

# TUG-GHOSTSCRIPT

RICHARD KOCH

## 1. GHOSTSCRIPT

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

## 2. OVERVIEW

Building Ghostscript is done in three steps. The first builds the binaries, which are placed in the folder “Raw Binaries.” Two binaries are built on each build platform, one with X11 support and one without this support. Binaries for 32 bits are made on Leopard PPC and Leopard Intel. Binaries for 64 bits are built on Snow Leopard. Binaries for the latest system are built on Yosemite. The script “liposcript” in Binaries then combines these binaries and places the resulting files in subfolders of Binaries.

In the second stage, Ghostscript is built and installed in /usr/local. Then a buildPackage script copies the resulting installation to the “root” folder in the Build Directory. This script replaces the binaries in this root folder with the carefully constructed combined build binaries in Binaries.

The script makes modifications for TeX Fonts by Arnold-Voisin and for Japanese fonts by Kuroki-Yusuke, Bruno Voisin, and Norbert Preining. The Arnold-Voisin additions were constructed in 2012. The Japanese additions were constructed in 2015 with very difficult testing by Bruno and Norbert.

In the third stage, this root folder is used to construct an install package using the GhostscriptBuild template.

## 3. CONTENTS

A clean package file will contain

- “About” folder with public documentation explaining how the package was constructed
- “Binaries” folder with empty subfolders “10.5”, “10.6” and the file “liposcript”

---

*Date:* May 11, 2016.

- Files “buildPackage.sh” and “buildTeXfonts.sh”
- Packagemaker template files GhostscriptBuild and GhostscriptBuild copy.
- Files “background.jpg”, “License.rtf”, “ReadMe.rtf”, and “Welcome.rtf”
- RawBinaries folder where binaries are first placed after builds on other systems
- “scripts” folder containing files “postinstall” and “setloginpath”
- signPackage.sh file
- “Bruno-Japanese” folder containing Bruno’s final version of updates for Japanese.
- “Bruno-TeXLive” folder containing Bruno’s 2012 additions for TeX Live fonts
- “TUG” folder with this document

To construct the package, download the latest Ghostscript tar file from <http://www.ghostscript.com/> and put the file in the “source” folder. Then unzip this file; the resulting folder is the source code for Ghostscript.

In 2015, very extensive work was done to improve support for CJK fonts by Bruno Voisin, Norbert Preining, and others. Various documents created during that work are preserved in the folder Bruno-Documentation, in case it is necessary to refer back to it.

#### 4. BUILDING BINARIES

Build binaries on Leopard PPC, Leopard Intel, Snow Leopard, and the latest OS. On each such platform, build a version with X11 support and a version without this support.

Before building, move /usr/local to /usr/local-temp. Leave /usr/X11 and /usr/X11R6 alone for X11 support, but move them to /usr/X11-temp and /usr/X11R6-temp to remove this support.

To build:

```
make clean
./configure --disable-compile-inits
make
sudo make install
```

Move all of these binaries to the “Raw Binaries” folder on the machine used to build the actual install package.

Change to the Binaries directory, and type

```
sh liposcript
```

This will lipo the various binaries together. The script will automatically throw away old copies and then create new copies of gs-noX11, gs-X11, gs-X11-64Bit and gs-noX11-64Bit. Leave them in the directories where the liposcript put them. The first two binaries are universal with 32 bit code for PPC and Intel. The last two binaries are universal with Intel 32 bit and 64 bit code. The 32 bit code is required because Snow Leopard runs on early Intel machines without 64 bit processors.

## 5. PREPARING THE ROOT FOLDER

Next edit the file “buildPackage.sh” to reflect the current version of Ghostscript. Currently this file contains “9.19”, referring to folders installed by Ghostscript. Change this number appropriately.

Next make sure the latest TeX Live is installed on this machine and type

```
sh buildTeXfonts.sh
```

This step will create a folder named “fonts” in “source/TeXfonts/” containing symbolic links to the pfb font files in this latest TeX Live distribution.

Next change directory to /usr and rename /usr/local to /usr/local-temp. Copy the Ghostscript source folder to your machine and install Ghostscript using

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

In this step we are not interested in the binaries, but instead in the support files for Ghostscript.

Then change to the main Ghostscript directory of the MacTeX build system and type

```
sudo sh buildPackage.sh
```

This step will copy Ghostscript support files from `/usr/local` to the root directory of MacTeX's Ghostscript build folder. It will add the binaries constructed earlier. Then it will make modifications from Arnold-Voisin so Ghostscript will understand TeX file. It will also make modifications from Kuroki-Yusuku, Bruno Voisin, and Norbert Preining so Ghostscript will understand Japanese files.

## 6. ADJUSTING FILES IN THE GHOSTSCRIPT INSTALL PACKAGE

This package contains some files which are shown to the user during installation. One of these files is a brief Welcome document, a second is a longer ReadMe file, and a third shows the License for the code. After the Ghostscript install package is constructed, you may need to revise these documents. For example, the Welcome document contains the release date. Usually TeX Live authors are overly ambitious and predict a release several months earlier than it actually occurs, and you'll need to change the release date from time to time.

None of these files appear in MacTeX, but they appear if you make an optional Ghostscript Install Package.

## 7. MAKING A STANDALONE INSTALL PACKAGE

This step is optional. If you want an install package containing only Ghostscript, construct it as follows.

This section of the MacTeX package contains a Pagemaker project file named "Ghostscript-Build". Edit this project file using the GUI interface of PackageBuilder 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: "Ghostscript-9.19"
  - (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 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: Ghostscript-9.19

- (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 Ghostscript-9.19 entry in the bottom left column. This item should show a “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.ghostscript9.19
  - (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 “Ghostscript-9.19-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 Ghostscript-9.19-Temp.pkg and “new-name” is Ghostscript-9.19.pkg