Identifying Files, Directories, and Volumes

Updated methods

Your application typically specifies a filename and location when it calls the **File Manager** to open or delete a file. It typically receives filenames and locations from the **Standard File Package** and the Finder, which handle the user interface for creating, saving, opening, and removing files.

File System Specifications

Conventions for identifying files, directories, and volumes have evolved as the **File Manager** has matured. Version 7.0 introduces a simple, standard form for identifying a file or directory, called a file system specification. You can use a file system specification whenever you must identify a file or directory for the **File Manager**.

The file system specification contains

- the volume reference number of the volume on which the file or directory resides
- the directory ID of the parent directory
- the name of the file or directory

For a complete description of the new data structure, the file system specification (**FSSpec**) record, see **Using FSSpec Records** under **Using the File Manager**.

The **Standard File Package** in system software version 7.0 uses **FSSpec** records to identify files to be saved or opened. The **File Manager** provides a new set of high-level routines that accept **FSSpec** records as input. Applications can pass the data directly from the **Standard File Package** to the **File Manager**. The **Alias Manager** and the **Edition Manager** accept file specifications only in the form of **FSSpec** records.

The Finder in version 7.0 uses alias records, which are resolved into **FSSpec** records, to identify files to be opened or printed. (The description of required Apple events in the **Apple Event Manager** explains how the Finder passes file information to your application and how your application retrieves it.)

Version 7.0 also introduces the **FSMakeFSSpec** function, which creates an **FSSpec** record for a file or directory. For a complete description of **FSMakeFSSpec**, see the sections entitled **Using FSSpec Records** under **Using the File Manager** and **Making FSSpec Records** under **High-Level File Manager Routines**.

File Specification Strategies

The original Macintosh File System (MFS) is a flat file system-that is, a system in which all files are stored at the same level on a volume (the volume is not subdivided into directories). To uniquely identify a file, you need to specify only a volume and a filename. In high-level MFS functions, you pass the specification in parameters called vRefNum and fileName.

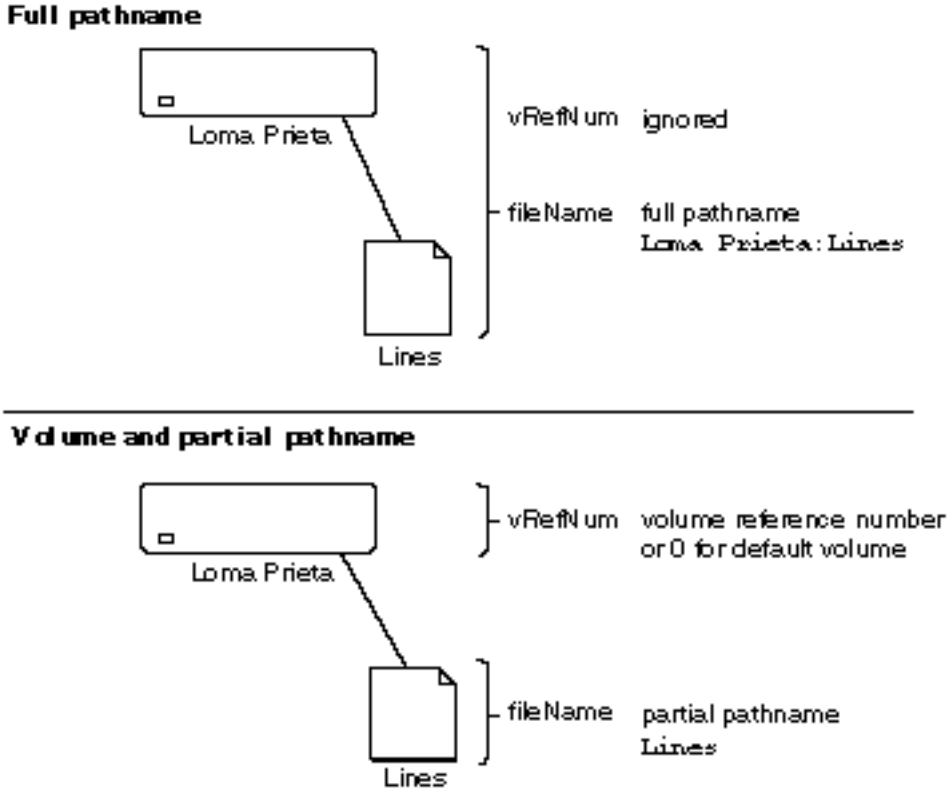
In MFS, you can specify a file in either of two ways:

- a full pathname, which contains the names of both the volume and the

file, in the fileName parameter

- filename by name in the fileName parameter and volume by volume reference number, a unique number assigned when the volume is mounted, in the vRefNum parameter

The following figure illustrates the two ways to identify a file in MFS.



To improve performance, especially with larger volumes, Apple Computer, Inc., introduced the Hierarchical File System (HFS) on the Macintosh Plus computer and later models. In HFS, a volume can be divided into smaller units known as **directories,** which can themselves contain files or other directories. Each file on an HFS volume is stored in a directory. To identify a file in HFS, you must specify its volume, its parent directory, and its name. The **File Manager** assigns each directory a directory ID, and the user or the system software assigns each directory a name. The Hierarchical File System **File Manager** routines include an additional parameter, the dirID parameter, to handle the directory specification.

In HFS, each folder created by the user is a directory. The folders represent a true hierarchy in the file system. In MFS, folders are an illusion maintained for the user by the system software. The first-level directory on a volume, the one that contains all of the other directories, is known as the root directory.

For compatibility between HFS and MFS, Apple introduced the concept of **working directories.** A working directory is a combined directory and volume specification. To make a directory into a working directory, the **File Manager** establishes a control block that contains both the volume and the directory ID of the target directory. The **File Manager** then returns a

unique working directory reference number, which you can use to identify the directory. You can use the working directory reference number in place of a volume specification in all of the MFS functions.

> **Note:** Working directories were introduced solely for compatibility between HFS and MFS. If you are writing an application to run in system software version 7.0, you do not need to use working directories.

In summary, HFS recognizes three kinds of file system objects: files, directories, and volumes. You can identify them using these labels:

File	**filename**
Directory	directory name
	directory ID
	Working directory reference number, which also implies a volume
Volume	volume name
	volume reference number
	working directory reference number, which also implies a directory

In HFS, you can pass a complete file specification in any of four ways:

- full pathname
- volume reference number and partial pathname
- working directory reference number and partial pathname
- volume reference number, directory ID, and partial pathname

A full pathname consists of the name of the volume, the names of all directories between the root directory and the target, and the name of the target. A full pathname starts with a character other than a colon and contains at least one colon. If the first character is a colon, or if the pathname contains no colons, it is a partial pathname. If a partial pathname starts with the name of a parent directory, the first character in the pathname must be a colon. If a partial pathname contains only the name of the target file or directory, the leading colon is optional.

You can identify a volume in the vRefNum parameter by volume reference number or drive number, but volume reference number is preferred. A value of 0 represents the default volume. A volume name in the pathname overrides any other volume specification. Unlike a volume name, a volume reference number is guaranteed to be unique. It changes, however, each time a volume is mounted.

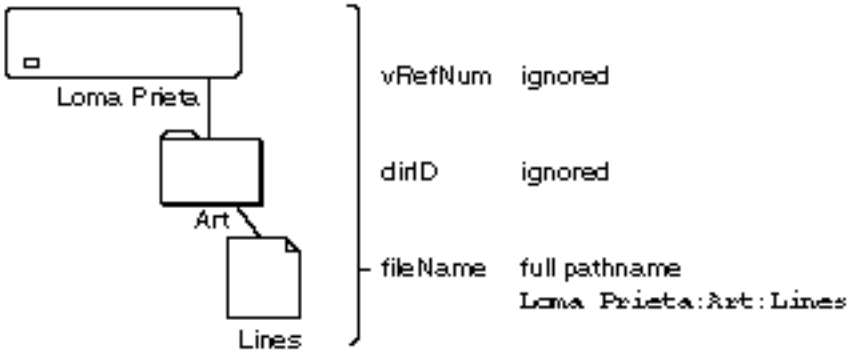
> **Note:** The system software that accompanied the release of HFS introduced the **PBHSetVol** function for setting the default volume on a hierarchical disk. Do not use the **PBHSetVol** function or the

high-level version, **HSetVol**, which is available in some development environments. If you need to set the default volume, use **SetVol** or **PBSetVol**.

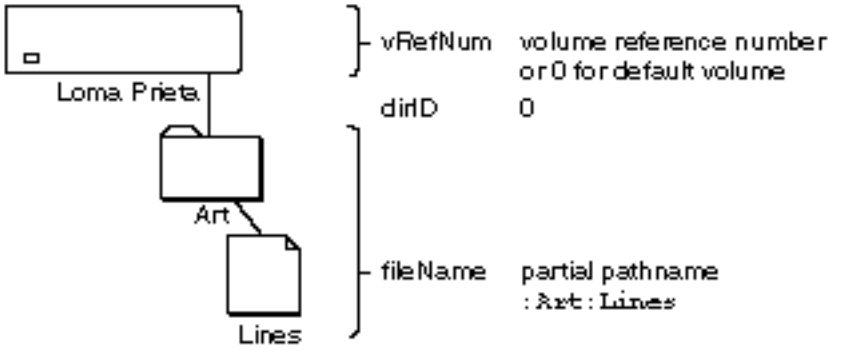
A working directory reference number represents both the directory ID and the volume reference number. If you specify any value other than 0 for the dirID parameter, that value overrides the directory ID implied by a working directory reference number in the volume parameter. The volume specification remains valid.

The four ways to identify a file in HFS are illustrated in the following figure.

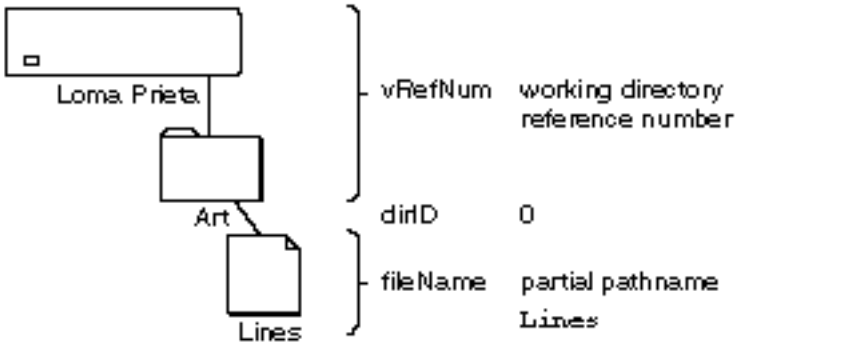
Full pathname



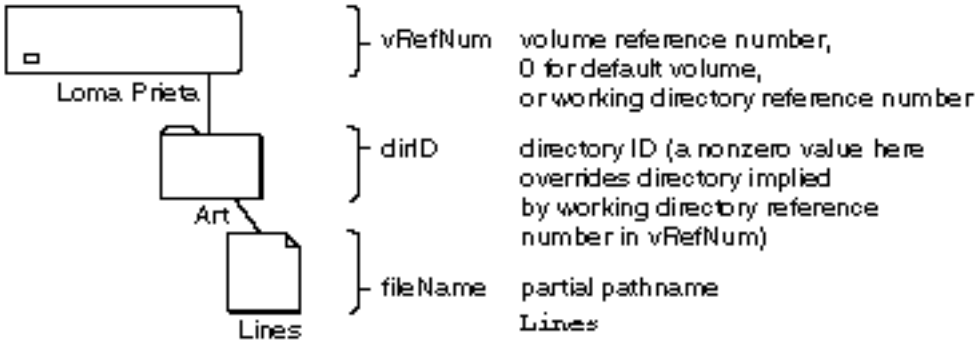
Volume and partial pathname



Working directory and partial pathname



Volume, directory ID, and partial pathname



The **FSSpec** record described in the previous section, "File System Specifications," replaces both the MFS and the HFS conventions for identifying files and directories in most cases. In system software version 7.0, you use the historical forms primarily when you are calling low-level **File Manager** functions.