

Volume (computing)

From Wikipedia, the free encyclopedia

In the context of computer operating systems, a **volume** or **logical drive** is a single accessible storage area with a single file system, typically (though not necessarily) resident on a single partition of a hard disk. Although a volume might be different from a physical disk drive, it can still be accessed with an operating system's logical interface. However, a volume differs from a partition. In Linux systems, volumes are usually handled by the Logical Volume Manager or the Enterprise Volume Management System and manipulated using `mount(8)` (<http://www.netadmintools.com/html/8mount.man.html>). In NT-based versions of Microsoft Windows, volumes are handled by the kernel and managed using the Disk Management MMC snap-in.

Contents

- 1 Differences between volume and partition
 - 1.1 Example
- 2 Nomenclature of volumes
 - 2.1 Windows-NT based operating systems
 - 2.2 Unix-like operating systems
- 3 Benefits of keeping files within one volume
 - 3.1 Speed of data management
 - 3.2 Special functions of advanced filesystems and volumes
- 4 Volume label and serial number
- 5 External links

Differences between volume and partition

A volume is not the same thing as a partition. For example, a floppy disk might be accessible as a volume, even though it does not contain a partition, as floppy disks cannot be partitioned with most modern computer software. Also, an OS can recognize a partition without recognizing any volume associated with it, as when the OS cannot interpret the filesystem stored there. This situation occurs, for example, when Windows NT-based OSes encounter disks with non-Microsoft OS partitions, such as the ext3 filesystem commonly used with Linux. Another example occurs in the Intel world with the "Extended Partition". While these are partitions, they cannot contain a filesystem directly. Instead, "logical drives" (aka volumes) must be created within them. This is also the case with NetWare volumes residing inside of a single partition. In short, volumes exist at the logical OS level, and partitions exist at the physical, media specific level. Sometimes there is a one-to-one correspondence, but this is not guaranteed.

It isn't uncommon to see a volume packed into a single file. Examples include ISO9660 disc images (CD/DVD images, commonly called "ISOs"), and installer volumes for Mac OS X (DMGs). As these volumes are files which reside within another volume, they certainly are *not* partitions.

Example

This example concerns a Windows XP system with two physical hard disks. The first hard disk has two partitions, the second has only one. The first partition of the first hard disk contains the operating system. Mount points have been left at defaults.

Physical disk	Partition	Filesystem	Drive letter
Hard Disk 1	Partition 1	NTFS	C:
	Partition 2	FAT32	D:
Hard Disk 2	Partition 1	FAT32	E:

In this example,

- "C:", "D:", and "E:" are volumes.
- Hard Disk 1 and Hard Disk 2 are physical disks.
- Any of these can be called a "drive".

Nomenclature of volumes

Windows-NT based operating systems

It is important to note that Windows NT-based OSES do not have a single root folder. As a result, Windows will assign at least one path to each mounted volume, which will take one of two forms:

- A drive letter, in the form of a single letter followed by a colon, such as "F:"
- A mount-point on an NTFS volume having a drive letter, such as "C:\Music"

In these two examples, a file called "Track 1.mp3" stored in the root directory of the mounted volume could be referred to as "F:\Track 1.mp3" or "C:\Music\Track 1.mp3" respectively.

In order to assign a mount point for a volume as a path within another volume, the following criteria must be met:

- The mounted-to volume must be formatted NTFS.
- A folder must exist at the root path. (As of Windows Vista, it can be any subfolder in a volume)
- That folder must be empty.

By default, Windows will assign drive letters to all drives, as follows:

- "A:" and "B:" to floppy disk drives, present or not
- "C:" and subsequent letters, as needed, to:
 - Hard disks
 - Removable disks, including optical media (e.g. CDs and DVDs)

Because of this legacy convention, the operating system startup drive is still most commonly assigned "C:", however this is not always the case. Since personal computers now no longer include floppies, and optical disc and other removable drives typically still start at "D:", letters A and B are available for manual assignment by a user with administrative privileges. This assignment will be remembered by the same OS on the same PC next time a removable volume is inserted, as long as there are no conflicts, and as long as the removable drive has not been reformatted on another computer (which changes its volume serial number), and as long as the OS has not been reinstalled on the computer.

On Windows XP, mount points may be managed through the Disk Management snap-in for the Microsoft Management Console. This can be most conveniently accessed through "Computer Management" in the "Administrative Tools" section of the Control Panel.

More than one drive letter can refer to a single volume, as when using the SUBST command.

Warning: removing drive letters or mount-points for a drive may break some programs, as some files may not be accessible under the known path. For example, if a program is installed at "D:\Program Files\Some Program", it may expect to find its data files at "D:\Program Files\Some Program\Data". If the logical disk previously called

"D:" has its drive letter changed to "E:", "Some Program" won't be able to find its data at "D:\Program Files\Some Program\Data", since the drive letter "D:" no longer represents that volume.

Unix-like operating systems

In Unix-like operating systems, volumes other than the boot volume have a mount-point somewhere within the filesystem, represented by a path. Logically, the directory tree stored on the volume is grafted in at the mountpoint. By convention, mount-points will often be placed in a directory called '/mnt', though '/media' and other terms are sometimes used.

As in Windows, to use a given path as a mount-point for another volume, an empty directory (sometimes called a folder) must exist there.

Unix-like operating systems use the `mount` command to manipulate mount points for volumes.

For example, if a CD-ROM drive containing a text file called 'info.txt' was mounted at '/mnt/iso9660', the text file would be accessible at '/mnt/iso9660/info.txt'.

Benefits of keeping files within one volume

Speed of data management


Files within a volume can generally be moved to any other place within that volume by manipulating the filesystem, without moving the actual data. However, if a file is to be moved outside the volume, the data itself must be relocated, which is a much more expensive operation.

In order to better visualize this concept, one might consider the example of a large library. If a non-fiction work is originally classified as having the subject "plants", but then has to be moved to the subject "flora", one does not need to refile the book, whose position on the shelf would be static, but rather, one needs only to replace the index card. However, to move the book to another library, adjusting index cards alone is insufficient. The entire book must be moved.

Special functions of advanced filesystems and volumes

Some filesystems, such as the Unix File System (ufs), Microsoft's NTFS filesystem, and ext3, allow multiple pseudonyms (known as "hard links") to be created for a single file within the same volume. Hard links allow a file to be referenced by two separate filenames, without its data being stored in two places on the disk (and thereby consuming twice as much space). Hard links cannot be created for files between volumes; this is comparable to moving the file in the library example above. To return to the library analogy, this is like filing two index cards for the same book: one could file the above book under both 'flora' and 'plants'. In general, deleting one hard link does not immediately effect other hard links, while deleting the final hard link for a file frees the disk space occupied by that file. However, modifying the data of the file referred to by one hard link will impact all other hard links as well. In the library, this is comparable to writing in the book. Hard links should not be confused with aliases (Mac OS), shortcuts (Windows), or soft links (another type of link under Unix and variants), which can refer to files on another volume or no file at all. FAT filesystems, such as FAT32, do not support hard or soft links as such, although the Windows operating system supports 'links', which are somewhat less capable.

Volume label and serial number

A **volume label** is the name given to a specific volume in a filesystem. In the FAT filesystem, the volume label was traditionally restricted to 11 characters (reflecting the 8.3 restrictions, but not divided into name and extension fields) even when long file name was enabled, stored as an entry within a disk's root directory with a special volume-label attribute bit set, and also copied to an 11-byte field within the Extended BIOS Parameter Block of the disk's boot sector. The label is always stored as uppercase in FAT and VFAT filesystems, and cannot contain special characters that are also disallowed for regular filenames. In the NTFS filesystem, the length of its volume label is restricted to 32 characters, and can include lowercase characters and even Unicode. The `label` command is used to change the label in DOS, Windows, and OS/2. For GUI systems like Windows Explorer,  can be pressed while the volume is highlighted, or a right-click on the name will bring up a context menu that allows it to be renamed, either of which is the same process as for renaming a file. Changing the label in Windows will also change the volume creation timestamp to the current date and time for FAT filesystems. NTFS partitions have the System Volume Information folder, whose creation timestamp is set when Windows creates the partition, or when it first recognizes a repartitioning (the creation of a new volume) by a separate disk utility.

In contrast to the label, the volume serial number is generally unique and is not normally changed by the user, and thus acts as a more consistent and reliable identifier of when a volume has been changed (as when a disk is removed and another inserted). Disk formatting changes the serial number, but relabeling does not. The `vol` command can be used from the command line to display the current label and serial number of a volume.

External links

- MSDN's article on Hard Links and Junctions (<http://msdn.microsoft.com/en-us/library/aa365006.aspx>)

Retrieved from "[http://en.wikipedia.org/w/index.php?title=Volume_\(computing\)&oldid=574781200](http://en.wikipedia.org/w/index.php?title=Volume_(computing)&oldid=574781200)"

Categories: Disk file systems

-
- This page was last modified on 27 September 2013 at 19:40.
 - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy.
- Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.