



Products Services Developer Resources Contact STS About STS



## Understanding Processes



Before you start working with code that manipulates processes, it is important to know just what **is** a process, and how can a process be identified.

### What is a Process?

Technically speaking, a process is any actively operating chunk of code on your computer. In practicality, however, we are mainly concerned with **application processes**, that is, processes that are currently running that may or may not provide an interface to the user, but have been designed to accomplish a predefined purpose. Keep in mind the difference between an **application** and an **application process**. A copy of SimpleText or Microsoft Word that is sitting on disk but not currently running is an **application**; start it up, and the code that is running is the **application process**.

There are other application processes that run on your computer that you don't see: things like the "Control Strip Extension" (MacOS 9), the "loginwindow" (MacOS X), or "winlogon.exe" (Windows). These are called **hidden** processes because, well, you can't see them.

### How Do I Identify an Application Process So I Can Work With It? ( )

To see the list of currently running processes, you can open up Apple's Script Editor, make sure the Result Window is showing, and enter and run this AppleScript:

```
tell application "Finder"
  get the processes
end tell
```

This will return a list; something like this (this is an example under OS X):

```
{application process "loginwindow" of application "Finder",
 application process "Dock" of application "Finder",
 application process "Script Editor" of application "Finder" ... }
```

*Note: In OS 9, the processes are all preceded with the term "process"; in OS X, the processes are all preceded with the term "application process". Just FYI.)*

In OS X, processes have a **process ID number** (or PID) that uniquely identifies one process from another, and through the Terminal (or equivalent) enables you to manipulate those processes. To see a list of running processes, you can use the process status command (ps). Open the Terminal application and type the following at the command prompt:

```
ps -aux -o command
```

This will display a list of running applications, including the paths to those applications.

But before you start working with Macintosh application processes, you need to understand the four different kinds of Mac applications:

- **Classic Application**

This application was most likely written in OS 9, and uses the Code Fragment Manager (CFM). It does not depend on anything that OS X provides, and when launched in OS X will run in the Classic environment.

**How You Know It's Classic:** Choose "Get Info" in the Finder — a Classic Application will show the Kind as "Classic Application".

**How You Can Identify Its Process Name:** The name of the application process for a Classic Application is simple; it is the name of the executable application file on disk. For example, if you launch QuickTime Player, the name of the process that is running is "QuickTime Player". If you change the name of the QuickTime Player on disk to "QuickTime Player Custom" and then launch it again, the name of the running process is now "QuickTime Player Custom".

- **Carbon Application (CFM)**

This application utilizes stuff in the Carbon Framework, but also uses the Code Fragment Manager (CFM). It depends on having the Carbon Libraries present (either as part of OS X or by using the CarbonLib extension that installs under OS 9). It can either run in the Classic Environment or natively in OS X. Note that just because an application *can* be set to open in the Classic environment, doesn't mean that the application developer intended to *allow it to run* in the Classic environment; if this is the case, you'll probably get a dialog box saying you need to open it in OS X.

**How You Know It's Carbon/CFM:** Choose "Get Info" in the Finder — if it has a checkbox that says "Open in the Classic environment", it is Carbon/CFM.

**How You Can Identify Its Process Name:** Like the Classic Application, it is identified by the name of the application file.

- **Carbon Application (Mach-O)**

This application also utilizes the Carbon Framework, but *cannot run* in the Classic environment; it can only run in OS X.

**How You Know It's Carbon/Mach-O:** Choose "Get Info" in the Finder — if it *does not have* a checkbox that says "Open in the Classic environment", it is Carbon/Mach-O.

**How You Can Identify Its Process Name:** OK. Here comes the fun... A Mach-O application has **TWO** identifiers; one for the *process* and one for the Application Menu that is displayed when the Mach-O program is running. If you are not careful, you might have an application called "MyApp" that has an application menu named "MyApp", but it can't be located through AppleScript because the process name is really "MetaCardCarbonMach-O".

If you Control-Click on a Mach-O application and choose "Show Package Contents", you will get to see *inside* a package. Look for the "Contents" folder and open it. Note that there is a file there called "Info.plist". Don't do anything with it yet, just know that it's there. Open the "MacOS" folder in the Contents folder. Inside is the name of the **bundle executable**, or as I like to call it, the **engine**. In the case of

MetaCard 2.4.3, this engine is called "MetaCardCarbonMach-O". The name of the application process that runs is based on the name of this engine file. Change the name of the engine, and you change the name of the process. But you can't just change the name of this file and call it a day. OS X needs to be *told* that you changed it, otherwise you'll get a dialog box that looks like this:



(This error dialog is displayed whenever there's something screwy in the package, BTW.) To tell OS X what the name of your executable is, you need to go back to that `Info.plist` file (just inside the Contents folder) and edit it in a text editor. Find the place where it says:

```
<key>CFBundleExecutable</key>
<string>MetaCardCarbonMach-O</string>
```

The string value here needs to be changed to exactly match the name you gave to the engine. If they are different, you'll get the error dialog above. Once you've changed this, save the `Info.plist` file and then try running your application. The Application Menu should say "MetaCard", but if you get a process list through AppleScript, you should see the name of your application as one of the processes.

To change the Application Menu name, you'll need to edit another key in the `Info.plist` file; the one that reads:

```
<key>CFBundleName</key>
<string>MetaCard</string>
```

If you change that to the name of your app and run it, you should now see the Application Menu reflecting your application accordingly. Note that if you had just changed the `CFBundleName` and **not** the `CFBundleExecutable` (and corresponding engine file), you would get an accurate Application Menu, but the process name would still say "MetaCardCarbonMach-O". So you see it is very important that you change **both** values in the `Info.plist` file, as well as changing the name of the engine file.

#### • Cocoa Application

This application utilizes the object-oriented Cocoa Framework. It can only run in OS X.

**How You Know It's Cocoa:** You can't tell through the Get Info window, and the sources I've looked at are in disagreement. But one test is to launch the program and hit the Help key on the keyboard; if it turns to a question mark, then it is probably a Cocoa app (try it with the Address Book or Clock).

**How You Can Identify Its Process Name:** This is exactly the same as for a Carbon/Mach-O application (see above).

#### How Do I Identify an Application Process So I Can Work With It? ( 🌐 )

To see the list of currently running processes, you can type Control-Shift-Escape; this should bring up the Windows Task Manager, and it should already be displaying the Processes tab, listing all the currently running processes. Application processes that are running usually end in `.exe`, and correspond to the name of the actual executable file on disk that was launched.

Of course in Windows, the name of the executable file and the name of the application itself are almost always different; Microsoft Word 2000 for example is identified as `WINWORD.EXE`.

Windows also keeps track of processes by process ID number (PID), so you can manipulate them as well using their unique ID number instead of the name of the application. PIDs are important, especially in Windows which allows you to have multiple instances of the same application running at the same time. These instances will have the same process name, but different PIDs.

This should give you a good understanding about processes in Mac and Windows; please keep the above in mind when working with application processes. Please let me know if I have misstated anything above so I can correct it.

*Posted 12/26/2002 by Ken Ray*



[News and Rumors](#)   [Products](#)   [Services](#)   [Developer Resources](#)   [Contact STS](#)   [About STS](#)

Copyright ©1997-2013 Sons of Thunder Software, Inc. All rights reserved.  
Send all comments to [webmaster@sonsofthunder.com](mailto:webmaster@sonsofthunder.com).