SignatureToApp

The Documentation

by Jens Peter Alfke
User Programming Group
Apple Computer, Inc.

DISCLAIMER: This software has not been extensively tested beyond a simple "it seems to work—ship it!" It *does* seem to work (and is being made a part of several Apple products including MacApp® 3.0) so we *are* shipping it, but you should still keep in mind that it has not undergone any sort of official testing and is not officially blessed by Apple. At least you're getting the source code, so if you think it's doing something wrong you can try to fix it yourself. (Or you can contact me — my AppleLink account is JENS.)

FINDING APPLICATIONS BY THEIR SIGNATURES

```
typedef enum{
      Sig2App_FindProcess           =0,
      Sig2App_FindApplication       =1,
      Sig2App_LaunchApplication     =2
} Sig2App_Mode;

OSErr
SignatureToApp(  OSType sig,
                 ProcessSerialNumber *psn, FSSpec *fileSpec,
                 Boolean *launched,
                 Sig2App_Mode mode,
                 LaunchFlags launchControlFlags )
```

The `SignatureToApp` function takes an application's signature or creator code (a four-byte identifier) and attempts to find a running application, or an application on disk, with that signature. It may launch the application if you allow it to and if the application is not already running.

s `SignatureToApp` works only in System 7.0 or later. s

The `mode` parameter tells what to do. If it is `Sig2App_FindProcess`, the process list will be searched for a running application with the right signature. If it is `Sig2App_FindApplication`, disks will also be searched (see below) if no such application is running. If it is `Sig2App_LaunchApplication`, an application found on disk will also be launched.

If no error is returned, `SignatureToApp` will return the application's process serial number in the `psn` parameter (`kNoProcess` if it is not running) and its location on disk in the `fileSpec` parameter. The `launched` parameter will be set to true if `SignatureToApp` launched the application.

Note: You may set the `psn`, `fileSpec` and/or `launched` parameters to NIL if you don't care about the values returned in them.

The `launchControlFlags` parameter controls how the application will be launched. These flags are described in the Process Management chapter of Inside Macintosh VI. (If the application was already running, only the `launchDontSwitch` flag is used: if not set, the application will be switched to the foreground.)

To find an appropriate application on disk, `SignatureToApp` asks each mounted volume's desktop database (as described in the Finder Interface chapter of Inside Macintosh VI) to find an application with the desired signature. It first checks the volume containing the active System file, then all other mounted volumes.

s `SignatureToApp` cannot find applications on floppy disks since volumes less than 2MB in size do not have a desktop database. s

s Occasionally `SignatureToApp` may not be able to find an application on a file server. This turns out to be the fault of the desktop database on the server. A lot of server volumes seem to have messed-up desktop databases which may not have entries for all the applications on the volume. The solution is for the administrator to rebuild the desktops on the file server volumes. s

Result codes:

- When `mode` is `Sig2App_FindProcess`:

noErr	0	no error
procNotFound	-600	No application with the right signature is running (possible only when the `mode` parameter is equal to `Sig2App_FindProcess`.)
appIsDaemon	-606	Application is background-only, and the `launchInhibitDaemon` flag was set (occurs only if the application is already running)

- When `mode` is `Sig2App_FindApplication`:

noErr	0	no error
appIsDaemon	-606	Application is background-only, and the `launchInhibitDaemon` flag was set (occurs only if the application is already running)
fnfErr	-43	No application with the right signature was found on disk

- When `mode` is `Sig2App_LaunchApplication`:

noErr	0	no error
appIsDaemon	-606	Application is background-only, and the `launchInhibitDaemon` flag was set
fnfErr	-43	No application with the right signature was found on disk
memFragErr	-601	Not enough room to launch application with special requirements
appModeErr	-602	Memory mode is 32-bit, but application is not 32-bit clean (use the `launchAllow24Bit` flag to defeat this)
appMemFullErr	-605	More memory for the partition size is required than the amount specified in the SIZE resource
memFullErr	-108	Not enough memory to allocate the partition size specified in the SIZE resource