



enfocus **CONNECT** 2021

Notarizing Connect ALL
Connectors

Contents

1. Introduction.....	3
2. Prerequisites.....	4
3. Procedure.....	5
3.1. Signing the Connector binaries.....	6
3.1.1. Signing QtWebEngineProcess.....	6
3.1.2. Signing the PitStop Library for Mac.....	7
3.1.3. Signing the Normalizer Library for Mac.....	9
3.2. Signing the Connector.....	11
3.3. Notarizing the Connector.....	12
4. Appendix.....	14
4.1. Overview of signed Connector binaries.....	14
5. Copyrights.....	15

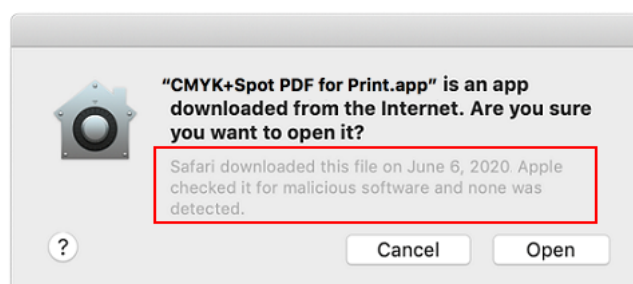
1. Introduction

This document describes the notarization procedure for Enfocus Connectors that are created using Connect ALL.

Note that this procedure applies to Mac Connectors only. Windows Connectors only have to be signed, which is a different procedure. For more information, refer to the [Signing Connect ALL Connectors](#) documentation on our website.

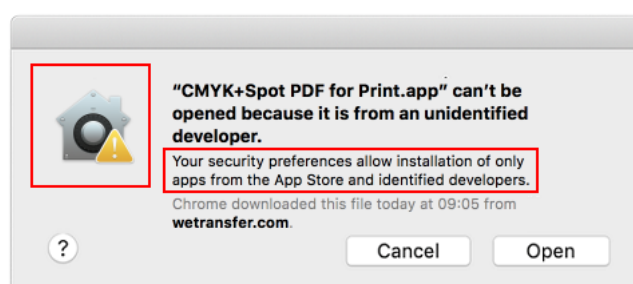
What is notarization?

Notarization is a process for applications that run on Mac and that are distributed outside the App Store. App developers are requested to submit their apps to Apple, so that Apple can scan them for malicious content and code-signing issues. If the apps are verified OK, the Apple notary service attaches a ticket to the software, so that, when a user installs it for the first time, a message is shown that confirms that the content is trustworthy.



Why notarize an Enfocus Connector?

When you create a Connector for use on Mac, we recommend notarizing it, to give your customers the confidence that the Connector you distribute to them has been verified by Apple. Otherwise your customers will get a warning that the Connector might be insecure.



More info about notarizing macOS software?

Refer to [Apple's developer site](#).

2. Prerequisites

Important remark for Windows users

Notarization is a process that applies to applications for use on MacOS. If you're running Connect ALL on a Windows machine, you can create a Connector for Mac on this Windows machine, but the notarization needs to be done on a Mac computer.

Terminal app

All the commands mentioned in this document must be executed in the Terminal app on your Mac. If you're not familiar with Terminal, consult [Apple's Terminal User Guide](#) for information on how to open, use and quit Terminal.

Prerequisites

Before you start the actual notarization process, make sure to do the following:

- Create an [Apple developer account](#).
- Get your [developer signing certificate](#).
- Install one of the recent (free) versions of **Xcode** for the macOS version you're using.

Xcode is an integrated development environment (IDE) for macOS. Developed by Apple, it contains a suite of tools for developing software for Apple's OS platforms such as macOS, iOS and iPadOS.

To know which version you need, refer to [Wikipedia](#). Note that older versions of Xcode are also available on [Apple's developer site](#) (sign in with your Apple developer account).

- Create and test your Connector Project using Connect ALL. Refer to the [Connect 2021 User Guide](#).

3. Procedure

The notarization procedure consists of the following **three steps**:

1. Signing the Connector binaries

Unfortunately, "deep signing" the Connector (which signs the Connector including all libraries in one go) is not possible, for various reasons.

As a consequence, there are three binaries that must be signed separately, and that must be done *first*.

Note that these binaries only need to be signed once, after which they will be used by all Connectors created with your Connect ALL application.

2. Signing the Connector

Once the Connector binaries have been signed, you can create the actual Connector, which will include the already signed binaries. Now this Connector needs to be signed as well.

3. Notarizing the Connector

Finally, the signed Connector must be compressed and uploaded to the Apple notarization system. When the verification by Apple is successful, you will receive a confirmation and a ticket that can be stapled to your Connector.

Remarks

- Are you a *Connect ALL user on Windows*? In that case, you should copy the files to a Mac computer (for signing the binaries), then copy them back to create the Connector on Windows, and then finalize the signing and notarization process on Mac.
- Have you created a *Remote Download Connector* (i.e. a small and portable version of the Connector)? In that case, you should notarize both the full Connector and the remote version ("remote_<Connector>.app").

For more information about the Remote Download option, refer to the [Connect 2021 User Guide](#).

- Do you want to notarize a Connector for which the *automatic updating mechanism* is enabled? In that case, you should copy the Mac Connector from the update server (where it resides after being created from Connect ALL) to a Mac computer, to carry out the whole procedure. Note that the Mac connector is compressed, so you should uncompress the ZIP file before starting the procedure, and compress it again when you're done. Make sure that the new ZIP file has the same structure as the original ZIP file.

When updating an existing Connector, make sure the update service is paused, to prevent remote customers from having their already installed Connectors updated, before you've completed the signing process.

For more information about the automatic updating mechanism, refer to the [Connect 2021 User Guide](#).

3.1. Signing the Connector binaries

Note that this is a one-time effort. Once the binaries have been signed, all newly created Connectors will make use of them.

3.1.1. Signing QtWebEngineProcess

The QtWebEngineProcess binary must be signed separately. This has to be done only once: afterwards, any newly generated Connector can reuse this signed executable.

To sign QtWebEngineProcess

1. Find the executable [QtWebEngineProcess].
 - Location on Mac: /Library/Application Support/Enfocus/Connect ALL/21/Connectors/Connector.app/Contents/QtWebEngine/QtWebEngineProcess
 - Location on Windows: C:\ProgramData\Enfocus\Connect ALL\21\Connectors\Connector.app\Contents\QtWebEngine\QtWebEngineProcess
2. Copy the executable to a folder on your Mac desktop.
3. In this folder, create a new file named QtWebEngineProcess.entitlements with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>com.apple.security.cs.disable-executable-page-protection</key>
  <true/>
</dict>
</plist>
```

For more information about the entitlements file, refer to <https://forum.qt.io/topic/102212/qtwebengine-signing-issues/5>

4. Open Terminal and enter the following command:

```
codesign --sign <dev-id> --entitlements <location of
QtWebEngineProcess.entitlements> --options=runtime --keychain <path to
signing.keychain file> --timestamp --prefix=<company prefix> <location of
QtWebEngineProcess>
```

5. When signing is complete, copy QtWebEngineProcess back to the original location (on Mac or on Windows).

6. To verify if the binary is signed correctly, enter the following command in Terminal:

```
codesign -dvv <path to binary>
```

3.1.2. Signing the PitStop Library for Mac

As the PitStop Library is compressed, it cannot be signed along with the Connector application. It must be uncompressed and signed separately. This has to be done only once: afterwards, any newly generated Connector can reuse this signed library.

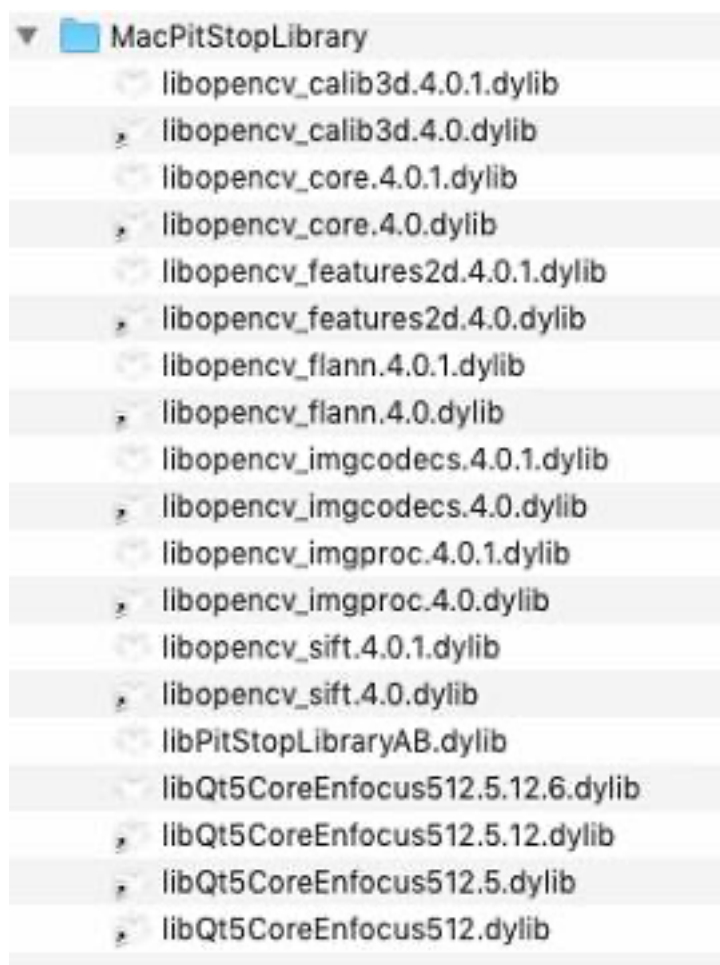
To sign the PitStop Library

1. Find the compressed PitStop Library (MacPitStopLibrary.zip)
 - Location on Mac: /Library/Application Support/Enfocus/Connect ALL/21/Connectors/MacPitStopLibrary.zip
 - Location on Windows: C:\ProgramData\Enfocus\Connect ALL\21\Connectors\MacPitStopLibrary.zip
2. Copy MacPitStopLibrary.zip to your Mac desktop.
3. To check the content of the ZIP file, enter the following command in Terminal:

```
zipinfo MacPitStopLibrary.zip
```

After signing, we need exactly the same files (but signed, see step 9), so that's why we recommend checking the content upfront.

4. Double-click the ZIP file to uncompress it.
This will create a folder named MacPitStopLibrary.



5. In the MacPitStopLibrary folder, enter the following command in Terminal to sign the libraries.

```
codesign --sign <dev-id> --options=runtime --keychain <path to signing.keychain file> --timestamp --prefix=<company prefix> <location of the library you want to sign>
```

Do this for each of the libraries:

- libPitStopLibraryAB.dylib
- libQt5CoreEnfocus512.5.12.6.dylib
- libopencv_calib3d.4.0.1.dyl
- libopencv_core.4.0.1.dylib
- libopencv_features2d.4.0.1.dylib
- libopencv_flann.4.0.1.dylib
- libopencv_imgcodecs.4.0.1.dylib

- libopencv_imgproc.4.0.1.dylib
 - libopencv_sift.4.0.1.dylib
6. Copy the signed binaries to a new folder.
 7. To move to this new folder in Terminal, enter the following on the command line:

```
cd <folder path>
```

8. To compress the content of this folder, enter the following in Terminal:

```
zip -r -q -y MacPitStopLibrary.zip *
```



Note: Do not forget the asterisk (*) at the end of the command!

9. Now check the content of the new ZIP file.

```
zipinfo MacPitStopLibrary.zip
```

The new ZIP file should contain the same files and folder structure as in step 4 of this procedure.

10. Copy the new ZIP file back to the original location:
 - Location on Mac: /Library/Application Support/Enfocus/Connect ALL/21/Connectors
 - Location on Windows: C:\ProgramData\Enfocus\Connect ALL\21\Connectors

3.1.3. Signing the Normalizer Library for Mac

As the Normalizer Library is compressed, it can't be signed along with the Connector application. It must be uncompressed and signed separately. This has to be done only once: afterwards, any newly generated Connector can reuse this signed library.

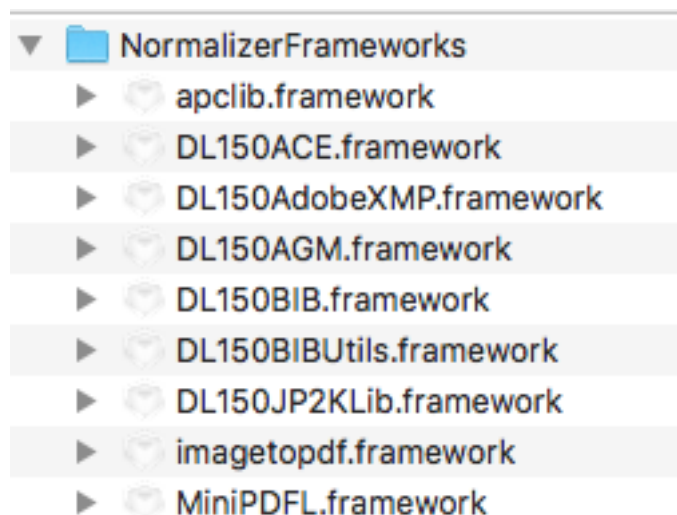
To sign the Normalizer Library for Mac

1. Find the Normalizer Library for Mac (NormalizerFrameworks.zip).
 - Location on Mac: /Library/Application Support/Enfocus/Connect ALL/21/Connectors/Frameworks/NormalizerFrameworks.zip
 - Location on Windows: C:\ProgramData\Enfocus\Connect ALL\21\Connectors\Frameworks\NormalizerFrameworks.zip
2. Copy NormalizerFrameworks.zip to your Mac desktop.
3. To check the content of the ZIP file, enter the following command in Terminal:

```
zipinfo NormalizerFrameworks.zip
```

After signing, we need exactly the same files (but signed, see step 9), so that's why we recommend checking the content upfront.

4. Double-click the ZIP file to uncompress it.
This will create a folder named NormalizerFrameworks.



5. In the NormalizerFrameworks folder, enter the following command in Terminal to sign the frameworks.

```
codesign --sign <dev-id> --deep --options=runtime --force --keychain <path to signing.keychain file> --timestamp --prefix=<company prefix> <location of NormalizerFrameworks/<framework name>.framework/Versions/A>
```

- As some of the frameworks are already signed, the option **--force** is included to override the signature.
- **<framework name>.framework**: All the following frameworks in the Normalizer Library must be signed using the above command:
 - apclib.framework/Versions/A
 - DL150ACE.framework/Versions/A
 - DL150AdobeXMP.framework/Versions/A
 - DL150AGM.framework/Versions/A
 - DL150BIB.framework/Versions/A
 - DL150BIBUtils.framework/Versions/A
 - DL150JP2KLib.framework/Versions/A
 - imagetopdf.framework/Versions/A

- MiniPDFL.framework/Versions/A

More info about signing of frameworks can be found on [Apple's developer site](#).

6. Copy the signed frameworks to a new folder.
7. To move to this new folder in Terminal, enter the following on the command line:

```
cd <folder path>
```

8. To compress the content of this folder, enter the following in Terminal:

```
zip -r -q -y NormalizerFrameworks.zip *.framework
```

9. Now check the content of the new ZIP file.

```
zipinfo MacPitStopLibrary.zip
```

The new ZIP file should contain the same files and the same folder structure as in step 4 of this procedure.

10. Copy the new ZIP file back to the original location:
 - Location on Mac: /Library/Application Support/Enfocus/Connect ALL/21/Connectors/Frameworks
 - Location on Windows: C:\ProgramData\Enfocus\Connect ALL\21\Connectors\Frameworks

3.2. Signing the Connector



Note: Signing a Connector is a mandatory step in the notarization of Connectors that will be distributed to customers, but there's no need to sign Connectors that are only used for testing purposes.

To sign the Connector

1. Create the Connect ALL Connector that you want to sign.
 - The Connector must include the earlier signed binaries (QtWebEngineProcess, PitStop Library, Normalizer Library), so make sure these binaries are signed before creating the Connector.
 - If you're using Connect ALL on Windows, you should first create the Connector for Mac on your Windows system, and then copy it to a Mac computer to execute step 2 of this procedure.
2. Enter the following command in Terminal:

```
codesign --sign <dev-id> --deep --options=runtime --force --keychain <path to signing.keychain file> --timestamp --prefix=<company prefix> <path of the Connector app>
```



Note: The `--deep` parameter indicates that all files that can be signed will be signed. Refer to [Overview of signed Connector binaries](#) on page 14.

3.3. Notarizing the Connector

To notarize the signed Connector

1. Compress the Connector.
2. Enter the following command in Terminal:

```
xcrun altool --notarize-app -t osx \
  --primary-bundle-id <Connector bundle ID> \
  -u <Apple ID> -p <password> \
  -f <Connector zip file path> \
  --output-format xml
```

Connector bundle ID: This is the bundle identifier of the application, which can be found in the Info.plist file of the application bundle. You can find Info.plist through the context menu of the app (**Show Package Contents**). In case of a Connector, this is actually the Unique ID field of the Connector Project (shown on the Description tab).

The output of the above command after successful upload will be something like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>notarization-upload</key>
  <dict>
    <key>RequestUUID</key>
    <string>12345678-12ab-49de-a3ae-1a2b3c4d5f6g</string>
  </dict>
  <key>os-version</key>
  <string>10.13.6</string>
  <key>success-message</key>
  <string>No errors uploading '<signed Connector.zip>'.</string>
  <key>tool-path</key>
  <string>...</string>
  <key>tool-version</key>
  <string>1.1.1138</string>
</dict>
</plist>
```

When the notarization process was successful, a confirmation email will be sent to the email address that was used when registering for the Apple Developer account. Awaiting this confirmation, you can check the notarization status yourself (see step 3).

3. To check for the status of the notarization process, enter the following command in Terminal:

```
xcrun altool --notarization-info 12345678-12ab-49de-a3ae-1a2b3c4d5f6g -u <Apple Id> -p <password>
```



Note: The ID used in this command can be found in the previous command output (step 2).

4. After successful notarization, to staple the ticket to your Connector, enter the following command in Terminal:

```
xcrun stapler staple <path of the signed Connector app>
```

The Connector is now notarized. When your customers download and launch it, they will see that the software is verified by Apple.

4. Appendix

4.1. Overview of signed Connector binaries

After completing the signing and notarization process the following binaries, bundled with the Connector are signed:

Application or binaries	Location	Remarks
QtWebEngineProcess	Connector.app/Contents/ QtWebEngine	First signed separately
MacPitStopLibrary.zip	Connector.app/Contents/ Helpers	First signed separately
NormalizerFrameworks.zip	Connector.app/Contents/ Helpers	First signed separately
PrinterProcessor.app	Connector.app/Contents/ Helpers	Signed as part of a Connector (deep sign)
EnfocusConnectorBackend	Connector.app/Contents/ Helpers	Signed as part of a Connector (deep sign)
libipworks.16.0.dylib libipworksssh.16.0.dylib	Connector.app/Contents/ Helpers/ IPWorks	Deep signed as part of a Connector
Connector	Connector.app/Contents/ MacOS	Deep signed as part of a Connector
Other Qt libraries within the Connect app	Connector.app/Contents/ MacOS	Deep signed as part of a Connector

5. Copyrights

© 2021 Enfocus BV all rights reserved. Enfocus is an Esko company.

Certified PDF is a registered trademark of Enfocus BV.

Enfocus PitStop Pro, Enfocus PitStop Workgroup Manager, Enfocus PitStop Server, Enfocus BoardingPass, Enfocus Connect YOU, Enfocus Connect ALL, Enfocus Connect SEND, Enfocus StatusCheck, Enfocus CertifiedPDF.net, Enfocus PDF Workflow Suite, Enfocus Switch, Enfocus SwitchClient, Enfocus SwitchScripter, Enfocus TestDrive, Enfocus SwitchScriptTool and Enfocus Browser are product names of Enfocus BV.

Adobe, Acrobat, Distiller, InDesign, Illustrator, Photoshop, FrameMaker, PDFWriter, PageMaker, Adobe PDF Library™, the Adobe logo, the Acrobat logo and PostScript are trademarks of Adobe Systems Incorporated.

Datalogics, the Datalogics logo, PDF2IMG™ and DLE™ are trademarks of Datalogics, Inc.

Apple, Mac, macOS, Macintosh, iPad and ColorSync are trademarks of Apple Computer, Inc. registered in the U.S. and other countries. Windows and Windows Server are registered trademarks of Microsoft Corporation.

PANTONE® Colors displayed here may not match PANTONE-identified standards. Consult current PANTONE Color Publications for accurate color. PANTONE® and other Pantone, Inc. trademarks are the property of Pantone, Inc. ©Pantone, Inc., 2006.

OPI is a trademark of Aldus Corporation.

Quark, QuarkXPress, QuarkXTensions, XTensions and the XTensions logo among others, are trademarks of Quark, Inc. and all applicable affiliated companies, Reg. U.S. Pat. & Tm. Off. and in many other countries.

This product and use of this product is under license from Markzware under U.S. Patent No. 5,963,641.

Other brand and product names may be trademarks or registered trademarks of their respective holders. All specifications, terms and descriptions of products and services are subject to change without notice or recourse.