

Mount points, volumes, and physical drives, oh my!

 devblogs.microsoft.com/oldnewthing/20201019-00

October 19, 2020



Raymond Chen

At the bottom of the storage hierarchy are physical drives. These are units of physical storage, access to which is governed by a single disk controller. Your SSD or hard drive is a physical drive.¹

The next layer up is the volume. A volume is region of storage that is managed by a single file system. The relationship between volumes and physical drives is typically one-to-one, but it doesn't have to be.

For example, a physical drive might not have any volume associated with it at all. For example, it could be a raw hard drive that hasn't been partitioned or formatted yet.

You might take your physical drive and create multiple partitions, and then format each partition separately. Each of those formatted partitions is its own volume.

Or you might get really fancy and use a feature like spanned volumes or Storage Spaces to take multiple physical drives and combine them into one giant volume.

Once you have your volumes, you need to make them accessible somehow.²

Mount points are places that volumes are inserted into the namespace and become paths. The most usual place to see them is as a drive letter. For example, your system boot volume is almost certainly mounted as `C:`.

Volumes don't have to be mounted as drive letters, though. You can also mount them inside a subdirectory of an existing volume, sort of like grafting one tree onto another. One way to do this is by going to the Disk Management tool, right-clicking a volume, and selecting *Change drive letter and paths*. From there, you can add a path for a volume, and the contents of the volume will be visible via that path.

Note that a volume can be mounted in multiple places, or it might not be mounted at all.

Next time, we'll look at how to navigate these concepts in code.

¹ Paradoxically, you can have virtual physical drives, like a RAM drive.

² Theoretically, you could eschew mounting the volume and just access it via its volume GUID.

```
C:\>type \\?\Volume{8c3513a4-d064-4c99-81fc-66e20810ec3c}\windows\win.ini
; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1
```

I mean, theoretically you could do that, but you'd also be a little bit crazy.

[Raymond Chen](#)

Follow

