

TUG-GUI-APPLICATIONS

RICHARD KOCH

1. GUI APPLICATIONS

This is the portion of the TUG documentation which deals with making the GUI Applications package.

A clean package file will contain

- “apps” folder with programs and documentation to be installed in /Applications/TeX
- File “buildPackage.sh”
- Files “background.jpg”, “License.rtf”, “ReadMe.rtf”, “Welcome.rtf”
- “TeX-Source” folder containing LaTeX source files for the documentation in apps
- “scripts” containing “postinstall”
- GUIApplicationsBuild project for Packagemaker
- “TUG” folder with this document
- File “signPackage.sh”

2. OVERVIEW

This package installs TeXShop, BibDesk, LaTeXiT, and TeX Live Utility in /Applications/Utilities. It installs the spell checker Excalibur. It also installs brief documentation for new features in that year’s TeX Live. All these pieces are added to the apps folder before building. They are moved to root/Applications/TeX by buildPackage.sh. Notice that buildPackage.sh must be edited if file names change or documentation is added or removed.

A key change in 2014 and 2015 is that the package identifiers are called org.tug.mactex.gui2015 and thus change from year to year. A consequence of this change is that installing the package does not remove applications and files from prior years. This is crucial; if the user has an old version of TeX Live Utility which runs on Leopard, we do not want to replace it with a newer version which only runs on Snow Leopard.

Date: May 7, 2015.

3. PROGRAMS IN APPS

To save space, the “apps” directory only contains a README file listing its assumed contents, but does not contain documentation or GUI applications. Thus before building, it is necessary to add Applications to this directory. The following is a list of these applications, and of web sites where the current versions can be obtained:

- TeXShop, available at <http://pages.uoregon.edu/texshop/> The folder contains two programs named TeXShop-Leopard and TeXShop-Lion. The first is the latest program in the 2.** series, but with the two Movies removed to save space. The second is the latest program in the 3.** series, which no longer contains movies and instead downloads them from my web site when needed. The first is a universal binary, and the second has 64 bit code and requires Lion or higher.
- BibDesk, available at <http://bibdesk.sourceforge.net/>; it is called BibDesk-SnowLeopard in apps because it requires Snow Leopard
- Excalibur-4.0.7, available at <http://excalibur.sourceforge.net/>. This is a folder named “Excalibur-4.0.7” containing the Excalibur program, Excalibur manual, gpl-2.0.pdf, Read Me.txt, and Standard Dictionary.
- LaTeXiT, available at http://chachatelier.fr/programmation/latexit_en.php
- TeX Live Utility, available at <http://mactlmgr.googlecode.com/>; it is called TeX Live Utility-SnowLeopard in apps because it requires Snow Leopard
- Setup-2015.pkg. This is a small install package which fixes problems when a system containing MacTeX is upgraded to a later version of the operating system, and also installs the Pref Pane and TeXDist structures for BasicTeX and TeX Live in case the user gets these from an install script.

4. BUILDING

The first step in building a new package is to update all the supplied programs. TeXShop, Bibdesk, and TeX Live Utility all use the Sparkle mechanism to update, so it suffices to read these programs and select the “Check for Updates...” menu. LaTeXiT may use the Sparkle mechanism as well, but in that case it seems to gray out the “Check for Updates...” menu when the program is up to date. To be sure, check

http://chachatelier.fr/programmation/latexit_en.php

The Excalibur spell checker is available at <http://excalibur.sourceforge.net/>. Note that it is provided in a folder containing the version number. If a new version is available, be sure to revise buildPackage.sh so it will copy the new folder to root/Applications/TeX.

Then update the documentation. Note that source code for the documentation is in the TeX-Source folder.

Finally, build by

```
sudo sh buildPackage.sh
```

5. DOCUMENTATION

Since Apple’s installer cannot install in user directories, there is no convenient place for it to install documentation. But sometimes new features of a release need to be pointed out. In that case, we install such documents in `/Applications/TeX`.

In general, it is a bad idea to put extensive documentation in `/Applications/TeX` because TeX Live has extensive and up to date information. Thus documentation we add tends to duplicate TeX Live documentation, and also tends not to be up to date. Currently, we provide the following documents:

- **READ ME FIRST:** This document is designed to help new users begin using the system
- **LuaTeX-and-XeTeX:** This document was added in 2010 to show users how to use system fonts in LuaTeX, and remind them that XeTeX has the same ability.
- **Eps-and-PdfLaTeX:** This was added to explain that TeX Live 2010 automatically converted eps files to pdf files when running pdfTeX.
- **What is installed:** An explanation of what is installed and where. Particularly useful when people want to remove things.

6. UPDATING FOR A NEW YEAR

Since the package name does not change from year to year, very little editing is require. The “Welcome.rtf” should be edited to reflect the release date.

7. BUILDING THE FINAL INSTALL PACKAGE WITH PACKAGEMAKER

The remaining steps are optional and not required for the final build of MacTeX.

Now edit `GUIApplicationsBuild` and `GUIApplicationsDVDBuild`, the Packagemaker project files, using the GUI interface, as follows. Both packages are built the same way, so we only list parameters of `GUIApplicationsBuild`.

- (1) Click the top item on the left. The right side will change to a view with three tabs. First select the Configuration tab. Edit so
 - (a) Title: GUI-Applications
 - (b) User Sees: Easy Install Only

- (c) Install Destination: System volume
 - (d) Certificate: Leave this blank. You might think that putting a certificate name here would correctly sign the package, but experiments show that it does not. The package needs to be signed after it is constructed.
 - (e) Description: Empty
- (2) The Requirements and Actions tab entries for the top item can be left blank
- (3) Click the entry GUI-Applications in the Contents section on the left. The right side will change to a view with two tabs. Select the Configuration tab. Edit so
- (a) Choice Name: GUI-Applications
 - (b) Identifier: choice1
 - (c) Initial State: Selected and Enabled
 - (d) Destination: Leave Blank
 - (e) Tooltip: Leave Blank
 - (f) Description: Leave Blank
- (4) The Requirements tab entries for this contents item can be left blank
- (5) Open the GUI-Applications entry in the bottom left column. This item should show an “Applications” directory, obtained by dragging root/Applications to the left column. The right side will change to a view with four tabs. It is not necessary to edit the items under the Contents tab. But it is *very important* to edit items in the Components tab, turning off all boxes labeled “Allow Relocation”. The items under the Configuration tab should be edited as follows:
- (a) Install: root/Applications (The item on the right should have selected “Relative To Project”)
 - (b) Destination: /Applications (The item on the right should have selected “Relative To Project”)
 - (c) Do not check “Allow custom location”
 - (d) Patch: No patch root selected
 - (e) Package Identifier: org.tug.mactex.gui2015
 - (f) Package Version: 1.0
 - (g) Restart Action: None
 - (h) Check “Require admin authentication”
 - (i) PackageLocation: Self-Contained

The items under the Scripts tab should be edited to read

- (a) Scripts directory: scripts (The item on the right should have selected “Relative To Project”)
- (b) Preinstall: Leave Blank
- (c) Postinstall: scripts/postinstall (The item on the right should have selected “Relative To Project”)

After verifying all these items, build the install package by clicking the “Build” icon. When asked to select a name, choose “GUI-Applications-Temp”. After the package has been made, sign it by executing the following line in Terminal:

```
sh signPackage.sh
```

This script simply calls

```
productsign --sign "Developer ID Installer: Richard Koch" old-name new-name
```

where “old-name” is GUI-Applications-Temp.pkg and “new-name” is GUI-Applications.pkg.

Similar remarks apply to building the second install package.