



Products Services Developer Resources Contact STS About STS



Creating Custom Icons for Your Standalones and Stacks



In the development of applications using MetaCard or Revolution, it is important to be able to get your custom icons just right. Here's some tips on how to do just that. (Note that some of these are MetaCard-specific and will need to be adjusted to work with Revolution.)

Creating Custom Icons (X)

You need to set the creator of the standalone, and the stacks [used with the standalone].

Step 1:

The resource fork of the standalone, and stacks. Firstly, the stacks themselves do NOT need a resource fork. But the standalone does. In the stack that will become the standalone, you should have an ic14, ic18, icns, ics4, and ics8 resource. If you edit ic18, the rest should be edited for you automatically. Put your standalone icon into resource 128 of ic18. Put your stack icon into resource 129 and 130.

Step 2:

When you use the Standalone Builder, there is a field "Creator" that has "MSTD" in it. This is Metacard's creator code.

You'll need to use a custom code for this. Creator codes are 4 characters, letters or numbers, small or capital. If you plan to distribute the standalone to others, you should make sure not to use a code already in use. You can get creator code info at:

<http://developer.apple.com/dev/cftype/faq.html>

I register all mine with Apple thru this link. And check for existing codes before using one, to make sure it is not already registered. For Blackjack Gold I used "bJg3".

Enter this code in the "Creator" field when you build the standalone.

Step 3:

You must save the icons as files, specifically ".icns" files, and put them into your "Resource" folder that Metacard OSX will create in the OSX package of your program. [Control-click on the application, choose "Open Package Contents" from the context menu.] Save one for the standalone, and one for the stacks. Replace the Metacard .icns files with your own, and enter the new info into the info.plist file:

The CFBundleTypeIconFile should be your icon for the stacks, for Blackjack Gold, I called the stack icon "Data.icns".

```
<key>CFBundleTypeIconFile</key>
<string>Data.icns</string>
```

The CFBundleIconFile should be your icon for the standalone, for Blackjack Gold, I called the standalone icon "Gold.icns"

```
<key>CFBundleIconFile</key>
<string>Gold.icns</string>
```

The CFBundleSignature is your creator code.

```
<key>CFBundleSignature</key>
<string>bJg3</string>
```

Step 4:

The pkgInfo file.

Open it, and change the text to "APPLXXXX", where XXXX is your creator code.

Step 5:

The standalone, and the stacks, should be in the MacOS folder of the package.

Step 6:

Open the resource fork of the standalone with ResEdit or a similar program. There should be a resource named your chosen creator code. There should be the ic** series, and each one should have 128 as the standalone icon, and 129-130 as the stack icon. Except for icns, which you just leave alone.

Open the BNDL resource and edit 128. The signature field should be your creator code. If it isn't, enter it there.

There should be two types: APPL and MSTK. APPL should be your standalone icon, MSTK should be your stack icon. If they aren't, double click to edit them.

Step 7:

Set the creator code of your stacks. To do this, save your external stacks with stackfiletype "XXXXMSTK" where "XXXX" is your creator code.

In the standalone, every place it saves a stack, set the stackFileType first:

```
set the stackFileType to "XXXXMSTK"
save stack someStack
You are done. Hopefully I haven't left anything out!
```

Creating Custom Icons ()

You need to set the creator of the standalone, and the stacks [used with the standalone].

Step 1:

The resource fork of the standalone, and stacks. Firstly, the stacks themselves do NOT need a resource fork. But the standalone does. In the stack that will become the standalone, you should have an ic14, ic18, icns, ics4, and ics8 resource. If you edit ic18, the rest should be edited for you automatically. Put your standalone icon into resource 128 of ic18. Put your stack icon into resource 129 and 130.

Step 2:

When you use the Standalone Builder, there is a field "Creator" that has "MSTD" in it. This is Metacard's creator code.

You'll need to use a custom code for this. Creator codes are 4 characters, letters or numbers, small or capital. If you plan to distribute the standalone to others, you should make sure not to use a code already in use. You can get creator code info at:

<http://developer.apple.com/dev/cftype/faq.html>

I register all mine with Apple thru this link. And check for existing codes before using one, to make sure it is not already registered. For Blackjack Gold I used "bJg3".

Enter this code in the "Creator" field when you build the standalone.

Step 3:

Open the resource fork of the standalone with ResEdit or a similar program. There should be a resource named your chosen creator code. There should be the ic** series, and each one should have 128 as the standalone icon, and 129-130 as the stack icon. Except for icns, which you just leave alone.

Open the BNDL resource and edit 128. The signature field should be your creator code. If it isn't, enter it there.

There should be two types: APPL and MSTK. APPL should be your standalone icon, MSTK should be your stack icon. If they aren't, double click to edit them.

Step 4:

Set the creator code of your stacks. To do this, save your external stacks with stackFileType "XXXXMSTK" where "XXXX" is your creator code.

In the standalone, every place it saves a stack, set the stackFileType first:

```
set the stackFileType to "XXXXMSTK"
save stack someStack
```

You are done. Hopefully I haven't left anything out!

Creating Custom Icons ()

My fella wouldn't let me give up last night, and he on the Windows computer and me on the Mac, spent hours trying to solve this. He found an icon Metacard liked, I opened it with GraphicConvertor, and did nothing except change the colors. There were 6 colors, and I simply changed black to green and grey to red, etc. Even that failed. The file size went from 766 bytes to 1022 bytes.

In a last ditch attempt, I remembered Chipp Walter's suggestion, regarding the "Creating a custom icon for Windows" info at:

<http://www.altuit.com/webs/altuit2/RunRev/Tutorials.htm>

I had read it, and followed every step, during the day long exercise in frustration, but I had ignored the final step.

It was not enough to create a 32x32 pixel image.
It was not enough to limit it to 16 colors.

Where all the fancy programs failed to turn the image into an icon Metacard would accept, the simple little program **QTam** did it.

The most important piece of the puzzle, was to use QTam to convert the image to the Windows icon. QTam does whatever internal voodoo is necessary, to make the image be 766 bytes.

Where all the fancy schmancy programs failed... Adobe PhotoDeluxe, GraphicConvertor, Iconographer, Axialis, and several other Mac and Windows programs... where they couldn't do it, QTam could.

My kudos to the creator! This software developer has definitely earned a shareware fee from me!

Download QTam from:

<http://shareit1.element5.com/programs.html?productid=134460&language=English>

Shari C

--
--Shareware Games for the Mac--
<http://www.gypsyware.com>

Update: 8/10/03

12/1/2020

Sons of Thunder Software - Creating Custom Icons for Your Standalones and Stacks

(Editor's Note: Klaus Major stated in a post that the Photoshop Plug-In "Icon Builder Pro" will create exact 766-byte icons that MC does not refuse. It can be downloaded from IconFactory at <http://www.iconfactory.com>).

Updated 8/10/03 by Ken Ray

 [Print this tip](#)

[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@sonsothunder.com.
