH command, this qualifier is not needed. Example:

```
DFU> SEARCH mydisk/...../frag=min=50/output=x.x
DFU> DEFRAG @x.x
```

/LBN=logical-block-number

Forces a single file to be moved to the specified logical block number. Do not use this qualifier when processing a list of files.

/STATISTICS

This qualifier displays the performance statistics: CPU time, Elapsed Time, I/O and PageFaults.

/WRITECHECK

Perform a writecheck on the resultant file. The default is /NOWRITECHECK. Note that a writecheck doubles the number of involved disk I/O's.

5.2 DELETE

This chapter describes the DELETE command with the related parameters and qualifiers.

5.2.1 Description

The DELETE command is designed for 2 purposes:

1. Delete a file by file id (with the /FILE qualifier). This function is needed to be able to delete files which no longer have a parent directory. This way 'lost' files and (sometimes) files marked for delete can be removed from a disk.
2. A fast method to delete a complete directory or a directory tree (with the /DIRECTORY and /TREE qualifier). First all entries of a directory are deleted without removing the directory entry; this saves a lot of unnecessary directory updates. Finally the directory file itself is deleted, unless the /KEEP qualifier is used. This way a directory is cleaned up many times faster compared with the DCL DELETE *.* command. On large directories DFU may be even 10 or more times faster than DCL. To delete a complete directory tree use the /TREE qualifier. The DCL command DELETE [.SUBDIR...]*.*;* will almost always fail to delete the intermediate subdirectory files, which forcing one to issue this command several times. The DFU DELETE command is smart enough to delete the files and the subdirectory files in the proper order; therefore only one command is sufficient to delete a complete directory tree.
   The /KEEP qualifier allows to preserve the directory tree. The contents are deleted but the directory file(s) remains intact.
   If a file cannot be deleted for some reason (such as a file access conflict) the file will stay at its place together with the parent directory.

DELETE

Deletes files or directories.

Format

DELETE device[:] or directory-file(s)

Parameters

device

The device on which to perform the DELETE/FILE=file-id command.

directory-file(s)

The filespecification of the directory on which to perform the DELETE/DIRECTORY command. Wildcard file specifications may be used. The format must be disk:[directory]XXX.DIR where XXX.DIR is the directory to be deleted. DFU will automatically add .DIR to the file-specification if a filetype is not provided.

Qualifiers

/DIRECTORY

Specify the directory file which must be deleted. Wildcards may be used but must be used very carefully. Make sure that the directory does not contain non-empty subdirectories. Note that the directory file itself will also be deleted, unless the /KEEP qualifier is used.

Example:

```
$ DIR EXAMPLE

Directory MYDISK:[RUBBISH]

EXAMPLE.DIR;1

$ DFU
DFU> DELETE/DIRECTORY EXAMPLE.DIR

%DFU-I-CLEANUP, Deleting MYDISK:[RUBBISH]EXAMPLE.DIR;1...
%DFU-S-DELETED, File DBGINI.COM;3 deleted
%DFU-S-DELETED, File ICON.COM;1 deleted
%DFU-S-DELETED, File I_FOOL.COM;1 deleted
%DFU-S-DELETED, File LOGICALS.COM;51 deleted
%DFU-S-DELETED, File LOGIN.COM;79 deleted
```

```
%DFU-S-DELETED, File LOGOUT.COM;4 deleted
%DFU-S-DELETED, File NOTE_COMPRESS.COM;3 deleted
%DFU-S-DELETED, File SHOWCL.COM;3 deleted
%DFU-S-DELETED, File SYMBOLS.COM;89 deleted
%DFU-S-DELETED, File TPUBUILD.COM;5 deleted
%DFU-S-TOTAL, 11 file(s) deleted
%DFU-S-DELETED, File EXAMPLE.DIR;1 deleted
```

/FILE=file-id

Specify a complete file-id (num,seq,rvn) of the file which must be deleted. This qualifier allows the deletion of lost files or files marked for delete. Example :

```
$ DFU
DFU> VERIFY MYDISK:

%DFU-S-CHKHOME, Home block info verified OK
%DFU-I-IFSCAN, Scanning INDEXF.SYS...
%DFU-E-INVBAKFID, file (86,17915,1) 0004CF04$BFS.;1 has invalid
backlink
%DFU-W-DELETED, file (537,2878,1) APPL_UTRYIT_BOOK.TMP;1 marked
for delete
 .
 .
 .

DFU> DELETE MYDISK/FILE=(86,17915,1)
%DFU-S-DELETED, File 0004CF04$BFS.;1 deleted

DFU> DELETE MYDISK/FILE=(537,2878,1)
%DFU-S-DELETED, File APPL_UTRYIT_BOOK.TMP;1 deleted
```

/KEEP

Preserve the directory tree; only delete the contents. /KEEP is only valid with the /DIRECTORY qualifier. /KEEP works for DELETE/DIR and DELETE/DIR/TREE.

/NOLOG

Do not log succesfull deletes to SYS$OUTPUT. /LOG is the default.

/NOREMOVE

Can only be used with /FILE. Default DELETE/FILE will try to remove the file from the parent directory. If the file's backlink no longer points to

a valid directory this will generate an error. The /NOREMOVE qualifier overrules this behaviour allowing such files to be deleted.

/STATISTICS

This qualifier displays the performance statistics: CPU time, Elapsed Time, I/O and PageFaults.

/TREE

Can only be used with /DIRECTORY. The delete command will delete all subdirectories within the directory file specified. DFU will first sort the subdirectory tree and then delete the files in the proper order.

5.3 DIRECTORY

This chapter describes the DIRECTORY command with the related parameters and qualifiers.

5.3.1 Introduction

The DIRECTORY command offers 7 major functions. 3 functions deal with all directories on a disk, the other 4 functions are used for manipulating individual directories. These functions are activated by special command qualifiers:

- Individual directory functions :
    1. /COMPRESS : Rebuild one or more directory files. Directories are alphabetically sorted lists of filenames and file id's. Because file names are randomly inserted and deleted, holes will occur in directory files. Such holes will only be reused if a file can be alphabetically fitted into it. As a result directory files will grow. The COMPRESS function will rebuild the complete directory file. The /TEST qualifier can be used to preview the results of a COMPRESS command, without making the change to the directory file.
    2. /DUMP : Make a dump of the directory blocks and formats the output. This option is useful for debugging directory files which are suspected or corrupted. This function is equivalent to the previously undocumented VMS command DUMP/DIRECTORY, although the layout of the output is different.
    3. /REBUILD_MFD : Rebuild a corrupted disk's master file directory ([000000]). This can also be used if the MFD is entered in another directory in stead of 000000.DIR.
    4. /RECOVER : Completely rebuild a directory. This option should only be used to recover corrupted directories. During the recovery procedure the directory will be inaccesible for some time. Therefore this command

should NEVER be used on critical system directories. (DFU refuses this command on directory names containing the string 'SYS' to avoid accidental usage).

- Disk-wide directory functions :
    1. /ALIAS : Scan all directories on the disk for alias file entries.
    2. /EMPTY : Scan the disk for empty directories.
    3. /VERSION=n : Scan all directories on the disk for files which have at least 'n' versions.

The diskwide function qualifiers can be used in one command; other combinations of qualifiers are not possible. Note that the diskwide directory functions can take several minutes to complete because a complete directory scan uses a lot of disk I/O (approximately 2 I/O's per directory).

An indirect command file containing a list of files can be used as a parameter to /COMPRESS or /DUMP. Such a list can be generated with the SEARCH command. Example:

```
DFU> SEARCH disk/FILE=*.DIR/SIZE=MIN=50/OUT=dir.lst
DFU> DIRECTORY/COMPRESS @dir.lst
```

Using the /COMPRESS function on critical system files is not recommended. The directory is not accessible for a short time during the execution of the compresss command; also the directory may be left corrupted, should the system crash in the middle of the compress command.

5.3.2 Discussion of directory compression

The DIRECTORY/COMPRESS function will improve performance on large directories. There are some options which can further enhance performance :

- /TRUNCATE: When a directory is compressed , the free space will be at the end of the directory. Therefore , if new files are added VMS is capable of using this space, and avoid a directory entension. The /TRUNCATE qualifier overrules this behaviour by immediately truncating the free space back to the disk.
- /FILL_FACTOR: Normally DFU tries to compress as much as possible. However, it may be better to deliberately create extra free space into each directory block. This can be done using the /FILL_FACTOR=n qualifier. Eg: a fill factor of 50% will result in approximately half of each block being free. Note

that a fill_factor lower tha 100% may result in an error DFU-E_EXTERR; in that case a higher factor must be choosen.

So the following approach should be used when compressing directories:

1. Inactive directories : use DIR/COMPRESS/TRUNCATE.
2. Not very active directories : use DIR/COMPRESS (/TRUNCATE=n).
3. Active directories, files added at the end (such as MAIL directories) : use DIR/COMPRESS without /TRUNCATE
4. Active directories, files added in random alphabetical order : use DIR/COMPRESS/FILL_FACTOR=n, n between 50 and 75.

5.3.3 Output formatting

The qualifier /FORMAT can be used with DIRECTORY/VERSION. This allows the build up of a command procedure directly from the output generated by the DIR/VERSION command. /FORMAT has the following restrictions :

1. /FORMAT is only valid with the /VERSION and the /OUTPUT qualifier.
2. The format string used must contain the !AS directive (in uppercase). The file found will be substituted at the !AS location

Example:

```
DFU> DIR/VERSION=4/OUTPUT=PURGE.COM/FORMAT="$PURGE/KEEP=3 !AS" mydisk
```

DIRECTORY

Performs directory functions.

Format

DIRECTORY device[:] or directory-file(s)

Parameters

device

The device on which to perform one of the diskwide directory functions.

directory-files(s)

The file specification of the directory on which to perform the /COMPRESS, /DUMP or /RECOVER function. /REBUILD_MFD requires a device name only (assumes 000000.DIR implicitly). Wildcard file specifications may be used. DFU will automatically add .DIR to the filespecification if a filetype is not provided.

Qualifiers

/ALIAS

The /ALIAS qualifier directs DFU to scan all directories for alias file entries on a disk. Normally only the system disk should contain alias files. This qualifier can be combined with /VERSION=n and /EMPTY.

/COMPRESS

Performs the directory compression unless /TEST is also specified. The output will show the results in terms of file sizes. Example:

```
DFU> DIR/COMP OWN$:[000000]MAIL

%DFU-S-DONE, OWN$:[000000]MAIL.DIR;1: 31 files; was : 4/9, now :
3/3 blocks

DFU>
```

/DEVICE

The device on which the files reside. The device name is added to the file name. If a file list is generated with the DFU SEARCH command this qualifier is not needed. This qualifier can only be used in combination with /COMPRESS or /DUMP.

/DUMP

Produce a block level dump of a directory. /DUMP will interpret the directory entries found in each block. A directory entry contains a size, version-limit, type and name field. Next the directory entry contains a list

of versions and file ID's for this entry. /DUMP will produce a rather low level output of this information. Example :

```
DFU> DIR/DUMP C.DIR

DUMP of directory block 1
Size: 22, Version limit: 3, Type: ODS-2, Name(10): CHKDSK.EXE
   Version: 3, FID : (1027,1456,0)
Size: 26, Version limit: 3, Type: ODS-2, Name(13): CHKDSK.README
   Version: 2, FID : (33892,1171,0)
   Version: 3, FID : (1256,11234,0)
Size: 22, Version limit: 3, Type: ODS-2, Name(10): CHKDSK.SAV
   Version: 2, FID : (13947,100,0)
Size: 24, Version limit: 3, Type: ODS-2, Name(11): CHKDSK.SAVE
   Version: 1, FID : (12531,2114,0)
Size: 26, Version limit: 3, Type: ODS-2, Name(14): CHKDSK_LNK.COM
   Version: 2, FID : (12314,72,0)
Size: 18, Version limit: 3, Type: ODS-2, Name(5): C_D.C
   Version: 3, FID : (32650,32,0)
Size: 20, Version limit: 3, Type: ODS-2, Name(7): C_D.EXE
   Version: 9, FID : (2675,64,0)
Size: 20, Version limit: 3, Type: ODS-2, Name(7): C_D.OBJ
   Version: 4, FID : (9835,117,0)
Size: 20, Version limit: 3, Type: ODS-2, Name(7): C_D_2.C
   Version: 15, FID : (12428,84,0)
Size: 22, Version limit: 3, Type: ODS-2, Name(9): C_D_2.OBJ
   Version: 9, FID : (8439,119,0)
Size: 22, Version limit: 3, Type: ODS-2, Name(9): C_D_3.CLD
   Version: 3, FID : (9777,178,0)
Size: 22, Version limit: 3, Type: ODS-2, Name(9): C_D_3.OBJ
   Version: 2, FID : (9737,161,0)
Size: 20, Version limit: 3, Type: ODS-2, Name(7): C_D_4.C
   Version: 3, FID : (10594,1580,0)
Size: 22, Version limit: 3, Type: ODS-2, Name(9): C_D_4.OBJ
   Version: 2, FID : (10177,109,0)

DUMP of directory block 2
Size: 18, Version limit: 3, Type: ODS-2, Name(5): DFU.C
   Version: 169, FID : (31123,52,0)

 .
 .
 .
%DFU-I-TOTAL, OWN$:[SOURCES.FORTRAN.DFUSRC]C.DIR;1: 42 files
```

/EMPTY

The /EMPTY qualifier will produce a list of empty directories. This qualifier can be combined with /ALIAS and /VERSION=n.

/FILL_FACTOR=n

Specifies ,in percentage, the filling of directory blocks during a /COMPRESS operation. Fill factor may be between 50 and 100. Omitting this qualifier is the same as /FILL_FACTOR=100 (maximum compression). If the directory has not enough allocated space a low fill factor may result in failure of the Compress operation. DFU will issue a DFU-E-EXTERR error, and the directory file will not be modified.

/FORMAT=format-string

Create an output file in a format described by the format string. The string must contain the !AS directive (this must be uppercase). At the !AS location the resultant filename will be filled in. The /OUTPUT qualifier is required.

/OUTPUT=filename

This qualifier redirects the output to a file. The output will also go to SYS$OUTPUT.

/REBUILD_MFD

Syntax : DIRECTORY/REBUILD_MFD <device> This qualifier causes the master file directory (000000.DIR) of the device to be rebuild and entered in itself.

/RECOVER

This qualifier rebuilds a corrupted directory. Do not use this command on critical or active directories.

/STATISTICS

This qualifier displays the performance statistics: CPU time, Elapsed Time, I/O and PageFaults.

/TEST

The /TEST can only be used in conjunction with /COMPRESS. The qualifier will show the results of the /COMPRESS but will not compress the directory file.

/TRUNCATE=blocksize

During a compress operation truncate the directory back to the blocksize specified. If blocksize is not specified, the file will be truncated back to

the end-of-file block number. Truncation will always be rounded up to the next highest multiple of the disk's cluster size.

/VERSION=n

The /VERSION qualifier directs DFU to scan all directories on the device to produce a list of files which have at least 'n' versions. This qualifier can be combined with /ALIAS and /EMPTY.

5.4 INDEXF

This chapter describes the INDEXF command with the related parameters and qualifiers.

5.4.1 Description

The INDEXF command can be used to manipulate the disk's INDEXF.SYS file. There are 4 options which can be invoked with the appropiate qualifiers :

- /ANALYZE (default) : report the fragments and mapping pointers for INDEXF.SYS. Also report the largest contiguous free space on the disk.
- /DEFRAG : defragment INDEXF.SYS
- /EXTEND=n : extend INDEXF.SYS with 1 new fragment of 'n' blocks; this is equivalent to 'n' new file headers.
- /TRUNCATE : reduces the size of INDEXF.SYS ; this can only be done if the allocated size is larger than the end-of-file size.

The INDEXF command is a very powerfull tool which can be used for solving some classic ODS2 problems (such as the SYSTEM-F-HEADERFULL error). It will save an image BACKUP/RESTORE operation or even a re-INIT of the disk. There are however some restrictions when using this command. Also issueing this command on a volume or shadow set can only be done after some preparation.

INDEXF/ANALYZE can always be performed on a on-line, mounted disk because it is a read-only function. The /DEFRAG, /EXTEND and /TRUNCATE option however requires that the disk is correctly dismounted from all systems in the cluster. Also VOLPRO privilege is required to execute these options. The procedure to defrag, extend or truncate INDEXF.SYS is as follows :

1. Perform an ANALYZE/DISK/REPAIR first to make sure that there are no structure errors on the disk
2. DISMOUNT the disk cluster wide (with /NOUNLOAD). In case of a volume or shadow set the complete set must be dismounted.
3. Issue the INDEXF/DEFRAG, /EXTEND=n or /TRUNCATE command for the disk. The syntax is : DFU> INDEXF/DEFRAG device: On a shadow set one must specify the virtual unit as the device: parameter and use the

/SHADOW_MEMBER=device: qualifier to specify ONE physical member of this set.

4. DFU will remount the disk privately and determine if the command can be executed.
5. If the command can be executed DFU will prompt for a confirmation.
6. If the command is confirmed DFU will remount the disk /FOREIGN and start with the operation. After completing all operations the disk will be dismounted.
7. The system manager must now manually remount the disk. In case of a volume set the complete set must be remounted. In case of a shadow set remounting the complete set will result in a correct shadow copy operation on the other members of the set.

There are some restrictions which will limit the use of the /DEFRAG, /EXTEND and /TRUNCATE options :

- The command can not be executed on the system disk
- There must be enough contiguous free space on the disk to be able to perform /DEFRAG or /EXTEND. If there is not enough space DFU will report it and cancel the operation.
- The INDEXF.SYS header must have enough space left to be able to execute the /EXTEND command. If not DFU will report a HEADERFULL error and advise to perform a /DEFRAG operation first.
- INDEXF.SYS cannot be extended beyond the MAXFILES parameter set for the disk. A $SHOW DEVICE /FULL command will report the maximum files allowed for the disk.
- The first 3 or 4 fragments of INDEXF.SYS (depending on the geometry) can never be moved. If there are not enough fragments available to be defragmented DFU will report a NOOPT warning and refuse to perform a DEFRAG operation.

5.4.2 Example of a DEFRAG operation

Below follows an example of a disk which is defragmented using the /DEFRAG qualifier (the /EXTEND qualifier is almost the same):

```
DFU> index/defrag DUA1:
%DFU-I-MOUNTING, Busy mounting disk DUA1:...
%DFU-I-ANALDISK, Analyzing INDEXF and BITMAP...
%DFU-I-TOTAL, Maparea maps 326 blocks in 9 fragments (11% used)
(1)
%DFU-I-FINDLBN, Largest free contiguous space 2351 blocks at LBN 2649
(2)
%DFU-I-MOVE, 305 blocks can be defragmented (5 fragments)
(3)
Continue to modify INDEXF.SYS ? (Y/N) [N] : y
%DFU-I-MOUNTFOR, Busy remounting disk LDA1: /FOREIGN...
%DFU-I-STARTDFR, Now copying fragments to new location...
```

```
%DFU-S-COPIED, 185 blocks copied (fragment 5)
(4)
%DFU-S-COPIED, 30 blocks copied (fragment 6)
%DFU-S-COPIED, 30 blocks copied (fragment 7)
%DFU-S-COPIED, 30 blocks copied (fragment 8)
%DFU-S-COPIED, 30 blocks copied (fragment 9)
%DFU-I-NEWTOTAL, New Maparea maps 326 blocks in 5 fragments
%DFU-S-REWRTIF, INDEXF.SYS File header rewritten !
(5)
%DFU-I-RBDBITMAP, Updating BITMAP.SYS...
(6)
%DFU-S-READY, all operations succesfully completed
(7)
%DFU-I-DISMNT, Volume dismounted
```

The meaning of these messages is as follows :

1. DFU reports the number of fragments and the percentage of space used in the file header of INDEXF.SYS. These figures can also be produced with the /ANALYZE qualifier.
2. DFU reports the largest number of contiguous free blocks on the disk.
3. DFU now proposes the largest chunk of INDEXF.SYS which can be defragmented into one new fragment.
4. After remounting the disk /FOREIGN DFU starts copying the individual fragments.
5. After all copy operations have been completed the new file header is rewritten.
6. Next BITMAP.SYS will be modified to reflect the new storage situation.
7. This message indicates that DFU has succesfully finished the operation.

5.4.3 Crash Recovery

During a DEFRAG, EXTEND or TRUNCATE operation there is a very small time interval in which a system crash or disk failure may result in a situation which requires manual intervention. This interval exists after DFU has rewritten the INDEXF.SYS new file header and before DFU has completed the rebuild of BITMAP.SYS. By taking a careful look at the log produced by DFU it is easy to determine which action to take. The critical interval exists after DFU issued the "%DFU-S-REWRTIF, INDEXF.SYS File header rewritten" message and before a "%DFU-S-READY, all operations succesfully completed" message has been reported. If the system or disk fails somewhere between this interval the disk must be remounted and immediately repaired with DFU> VERIFY/REBUILD (a ANALYZE/DISK/REPAIR or a SET VOLUME/REBUILD=FORCE command will do as well). Failing to do so may result in a corrupted disk.

Any error, system or disk failure outside this interval does not require further action (apart from manually remounting the disk).

5.4.4 Disclaimer

Despite careful testing on several disk types and configurations HP cannot absolutely garantuee that defragmenting or extending INDEXF.SYS will not result in a corrupted disk. Therefore it is strongly recommended that a defragment or extend operation only be performed on a disk if a valid and recently made disk backup is available.

INDEXF

Analyze, Defragment or Extend INDEXF.SYS

Format

INDEXF device[:]

Parameters

device[:]

device which holds the INDEXF.SYS file

Qualifiers

/ANALYZE

Displays information about the number of fragments in INDEXF.SYS and the largest contiguous free space. If INDEXF.SYS can not be defragmented a "%DFU-W-NOOPT" message will be displayed.

/DEFRAG

Starts a defragment operation on INDEXF.SYS. The disk must be clusterwide dismounted.

/EXTEND=n

Extends INDEXF.SYS with 'n' blocks in 1 new fragment. This is equivalent to 'n' extra file headers. The disk must be clusterwide dismounted. 'n' will be rounded up to be a multiple of the disk's cluster size.

/SHADOW_MEMBER=device:

The physical member of the shadowset on which to perform the EXTEND or DEFRAG operation. This qualifier is only required in combination with /DEFRAG and /EXTEND. Example :

```
DFU> INDEXF/DEFRAG DSA0:/SHADOW_MEMBER=$1$DUA104:
```

/SHOW_POINTERS

Displays all mapping pointer information. See example :

```
DFU> index/analyze/show $1$dua11
%DFU-I-ANALDISK, Analyzing INDEXF and BITMAP...
%DFU-I-MAPPTR, Retrieval ptr ( 1) Size :       6 , LBN :       0
%DFU-I-MAPPTR, Retrieval ptr ( 2) Size :       3 , LBN :    1023
%DFU-I-MAPPTR, Retrieval ptr ( 3) Size :       3 , LBN : 1498254
%DFU-I-MAPPTR, Retrieval ptr ( 4) Size :   27534 , LBN : 1470720
%DFU-I-MAPPTR, Retrieval ptr ( 5) Size :    1002 , LBN :  345642
%DFU-I-MAPPTR, Retrieval ptr ( 6) Size :    1002 , LBN : 1016796
%DFU-I-MAPPTR, Retrieval ptr ( 7) Size :    1002 , LBN : 1467444
%DFU-I-MAPPTR, Retrieval ptr ( 8) Size :    1002 , LBN : 1925727
  .
  .
  .
%DFU-I-MAPPTR, Retrieval ptr (19) Size :    1002 , LBN :  544224
%DFU-I-MAPPTR, Retrieval ptr (20) Size :   10215 , LBN : 2565624
%DFU-I-TOTAL, Maparea maps 52791 blocks in 20 fragments (37%
used)
%DFU-I-FINDLBN, Largest free contiguous space 9132 blocks at LBN
1648761
%DFU-I-MOVE, 9018 blocks can be defragmented (9 fragments)
```

/STATISTICS

This qualifier displays the performance statistics: CPU time, Elapsed Time, I/O and PageFaults.

/TRUNCATE

This qualifier truncates the INDEXF.SYS file back to the end-of-file size. Thus over-allocated blocks can be recovered.

5.5 REPORT

This chapter describes the REPORT command with the related parameters and qualifiers.

5.5.1 Description

The REPORT command generates a report of the file and free space fragmentation of the disk. Also a graph may be generated by the /GRAPH qualifier; this visualises the free space distribution on the disk. The /USAGE qualifier will generate a diskspace usage report. This is especially useful if DISKQUOTA is not enabled on the disk.

The default report contains information of the volume, files and the bitmap. This output can be suppressed by the /NOVOLUME, NOFILE and NOBITMAP qualifiers. Also the /USAGE qualifier can be used for displaying information of a specific UIC or Identifier simply by specifying /USAGE=<uic> or /USAGE=<identifier>.

5.5.2 Syntax and Output

The syntax of the report command is:

```
DFU> REPORT device/QUALIFIERS
```

Below follows an example and output of the report command:

```
DFU> REPORT USER3:/GRAPH/USAGE

%DFU-I-REPORT, Reporting on USER3: ($1$DUA102:)

  ***** Volume info for USER3: (from HOME block) *****
 Volume name                     :  USER3
 Volume owner                    :  SYSTEM
 Volume set name                 :
 Highwater mark. / Erase on del.  :  No / No
 Clustersize                     :  3
 Maximum # files                 :  367618
 Header count                    :  51547 (1)
 First header VBN                :  103
 Free headers                    :  10951 (2)

  ***** File statistics (from INDEXF.SYS) *****
 INDEXF.SYS fragments /map_in_use :  21 / 61 (39% used)   (3)
 Total files (ODS2 / ODS5)       :  40595 / 0
 Empty files                     :  177
```

```
Files with allocation            :  40418
Files with extension headers     :  0
Files marked for delete          :  19 , size : 25/63 blocks
Directory files                  :  1957
Contiguous files                 :  40326 (4)
Total size used /allocated       :  2492431 /2533209 blocks
Total headers / fragments        :  42345 / 40770
Average fragments per file       :  1.009  (5)
File fragmentation index         :  0.692 (excellent) (6)
Average size per fragment        :  62 blocks
Most fragmented file             :
   $1$DUA102:[LARGEUSER]A.DAT;1 ( 180/630 blocks; 30  fragments)

 ***** Free space statistics (from BITMAP.SYS) *****
Total blocks on disk             :  2940951
Total free blocks                :  407742
Percentage free (rounded)        :  13
Total free extents               :  260
Largest free extent (blocks)     :  42912 at LBN: 2463597 (7)
Average extent size (blocks)     :  1568
Free space fragmentation index   :  0.191 (excellent) (6)


    LBN   +-----------------------------------------------------------
-+   %
      0-
>|***....**********.***********.******...*********..********|  6%
   208360-
>|************************************************************| 13%
   416721-
>|************************************************************| 20%
   625081-
>|*****************************************************.*****| 26%
   833442-
>|*******..*********************************..****************| 33%
  1041802-
>|******.**.*********...********.*****.*******...*.**..*.***..| 40%
  1250163->|.*..*****************....*.....**.*. ..*****.***..****.
..*| 46%
  1458523->|**********.
.****************************************.**.| 53%
  1666884->|.**...  .......
.*******...**********.*..***********..******| 60%
  1875244->|***..  ....*...**********.   ..*********.
.**************| 66%
  2083605->|*********************************.*****..   .****... . ....
| 73%
  2291965->| ..*******....
......************************************| 80%
  2500326->|****..*****.*****.**.
.********...*.*****..*******.*..*| 86%
  2708686->|.*******..*******.***..*.****************.*.
.*...**********| 93%
  2917047->|..**.    .*********************.******..  .*******.
|100%
          +-----------------------------------------------------------
```

```
-+
 * : Fully allocated, . : Partial allocated, <space> : Free, 3472
blocks each

 Free space distribution, each * =  2100  free blocks

  ***** Disk Usage Table (from INDEXF.SYS and QUOTA.SYS) *****
 Identifier/UIC                Used/Allocated   Headers   Quota
Used/Perm
 ------------------------------------------------------------------------
---
 [SSG,MRX]                      42079/46356      2259      48615/60000
 [SSG,TESTJE]                  189306/254742     4935
259677/300000
 [SSG,LARGEUSER]                47961/50238      1648      51886/65000
 [SYSTEM]                      499554/505626     4954
510580/2000000
 .
 .
 .
```

The various items have the following meaning :

1. The header count is calculated based on the size of INDEXF.SYS. If more headers are needed INDEXF.SYS has to be extended.
2. The free headers gives the number of free entries in INDEXF.SYS before it has to extend.
3. The INDEXF.SYS number of fragments and mapwords in use are very important figures to determine if the INDEXF.SYS file can be extended. The theoretical maximum number of mapwords is 155. This value may be less if there are ACL's on INDEXF.SYS. Each fragment, and therefore each retrieval pointer must be in the mapword area. As the size of a retrieval pointer depends on the physical size of a disk, it is not easy to say how many fragments INDEXF.SYS can store in the mapwords area. If the mapword area is full the INDEXF file can no longer extend, and creating new files may result in a SYSTEM-W-HEADERFULL error.
4. Files which have exactly 1 retrieval pointer are considered to be contiguous. This doesn't mean that the CONTIGUOUS bit is set in the file header.
5. The average fragments per file give some indication about the total file fragmentation.
6. The file and free space fragmentation index classifies the disk as follows :
    - 0-1 : Excellent
    - 1-2 : Good
    - 2-3 : Fair
    - 3-4 : Poor
    - >4 : Bad

   If the index is greater than 3 one should consider defragmenting the disk using HP's DFO software, or by performing an Image BACKUP / Restore.

7. The largest free extent is a useful figure for defragmentation purposes. A file cannot be defragmented when its size is larger than this value.

When using /GRAPH the disk's free space distribution is shown as a bitmap image. Each position in the graph represents a certain number of blocks (in the example above 3472 blocks). This bitmap image gives a quick impression about the free space distribution on the disk.

If the /USAGE qualifier is used, a sorted usage table will be added. Per Identifier / UIC the blocks used/allocated, and the number of file headers are shown. If Diskquota is enabled a third column will be included which shows the blocks used/permitted allowing to QUOTA.SYS. Normally quota should satisfy the following rule :

```
Quota used = blocks allocated + # of file headers
```

REPORT

Create a file and free space report of a disk device.

Format

REPORT device[:]

Parameters

device[:]

Device to be reported. (May use a logical name)

Qualifiers

/APPEND=filename

This qualifier redirects the output to be appended to an already existing file. If the output file does not exists it will be created. Use /APPEND or /OUTPUT but not both.

/GRAPH

This qualifier generates a graph table which visualizes the free space distribution on the disk. /NOGRAPH is the default.

/NOBITMAP

The /NOBITMAP qualifier suppresses the "Free space statistics" output.

/NOFILE

The /NOFILE qualifier suppressed the "File Statistics" output.

/NOVOLUME

The /NOVOLUME qualifier suppresses the "Volume info" output.

/OUTPUT=filename

This qualifier redirects the output to a file. The output will also go to SYS$OUTPUT.

/STATISTICS

This qualifier displays the performance statistics: CPU time, Elapsed Time, I/O and PageFaults.

/UNIT

- /UNIT=BLOCKS (default)
- /UNIT=BYTES

The file and fragment sizes are reported (default) in units of blocks. To report the sizes in bytes, the /UNIT=BYTES qualifier must be used.

/USAGE(=uic or identifier)

The /USAGE qualifier will generate a diskspace usage report based on Identifiers/UIC. If diskquota is enabled on the disk the blocks used/permitted allowing to QUOTA.SYS will also be shown. /USAGE is very useful when diskquota is not enabled on the disk. When a UIC or identifier is specified only the information for this UIC or identifier will be displayed.

5.6 SEARCH

This chapter describes the SEARCH command with the related parameters and qualifiers.

5.6.1 Description

The search function allows very quick disk-wide searches for specific files. Searches can be performed based on almost any file attribute or attributes. Each attribute has a corresponding qualifier. More than one qualifier can be used to limit the search width.

5.6.2 Syntax and Output

The command syntax for SEARCH is:

```
DFU> SEARCH device/qualifiers
```

The output is displayed on the terminal. It can be sorted with the /SORT qualifier. The output is shown in 2 or 3 columns eg:

```
DFU> SEARCH MYDISK/FILE=X.X/FRAGMENT
 .
 .
$1$DUA102:[USER.COMMAND]X.X;1             1/3               1/1
 .
 .
```

The first column shows the full file name . The device name is included in the file name. The second columns shows the file size as actual/allocated size. The 3rd column is optional and will only be shown when the /FRAGMENT qualifier is used. It shows the number of fileheaders / number of file fragments.

5.6.3 Volume Set processing

SEARCH checks if the device to-be-searched is member of a volume set. If so, the complete volume set will be processed, starting with Relative Volume Number 1 up to the last member in set. This behaviour can be inhibited with the /NOVOLSET qualifier.

5.6.4 Output formatting

The qualifier /FORMAT can be used with SEARCH. This allows the build up of a command procedure directly from the output generated by the SEARCH command. /FORMAT has the following restrictions :

1. /FORMAT is only valid with the /OUTPUT qualifier, and can not be used together with the /FULL or the /SORT qualifiers.
2. The format string used must contain the !AS directive (in uppercase). The file found will be substituted at the !AS location

Example:

```
DFU> SEARCH/OUTP=DEL.COM/FORMAT="$DELETE/CONF !AS"/FILE=*.LOG mydisk
```

SEARCH

Fast file search through INDEXF.SYS.

Format

SEARCH device[:]

Parameters

device[:]

Device to be searched. (May use a logical name)

Description

The SEARCH command is used for quick disk-wide searches for specific files. The qualifiers describe the file attributes used as the search criteria. A combination of almost all of the qualifiers is possible, unless otherwise specified. Eg.:

```
DFU> SEARCH DISK1/SIZE=MIN=10/OWN=[1,4]/CREATED=SINCE=YESTERDAY
```

is a valid command. Note that qualifiers will be used in a logical AND manner, that is, SEARCH will only display files which match all qualifiers and options specified.

Qualifiers

/ACCESS=option(,option)

The /ACCESS qualifier is used to search files depending on their last ACCESS date. The possible options are :

- /ACCESS=BEFORE=date
- /ACCESS=SINCE=date
- /ACCESS=NONE

/ACE=identifier

Search for files which have an ACE containing the specific identifier. Only ACE's of the format "IDENTIFIER=<identifier>,..." will be searched.

/ALLOCATED

Default action for DFU SEARCH is to use the actual file size. The /ALLOCATED qualifier forces SEARCH to use the allocated filesize. This qualifier is only meaningfull if combined with /SIZE.

/APPEND=filename

This qualifier redirects the output to be appended to an already existing file. If the output file does not exists it will be created. Use /APPEND or /OUTPUT but not both.

/ATTRIBUTE=option(,option)

The /ATTRIBUTE qualifier is used to search files depending on their last attribute change date. The possible options are :

- /ATTRIBUTE=BEFORE=date
- /ATTRIBUTE=SINCE=date
- /ATTRIBUTE=NONE

/BACKUP=option(,option)

The /BACKUP qualifier is used to search files depending on their backup date. The possible options are :

- /BACKUP=BEFORE=date
- /BACKUP=SINCE=date
- /BACKUP=NONE

The NONE option gives files which don't have a backup date recorded.The other 2 options can be used to get files which have a backup date before or after a specific date. The BEFORE and SINCE option can be combined in one command.

/BAKFID=backlink-file-id

Use this qualifier to look for a file with a specific backlink file id. Only the first part of the file id must be specified, that is, if the file id is (x,y,z) one must specify 'x' as the file id. This qualifier can be used to get files from a specific directory. If /BAKFID=0 is used, DFU will report files not belonging to a directory (often temporary files).

/CHARACTERISTIC=(char1,char2...)

This qualifier is used to get files with specific file characteristics.The characteristics can have the following values:

- Directory : directory files
- Nobackup : files marked nobackup
- Contiguous : files marked as contiguous
- Erase : erase file before deletion
- Spool : spool files
- Badblock : files which contain suspected bad block(s)
- Badacl : files with a corrupted ACL
- Besttry : files marked contiguous-best-try
- Scratch : files marked as scratch files
- Nocharge : files not charged against quota
- Nomove : files marked NoMove
- Locked : files with the deaccess lock bit set
- Marked : files marked for deletion
- Noshelvable : files which can not be shelved
- Isshelved : files which are shelved

The several characteristics can be combined in one command. When more than one characteristic is specified DFU uses a logical AND, but this can be changed by using the special option MATCH=OR eg:

```
SEARCH disk/CHAR=(NOMOVE,DIR,MATCH=OR)
```

/CREATED=option(,option)

The /CREATED qualifier is used to search files depending on their creation date. The options are :

- /CREATED=BEFORE=date
- /CREATED=SINCE=date

These 2 options can be used to get files which have a creation date before or after a specific date. The BEFORE and SINCE option can be combined in one command.

/EXCLUDE=(file1,file2...)

This qualifier is used to exclude files from the search. Wildcard filenames are allowed.

/EXPIRED=option(,option)

The /EXPIRED qualifier is used to search files depending on their expiration date. The options are :

- /EXPIRED=BEFORE=date
- /EXPIRED=SINCE=date
- /EXPIRED=NONE

The NONE option gives files which don't have a expiration date recorded.The other 2 options can be used to get files which have a expiration date before or after a specific date. The BEFORE and SINCE option can be combined in one command.

/FID=file-id

The /FID qualifier is a special function of SEARCH. It directs SEARCH to go directly to the specified file without processing the rest of the Index file. Therefore, this qualifier can not be combined with other search qualifiers such as /FILE. Normally a file-id consists is in the form (x,y,z) where x is the number of the file header within INDEXF.SYS y is the sequence number, and z is the Relative Volume Number. One only needs to specifiy the x value in the /FID qualifier.

/FILE=(file1,file2...)

This qualifier is used to search for files with a specific file name. Wildcard file names are allowed.

/FORMAT=format-string

Create an output file in a format described by the format string. The string must contain the !AS directive (this must be uppercase). At the !AS location the resultant filename will be filled in. The /OUTPUT qualifier is required.

/FRAGMENT=(MINIMUM=nr,MAXIMUM=nr)

This qualifier displays the number of fragments and file headers of each file found. This can be further delimited by using the MINIMUM=n and MAXIMUM=m option. Example:

```
DFU> disk/FRAG
DFU> disk/FRAG=min=10
DFU> disk/FRAG=(min=10,max=100)
```

/FULL

If this qualifier is used, SEARCH will give a full output for each file found. This output is a look-alike of the output from a DIRECTORY/FULL command. This qualifier can not be combined with /SORT or /SUMMARY.

/HOME

This qualifier directs search to give output from the disk's home block. SEARCH uses the home block for calculating the free and used file headers within INDEXF.SYS.

/(NO)IDENT=identifier or uic

Search for files owned by a specific identifier or UIC. Any valid UIC or identifier format may be used. Another way is the /OWNER_UIC qualifier; these 2 qualifiers cannot be combined in one SEARCH command. Issueing /NOIDENT directs DFU to search for files NOT owned by this IDENTIFIER, eg: DFU> SEARCH sys$sysdevice/NOIDENT=SYSTEM.

/LBN=logical-block-number

The /LBN qualifier is a special function of SEARCH. This allows to find a file which contains a specific LBN. This may be useful if bad blocks are logged in the error log. /LBN can not be combined with other search qualifiers such as /FILE.

/LIMIT=(MINIMUM=n,MAXAXIMUM=m)

Searches DIRECTORY files which have a default version limit between n and m (including n and m). Either MINIMUM or MAXIMUM is required.

/NAME_TYPE=ODS2 | ISL1 | UCS2

Search for files which have the specified name type attribute. ODS2 if for classic VMS file names, ISL1 is for Iso-Latin-1 names and UCS2 is for Unicode names. This qualifier is only meaningfull for ODS5 disks.

/(NO)OWNER_UIC=uic or identifier

This qualifier is used to get files owned by a specific UIC or identifier. Any valid UIC or identifier format may be used. /UIC and /IDENT can not be combined. Specifying /NOOWNER directs DFU to search for files NOT owned by this UIC or identifier.

/NOSEARCH

This qualifier can only be used in combination with /HOME. It is used to get the Homeblock info, without searching the entire disk.

/NOVOLSET

The default behaviour of SEARCH is to process an entire volume set. This can be changed with the /NOVOLSET qualifier. This may be useful when doing a /LBN search.

/MODIFIED=option(,option)

The /MODIFIED qualifier is used to search files depending on their modification date. The options are :

- /MODIFIED=BEFORE=date
- /MODIFIED=SINCE=date

These 2 options can be used to get files which have a modification date before or after a specific date. The BEFORE and SINCE option can be combined in one command.

/MULTIPLE

This qualifier searches for files which have more than 1 file header.

/ORGANIZATION=INDEXED | SEQUENTIAL | RELATIVE | DIRECT

Search for files with the specified RMS file organization.

/OUTPUT=file

This qualifier redirects the output from SEARCH to an output file.This file can later be used as input to the DEFRAG, DIRECTORY or SET command.

/OVER_ALLOCATED=n

Searches for files where the difference between the used and allocated size is at least 'n' blocks.

/PLACED

Checks for files which have a placement control retrieval pointer.This is sometimes set by non-DEC disk defragmentation programs.

/SIZE=(minimum=size1,maximum=size2)

The /SIZE qualifier is used to limit files found by their file size.Use the minimum=n or maximum=m or both. SEARCH will use the actual file size for selecting files, except when the /ALLOCATED qualifier is used.

/SORT

This qualifier will sort the files found before being output.

/STATISTICS

This qualifier displays the performance statistics: CPU time, Elapsed Time, I/O and PageFaults.

/SUMMARY

This qualifier will limit the output to the total number of files found, with their size (and optional the fragmentation when /FRAG is used). /SUMMARY can not be combined with /FULL.

/TYPE=option

The /TYPE qualifier has 2 options : /TYPE=ODS2 or /TYPE=ODS5. This qualifier can be used to search for files with either an ODS2 or an ODS5 file header. This qualifier is only meaningfull on an ODS5 disk.

/VERSION_NUMBER=(minimum=n,maximum=m)

The /VERSION_NUMBER qualifier is used search for files within a range (n thru m) of file version numbers. Logfiles which are produced with the same name may pose problems when they reach the maximum version number (32767). Using /VERSION=MIN=32000 is a quick way to find such files.

5.7 SET

This chapter describes the SET command with the related parameters and qualifiers.

5.7.1 Description

The SET command allows to modify file attributes which can't be modified through DCL commands. This command should be used with care to avoid corrupting files. Most a the DFU SET command options are also available with the DCL command $SET FILE/ATTRIBUTES. Still DFU's SET has a few more options which are not in SET FILE/ATTRIBUTES (such as setting a BACKUP date).

The qualifier /IGNORE=INTERLOCK allows setting file attributes even on open or locked files.

SET

Modify file attributes.

Format

SET file1,file2,...,@file

Parameters

file1,file2,...,@file

The files to be modified. The attributes to be modified are specified with qualifiers. Wildcards area allowed in the filename. An indirect file can be used by using the @ sign. This allows processing of a file list produced by a DFU SEARCH command.

Qualifiers

/ACCESS_DATE=date

/NOACCESS_DATE

Controls whether a new last-access date is assigned to the specified files. Specify the date according to the rules described in Chapter 1 of the VMS DCL Concepts Manual. Absolute date keywords are allowed. If the date is specified as '0' today's date is used. If /NOACCESS_DATE is specified, the last access date field will be cleared.

/ATTRIBUTE_DATE=date

/NOATTRIBUTE_DATE

Controls whether a new last attribute change date is assigned to the specified files. Specify the date according to the rules described in Chapter 1 of the VMS DCL Concepts Manual. Absolute date keywords are allowed. If the date is specified as '0' today's date is used. If /NOATTRIBUTE_DATE is specified, the last attribute change date field will be cleared.

/BACKUP_DATE=date

/NOBACKUP_DATE

Controls whether a new backup date is assigned to the specified files. Specify the date according to the rules described in Chapter 1 of the VMS DCL Concepts Manual. Absolute date keywords are allowed. If the date is specified as '0', today's date is used. If /NOBACKUP_DATE is specified, the Backup date field will be cleared.

/BADACL

/NOBADACL

Sets or resets the 'BADACL' flag in the file header. This enables deletion of a file with a corrupted ACL.

/BCK

/NOBCK

Clears or sets the files NOBACKUP bit. Setting a file to NOBACKUP with /NOBCK causes it to be skipped by a BACKUP operation.

/BUCKETSIZE=size

Sets a new value for the bucket size in the file header.

/CONFIRM

/NOCONFIRM (default)

Controls whether a request is issued before each individual SET operation to confirm that the operation should be performed on that file. When the system issues the prompt, the following responses can be used:

```
YES       NO      QUIT
TRUE      FALSE   CTRL/Z
1         0       ALL
          <RET>
```

QUIT or CTRL/Z issues DFU to stop processing the command at that point. When using ALL as the response, the command continues to process, but no further prompts are given.

/CONTIGUOUS_BEST_TRY

/NOCONTIGUOUS_BEST_TRY

Sets or resets the 'CONTIGUOUS_BEST_TRY' bit in the file header.

/CREATION_DATE=date

/NOCREATION_DATE

Controls whether a new creation date is assigned to the specified files. Specify the date according to the rules described in Chapter 1 of the VMS DCL Concepts Manual. Absolute date keywords are allowed. If the date is specified as '0', today's date is used.

/DIRECTORY

/NODIRECTORY

Sets or resets the directory attribute of a file. This qualifier allows to set the directory bit of a file which was mistakingly reset by the 'SET FILE/NODIRECTORY' command. If it is done on a non-directory file, then access to that directory will give a 'BADIRECTORY' error.

/EBLOCK[=block]

This qualifier will reset the end-of-file mark to the highest block allocated if no block has been specified. Otherwise the end-of-file mark will be set to the specified block.

/EBYTE[=byte]

This qualifier will set the end-of-file byte mark to the highest byte if it has not been specified. Otherwise the end-of-file byte mark will be set to the specified byte.

/EXPIRATION_DATE=date

/NOEXPIRATION_DATE

Controls whether an expiration date is assigned to the specified files. Specify the date according to the rules described in Chapter 1 of the VMS DCL Concepts Manual. Absolute date keywords are allowed. If the date is specified as '0', today's date is used.

/IDENT=identifier or uic

Modify the file ownership. See also /OWNER_UIC. This allows modification of the file-ownership even if the file is open, eg. INDEXF.SYS. /IDENT and /OWNER_UIC can not be used together.

/IGNORE=INTERLOCK

Perform the SET command on open or locked files. Default DFU will return a %SYSTEM-E-ACCONFLICT on open files. The /IGNORE=INTERLOCK option can overrule this behaviour.

/LOCKED

/NOLOCKED

This qualifier will lock a file for future use. Nothing else can then be done with the file, until it is unlocked (which can also be done with the VMS 'UNLOCK' command.

/LOG (default)

/NOLOG

Controls whether the SET command displays the file specification of each file after the modification is made.

/MAXREC=record

Sets a new value for the maximum record number in the file header.

/NOMOVE

This qualifier disables or enables (/NONOMOVE) the MoveFile attribute on files. It is the same as the DCL command SET FILE/NOMOVE, but in conjunction with /IGNORE=INTERLOCK DFU can change the setting on Open or Locked files.

/ORGANIZATION=keyword

The following keywords are used as parameters for the ORGANIZATION qualifier: DIRECT, INDEXED, RELATIVE and SEQUENTIAL. This allows one to modify the file organization type in the file header. Of course this won't change the real organization of the file.

/OWNER_UIC=uic or identifier

Modify the file ownership to an UIC or identifier. See also /IDENT. This allows modification of the file-ownership even if the file is open, eg. INDEXF.SYS. /IDENT and /OWNER_UIC can not be used together.

/RECATTRIBUTES=keyword

The following keywords are used as parameters for the RECATTRIBUTES qualifier: NONE, FORTRAN, IMPLIED, PRINT and NOSPAN. This allows one to modify the file's record attributes in the file header. NONE, FORTRAN, IMPLIED and PRINT are mutually exclusive, but can be used in combination with NOSPAN. When NOSPAN is omitted SPAN is assumed (the default is to allow records to cross block boundaries).

/RECSIZE=size

Sets a new value for the record size in the file header.

/RECTYPE=keyword

The following keywords are used as parameters for the RECTYPE qualifier: FIXED, STREAM, STREAMCR, STREAMLF, UNDEFINED, VARIABLE and VFC. This allows one to modify the file's record type in the file header.

/REVISION_DATE=date

/NOREVISION_DATE

Controls whether a revision date is assigned to the specified files. Specify the date according to the rules described in Chapter 1 of the VMS DCL Concepts Manual. Absolute date keywords are allowed. If the date is specified as '0', today's date is used.

/RVCOUNT=count

Sets a new value for the revision count in the file header.

/UPDATE

/NOUPDATE (default)

Normally the file's revision date will be updated after any modification to it. SET however disables this update (otherwise the REVISION date could not be set). Specify this qualifier if the revision date is to be updated.

/VERSION_LIMIT=n

Changes the file version limit (n must be between 0 and 32767). This qualifiifier can be used also to change the version limit on open files (such

as log files); in that case the qualifier must be used in conjunction with /IGNORE=INTERLOCK.

/VFCSIZE=size

Sets a new value for the VFC size in the file header. This value will only be used with the VFC record type.

5.8 UNDELETE

This chapter describes the UNDELETE command with the related parameters and qualifiers.

5.8.1 Description

The UNDELETE function is designed to recover deleted files, if possible. UNDELETE operates in a safe mode such that it first checks if the deleted file header and diskblocks are still available. Only in that case will the file be recovered. Otherwise UNDELETE will leave the disk unmodified. UNDELETE has some powerfull options:

- Generate a list of recoverable files with the /LIST qualifier. This qualifier will not undelete any file, and therefore it does not lock up the disk.
- Undelete multiple files in one pass.
- Make file selections based on the owner with the /OWNER or /IDENT qualifier.
- Undeleted files which cannot be entered back in their original directory will be moved to the [SYSLOST] directory. This saves an additional ANALYZE/DISK/REPAIR action.

5.9 File specification syntax

Because a deleted file no longer 'knows' its parent directory, do NOT enter a directory specification in the file name. Therefore , the syntax to undelete a specific file is:

```
   DFU> UNDELETE <device:>/file=<filename.ext>
eg:
   DFU> UNDELETE $1$DIA1:/file=FOO.BAR
```

DFU will try to re-enter the file in its original directory; it that fails the file will be entered in the [SYSLOST] directory.

5.9.1 How Undelete works

The UNDELETE command operates as follows :

1. Lock the volume. In the same manner as ANALYZE/DISK/REPAIR and SET VOLUME /REBUILD Undelete locks the disk for other writers such that no modifications to the INDEXF, BITMAP or QUOTA file can be made.
2. Search the INDEXF.SYS file if the requested file is still there. A check is made if it really is a valid deleted file header.
3. Checks if the deleted file header and all possible extension headers are still valid.
4. Checks if the blocks previously owned by this file are still free.
5. If all these conditions are met DFU will display the file and asks for a confirmation.
6. Rebuilds all of the deleted file headers and bitmaps in memory. If no errors are found all the file headers and bitmaps are written back to disk. The files original backlink is saved in a table in memory.
7. If appropiate the blocks and headers are charged against diskquota.
8. When all of INDEXF.SYS has been scanned the volume is unlocked.
9. The file(s) is (are) entered in the original directory if possible, using the table build up in memory. If the enter command fails (probably because the original directory is gone) the file will be entered in [SYSLOST]. DFU will create this directory if needed.

If the /LIST qualifier is used, DFU will just list the recoverable files without performing any action on the disk. The disk will not be locked.

It is important to note that during the bitmap and file header processing any error will immediately terminate the recovery process, and unlock the disk. Files which have been recovered up to that point will still be entered in the appropiate directory. As no wrong information has been written back to the disk this should leave the disk in a proper state.

NOTE: The blocks recovered will NOT be subtracted from the Volume's free block count. To get the actual freeblock count a SET VOLUME/REBUILD=FORCE is necessary. Also a complete reMOUNT will reset the freeblock count.

IMPORTANT: DFU takes care to recover only files whose original blocks are free. However these blocks may have been modified in the meantime by another file which also has been deleted. Therefore each recovered file must be checked manually to check its integrity.

Example of a Undelete session :

```
DFU> undel $1$dua102:/list

Recoverable file [TEST]CHANGE_UIC.FOR;2 found
Recoverable file [TEST]CHECK_ID.FOR;1 found
Recoverable file [TEST]CHKPRDIMG.FOR;1 found
```

```
Recoverable file [TEST]CHRLEN.FOR;1 found

DFU> undel/file=*.for $1$DUA102:

%DFU-I-READBMAP, Reading BITMAP.SYS...
%DFU-W-LOCKED, Volume now LOCKED for write
%DFU-I-UNDEL, Start search on $1$DUA102:

Recoverable file [TEST]CHANGE_UIC.FOR;2 found
Recover this file? (Y/N) [N] : n
Recoverable file [TEST]CHECK_ID.FOR;1 found
Recover this file? (Y/N) [N] : y
%DFU-S-RECOVER, File succesfully recovered
%DFU-I-ADDQUOTA, updating diskquota...

Recoverable file [TEST]CHKPRDIMG.FOR;1 found
Recover this file? (Y/N) [N] : n
Recoverable file [TEST]CHRLEN.FOR;1 found
Recover this file? (Y/N) [N] : y

%DFU-S-RECOVER, File succesfully recovered
%DFU-I-ADDQUOTA, updating diskquota...
%DFU-I-UNLOCK, Volume unlocked
%DFU-I-ENTER, Entering file(s) in directory...
%DFU-S-ENTERED, File CHECK_ID.FOR entered in original directory
%DFU-S-ENTERED, File CHRLEN.FOR entered in original directory

DFU> EXIT
```

5.9.2 Files marked for delete

Normally DFU does not undelete files marked for delete. To recover files marked for delete, use the /MARKED qualifier. This will specifically undelete files marked for delete. This allows the recovery of files which are deleted but still open, such as INSTALLed files, or even recovery of the system dump file SYSDUMP.DMP whenever such a file is accidentially deleted.

UNDELETE

Recover one or more deleted files on a device.

Format

UNDELETE device[:]

Parameters

device

The device on which to undelete one or more files. The device will be write-locked during the undelete process.

Qualifiers

/FILE=filename

The file to be recovered. Wildcards may be used. If a matching file is found Undelete will ask a confirmation. If a certain file is confirmed, that file will be recovered if possible, and control will be returned to the DFU> prompt. If the /FILE qualifier is omitted DFU will assume *.*;* .

/IDENT=identifier or uic

Used to search for files owned by a specific identifier or UIC. Another way is the /OWNER_UIC qualifier; these 2 qualifier can not be combined in one UNDELETE command.

/LIST(=output-file-name)

Generate a list of recoverable files. No undelete will take place, and the disk will not be locked. The defaultoutput is SYS$OUTPUT.

/MARKED

Consider only files marked-for-delete. Such files are usuallly still open by some utility, such as INSTALLed files or the system dump file.

/OWNER_UIC=uic or identifier

This qualifier is used to select files by a UIC or identifier . This qualifier can not be combined with /IDENT.

/STATISTICS

This qualifier displays the performance statistics: CPU time, Elapsed Time, I/O and PageFaults.

5.10 SPAWN

The SPAWN command creates an interactive subprocess. This allows a quick escape to the DCL level from an interactive DFU session. The subprocess has a default prompt DFU_sub$.

5.11 VERIFY

This chapter describes the VERIFY command with the related parameters and qualifiers.

5.11.1 Description

The VERIFY command makes an analysis of the disk, scanning for file and disk structure errors. VERIFY performs almost all of the checks normally done by a ANALYZE/DISK command. But VERIFY is several times faster than ANALYZE/DISK, and uses less resources.

VERIFY checks and reports the following errors:

- Checks the logical information contained in the disk's HOME block
- Reports files marked for deletion
- Reports mismatches in the index file's bitmap
- Checks the VMS$COMMON.DIR backlink on system disks
- Checks all backlinks.
- Reports multiple allocated blocks
- Reports blocks incorrectly marked free and allocated
- Reports all lost files
- Checks for mismatches between blocks used and blocks charged in QUOTA.SYS. (Only done when diskquota is enabled)

5.11.2 Basic repair actions

The /FIX qualifier can be used to perform some basic repair actions. Unlike ANALYZE/DISK/REPAIR this does not lock the disk! /FIX repairs the following errors:

- Deletes file marked for delete (if possible).
- Recovers lost files and directories into [SYSLOST].
- Recovers a very rare problem which will get ANALYZE/DISK/REPAIR into a computable loop (This problem is fixed in VMS V6.2 or with the VERI02_061 patch kit).
- Recovers a corrupted backlink for the Master File Directory 000000.DIR.
- Recovers a wrong backlink for VMS$COMMON.DIR files on the system disk

5.11.3 Advanced repair actions

The /REBUILD qualifier can be used to perform some more repair actions. This will however lock the disk (like a DCL $SET VOLUME/REBUILD=FORCE command) for a short period of time. /REBUILD repairs the following errors :

- Mismatches in the INDEXF.SYS bitmap (files reported with the NOBITSET and NOBITCLR errors)
- Blocks incorrectly marked allocated or free in the BITMAP.SYS.
- Mismatches in the diskquota file.

Note that /REBUILD does NOT change the disk's free block count.

5.11.4 Directory scanning

The /DIRECTORY_SCAN qualifier will force DFU to also verify and repair (when using /FIX) directory problems. This qualifier directs DFU to scan all directories on the disk. This has some advantages :

- Better detection of some cases of 'lost' files. Combined with the /FIX qualifier DFU may be able to recover such files into the correct directory instead of moving such files to [SYSLOST].
- Detects and fixes file-id mismatches between directories and the INDEXF.SYS file.
- Using the /DIRECTORY_SCAN qualifier makes DFU fully compatible with ANALYZE/DISK.

Please note that a complete directory scan may take several minutes to complete.

5.11.5 Interpreting errors

On a system (cluster) wide mounted disk there will probably be concurrent disk activity during the VERIFY command. Therefore VERIFY may report some errors which are not really errors. Try running VERIFY 2 or 3 times to see if the errors are reported again. To get a really consistent report from VERIFY use the /LOCK qualifier. This will write-lock the disk during the VERIFY run, so it is not recommended to use this qualifier too often. (The /REPAIR qualifier of ANALYZE/DISK also write-locks the disk).

5.11.6 Error reporting

An example of a typical VERIFY run follows:

```
DFU> VERIFY mydisk
%DFU-I-VERIFY, Verifying MYDISK:
%DFU-S-CHKHOME, Home block info verified OK
%DFU-I-IFSCAN, Scanning INDEXF.SYS ...
%DFU-I-CHKBITMAP, Checking BITMAP.SYS...
%DFU-I-CHKLOST, Checking for lost files...
%DFU-I-CHKQUOTA, Checking QUOTA.SYS...

DFU> VERIFY/REBUILD sys$sysdevice
%DFU-W-LOCKED, Volume now write locked
%DFU=I-VERIFY, Verifying SYS$SYSDEVICE:
%DFU-S-CHKHOME, Home block info verified OK
%DFU-W-DELETED, file (620,351,1) RSF_DI_RSF_SERVER0.TMP;3 marked
 for delete
%DFU-W-DELETED, file (6349,173,1) DCLTABLES.EXE;937 marked
 for delete
%DFU-W-DELETED, file (19745,35,1) RSF_DI_RSF_SERVER0.TMP;3 marked
 for delete
%DFU-I-CHKBITMAP, Checking BITMAP.SYS...
%DFU-E-ALLOCCLR, blocks LBN 2667141 through 2667143 incorrectly
 marked allocated
%DFU-E-ALLOCCLR, blocks LBN 2667153 through 2667161 incorrectly
 marked allocated
  .
  .
%DFU-S-RBDBITMAP, BITMAP.SYS succesfully rebuild
%DFU-I-CHKLOST, Checking for lost files...
%DFU-I-UNLOCK, Volume unlocked

DFU>
```

VERIFY can report the following errors:

ALLOCCLR, blocks incorrectly marked allocated,

Severity: ERROR
Explanation: Blocks were found which are not allocated by a file but set in the BITMAP file. This error is often seen when VERIFY is run and there is concurrent file activity on the disk. Also, if after a system crash the disk is mounted with /NOREBUILD such blocks may exist. ReRun VERIFY with /LOCK to see if the error is still there.
User Action: Run VERIFY/REBUILD to rebuild the BITMAP file.
ALLOCSET, blocks incorrectly marked free,

Severity: ERROR
Explanation: Blocks were found which are allocated by a file but not set in the BITMAP file. This error is often seen when VERIFY is run and there is concurrent file activity on the disk. Also, if after a system crash the disk is mounted with /NOREBUILD such blocks may exist. ReRun VERIFY with /LOCK to see if the error is still there.
User Action: Run VERIFY/REBUILD to rebuild the BITMAP file.
BADBLOCK, file has suspected bad blocks,

Severity: ERROR
Explanation: A file has suspected bad blocks. This is likely to be caused by hardware errors on the disk.
User Action: Try to copy the file to another location. Also check the Error Log.
BADEXTLNK, link to extension header broken,

Severity: ERROR
Explanation: VERIFY tries to follow the extension link of the file and it failed, or the extension header is invalid.
User Action: Try ANALYZE/DISK/REPAIR. However, this may be an unrepairable error, in which case the file will be corrupted and unreliable.
BADMFDLNK, directory has backlink to 000000.DIR on RVN n,

Severity: ERROR
Explanation: A directory has a backlink to a Master file directory other than the one on RVN 1. This is incorrect, but the only way to recover is to reenter the directory manually into the 000000.DIR on RVN 1.
User Action: Do a SET FILE/ENTER into the 000000.DIR directory on RVN 1, then do a SET FILE/REMOVE from the wrong 000000.DIR.
DELETED, file marked for delete,

Severity: WARNING
Explanation: A file was found marked for deletion. Such files usually exists as a result of file being deleted while it was still INSTALLed, or as a result of a system crash. It is not considered as an error. VERIFY/FIX may be able to cleanup such files.
ERRHOME, Home block info not OK,

Severity: ERROR
Explanation: The Info contained in the Home Block was not consistent or corrupted.
User Action: Run ANALYZE/DISK/REPAIR to try to fix the error.
INVBAKFID, file has invalid backlink,

Severity: ERROR
Explanation: The files backlink file id is not filled in. This is likely to be a lost file. (Note that once VERIFY reports this error, it will not be reported again during the LOSTFILE check). But is also possible that the file is in a valid directory, but that the Backlink file id is not filled in.
User Action: Run ANALYZE/DISK/REPAIR. If the file is in a valid directory the backlink will be repaired. Otherwise the file will be moved to the [SYSLOST] directory.
LINKCOUNT, linkcount set to n, must be m,

Severity: WARNING
Explanation: This error indicates that DFU has found files, for which the linkcount does not match the number of found directory entries for this file. This error will only be reported on a disk with hardlinks enabled, and only when using the

/DIRECTORY_SCAN qualifier.
User Action: Run VERIFY/DIR/FIX to repair the linkcount.
LOCKED, file is deaccess locked,

Severity: WARNING
Explanation: The file is deaccess locked. This may be a result of a system crash.
User Action: Try to unlock the file with the DFU SET command, or the DCL UNLOCK command.
LOSTHDR1, file found in nonexistent directory,

Severity: WARNING
Explanation: During the lost file check a file was discovered in a nonexistent directory. This error can be caused as follows : set a directory file to NODIRECTORY and delete it.
User Action: Run VERIFY/FIX to move the file to the [SYSLOST] directory.
LOSTHDR2, file found in directory with bad backlink,

Severity: WARNING
Explanation: During the lost file check a file was discovered in a valid directory. However, the directory has an invalid backlink. This error can be caused by doing a SET FILE/REMOVE of a directory file.
User Action: Run VERIFY/FIX to move the directory to the [SYSLOST] directory. Then move the directory back to the correct location. The files in the directory should be accesible again.
LOSTHDR3, file found in invalid directory,

Severity: WARNING
Explanation: During the lost file check a file was discovered in a directory which is not a valid directory. Either the file has a wrong backlink, or the directory file has the directory file attribute not set.
User Action: Look up the directory and use DFU SET to set the directory bit. If the directory file was OK then run VERIFY/FIX to recover the lost file.
LOSTHDR4, not found in a directory,

Severity: WARNING
Explanation: When the /DIRECTORY_SCAN qualifier is used, DFU detects files which have a valid backlink but which are not seen in the directory. This can be repaired with /FIX.
User Action: Using /FIX will enter these files in the correct directory.
MULTALLOC, blocks multiple allocated,

Severity: ERROR
Explanation: A file has blocks allocated which already belong to another file. Note that VERIFY will make a second pass through INDEXF.SYS to report all the files involved. This is a serious error, and may be a result of a disk being mounted on two seperated VAXClusters (or a partitioned VAXCluster).
User Action: Copy all the files found to another location. Next all the files must be

deleted. Run VERIFY/REBUILD to repair the BITMAP. All the afflicted files must be manually inspected to see which files are valid and which are corrupted.
NOBITCLR, Deleted file header marked BUSY,

Severity: WARNING
Explanation: A file was deleted. The corresponding bit in the INDEXF bitmap should be cleared but is still set. This is not a serious error.
User Action: Run VERIFY/REBUILD to fix the error.
NOBITSET, index file bitmap bit not set,

Severity: WARNING
Explanation: A valid file was found, but the corresponding index file bitmap bit is clear. This is not a serious error.
User Action: Run VERIFY/REBUILD to fix the error.
NOOWNER, file has no owner,

Severity: WARNING
Explanation: A file was found with UIC [0,0] as the owner. This is not a normal situation.
User Action: Look up the file and modify the file owner.
QUOTAERR, UIC [x,y] has n blocks used, QUOTA indicates m blocks,

Severity: WARNING
Explanation: If a disk has diskquota enabled, VERIFY calculates the blocks used and compares them with the value in the QUOTA.SYS file. Any mismatch is reported. This error is often seen when VERIFY is run, and there is concurrent file activity on the disk. Also, if after a system crash the disk is mounted with /NOREBUILD such Quota mismatches may exist. ReRun VERIFY with /LOCK to see if the error is still there.
User Action: Run VERIFY/REBUILD to rebuild the QUOTA file. Or run a DISKQUOTA> REBUILD command.
SLFBAKFID, backlink points to itself,

Severity: WARNING
Explanation: This error indicates that DFU has found a directory with a backlink pointing to itself. This feature is only allowed for the MFD 000000.DIR. This error will result in ANALYZE/DISK going into a COMputable loop.
User Action: Run VERIFY/FIX to move the directory to the [SYSLOST] directory. Then remove the directory entry from itself with a SET FILE/REMOVE command. The files in the directory should be accesible again.

VERIFY

VERIFY a device for disk structure errors.

Format

VERIFY device[:]

Parameters

device

The device to be verified.

Qualifiers

/APPEND=filename

This qualifier redirects the output to be appended to an already existing file. If the output file does not exists it will be created. Use /APPEND or /OUTPUT but not both.

/DIRECTORY_SCAN

Performs a full directory scan. This may take up some time, but it allows detection of mismatches between the directories and INDEXF.SYS.

/FIX

/NOFIX (default)

Repair errors on the disk. File marked for delete will be deleted and lost files and directories will be moved to [SYSLOST].

/LOCK

/NOLOCK (default)

Locks the disk for file allocation/deletion. This gives a consistent report for the disk, but other users may experience a delay when accessing the disk. The device will be unlocked a the end of the VERIFY command.

/OUTPUT=filename

This qualifier redirects the output to a file. The output will also go to SYS$OUTPUT. CAUTION: if one uses /LOCK in combination with /OUTPUT , be careful to specify an outputfile on ANOTHER disk, or DFU will finish with a severe error.

/REBUILD

/NOREBUILD (default)

Perform a volume rebuild. Using this qualifier will temporarily lock the disk for other users.

/STATISTICS

This qualifier displays the performance statistics: CPU time, Elapsed Time, I/O and PageFaults.