About Alias Records Data structures that describe a file, directory or volume

An AliasRecord is a data structure that describes a file, directory, or volume. The record contains:

- location information, such as name and parent directory ID
- verification information, such as creation date, file type, and creator
- volume mounting information (that is, server and zone), if applicable

By storing an AliasRecord, you can allow your users to create a robust connection to a file-that is, a connection that can survive the moving or renaming of the target file. The Finder in system software version 7.0, for example, stores an AliasRecord in aliases created by the user to represent other files or folders. The **Edition Manager** uses an AliasRecord to support data sharing among separate documents. (The **Finder Interface** and **Edition Manager** describe those features in detail.)

An AliasRecord is a reliable way to identify a file system object when your application is communicating with a process that might be running on a different machine.

The creation of an AliasRecord has no effect on the target of the record, except to establish a file ID if one did not previously exist for the target file. (See the **File Manager** for a description of file IDs.)

The AliasRecord contains only two fields of public information available to your application. The bulk of the record is managed privately by the **Alias Manager**.

Your application can use the userType field to store its own signature or any other data that fits into 4 bytes. When the **Alias Manager** creates an AliasRecord, it stores 0 in that field.

The **Alias Manager** stores the size of the record when it was created in the aliasSize field. Knowing the starting size allows you to store and retrieve data of your own at the end of the record (see **Customizing Alias Records** under **Using the Alias Manager**). An AliasRecord is typically 200 to 300 bytes long.

The private **Alias Manager** data includes all of the location, verification, and mounting information needed to resolve the AliasRecord with the various search strategies described in **Search Strategies for Resolving Alias Records**.

When you create an AliasRecord, you have the option of recording a **relative path**, that is, a path to the **target** from another file or directory on the same volume. (Relative paths don't work across volumes.) The beginning point of a relative path is called the fromFile. To record a relative path, the **Alias Manager** saves the distances from the target and the fromFile to their **common parent,** that is, the lowest-level directory that appears in the pathnames of both. The **Alias Manager** can later use those distances in conjunction with the full pathname to conduct a relative search.

Suppose, for example, that you are writing a word-processing application that allows the user to build a customized, supplemental dictionary for each

document. You create the dictionary as a separate document in the same directory as the document it serves.

When resolving the **AliasRecord** by using a relative path, the **Alias Manager** starts at the directory that is the specified distance above the fromFile, then constructs a partial pathname by extracting one field of the absolute pathname for each step from the target to the common parent. In this example, the distance is one, so the pathname contains only the name of the target document, Dictionary.

In some circumstances, a relative search identifies the correct target when a direct search cannot. For example, suppose the user of your word-processing application creates a working copy of a document and dictionary by copying the entire folder Sample to another disk. The user later updates the original document and dictionary by copying the folder from the working disk. All of the underlying file and directory identifications change, but the filenames and relative path remain the same. When the user later runs the spelling checker on the document, a relative-path search finds the correct target dictionary.

The **Alias Manager** accepts and returns file specifications in the form of file system specification records (**FSSpec** records), described in the **File Manager** section. The **FSSpec** record represents a standard, complete description of a file system object. It contains a volume reference number, a parent directory ID, and a name.