DOC-MACTEX-2020

RICHARD KOCH

1. MacTeX

This is the portion of the TUG documentation which deals with making the MacTeX-2020 package. It is the final product of the build directory, built for internet distribution.

As a first step, move

```
TeXLive2020-Start.pkg
Ghostscript-9.50-Start.pkg
Ghostscript-9.50libgs-Start.pkg
GUI-Applications-Start.pkg
```

to the CreateInstallPackage folder of this folder.

2. Signing, Timestamping, and Adopting Hardened Runtimes for Binaries

Binaries in this package have already been signed and notarized, and have already adopted a hardened runtime.

3. Building the Final Install Package

The install package is constructed using scripts in the folder "CreateInstallPackage." This folder also contains Resources and Scripts which will become part of the package. We first build "MacTeX-2020-Temp.pkg" by

```
sudo sh productbuildscript.sh
```

We then create a signed package "MacTeX-2020.pkg" by running the script

```
sh signPackage.sh
```

This script contains my Apple signing identity; future builders will first have to join Apple's Developer Support and get their own signing identity.

The CreateInstallPackage folder contains the packages for the three component pieces: Ghostscript-9.50-Start.pkg, Ghostscript-9.50libgs-Start.pkg, GUI-Applications-Start.pkg, and TeXLive-2020-Start.pkg.

The CreateInstallPackage folder also contains two other files and one other script. The files are requirement.xml and distribution.xml, and the script is create_distribution.sh.

Date: March 12, 2020.

The script used requirement.xml to restrict to installation on macOS 10.13 and above, and created distribution.xml. This time, a fair amount of hand editing of the resulting distribution file was required.

4. Notarizing the Final Install Package

Scripts to notarize "MacTeX-2020.pkg" are in the folder "NotarizePackage" inside "CreateInstallPackage." The actual "MacTeX-2020.pkg" created above can be left where it is; it need not be moved to the new folder.

To notarize,

```
sh sendnotarizerequest.sh
```

This script has the flag –verbose, so information will appear in the console as the package is slowly uploaded to Apple. That is likely to take half an hour or longer. When the upload is complete, a message of the form

```
RequestUUID = 96f0932f-b50f-4d73-ad65-1220fd9c8efe
```

will be printed. This magic number can be used to discover what went wrong if notarization fails, so I recommend copying it to the file requestUUID.tex. There will be a pause of about ten minutes and then a notification from Apple and an email from Apple will arrive. If all is well, this message says that notarization succeeded. To add the certificate to the install package

```
sh stapleresult.sh
```

The package "MacTeX-2020.pkg" is then ready for uploading and release.

In case of failure, open the script detailedinfo.sh and notice that the magic UUID number is part of this script. Copy and paste the new number from requestUUID.tex to this script. Then run it.

```
sh detailedinfo.sh
```

This command will return a web url. Use Safari to go to that url, where a very detailed error report will clearly state what went wrong.

The detailedinfo script runs a command named xcrun which communicates with developer support. Apple has adopted two-factor authentication security for such communication, but this security is not very convenient when running shell scripts. So developers can log into their Apple developer account and assign a password to xcrun and a few similar commands. Then when the script runs, typing this password is enough to run the command. Note that the detailedinfo script contains my account name and password. When someone else makes these packages, they will replace this information with their own personal information.