livecode

Setting Document Associations With OS X

Here's what you need to do in OS X... I'm assuming a creator code of "STS1" and an extension of "sts" - remember the original app was called "TestAE":

- Create your standalone (in my case it was "TestAE.app").
- Find your standalone in the Finder, control-click it and choose "Show Package Contents".
- In the folder that opens, open the folder called "Contents".
- In the Contents folder, double-click on "PkgInfo". It will launch TextEdit and it should contain the simple string "APPLMCRD". Change this to "APPLSTS1", save the file.
- The next step is to edit the "Info.plist" file; if you installed the Developer Tools that came with OS X, you will have an application called "PropertyList Editor" on your hard drive, which provides a UI for editing this file. If not, you can edit it in TextEdit, but you'll be manipulating XML. I've outlined both ways below:

If you have PropertyList Editor:

- Quit out of TextEdit (since you won't need it).
- Double-click "Info.plist", which will launch Property Editor.
- Expand "Root".
- Change the "CFBundleSignature" to "STS1".
- Expand "CFBundleDocumentTypes", then expand "0".
- Change "CFBundleTypeName" to a name you want your document to show in the Finder. I've changed mine to "TestAE document".
- Expand "CFBundleTypeExtensions".
- Change the value for "0" from "mc" to "sts".
- Save changes and quit the PropertyList Editor.

If you *don't* have PropertyList Editor:

- With TextEdit still open, open the file "Info.plist".
- Under the root level <dict>, find the subelement <array>, then the subelement of <array> called <dict>, then the subelement of <dict> called <array>, then the subelement of <array> called <string> that contains the value "mc". (For the purposes of the rest of this email, paths will be referred to using backslahes, so this location in the XML document would be \dict\array\dict\array\string.)
- Change the value from "mc" to "sts".
- Locate the \dict\array\dict\key with the value "CFBundleTypeName". Underneath that is a <string>. Change its value from "Metacard stack" to "TestAE document".
- Locate the \dict\key with the value "CFBundleSignature". Underneath that is a . Change its value from "MCRD" to "STS1".
- Save changes and quit TextEdit.

If you like, you can also change other data, such as copyright info ("NSHumanReadableCopyright") and version strings ("CFBundleShortVersionString", "CFBundleLongVersionString"). 6) Create your test stack, adding a ".sts" extension (mine was called "Dummy.sts").

Now comes the important part — reboot *twice*. Perhaps its the way I did things (create the dummy stack with the ".sts" extension first and then tweak the settings of TestAE.app), but I found that if I rebooted once, the dummy stack I'd created did not associate properly with TestAE.app.

BTW: You'll know if the association works if the "Kind" column in the Finder shows "TestAE document" and not "Document" or "Metacard stack".

One final note: You can change the icons that your standalone uses for documents by downloading an icon editor for OS X (I tried a cheap one called Icon Machine III), and opening the file "MetaCardDoc.icns" in the Resources folder of the Contents folder. I would assume that for professionally shipping applications you could change the names of the '.icns' files, so long as you changed the references in the Info.plist file ("CFBundleIconFile", "CFBundleTypeIconFile").

Hope this works for you; please report back to the list and let us know if these instructions are sound. If they are, I'll formalize them and put them as a Tip on my site that anyone can access as needed.

When a document with the proper association is double-clicked in the Finder, your application will launch, but you will need to use some AppleScript to determine the path of the file that was opened:

```
on appleEvent theClass,theID
   if theClass is "aevt" and theID is "odoc" then
      request appleEvent data
      put it into theFiles ## files OS is requesting your application opens, one per line
      if theFiles is not "not found" and theFiles is not empty then
         ## code to open theFiles
      end if
   else
      pass appleEvent
   end if
end appleEvent
```

Put that in your preOpenCard or preOpenStack script and you should be able to work with files that were opened from the Finder.

Posted 7/24/2002 by Ken Ray to the Use Revolution List (See the complete post/thread)
Updated 6/15/2013 by Ken Ray

Print this tip