Development-Frameworks

Overview

A Framework is Apple's mechanism for packaging and distributing shared functionality. Frameworks can contain not only code, but also Resources and Web Server Resources much like a full-blown WOA.

Images deployed in a Framework

If a framework contains images, accessed from your main application, you must remember to change the framework binding in some of your WOComponents. The failure to do so will become evident in development. The default is "app", but you must change it to the name of your framework. When it comes time to deploy, you application frameworks must be included on the deployment machine. Images do not seem to display properly if you only put your frameworks in this location: /Library/Frameworks. You'll also want to put your webserver resources from your framework in the /WebServer/Documents/WebObjects/Frameworks folder. I just included the whole .framework folder in that directory and the images were served from the webserver properly. All the above paths are for MacOS X.

Framework Initialization

I have a framework that I would like to do some explicit initialization on. Something akin to doing a class static initializer in Java. Where is the proper place to place such code? Or can I just use a class static initializer?

Gary Teter

Add a text file called CustomInfo.plist to your framework's resources. Add this text to the file:

```
{
    NSPrincipalClass = MyFrameworkInitializer;
}
```

Create a java file, MyFrameworkInitializer.java and initialize any classes in your framework in a static block in this class. Of course you can call the class whatever you want, just make sure you put the full class name, package and all in the .plist file.

This is a hangover from NeXT days and may stop working if frameworks ever go away and WO switches to NORMAL java deployment. Basically the framework loader will reference every class declared as NSPrincipalClass? in this way. The neat part is your framework get's initialized w/o having to remember to do it yourself in th app constructor.

Arturo Pérez

I could not get things to work as described above. The problems:

- The Info.plist generated does not contain that NSPrincipalClass? reference and apparently nothing happens. If you do it as described, you actually end up with a CustomInfo.plist. I had expected the CustomInfo.plist to be merged into the Info.plist. I found that if you do want it merged into the Info.plist then you need to "Project-Edit Active Target" then under the Info.plist twistee there's an entry called "Cocoa Specific". There you find the entry for "Principal Class" and you put the initializer class there.

- The initializer class is called *way* early. I expected it to work like the database connections; i.e. after the application had been initialized. In fact, in my case, the framework is initialized right after the main. If you have the run log opened the initialization happens when the first <main> is printed after the classpath list is printed. So, I came to believe it wasn't happening because I was looking for the initialization much later.

HTH the next intrepid developer. BTW, the Xcode documentation for this is only tangentially useful to a WO developer. The panels described don't actually exist when you're doing a WO framework. The information described in terms of the necessary entries helps, though.

King Chung Huang

Tip: I often have my Framework initializers create a worker object to listen for ApplicationWillLaunchNotification or ApplicationDidLaunchNotification. if I need to do initialization just before or after the WOApplication instance is constructed.