DFSee concepts and Q&A

Jan van Wijk

How to use the DFSee program more effectively by understanding its capabilities and some of the internal workings
Presentation contents

- Who am I
- DFSSee functional and technical view
- DFSSee architecture and working
  - Accessing 'sectors' in 'stores'
  - Different types of media
  - Virtual disks, for analysis purposes
  - Generic implementation plus FS (mode) specific
- Examples using DFSSee ...
Who am I ?

Jan van Wijk

- Software Engineer, C, Rexx, Assembly, PHP
- Founded FSYS Software in 2001, developing and supporting DFSee from version 4 to the latest
- First OS/2 experience in 1987, developing parts of OS/2 1.0 EE (Query Manager, later DB2)
- Used to be a systems-integration architect at a large bank, 500 servers and 7500 workstations
- Developing embedded software for machine control and appliances from 2007 onwards

Home page:  https://www.dfsee.com/
What is DFSee, functional view

- DFSee is an OS neutral utility similar to FDISK, LVM, PQ-Partition Magic, PQ-Drive-Image, Norton-Ghost, Norton-Commander, Acronis True Image and more...

- Main areas of functionality:
  - Backup and restore of partitioning information
  - Search missing partitions and recreate them
  - FDISK/LVM create and maintain partitions
  - Imaging, disk-areas to/from (compressed) files
  - Cloning, disk-areas to/from other disk-areas
  - FS-specific: Check, Display, Undelete and Fix
  - Browse directory/files, with copy, view, edit …
  - Access disk/partition images incl browse (.IMZ/.VDI)
  - Disk data analysis and update (binary edit, disasm)
What is DFSee, technical view

- DFSee is a tool to examine and possibly modify data on a variety of storage media

- Types of storage supported:
  - Physical disks, when access supported by the OS
  - Disk partitions on partitionable media, MBR or GPT
  - Volumes (drive letters on PC) or Devices on Linux
  - Regular files, like RAW disk images or binary files
  - DFSee compressed disk/partition images (.IMZ)
  - VirtualBox static or dynamic disk images (.VDI)

- Data can be viewed RAW or formatted for:
  - FDISK/GPT usage, partition tables, boot sectors, LVM-info
  - Filesystem structures, FAT, HPFS, NTFS, JFS, HFS, EXTn …
  - And viewed as ASCII, Disassembly or HEX (incl. editing)
DFSee versions and user interface

- DFSee is available for OS/2 (ArcaOS/eCS), DOS(32), Windows-XP/7/8/10, most Linux distributions and macOS (Intel 64-bit only)
- It is a non-graphical text based program, able to run in simple environments like a boot diskette, CDROM or USB-stick
- Most functions can be run from a windowed MENU interface with additional dialogs
- Even more through the command-line
- Output can go to the screen AND a logfile
DFSee architecture

- DFSee considers all storage as a collection of sectors (typical 512 bytes) called a STORE

- Many generic commands are available to work on any type of FS or disk (see DFSCMDS.TXT)

- On opening, DFSee will analyze the first sector(s) and select a suitable mode with specific commands and menu selections

- The most important modes (or filesystems) are: FDISK, FAT, HPFS, NTFS, JFS, HFS+, EXTn ...
The DFSee STORE concept

Open/Read/Write from DFSee functions

DFSee store nr 0 (system)

DFSee store nr 2 (alternate)

DFSee store nr 1 (default)
Collection of sectors 0 .. N

Filesystem
In PC-style driveletter or a Linux device

DFSee filesystem image file
*.IMZ
(as backup)

DISK Image
*.IMZ
*.VDI

Physical disk
SCSI - PATA
SATA - USB

DFSee in-memory virtual disk

RAW image file

Filesystem in a disk partition
On MBR/GPT partitioned media
A store can be associated with a medium using an OPEN menu-item or command.

The store keeps additional information like the geometry and some statistics:
- See the **STORE** command/menu-item for details.

Partitions on (virtual) disks are supported by defining a non-zero 'base' sector number as the disk-sector considered to be 'sector 0':
- See the **BASE** command/menu-item for details.

Operations like CLONE copy sectors between 2 stores.
Virtual disks in DFSee

- Exists in MEMORY within DFSee only
- Behaves (almost) the same as a real disk

Can be created in 2 ways:
- By specifying a size and/or disk geometry
- By using a set of .Pdx files as a template (often used with the DFSDISK*.* result files)

Can be used to:
- Learn DFSee commands and functions
- Test recovery scenarios and scripts
DFSee concepts and Q&A

Questions?