

TUG-IMAGEMAGICK

RICHARD KOCH

1. IMPORTANT WARNINGS

This section contains lessons learned in 2013. The material here is important for all packages made with the new PackageMaker.

Folders in which packages are built should live on the hard disk rather than an external disk. Building constructs subfolders named “root” which contain copies of material to be installed. This root folder and its contents are usually owned by root. But material on an external drive is owned by the user instead, so owners will be incorrect after installing.

The final MacTeX template refers to root folders of the various sub projects. In the template, each sub project is listed in the left column; clicking on an arrow there reveals a panel in which the root folder is determined and configured.

This root item must be reconfigured if the contents of the corresponding root are changed. Remove the item on the left listing the root (for example, /Applications/TeX) by selecting it and pushing “Delete”. Then drag a new copy of the root folder to this spot. Wait until PackageMaker has digested the new comments by watching a spinning cursor. Then reconfigure that root material. Each project contains *two* copies of the template to make this easy. Copy the configuration material from the “copy” of the template to the “active” template.

Configuration of a root folder is governed by a panel with four tabs: Configuration, Contents, Components, and Scripts. The material in the first and last tabs is important and should be entered carefully. The second Contents tab allows owners, groups, and permissions of individual files in the folder to be set. This task is impossible for a huge root folder like TeXLive, but luckily Apple provides a button “Apply Recommendations” which does it automatically following Apple’s “best practices policy.” Always click this button.

For binary files like texlive/2013/bin/universal-darwin, the recommendations are unexpected. Actual binaries have owner root and group wheel, but symbolic links have owner an individual user and group wheel. Permissions are -rwxr-xr-x. Experiments and email conversations show that these choices work well.

For the GUI-package, the third tab, “Components”, is very important. It lists actual apps and offers to relocate them. *Turn this off.* Otherwise the installer won’t upgrade programs

Date: April 18, 2013.

in /Applications/TeX, but instead will search the disk for a copy to update. Always check that these items are off.

2. IMAGEMAGICK

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

Warning: It is important that the folders in which packages are built live on the hard disk rather than an external disk. Building constructs folders named “root” which are copies of material to be installed. This root folder and its contents are often owned by root. But material on an external drive is owned by the user instead.

A clean package file will contain

- “Building This Package” folder with document explaining in detail how to build the package
- File “buildRoot.sh”
- Files “background.jpg”, “License.rtf”, “ReadMe.rtf”, and “Welcome.rtf”
- “Code2013” folder containing tar files with the sources for ImageMagick.
- “Universal” folder with two files named liposcript and tarscript, and three empty folders.
- scripts folder containing files “postinstall”, “setloginpath”, and “settexpath”
- ConvertIMBuild project file for Packagemaker
- “TUG” folder with this document
- “OldStyle” folder containing material to build a legacy copy of Convert-IM.pkg

3. OVERVIEW

We do not ship ImageMagick, but only the “convert” utility inside it. This is necessary because convert is used by TeX4ht and yet is not part of TeX Live. Also TeX can be instructed to call convert automatically if tiff files are encountered.

Normally ImageMagick ships as a series of libraries called by the main program. We don’t want to install any libraries. So we compile convert to use static libraries. This requires some tricky maneuvers as the program is constructed.

4. ADJUSTING FOR A NEW YEAR

To switch to the new name:

- Edit the “buildPackage.sh” script, noting that the version number of the program is contained in the script.
- The files which will be shown to the user during installation: License.rtf, ReadMe.rtf, and Welcome.rtf. Edit these files to reflect the new package name and release date. Make other changes as appropriate.

5. COMPILING

Compiling convert is tricky. Image Magick constructs shared libraries which its components all call, and it requires other libraries: libjpeg, png, freetype, tiff, zlib. We don’t want to install the Image Magick shared libraries or these extra libraries. The trick is to compile Image Magick using static libraries and then pull out convert.

Start by renaming the old /usr/local in /usr, and rename X11 and/or X11R6 so X11 support won’t be added.

We compile on OS X 10.5 for PPC and for Intel.

First install the required libraries in /usr/local. We need the following libraries:

zlib	http://www.zlib.net/	Ver 1.2.7: May 2, 2012
libjpeg	http://www.ijg.org	Ver 8d: January 15, 2012
freetype	http://freetype.sourceforge.net/index2.html	Ver 2.4.11: December 20, 2012
libpng	http://www.libpng.org/pub/png/libpng.html	Ver 1.5.14: January 24, 2013
libtiff	http://www.remotesensing.org/libtiff/	Ver 4.0.3 September 22, 2012
bzip2	http://bzip.org/	Ver 1.0.6; September 20, 2010

The latest version of libjpeg is Ver 9, but that version has a bug which affects libtiff. So we downgrade to Ver 8d.

Compile and install libjpeg first. Use the following instructions:

```
make clean
./configure --enable-shared --enable-static

rm libtool
ln -s 'which glibtool' ./libtool

make

sudo mkdir -p /usr/local/include

sudo mkdir -p /usr/local/lib
sudo mkdir -p /usr/local/man/man1
sudo mkdir -p /usr/local/bin

sudo make install
```

Then change to /usr/local/lib and remove all files except libjpeg.a.

Compile zlib, libtiff, libpng, freetype, and bzip2 in that order, using the standard commands (bzip2 doesn't have a .configure step):

```
make clean
./configure
make
sudo make install
```

After each step, remove all .dylib and .la files in /usr/local/lib.

Next go to /usr/local and remove everything in the bin directory. Also remove the man, and share directories, leaving only the include and lib directories.

Obtain the ImageMagick source code at

<http://www.imagemagick.org/script/install-source.php#unix>

This contains version 6.8.3-3 released February 19, 2013.

Go to IM/ImageMagick-6.8.3-3 and compile Image Magick using the sequence

```
make clean
./configure --disable-shared
make
sudo make install
```

It helps to look at the configure output and confirm that shared libraries are disabled.

Now go to `/usr/local` and remove the include library. Also in `/usr/local/lib`, remove the “old” directory and all libraries, leaving only the folder named “ImageMagick-6.8.3”.

Finally, move the entire remaining contents of `/usr/local` to `Universal/local-i386` or `Universal/local-ppc` depending on whether you compiled on Intel or on PPC.

Make sure the Universal folder has a subfolder named “local” containing a subfolder named “bin”. Change directory to Universal, strip the binaries, and lipo them together using the following commands:

```
cd local-i386/bin
strip *
cd ../..
cd local-ppc/bin
strip *
cd ../..
sh liposcript
```

Finally, execute the build instruction using the command

```
sudo sh buildRoot.sh
```

6. BUILDING OLD STYLE LEGACY CONVERT-IM.PKG

To build an old style version, change directory to “OldStyle” and issue the command

```
sudo sh buildOldStyle.sh
```

7. BUILDING THE FINAL INSTALL PACKAGE WITH PACKAGEMAKER

The remaining tasks are not necessary unless you want to make a copy of Convert-IM for preliminary testing.

Now edit `ConvertIMBuild`, the Packagemaker project file, using the GUI interface, as follows.

- (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: ImageMagick-Convert-6.8.3-3
 - (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 ImageMagick-Convert-6.7.6-9 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: ImageMagick-Convert-6.8.3-3
 - (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 ImageMagick-Convert-6.8.3-3 entry in the bottom left column. This item should show a “/usr/local” directory, obtained by dragging root/user/local to the left column. The right side will change to a view with four tabs. The item under the Components tab can be left blank. It is not necessary to edit the items under the Contents tab. The items under the Configuration tab should be edited as follows:
 - (a) Install: root/usr/local (The item on the right should have selected “Relative To Project”)
 - (b) Destination: /usr/local (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.imagemagick-convert-6.8.3-3
 - (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 “ImageMagick-Convert-6.7.6-9-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 ImageMagick-Convert-6.8.3-3-Temp.pkg and “new-name” is ImageMagick-Convert-6.8.3-3.pkg