Compact Disc Digital Audio (CDDA) Filesystem Module Requirements Specification

Author: Chris Sarcone
Group: Mass Storage Software
Status: Version 1.0
Date: 11/03/09

Change History

Version	*Section/Page Affected*	*Change Description*	*Author(s)*	*Release Date*
Draft 1	*All pages*	*First version*	*C. Sarcone*	*11/03/09*

1.0 Definitions, Acronyms, Abbreviations

Address Space	- The memory allotment for a task (sometimes called a process)
AIFF	- Audio Interchange File Format
API	- Application Programming Interface
ATAPI	- Advanced Technology Attachment Packet Interface
Block	- The smallest amount of data which can be read from a storage device
BlockSize	- The size (in bytes) of a block
BSD	- Berkeley Standard Distribution (a UNIX variant)
CD	- Compact Disc
CD-ROM	- Compact Disc Read Only Memory
CDDA	- Compact Disc Digital Audio (sometimes written as CD-DA)
daemon	- pronounced de-mon, this is an application which is constantly running on a UNIX-like operating system
DVD	- Digital Versatile Disc (sometimes incorrectly called Digital Video Disc)
DVD-RAM	- Digital Versatile Disc Random Access Memory
DVD-ROM	- Digital Versatile Disc Read Only Memory
IEEE-1394	- A high speed serial bus used for data transfer
FireWire	- Apple's implementation of IEEE-1394
Frame	- Approximately 1/75 of a second
HTTP	- Hyper Text Transfer Protocol
KB	- KiloBytes (1024 bytes)
Kernel	- The "guts" of OS X which includes mach, virtual memory, IOKit, and anything else residing in the kernel's address space
Mach	- The Microkernel designed at CMU by Avie Tevanian and colleagues The term Mach really only describes the microkernel, but is sometimes colloquially used to mean the entire kernel address space and execution layer
Mach/BSD	- Term used to describe the aggregate foundation on which OS X is built. Some of the foundation comes from BSD, and the core of BSD has been replaced with the Mach microkernel.
MB	- MegaBytes (1024 KB)
MSF	- Minutes, Seconds, Frames
SCSI	- Small Computer Systems Interface
Track	- A song on an Audio CD
TCP/IP	- Transmission Control Protocol / Internet Protocol
UNIX	- An Operating System developed in the 1970s a type of which (BSD) is at the core of OS X
URL	- Uniform Resource Locator
VM	- Virtual Memory. Term used to describe the use of secondary memory (e.g. hard disk) as physical primary memory (e.g. RAM).

VFS - Virtual FileSystem
VNODE - Virtual node

1.1 References

1.1.1 MacOS X References:

IOKit documentation found at <coreos.apple.com/iokit/>
Mach 3.0 documentation found at <www.cs.cmu.edu/afs/cs/project/mach/public/www/doc/documents_top.html>
BSD documentation found at <www.freebsd.org>, <www.netbsd.org>, <www.openbsd.org>
Core Foundation documentation found at <developer.apple.com>

1.1.2 Stackable FileSystem References

McKusick et al., The Design and Implementation of the 4.4 BSD Operating System, Addison-Wesley

Heidemann and Popek, "File-System Development with Stackable Layers," ACM Transactions on Computer Systems, vol. 12, no. 1, pp.58-59, February 1994.

1.1.2 CD/DVD References

ATAPI SFF-8020i
ATAPI SFF-8090i
SCSI Multimedia Commands (MMC-2) www.t10.org

2.0 Problem Set

Mac OS X needs to be able to mount Audio CDs as discs in the Finder and allow users to copy the music tracks on the discs to another location, to play the music from the tracks in applications such as iTunes and QuickTimeX, and to view the names of the tracks in the Finder window. All applications shall be able to access the musical tracks while the disc is mounted.

The goal of this project is to create a filesystem, the CDDA FileSystem, which mounts the digital audio portions of a CD and allows the Finder and other applications to read this digital data as they would any other file. The files will be presented as AIFF-C-encoded sound files with the attributes of 16-bit stereo sound and 44.1kHz sampling. This means the Finder can directly copy files from a CDDA filesystem to another filesystem (e.g. HFS+) and the data will be preserved in the copy. The CDDA FileSystem will support all required VFS operations for BSD compatibility which in turn guarantees compatibility with Cocoa and Carbon clients as well.

2.1 User experience

When the user inserts a disc with CD-DA tracks encoded on it, the system is expected to mount the disc automatically, like it does with other CD-ROM data, and present the user with files which correspond to each track on the disc which contains digital audio data. It is important to

note that digital audio tracks can be mixed in with data tracks (Enhanced CDs, CD+ discs, etc.), so any disc that has any tacks where the Table Of Contents of the disc indicates digital audio tracks are valid, shall be mounted via this filesystem.

The user shall be able to do the following things with each individual track:

1) Listen to the track in the Finder / QuickLook preview.
2) Listen to the track in QuickTimeX, or any other application which knows how to play AIFF content.
3) Listen to the track in iTunes.
4) Drag copy the track in Finder to another disk
5) Play said copies in Finder, QuickLook preview, QuickTimeX, iTunes, and any other player that understands AIFF content.
6) Eject the disc using the Finder to eject it.

3.0 Performance Target

A major goal of this filesystem is to stream data directly from the optical disc whenever possible. Since playback does not require much bandwidth (~170Kbps/sec), the real performance target to meet is copying the files. The filesystem shall be able to copy the files at the speed of the optical disc drive used (typically ~3MB/s). A typical 4 minute song is ~44MB, so it shall take ~15 seconds to copy a typical track from the disc to another disk.

4.0 Supported Features

The filesystem shall support all of the use cases outlined in §2.1. In addition, if iTunes has track/album title information, those names shall show up in the filesystem. Finally, you shall be able to use the Terminal to copy files from the disc to another disk.

5.0 Target Footprint

The code shall be small (< 100K for multi-architecture builds), since it simply has to implement a few methods to be a filesystem and the runtime footprint shall be small as well (<50K), since only a handful of data structures are required for the filesystem to do its work.