

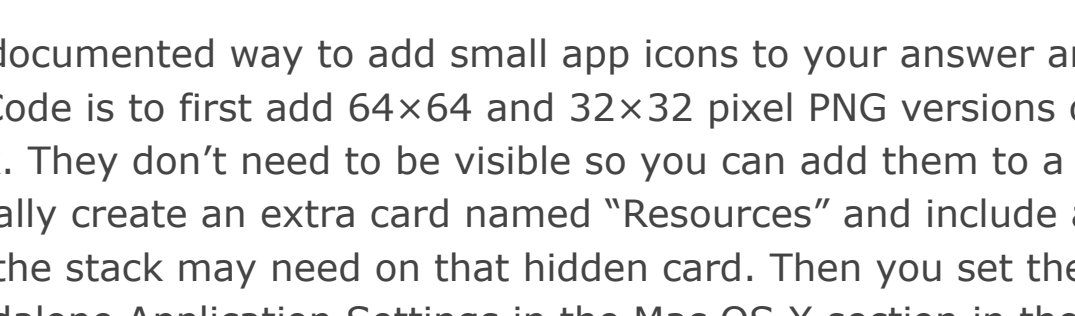
Adding Mac OS X Small Icons to Answer Dialogs LiveCode Hack Fix

by Michael Scaramozzino | posted in: How to Tutorials, Programming | 0

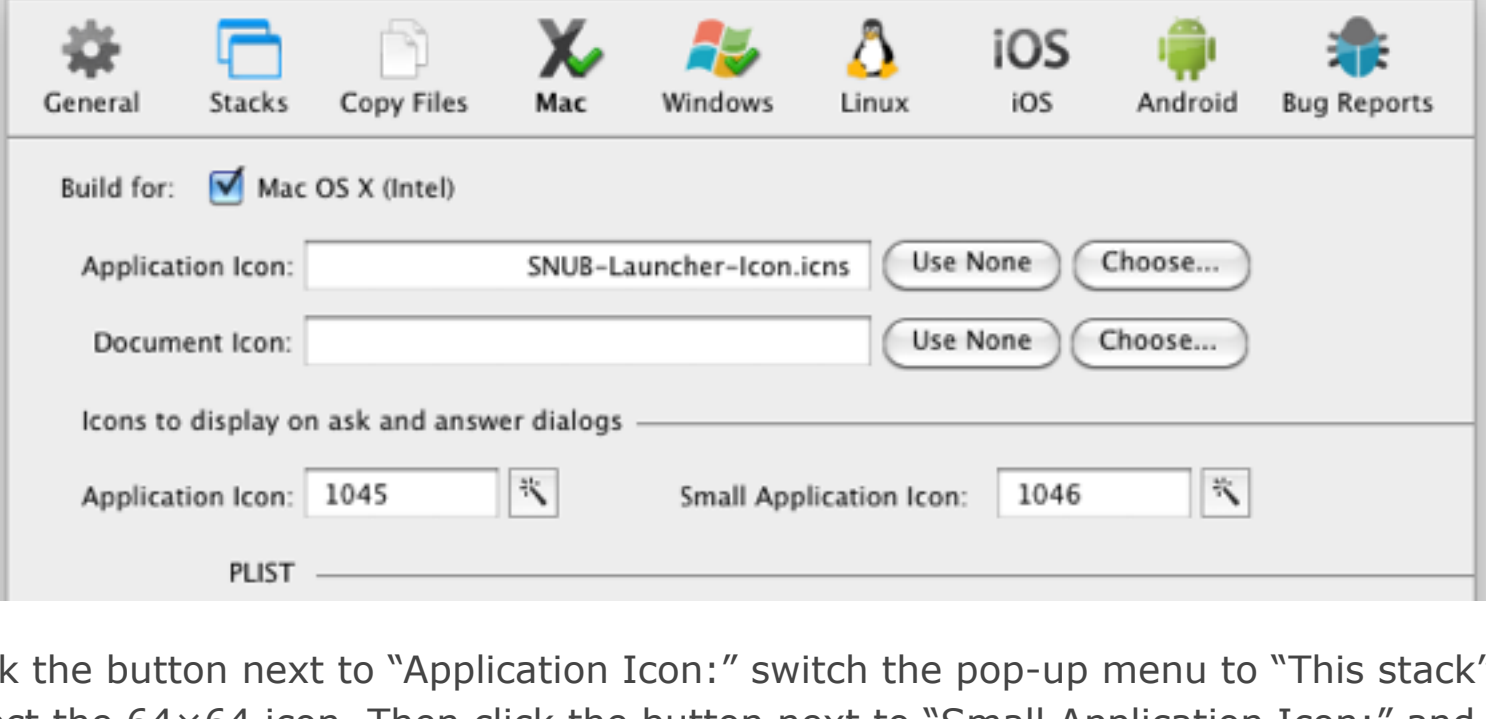
This is a quick how to tutorial – adding Mac OS X small icons to answer dialogs LiveCode hack fix. One of the coolest things about LiveCode is that it's built with LiveCode itself. The entire IDE is built from LiveCode stacks. This means the entire thing is easily editable and hackable. So if something doesn't work right or doesn't work the way you like, you can change it yourself.

[Disclaimer: This post is intended as an educational example. Before editing or hacking LiveCode itself, be sure to backup LiveCode and all your data files first, in case you mess things up. Proceed at your own risk. DreamLight and Michael Scaramozzino make no warranties whatsoever.]

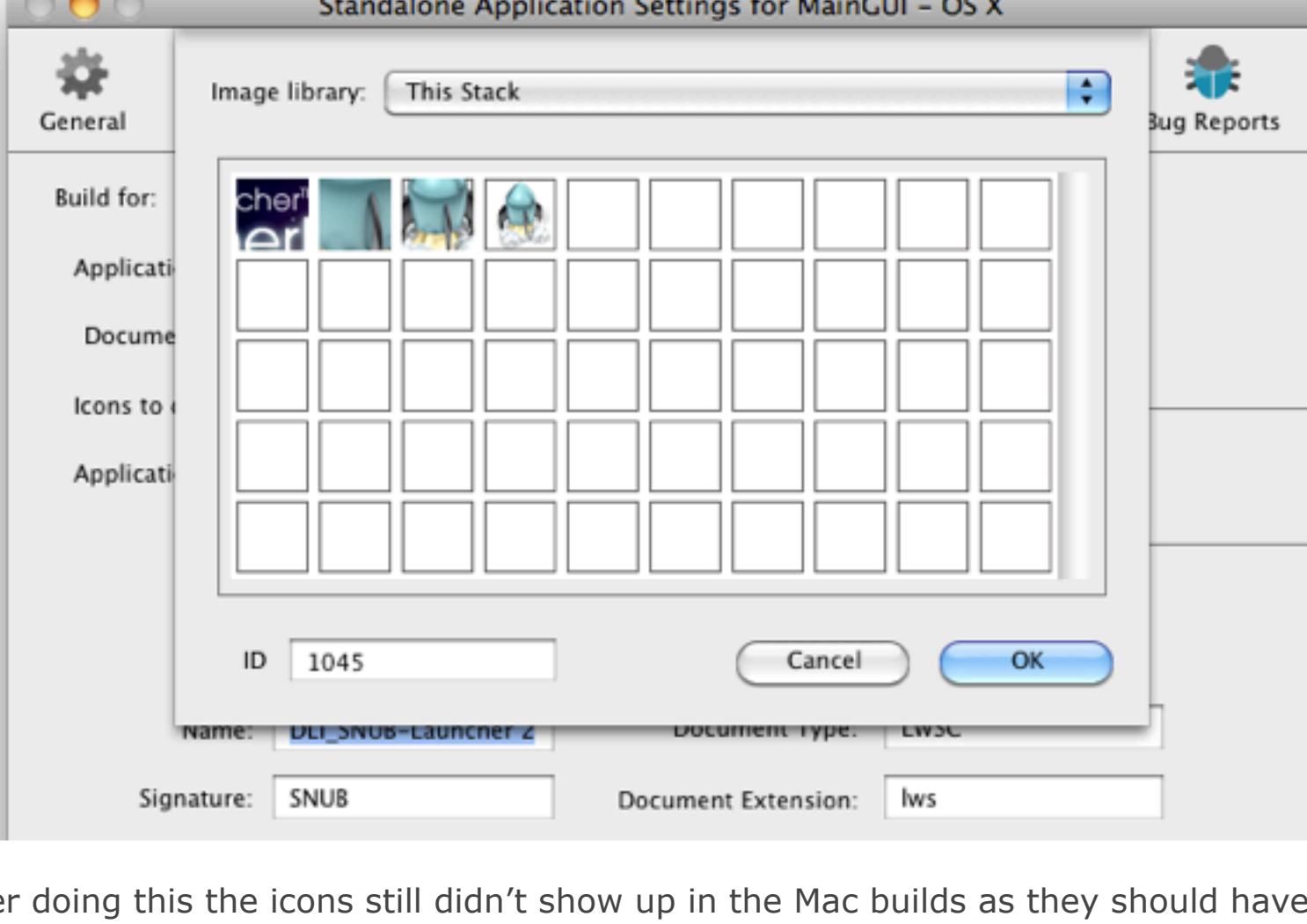
On Mac OS X, dialogs typically have a small version of the application icon overlaid on the dialog box icon to help identify which app the dialog belongs to.



The documented way to add small app icons to your answer and ask dialogs in LiveCode is to first add 64x64 and 32x32 pixel PNG versions of your app icon to the stack. They don't need to be visible so you can add them to a card that never shows. I typically create an extra card named "Resources" and include any images and icons that the stack may need on that hidden card. Then you set the icons on the Standalone Application Settings in the Mac OS X section in the items under "Icons to display on ask and answer dialogs."



Click the button next to "Application Icon:" switch the pop-up menu to "This stack" and select the 64x64 icon. Then click the button next to "Small Application Icon:" and set it to the 32x32 icon.



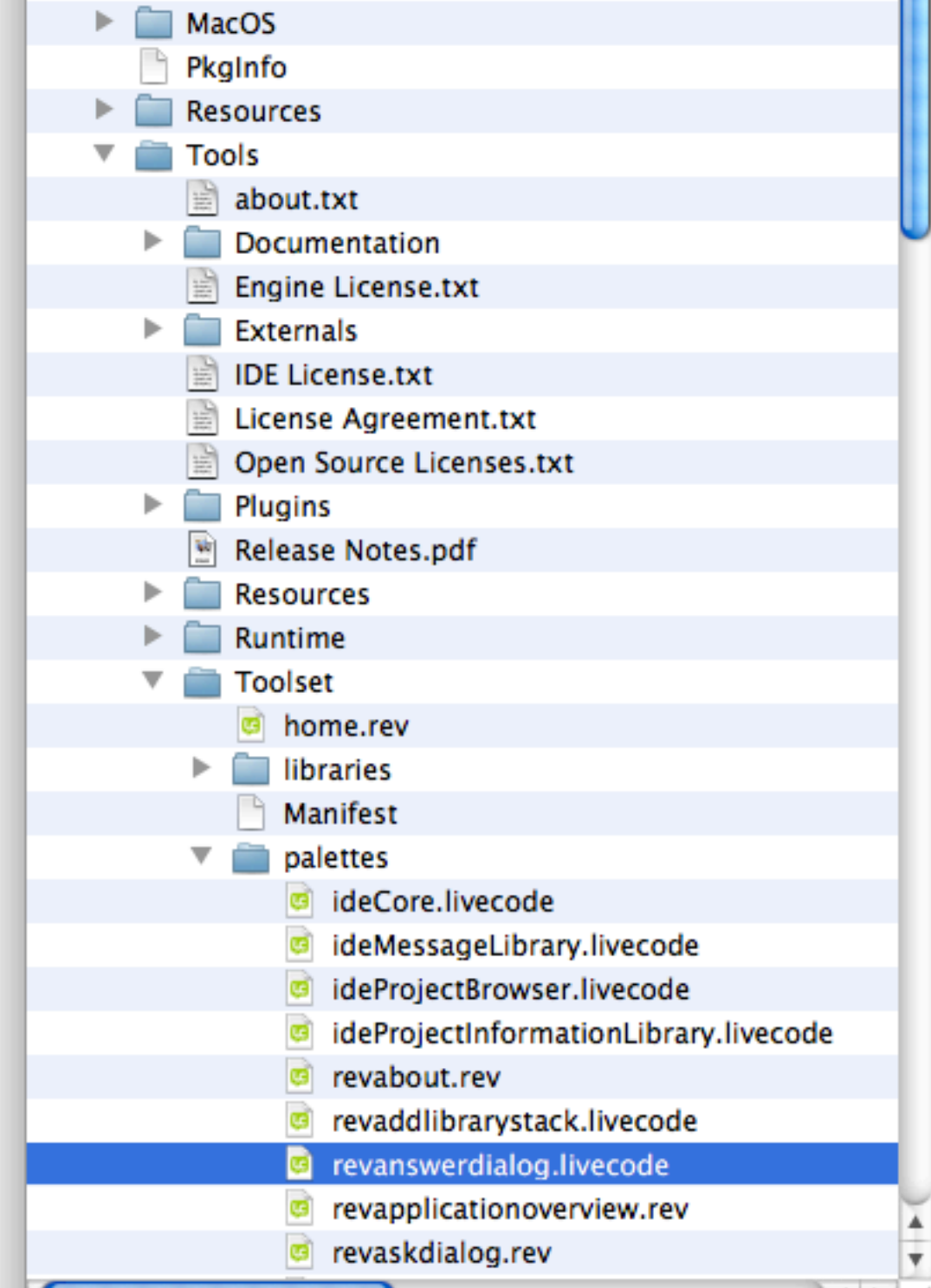
After doing this the icons still didn't show up in the Mac builds as they should have. I spent quite a while trying to figure out why small icons weren't working as documented in LiveCode's answer dialogs. I read and reread the documentation many times thinking I may have been making a mistake somewhere. Next I figured that setting it up in the Standalone Settings would only show up in the standalone and not in the IDE, which make things more difficult to troubleshoot because you have to keep making standalone builds to test. So digging a little deeper I found that those settings are really just putting the icon IDs into two global variables gREVAppIcon and gREVSmallAppIcon. I added the following code to set those variables up even when running in the IDE so I could more easily test what was going on.

```
-- set the icons for answer/ask dialogs
put the short ID of image "SNUB-Icon-64.png" of card "Resources" in
to gREVAppIcon
put the short ID of image "SNUB-Icon-32.png" of card "Resources" in
to gREVSmallAppIcon
```

The Script for Adding Mac OS X Small Icons to Answer Dialogs LiveCode Hack Fix

But even with this code, I could see that the IDs were properly set, but still no small icons appeared on the answer or ask dialogs. After not being able to get them to work as documented, I logged in and checked the open bug list. It turns out that the Mac OS X small icon functionality has been broken and reported since 2011. It looks like that functionality was removed to make it consistent with Windows and Unix builds, which don't have such a feature, so it probably wouldn't be fixed any time soon. Because LiveCode is itself built from stacks, including the answer and ask dialogs, I figured I could just hack up a quick fix myself. Looking at the answer dialog stack I could see that it must have been implemented in the past because the gREVSmallAppIcon global variable was declared, and there was a hidden second button icon, but they are never used in the current implementation. So to enable them is a trivial fix.

The trickiest part was just locating the dialog stack. On Mac OS X applications are really just folders called packages with all the necessary files packed inside. After properly backing up your LiveCode file and all data files just in case, simply right-click on the LiveCode icon and select "Show Package Contents" which will open the folder.

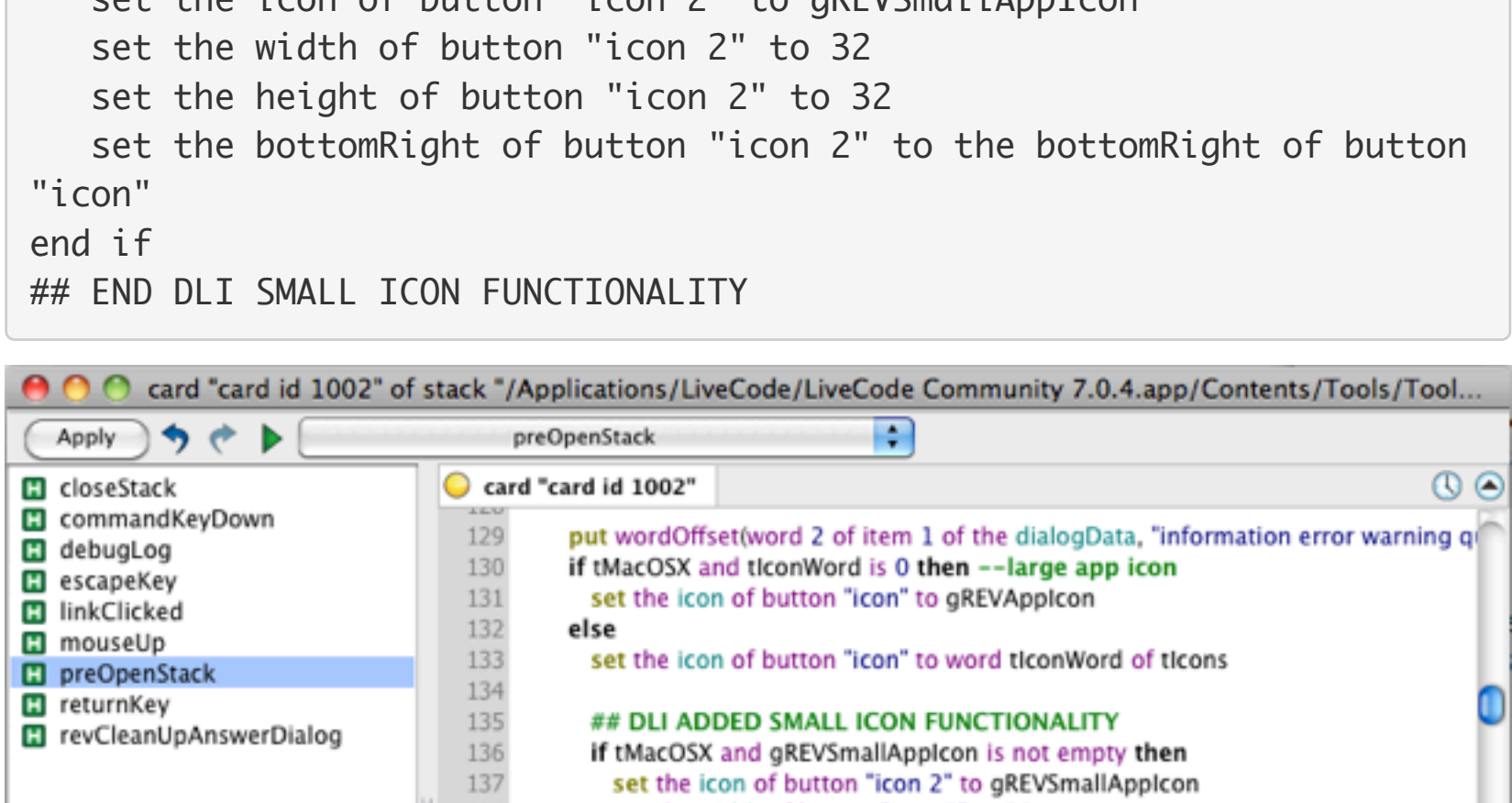


Inside the palettes folder you'll find the revanswerdialog and revaskdialog files. Those are the stacks for the answer and ask dialogs. Again, before editing these files, especially because you're editing files in the running app, backup the LiveCode app and all your data files first! You can open the revanswerdialog stack by double-clicking on this file. Another way to open this is from within the IDE by typing the following command into the message box:

```
open stack "Answer Dialog"
```

One thing to be aware of is that because this answer dialog is part of the IDE it won't show up in the Application Browser because it's not normally something you would want listed in there for editing. Once the stack is open though you can go to the "Object" menu and select the "Card Script" command to open the card script. Then add the following code between the ## DLI comments in the preOpenStack handler located as shown in the image.

```
## DLI ADDED SMALL ICON FUNCTIONALITY
if tMacOSX and gREVSmallAppIcon is not empty then
  set the icon of button "icon 2" to gREVSmallAppIcon
  set the width of button "icon 2" to 32
  set the height of button "icon 2" to 32
  set the bottomRight of button "icon 2" to the bottomRight of button
  "icon"
end if
## END DLI SMALL ICON FUNCTIONALITY
```



Then just save the stack and you're good to go! The small icons will then work in the IDE as well as being included in any Mac OS X standalone builds that you make. A similar hack could also be applied to the ask dialog if you wish. The biggest drawback to such a hack fix is that you'd need to reapply it again if you upgrade LiveCode itself.

Related Links

[Bug 9421 – gREVSmallAppIcon no longer works](#)


If you enjoyed this tutorial please consider making a donation to fund more!



bug fix, DreamLight, gREVSmallAppIcon, How to, LiveCode, LiveCode Hack, Mac OS X, Programming, Small Icons, Tutorial

Previous Post Next Post

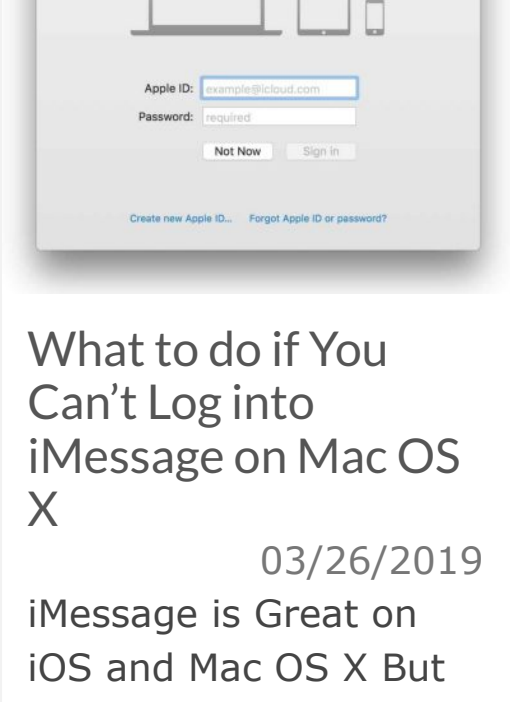
Similar Posts



Web Color Management for Images and CSS Colors

05/31/2019

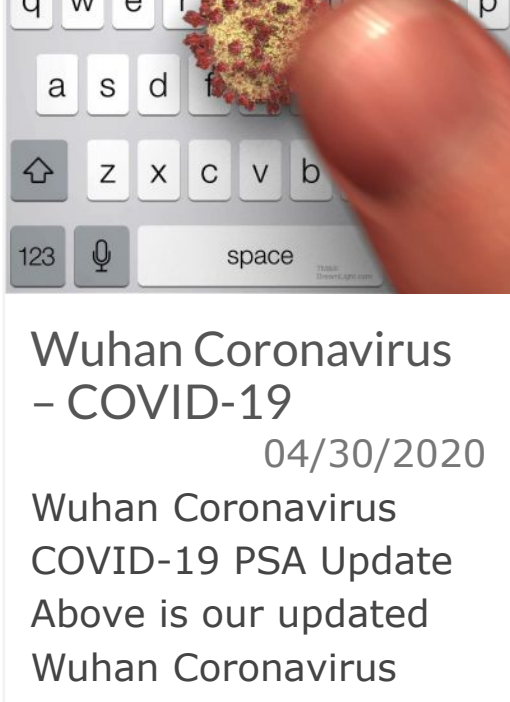
Understanding and Testing Web Color Management for Images and CSS Colors in Web Browsers Web...



What to do if You Can't Log into iMessage on Mac OS X

03/26/2019

iMessage is Great on iOS and Mac OS X But Here's What to do if...



Wuhan Coronavirus – COVID-19

04/30/2020

Wuhan Coronavirus COVID-19 PSA Update Above is our updated Wuhan Coronavirus COVID-19 PSA. Please share...

Leave a Reply

Comment

☐

Save my name, email, and website in this browser for the next time I comment.

two + 4 =

Post Comment

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)