

Copyright 1993 Jeremy Slade.

You are free to use all or any parts of the Locus project however you wish, just give credit where credit is due. The author (Jeremy Slade) shall not be held responsible for any damages that result out of use or misuse of any part of this project.

JGS Fri Apr 3 21:20:14 GMT-0700 1992

Launching Items

When the ItemCells are told to Launch, the procedure used to do this is different depending on the type of file the item represents. The launching methods for each file type is discussed below.

Launching Files -- FT_FILE Type

Regular files are opened using the NXWorkspaceRequestProtocol, and in specific the

-openFile:withApplication:andDeactivate: method. If the Item's specificApp flag is set (meaning that it has been specified to open with a certain application), then the appName is passed as the withApplication: parameter, otherwise, NULL is used to indicate that the default application for that item should be used.

Launching Applications -- FT_APPLICATION Type

Launching applications is the trickiest part of the operation. The problem arises in the fact that the NXWorkspaceRequestProtocol's -launchApplication:showTile:autoLaunch: method doesn't quite perform as expected. The first time an application is launched, it works perfectly. Subsequent request to 'launch' (and the resulting call to this method) should result in the already-running application to be activated. Instead, another copy of the application is launched, with the side effect that the second copy does not have its own icon! This is most likely a problem with the Workspace, that will hopefully be fixed at some point.

In the mean time, I have been forced to implement a workaround to this problem. The first step is to determine if the app is already running, using the NXPortNameLookup() on the local host. If the app is not running then we go ahead and launch it using the workspace request method. If it is running, then we call [[NXApp appSpeaker]

launchProgram:app ok:flag], which is an undocumented method in 3.0, left over for existence with 2.x. This method does what I am trying to accomplish, and it will be there as long as future versions of NS have support for 2.x applications.

I've tried various other methods to accomplish this, such as sending the running app an activateSelf: or unhide: message through the Speaker/Listener classes, but these methods simply didn't do anything. This solution seems to be the best way to go for now.

Launching Folders -- FT_SUBDIR Type

Subdirectories/Folders are opened using the -selectFile:inViewerRootedAt: method of the NXWorkspaceRequestProtocol, which selects the specified directory in one of the Workspace Manager's File Viewers. If the Item's specificApp flag is set, it is launched just like a regular FT_FILE-type item (See above).

If the Item's createBrowser flag is set, then the directory is selected in a new File Viewer, which is rooted at the path specified by rootAt, or at the root directory ('/') if rootAt is not set. If createBrowser is not set, the directory is selected in the main File Viewer.