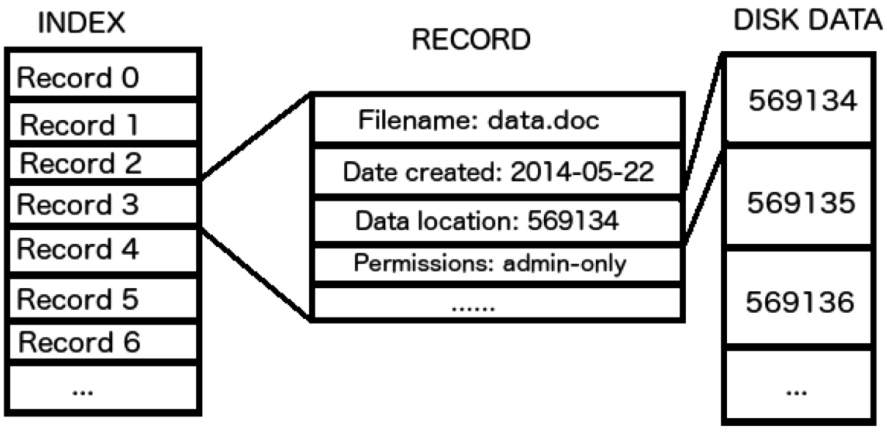
Fixity and Filesystems: Enhanced System Monitoring via inodes

by alex duryee
digital & metadata preservation specialist
avpreserve

AVPreserve | Media Archiving & Data Management Consultants
350 7th Avenue, Suite 1605 | New York, New York | 10001
917.475.9630 | www.avpreserve.com | info@avpreserve.com

If you have explored our checksum tool, Fixity, you may have noticed that it tracks a third value alongside the expected filepath and checksum value of files. This value is the file's index location, which is crucial to the operation of the filesystem. On OSX and Linux, the index value is called the inode of a file; on Windows, it is the file identifier.

Put abstractly, a filesystem typically stores file metadata (name, permissions, filepath, etc.) in a filesystem index. These index entries also document the physical location on the disk where the data is located, and thus link the file metadata to its content data. For purposes of efficiency, the filesystem prefers to change the file metadata to altering its on-disk data - if you have ever wondered why moving files within a filesystem is almost instantaneous, whereas copying is slow, it is because files are 'moved' by changing the filepath metadata in the index, not the physical location on the disk. When a file's data is changed, such as when a document is edited and saved, the metadata remains in place while the new data is written to disk. This method of handling data allows for quick discovery of directories and files, as well as maintaining separation between data and metadata.



Simplified example of the relationship between the filesystem index and data

Fixity takes advantage of this (meta)data structure by tracking both the index location of a file and the checksum of its on-disk data. One of the inspirations for developing Fixity was the lack of powerful and easy-to-use file monitoring tools in both the open-source and commercial marketplaces. Consider the following common scenario: an archive decides to implement a new directory standard for its digital repository, and changes the locations of its digital assets accordingly. Most checksum tools, which use filepaths as the unique identifier of a file, will cease to recognize the digital objects post-move - they will report that one object was deleted and one object was created. For a repository environment, where tracking assets is more important than tracking filenames, this is unacceptable. Thus, a more sophisticated system was implemented in Fixity.

Fixity offers the unique capability of tracking file attendance as well as file integrity. Since a file's index location only changes when a file is deleted or moved to a new storage device, it can be used as a unique identifier for a file, no matter its location or name on a given filesystem. Therefore, if a file is moved from one directory to another, it will keep the same index location and data content, but its filepath will be updated in the metadata. In this instance Fixity will triangulate the three pieces of information that it maintains about an object - filepath, checksum, and index number - and report that the object was moved from the old location to the new

one. The file is not considered missing and will continue to be tracked as the same file in a new directory location. Similarly, if one object's filepath is assigned to another existing file, Fixity will recognize them as different objects and not confuse the two.

The additional tracking of index locations raises the question of index collisions cropping up in between Fixity scans. If, for example, a file is deleted and a new file is created between scans, will the new file have the same index location as the old one? This is a reasonable concern, and was tested by AVPreserve on an OSX system (using the HFS+ file system) and on Windows 7 (using the NTFS filesystem). Our testing methodology involved the creation of ten thousand 10KiB files, followed by their deletion, and the creation of ten thousand more 10KiB files immediately afterwards. Through this process, we found that HFS+ assigns new index locations for files sequentially - no index locations used by the first set of files were used again by the second. NTFS produced the same results - after creation, deletion, and creation again, there were no index collisions between the two sets of ten thousand 10KiB files. Given that most filesystems assign index values sequentially, only repeating after exhausting all possible values, a collision within a repository monitored by Fixity is functionally impossible between two scans.

There are some notable exceptions to the handling of index values within file systems. The behavior of certain programs will change the index location of a file. Many programs, such as text editors, do not edit files in place on the filesystem. Instead, they create temporary copies of the file, which then store the changes to the file being made within the program. When the changes are committed to disk, the temporary file (containing the changes) is copied over the original. This process is used by programs in certain cases because it allows for greater crash resilience. However, the temporary copy is a completely new file and it is given a new file index location. As such, after using a program such as TextEdit, you may find that Fixity gives an unexpected result for the file. This is due to the editing program's behavior, and not an issue with Fixity or your file.

Finally, as stated above, index values change when the storage device changes. This raises the question of how Fixity handles the scenario of a movement of directories to a new storage device, such as the movement of a collection of files to a new server. When a user changes the path of a directory to scan within Fixity, the user is prompted with the question of whether or not the user intends to change the filepath (as opposed to deleting and adding a different one). If the user answers "yes", Fixity will expect that the index value will change and will use the filepath and checksum for verification on the first scan performed after the change. During this scan, the new index values are captured and used moving forward.

Taking advantage of index values allows for Fixity to better monitor the digital assets under its watch. We hope that its combination of powerful monitoring and ease of use will improve the level of data integrity in digital collections. Fixity is available for free download at **avpreserve.com/tools**