Aliases and the Finder

The **Finder** for System 7.0 allows the user to create multiple desktop icons to represent a single document or other desktop object (such as a disk, a folder, or the Trash). One of the icons represents the actual file; the others are aliases that point to the file. An **alias** is an object on the desktop that represents some other file, directory, or volume. An alias looks like the icon of its target, but its name is displayed in a different style. The style depends on the system script; for Roman and most other scripts, alias names are displayed in italic.

To the user, the icons of the actual file and its aliases are functionally identical. Aliases give the user more flexibility in organizing the desktop and offer a convenient way to store a local copy of a large or dynamic file that resides on a file server.

As a desktop object, the alias depicts a file called the alias file**,** which contains a record that points to the file, directory, or volume represented by the icon. Alias files are created and managed by the user through the **Finder**. Although your application should not create alias files or change users' desktop aliases, your application can create and use its own alias records for storing identifying information about files or directories. An **alias record** is a data structure that identifies a file, folder, or volume. Whenever your application needs to store file or directory information, you can record the location and other identifying information in an alias record. The next time your application needs the file or directory, you can use the **Alias Manager** to locate it, even if the user has renamed it, copied it, restored it from backup, or moved it. You can also use alias records to identify objects on other volumes, including **AppleShare** volumes. See the **Alias Manager** for details about creating and managing information in alias records.

Ordinarily, when the user wants to open or print files, your application does not need to be concerned with whether they are aliases because both the **Finder** and the **Standard File Package** resolve aliases before passing them to your application. If the user opens an alias that represents a document created by your application, the **Finder** passes your application the name and location of the document itself, not the alias. Similarly, when the user opens an alias from within your application, the **Standard File Package** passes your application the name of the target document.

If your application bypasses the **Finder** or the **Standard File Package** when manipulating documents, it should check for and resolve aliases itself by using the **ResolveAliasFile** function.