Creators, File Types, and the Signature Resource

The **Finder** identifies your application through its **signature,** a unique, four-character sequence. The signature must not conflict with the signature of any other application. To ensure uniqueness, you must register your application's signature with Apple at Macintosh Developer Technical Support.

> **Note:** There is no need to register your own resource types because they're usually used in your own applications or documents only.

You must include in your resource file a special resource that has the application's signature as its resource type. By convention, the signature resource has a resource ID of 0. The signature resource typically contains a string that specifies the name, version number, and release date of your application. If you do not provide specific version information through a 'vers' resource (described in Version Resources), the **Finder** displays the string stored in the signature resource when the user selects your application and chooses Get Info from the File menu.

The following sample code illustrates a signature resource in Rez input format.

```
type 'WAVE' as 'STR ';                  /* WAVE is the signature */
resource 'WAVE' (0, purgeable) {            /* resource ID is 0 */
    "MyApplication 2.0 © 1991"              /* default Get Info string */
};
```

> **Note:** The signature resource alone is not sufficient to fully establish your application's signature. You must also supply a bundle resource, described in **The Bundle Resource**.

Whenever your application creates a document, it assigns the document a creator and a file type. Typically, as described in **Finder Information in the Volume Catalog**, your application sets its signature as the document's creator. When a user double-clicks a document or selects it and chooses Open or Print from the **Finder**'s File menu, the **Finder** reads the creator field of that file to find the document's creator. The **Finder** then searches for an application with a signature by that name. When it finds that application, the **Finder** launches that application.

If the document's creator is your application's signature, for example, the **Finder** calls the **Process Manager** to start your application. The **Finder** sets up the information your application needs to open or print the document. In System 7.0, applications that support high-level events (that is, have the isHighLevelEventAware flag set in the 'SIZE' resource) receive the **Finder** information through Apple events.

Applications that do not support high-level events can use the **CountAppFiles**, **GetAppFiles**, and **ClrAppFiles** procedures or the **GetAppParms** procedure to get the **Finder** information. For information on these routines, see the **Segment Loader**.

As described in **Finder Information in the Volume Catalog**, your appli-cation typically assigns a file type to a document when it creates one. The file type can be a type especially defined for your application, or it can be one of the existing general types, such as those listed here.

File type	Description
'APPL'	Launchable application
'DFIL'	File for storing desk accessories
'DRVR'	Driver
'FFIL'	File for storing fonts
'INIT'	System extension
'PICT'	**QuickDraw** picture
'PRER'	Printer driver
'RDEV'	Chooser extension
'TEXT'	Stream of ASCII characters
'adev'	Network extension (like EtherTalk 2.0)
'appe'	Background-only application
'cdev'	Control panel
'edtp'	Edition for sharing graphics-oriented data
'edts'	Edition for sharing sound-oriented data
'edtt'	Edition for sharing text-oriented data
'ffil'	Font
'ifil'	Script system resource collection
'kfil'	Keyboard layout
'pref'	Preferences file
'qery'	Query document for database access
'scri'	System extension for script systems
'sfil'	Sound
'tfil'	TrueType font
'ttro'	**TeachText** read-only file
'zsys'	A system file (such as the System file itself)

Note: Apple reserves the use of all signatures and file types whose names contain only lowercase and nonalphabetic characters. Your signature and the file types created especially for your application must contain at least one uppercase character. Since the system software never displays signatures and file types to users, signatures and file types can consist of character combinations that are incomprehensible to anyone but you.

Like signatures, file types must be registered with Apple. Your application must have a file type of 'APPL'. The creator field of your application file should contain its own signature. Most programming environments provide a simple tool for setting the creator field of your application file.

Your application can create documents of any type, and it can specify any application as the creator. You could write a utility application, for example, that creates a new document by opening one text file and appending onto it another text file. The application would give the new document the same creator as the first original text file so that the **Finder** can call on that application when the user wants to open or print the new document.

Assign the standard file type 'TEXT' to files that consist only of text-that is, a stream of characters with return characters at the ends of paragraphs. Most word processors allow the user to create text-only files. A document of file type 'TEXT' can be opened or printed by any application that accepts such file types. Your application can still assign its own signature as the file's creator so that the **Finder** can call on it to open or print the file when appropriate. Users can also open a document created by your application-as well as a document of a file type supported by your application-by selecting its icon and dragging it to your application's icon. Because the document's file type is

stored in the catalog and the **Finder** stores a list of your application's supported file types in the desktop database, the **Finder** can determine whether to launch your application. If the document's file type is supported by your application, the **Finder** launches your application and passes it the name of the document.

For example, if your application is a page layout program, it might create documents of its own file type while also supporting documents of 'TEXT' and 'PICT' file types. A user can launch your application by dragging a document of any of these file types to your application icon.

Your application also relies on file types to determine which files to let the user open when your application is running. When your application calls the **Standard File Package** to open a file, your application supplies either a list of the file types that your application can open or a filter function for those types. The open file dialog box then displays only files of the specified types. (See the **Standard File Package** description for details.)