(NeXT Tip #8) Application File Packages

Christopher Lane (*lane[at]sumex-aim.stanford.edu*)
Mon, 30 Nov 1992 15:35:57 -0800 (PST)

Starting in the early NeXT releases, there has been an alternate to the standard executable called the 'file package'. Instead of a single binary with things like InterfaceBuilder data in separate mach segments, the file package is a directory, with the extension *.app, that contains the executable, *.nib data and other components as separate files. The Workspace browser generally treats the *.app directories as a file and doesn't let you travel into them like normal directories. (You can view a *.app file package as a directory using the 'File>Open as Folder' menu command in Workspace.)

Although these have been around for a while, and are an option in Interface Builder under 2.1, all applications compiled under the 3.0 release are file packages. If you don't add any special mach segments to your application, via a Makefile.preamble for example, then this shouldn't cause you any concern as the normal process of upgrading an application through InterfaceBuilder (and ProjectBuilder which has been separated out under 3.0) should take care of things for you. If you did have special segments, you'll need to make these resources under ProjectBuilder instead and remove your special Makefiles.

To aid developers in using file packages, NeXT has added the NXBundle object that lets you get at the files in the package by type without knowing the full path to the application. See the 3.0 release notes for more details. When building an application, if only *.nib, icons, sounds, etc. have changed but not code, then remaking is much faster. Of course, you have to remember to run 'gdb' on the executable in the *.app directory, not the *.app itself! :-)

>From a user's point of view, some applications have changed their name under 3.0, e.g 'Draw' becomes 'Draw.app'. The 3.0 release will still execute the 2.1 multi-segment binaries (and you'll find lots of them still around), it just won't generate them. This means that you can't use InterfaceBuilder in 3.0 to fiddle with the __NIB segment of a 2.1 binary -- though some sites are running the 2.1 InterfaceBuilder under 3.0 to do 2.1 binary tuning.

So, the tip is 'the sooner you get comfortable with taking advantage of file packages the better'. Over time, more and more things in the NeXT world are going to be 'bundled' into directories that appear to the user as individual entities. Consider where your application can take advantage of this.

- Christopher