

TUG-BINARY

RICHARD KOCH

1. TEX LIVE BINARIES

This is the portion of the TUG documentation which deals with making the TeX Live binaries. Building is a two stage process. First TeX Live and asymptote are compiled on PPC Leopard, Intel Leopard and Intel Snow Leopard. Then the resulting binaries are processed in this directory, forming the universal binaries sent to TUG for inclusion in TeX Live. We also build TeX Live on the latest operating system, currently El Capitan, to check that users who compile themselves will be happy.

A clean package file will contain

- the folder AsymptoteBuild containing information for building asymptote.
- the folder BuildingBinaries, containing only the the key document *BuildStatus-2016*
- an empty folder named “RawCodeFromCompile”
- two scripts, liposcript-universal and liposcript-x86_64, which combine the raw binaries into final form
- “TUG” folder with this document
- xindy-2010 folder with several xindy.mem and xindy.run files

A few other items are in the folder, but less important. The script liposcript-asy is useful in case asy, but nothing else, must be uploaded to TUG. The script compareRawFinal might be useful for comparing 32 bit and 64 bit binaries, and clisp-sources are included just in case.

The main MacTeX folder containing Binary must contain “xz”, the binary which compresses packages, since the scripts in the Binary directory call it.

2. COMPILING THE BINARIES

The “BuildingBinaries” folder contains a crucial document named

BuildStatus-2016

Date: May 11, 2016.

This document contains the precise shell commands needed to build TeX Live binaries on PPC Leopard, Intel Leopard, and Intel Snow Leopard. The document is a text file rather than a TeX file so it can be easily edited. Any change in compile commands should immediately be added to the document. I find it convenient to put a copy of the document on the desktop of each machine, merging editing changes back into the master document from time to time.

Note that we do not make luajit, mflua, asy, or biber on PPC.

Building xindy requires obtaining and compiling CLISP. The document explains exactly how that is done. It only needs to be done once on each platform.

Building asymptote requires obtaining, compiling, and installing three libraries. Complete instructions are included in the AsymptoteBuild folder, in the document

AsymptoteBuild-2016

The end result of this process on Intel Leopard will be a folder of binaries and an isolated binary

- i386-apple-darwin9.8.0
- asy-Intel

The end result of this process on PPC Leopard will be a folder of binaries.

- powerpc-apple-darwin9.8.0

The end result on Intel Snow Leopard will be

- x86_64-apple-darwin10.8.0
- asy-x86_64

Move these folders and files to the Binary/RawCodeFromCompile directory of the MacTeX build system on the machine used to make final packages.

3. MAKING UNIVERSAL BINARIES

To build the universal-darwin binary directory, go to the Binary directory and execute the command

```
sudo sh liposcript-universal
```

This will create a folder named universal-darwin and a file named universal-darwin.tar.xz. Send the xz file to Karl Berry.

To build the x86_64-darwin binary directory, go to the Binary directory and execute the command

```
sudo sh liposcript-x86_64
```

This will create a folder named x86_64-darwin and a file named x86_64-darwin.tar.xz. Send the xz file to Karl Berry.

4. XINDY

In 2011, due to a strange interaction between Xindy and CLisp, the xindy.run and xindy.mem files were corrupt. But the 2010 versions of these files work, and the files have not been changed. So the two liposcripts replace 2011 versions of the files with the 2010 versions. CLisp from 2010 continues to work fine and has not been recompiled in more recent versions through 2016.