

Code Signing FileMaker Runtimes

Everyone producing runtimes with FileMaker 13 for Mac OS X has a problem. You need to sign the app for Gate Keeper. If your runtime is not signed, the Finder will ask the user when launching the application after download to delete the app! To avoid users get confused or not use the runtime, we sign the runtime.



How?

Signing the runtime got harder with new requirements in Mac OS X 10.9 and the upcoming Yosemite release. We not just need to sign the actual application, but also make sure we sign all the components. For the frameworks inside the runtime, we need to fix them to give them a few standardsymlinks which are missing. Without those modifications, the bundle will not sign. Luckily all the PPC code and duplicate libraries got removed in FileMaker 13.0v3, so we don't need to do this yourself like in FileMaker 12. If you like to save a few bytes, you can delete the icon files: FM12Dict.icns, FM12Label.icns and FM12Plug.icns inside the runtime. Those are not referenced and never used.

Developer Account

In order to sign, you need to sign up for a Mac Developer Account with Apple. This costs \$99 per year. In the Certificates, Identifiers & Profiles section, please go in Mac Apps section and there in Certificates. Create a new Developer ID, follow the instructions and download the certificate file. Double click it to add it to Keychain Manager and voila, it is installed and ready to use.

Sign Script

The complete script is below.

When you edit the script in a text editor (like BBEdit or TextWranger), you can change the path to the runtime, the name of the app and the name of the certificate. Please note that this script will not work without correct values. If your file name or path contain special character, escaping may be needed. For example a space character needs to be escaped with putting a backslash character just before the space character. You can learn how to escape a path by dragging and dropping

the file or folder into the window of the Terminal application. The path is inserted and you can copy & paste it.

Now when everything is setup for your runtime in the script, you can drop the script on a Terminal window and press return key to start it. You'll see a couple of messages. This may include some complains from rm command trying to delete files which are not there. Nothing to worry about, the script just makes sure everything is correct. Further you see a couple of sign message for various parts of the application. The final line should show "signed bundle with Mach-O thin" and report success for signing your runtime app.

Verification

To check if app is okay for Gate Keeper you can first verify code signature using a call to codesign with a couple of v for more details and -d to display certificate:

```
codesign -d -vvvv /Users/cs/Desktop/Test/test.app
```

Next with spctl utility you can show if app is accepted. So we run spctl utility with verbose messages and -a parameter:

```
spctl -a -vv /Users/cs/Desktop/Test/test.app
```

Please change path to your runtimes before running above commands. The output should say "accepted" and now you are lucky and can archive and upload your runtime. Good luck!

The script

```
#!/bin/bash
```

```
# replace path to app: /Users/cs/Desktop
```

```
# replace name of app: test.app
```

```
# replace name of certificate: "Developer ID Application: Christian  
Schmitz Software GmbH"
```

```
# Special characters like spaces need to be escaped in paths. e.g. space  
character gets a backslash before the space.
```

```
cd /Users/cs/Desktop/Test/Test.app/Contents
```

```
rm -Rfv _CodeSignature
```

```
# fix the frameworks
```

```
# rm commands to make sure we recreate things correctly.
```

```
cd Frameworks/
```

```
cd DBEngine.framework
```

```
rm -Rfv _CodeSignature
```

```
rm DBEngine
```

```
rm Resources
```

```
In -s Versions/A/Resources Resources
```

```
In -s Versions/A/DBEngine DBEngine
```

```
In -s A Versions/Current
```

```
cd ..
```

```
cd FMEngine.framework
```

```
rm -Rfv _CodeSignature
```

```
rm FMEngine
```

```
rm Resources
```

```
In -s Versions/A/Resources Resources
```

```
In -s Versions/A/FMEngine FMEngine
```

```
In -s A Versions/Current
```

```
cd ..
```

```
cd FMWrapper.framework
```

```
rm -Rfv _CodeSignature
```

```
rm FMWrapper
```

```
rm Resources
```

```
rm -rf Versions/A/Headers
```

```
In -s Versions/A/Resources Resources
```

```
In -s Versions/A/FMWrapper FMWrapper
In -s A Versions/Current
cd ..
```

```
cd OmniORB4.framework
rm -Rfv _CodeSignature
rm OmniORB4
rm Resources
In -s Versions/A/Resources Resources
In -s Versions/A/OmniORB4 OmniORB4
In -s A Versions/Current
cd ..
```

```
cd OpenSSL.framework
rm -Rfv _CodeSignature
rm OpenSSL
rm Resources
In -s Versions/A/Resources Resources
In -s Versions/A/OpenSSL OpenSSL
In -s A Versions/Current
cd ..
```

```
cd Support.framework
rm -Rfv _CodeSignature
rm Resources
rm Support
In -s Versions/A/Resources Resources
In -s Versions/A/Support Support
In -s A Versions/Current
cd ..
```

```
# now we sign all the parts
cd /Users/cs/Desktop/Test/
```

```
codesign -f -vvvv -s "Developer ID Application: Christian Schmitz
Software GmbH" test.app/Contents/Frameworks/DBEngine.framework/
Versions/A
```

```
codesign -f -vvvv -s "Developer ID Application: Christian Schmitz
Software GmbH" test.app/Contents/Frameworks/FMEngine.framework/
Versions/A
```

```
codesign -f -vvvv -s "Developer ID Application: Christian Schmitz
Software GmbH" test.app/Contents/Frameworks/FMWrapper.framework/
Versions/A
```

```
codesign -f -vvvv -s "Developer ID Application: Christian Schmitz  
Software GmbH" test.app/Contents/Frameworks/OmniORB4.framework/  
Versions/A
```

```
codesign -f -vvvv -s "Developer ID Application: Christian Schmitz  
Software GmbH" test.app/Contents/Frameworks/OpenSSL.framework/  
Versions/A
```

```
codesign -f -vvvv -s "Developer ID Application: Christian Schmitz  
Software GmbH" test.app/Contents/Frameworks/Support.framework/  
Versions/A
```

```
codesign -f -vvvv -s "Developer ID Application: Christian Schmitz  
Software GmbH" test.app/Contents/Frameworks/*.dylib
```

```
codesign -f -vvvv -s "Developer ID Application: Christian Schmitz  
Software GmbH" test.app/Contents/XPCServices/*.xpc
```

```
codesign -f -vvvv -s "Developer ID Application: Christian Schmitz  
Software GmbH" test.app/Contents/MacOS/Runtime
```

```
codesign -f -vvvv -s "Developer ID Application: Christian Schmitz  
Software GmbH" test.app
```