

FSYS - *software*



ALL05: DFSee

.....
Introduction and concepts



Who am I

■ Jan van Wijk

- ▶ Software Engineer, C, REXX, Assembly
- ▶ Founded FSYS software in july 2001 ...
- ▶ Started with OS/2 in 1987, development of OS/2 1.0 Extended Edition (Query Manager)
- ▶ Used to be a system-integration architect at a large Bank, 500 servers en 7500 workstations.
- ▶ Home Pages: <http://www.fsys.nl>
<http://www.dfsee.com>



FSYS Software offerings

- DFSee, OS/2, DOS, Win-NT
 - ▶ Partitions, (un)FDISK and LVM functionality
 - ▶ HPFS, FAT, FAT32, NTFS, JFS ...
 - ▶ CHECK, UNDELETE, CLONE, IMAGE
- LPTool, OS/2 only
 - Parallel port connectivity utility
 - ▶ File transfer
 - ▶ Remote execution
- Several small (OS/2) utilities



Presentation contents

- Short introduction to DFSee, usage
- DFSee architecture and working
 - Accessing 'sectors' in 'stores'
 - Different types of media
 - Virtual disks and UNDO/REDO buffering
 - Generic implementation plus FS specifics
- Some hands-on in disk analysis and repair
 - DFSee, FS display, analysis and repair; JvW



What is DFSee, functional view

- DFSee is an operating-system neutral disk utility with functionality comparable to FDISK, LVM, PartitionMagic, DriveImage, DriveCopy, Norton-Ghost, Undelete and more ...
- Main areas of functionality:
 - Backup and Restore of all partitioning information
 - FDISK/LVM, create, delete, change partitions and UNFDISK
 - IMAGING, disk-areas to and from (compressed) image files
 - CLONING, disk-areas to and from other disk-areas
 - Filesystem specific, Check, Display, Undelete and Fix



What is DFSee. technical view

- DFSee is a tool to examine and possibly modify data on a variety of storage media like disks and partitions.
- Types of storage supported:
 - Physical disks, selection with the "disk" or "walk" command
 - Disk partitions, selection with the "part" command
 - Volumes (disk, diskette, CD ...), select with "vol" command
 - Regular files (images) selected with the "im" command
- Data can be viewed 'raw' binary or formatted for:
 - FDISK usage, partition-tables bootsectors ...
 - Filesystem structures, FAT, HPFS, NTFS, JFS ...



DFSee versions and user interface

- DFSee is available in an OS/2, DOS and Windows-NT version, which are almost identical in functionality.
- Text-based utility, able to run in simple environments
- You control DFSee by entering commands
- Output from the commands will be printed to the screen and, when activated, to a logfile as well
- At normal startup DFSee will work in a 'windowed' mode with an output area, a command-line and help
- Full menu-based user interface is under development



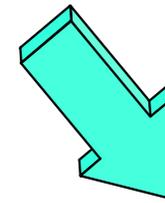
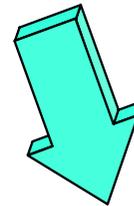
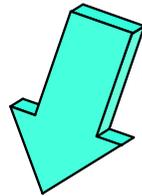
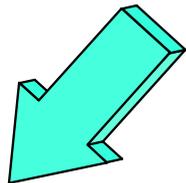
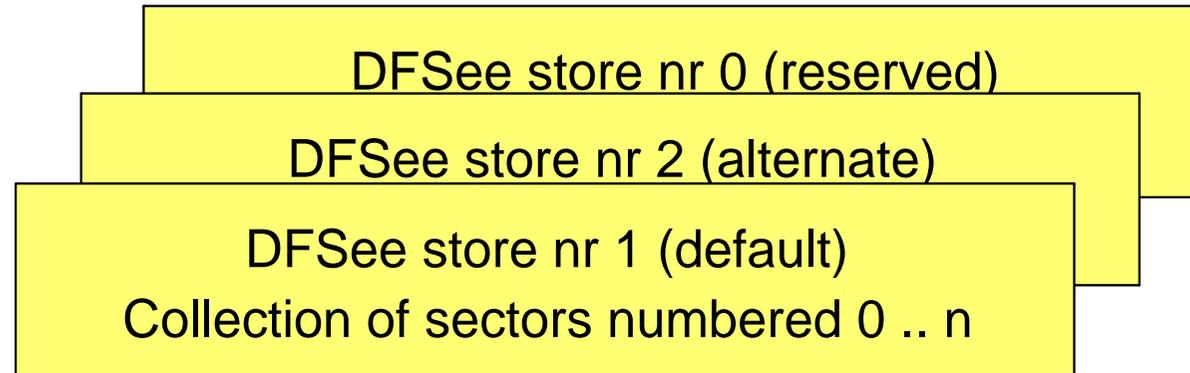
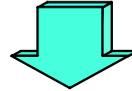
Architecture of the DFSee utility

- DFSee considers every storage opened as a collection of sectors (typical size is 512 bytes), called a STORE
- Many generic commands are available to work with any type of filesystem or disk. (see dfscmds.txt)
- On opening a storage medium, DFSee will analyse the first sector(s) to determine if it is of a known format. If so, specific functionality and extra commands will be activated for this type of data. Example formats:
 - FDISK, HPFS, FAT, NTFS, JFS (see dfsXXXX.txt)
- Use the "?", "??" and "???" commands to list available commands and sector-types for the current format



Store concept

Open/Read/Write from DFSee functions



Volume (dr-letter)
VOL cmd

Image file
IM cmd

Physical disk
DISK/WALK cmd

Virtual disk (VMA)
DISK/WALK cmd

Disk-partition
PART cmd



Store concept, details

- A store can be associated with a medium
- The store also keeps additional information like the current geometry (Cyls/Head/Sectors)
- Partitions on (virtual) disks are supported by defining a 'base' sector-nr, this is the sector-number on the disk that will be considered sector-0 for the partition.
 - When a volume, file or whole disk is opened, the base will be zero. (as shown in the DFSee status line)



DFSee command handling

- Commands in DFSee are handled on a few levels:
 - 1) Generic commands (TPlibrary) like CD, LOG and SET
 - 2) Filesystem (mode) specific commands, like:
FIXBOOT, ALLOC, DELFIND ...
 - 3) Other DFSee commands like: PART, DISK, WALK,
CHS, MAP, LIST, BASE, WIPE, SIM, WRIM ...
- Because level (2) is dedicated, these commands are not always available! As an example: FIXBOOT only works in FAT, HPFS, NTFS or JFS mode.



Supported filesystems (mode)

- **FDISK:** view and modify partition-tables, LVM, bootmanager and related information
- **HPFS:** (OS/2) HPFS partitions
- **FAT:** FAT, including VFAT and FAT32
- **NTFS:** Win-NT/W2K and XP specific filesystem
- **JFS:** (OS/2) journalling filesystem
- **AUX:** Auxillary, when nothing else

- The mode (or FS) is set automatically most of the time, but can be forced with the 'FS xxx' command. It is shown on the DFSee status line displayed after completion of each command.



Commands: general

- LOG: append all screen output to a file
- PART: show table of all partitions (all disks) or select the specified partition
- DISK: select the specified disk
- WRIM: write image, reading sectors from file
- BSFIND: find bootsectors (MBR/EBR etc)
- WIPE: clear sectors using a pattern (or 00)
- FIND: find any data on disk or partition
- EDIT: binary edit sector contents
- RUN: run a script with DFSee commands



Commands: Imaging and Cloning

- IM: open imagefile for analysis
- SIM: save image, writing sectors to file
- WRIM: write image, reading sectors from file
- CLONE: copy disk-area to another disk-area
- COMP: compare sectors of two disk-areas

- EXPORT: write sector list to a file
- IMPORT: read sector list from a file
- LIST: display sector list in various formats



FDISK partition table update

- CREATE: create a new partition
- DELETE: remove a partition
- SETTYPE: change type of a partition
- SETACCESS: change visibility of a partition (hide)
- STARTABLE: make partition startable (active)
- NEWMBR: refresh MBR code, clear part. tables
- FIXEXT: change extended type **0F** to **05**
- FIXCHS: write correct CHS values to tables



FDISK: other commands

- **NEWMBR:** write new MBR startup code, with optional clearing of partition tables
- **PLIST:** list selected partition information
- **PSAVE:** save partition information in a file
- **PRESTORE:** restore psaved info from a file
- **LVMSET:** update LVM-info for partition
- **LVMCRC:** Update all LVM sector CRC values
- **SETBOOT:** boot specified partition (IBM-bmgr) or update bootmanager information
- **VMAKE:** create virtual DFSee disk for testing



HPFS

- **ALLOC:** show pseudo-graphic map of allocation
- **CHECK:** check FS integrity, display errors
- **DIRTY:** set/reset the dirty bit (avoid CHKDSK)
- **DELFIND:** find all deleted files (fnodes)
- **RECOVER:** recover selected files (undelete)
- **SAVETO:** recover one (current) file
- **FIXBOOT:** create a new bootsector
- **FINDxxx:** find root or codepage sectors
- **FIXxxx:** update superblocks with find results



FAT(32)

- ALLOC: show pseudo-graphic map of allocation
- CHECK: check FS integrity, display errors
- CL: show info at specified cluster
- FATSIM: save FAT sectors in a file
- FATWRIM: restore FAT sectors from a file
- FIXBOOT: create a new bootsector
- FINDROOT: find rootcluster for FAT32
- SETROOT: update bootsector with find result



NTFS

- ALLOC: show pseudo-graphic map of allocation
- CHECK: check FS integrity, display errors
- CL: show info at specified cluster
- DELFIND: find all deleted files (fnodes)
- RECOVER: recover selected files (undelete)
- SAVETO: recover one (current) file
- FINDROOT: find root directory MFT
- FIXBOOT: create a new bootsector (from spare)
- MFT: show master-file-table information



JFS

- ALLOC: show pseudo-graphic map of allocation
- BL: show info at specified block
- DIRTY: set the dirty status (avoid CHKDSK)
- FIXBOOT: create a new bootsector
- SYNC: synchronize 1st and 2nd superblocks



Generic view on OS interfaces

- Operating systems offer services to applications in several interfaces called API's
- DFSee uses API's on several levels:
 - C-library functions (normal file-i/o etc)
 - Specific OS API's (VIDEO, DISK ...)
 - Low-level API's (BIOS, driver IOct1 ...)
- Except for the C-library, these are quite different for the target platforms (DOS, OS/2 and Win-NT)
- DFSee uses **NO** direct interface to disk-controllers, it **NEEDS** the operating system for access
 - This requires working (SCSI/ATAPI/RAID/BIOS) drivers ...



Simplified API view

OS/2 examples

C-runtime
TPIlibrary

HPFS.IFS
FAT32.IFS

OS2DASD
OS2LVM

IBM1S506
AHA174x

EIDE contr/disk
Adaptec 2940

JvW 2002

C-library functions

Applications + runtimes

OS (FILE) SYSTEM API's

FS-drivers (HPFS / FAT ...)

VOLUME MANAGEMENT

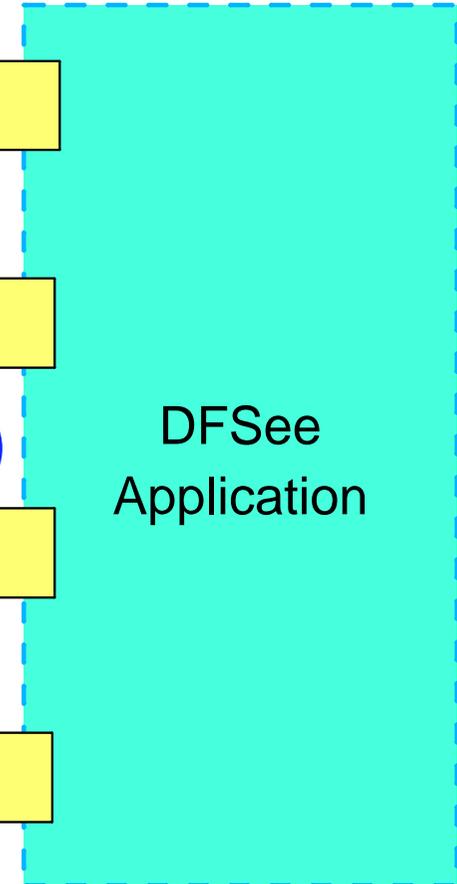
Volume / disk manager

DD STRATEGY .. / IOctl ...

Device drivers

SCSI / ATAPI / ...

HARDWARE (disk controllers)



OS interfaces used by DFSee

- The majority of DFSee disk-related API-usage is on the OS and Filesystem level, with API's such as:
 - OS/2: DosOpen, DosRead, DosWrite, FsCtrl
 - DOS: (extended) INT13, INT21 DOS functions
 - Win-NT: CreateFile, ReadFile, WriteFile ...
- Most operating system specific code is concentrated in a single module, to make porting easier.
- Really generic stuff is in the TPLibrary, used by other programs as well ...



Planned: UNDO and REDO

- Until now all DFSee commands always work **DIRECTLY** on the medium involved.
Every change made is written to the disk or volume.
- Some commands are hard to undo by giving additional (reversing) commands.
- One of the next DFSee versions will have a multiple UNDO/REDO facility that is based on the use of a virtual disk, linked to the actual physical medium



UNDO/REDO concept

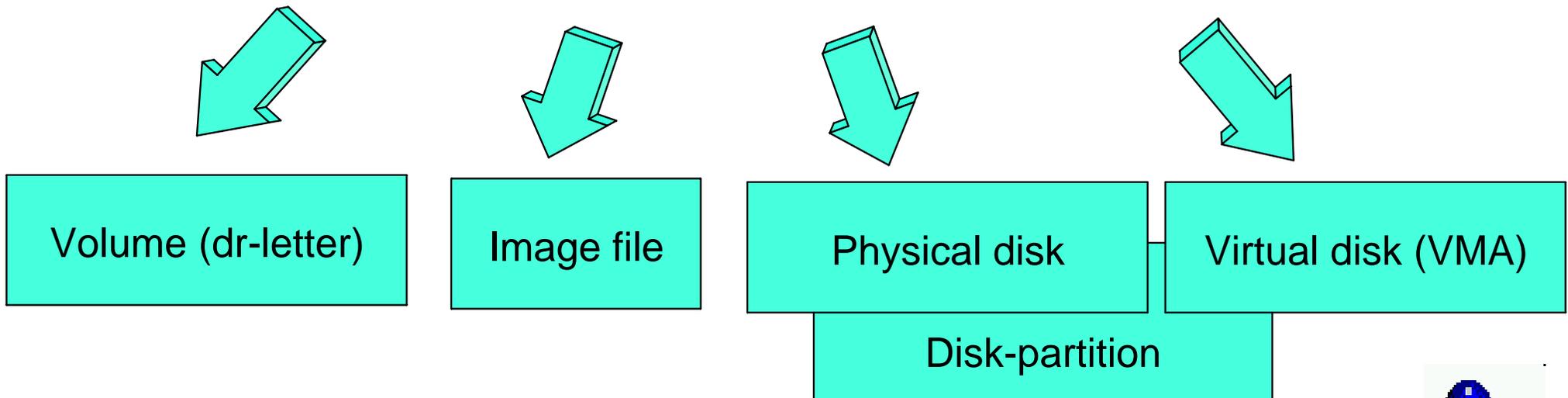
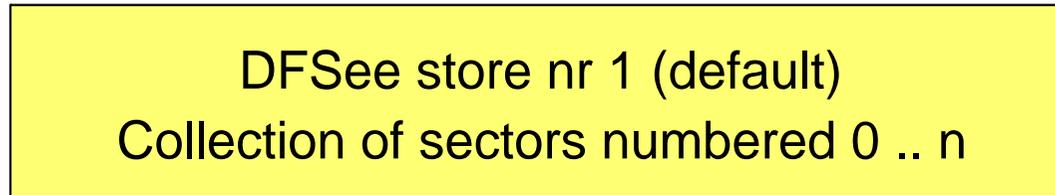
Open:
Initialize URB
for medium

Open/Read/Write from DFSee

Read:
From URB or
from medium

Close:
Write changes
to medium

Write:
Always to
URB



Undo/Redo concept, details

- The URB records a 'level' for each sector stored, the level increases after each command.
- Read will get data from the current URB level, if available and from the medium otherwise
- Write will write at the current level in the URB
- Undo decrements the level (but no erase of data)
- Redo increments the level (upto last level)
- Close will commit all stored data from level 0 upto the current level, and store 'old' contents in a commit-log
 - An undo of the whole session is still possible after quitting DFSee by using this commit-log (similar to PSAVE file).



Some remarks on DFSee usage

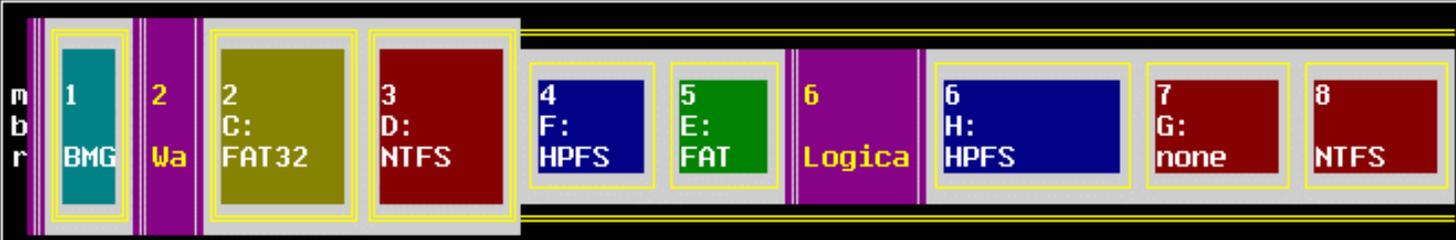
- DFSee will normally ask before writing anything, however after confirmation the change is written immediately! (UNDO/REDO is planned)
- Remember to use the "log" command to get DFSee output in a file. Never use simple output redirection
- For fixing partition (FDISK) disasters first use the DFSUNFD.CMD (or .BAT) to collect information



```

dfs.exe
DFSee-OS2: 5.03 25-09-2002 (c) 1994-2002; Jan van Wijk
multi-page output window
DFSee-OS2 version : 5.03 executing: map -r -g
Number of disks : 1
LogGeo Disk 1 Cyl : 2584 H:240 S:63 = 7.38 Mb DiskSize : 02542980 = 19077 Mb

```



```

Store 1 : Unused : -- mode=--none-- 260902-121819
RC:0 <Enter> : 00000000 u=00000000 x=00000000 this=00000000 Base=00000000
DFSee-OS2 version : 5.03 executing: fdisk -r-
Pid02=C: WARNING : Primary does not start on head-0 (cylinder boundary)

```

id	PD	Vol	Type, description	Format	Creator	VolumeLabel	LVM Volume, Par	Size Mb
01	1	>	Prim 0a Boot-Mgr	BMGR	fdisk	«eComStati»	, [BOOT MANAGE	7.4
02	1	*C:	Prim 0b FAT32	FAT32	MSWIN4.1	WINDOWS 98	Win98, Preload	3071.2
03	1	*D:	Hide 17 Inst-FSys	NTFS	Win NT		WinXP, Boot-Win	3004.8
04	1	*F:	Log 07 Inst-FSys	HPFS	IBM 4.50	ECOMSTATION	eComStation, eC	967.1
05	1	E:	Log 06 FAT16	FAT	IBM 4.50	EXCHANGE	Exchange, excha	509.4
06	1		Logical FreeSpace	-- -- --	-- -- --	-- -- --		723.5
06	1	H:	Log 07 Inst-FSys	HPFS	IBM 4.50	OS2 DATA	HPFS data, big	4902.2
07	1	G:	Log 07 Inst-FSys	none	fdisk		NTFS data, big	2879.3
08	1		Log 07 Inst-FSys	NTFS	Win NT		XP-boot, backup	3012.2

```

Line 41 of 45 <Ctrl> + ↑↓ = Scroll <PgUp> <PgDn>
F1=help F2=cmds F3=quit F4=save F5/F6=part-tbl F7=vol F8=map F9=alloc Esc=abort

```

Time for Play ...

Some hands-on

And questions?



FSYS - *software*



DFSee



OS/2 **WARP**
Server
for e-business

More Questions?

