Developing and Distributing Cross-Platform Software with PyInstaller

Claude Rubinson
University of Houston—Downtown
rubinsonc@uhd.edu
cjr@grundrisse.org

Slides and files available at:
http://grundrisse.org/cjr/pyhou-201403/

PyHou
Houston, Texas
March 18, 2014

Options for Distributing Python Programs

From https://wiki.python.org/moin/DistributionUtilities

Tool	OS Targets	Limitations
bbfreeze	Windows and *nix	Python 2 only
cxFreeze	All	Library support
Freeze	*nix	Library support
EXE maker	Windows	
py2exe	Windows	
py2app	OSX	
McMillian's	?	No longer exists
PyInstaller	Windows, OSX, Linux, Solaris and AIX (experimental)	

Note: See esky for an auto-update framework (supports py2exe, py2app, cxFreeze, and bbfreeze)

Why did I choose PyInstaller?

- Support for Windows, OSX, and Linux
- Support for 3rd-party packages (specifically sip/PyQt)
- **Laziness, Impatience,** Hubris

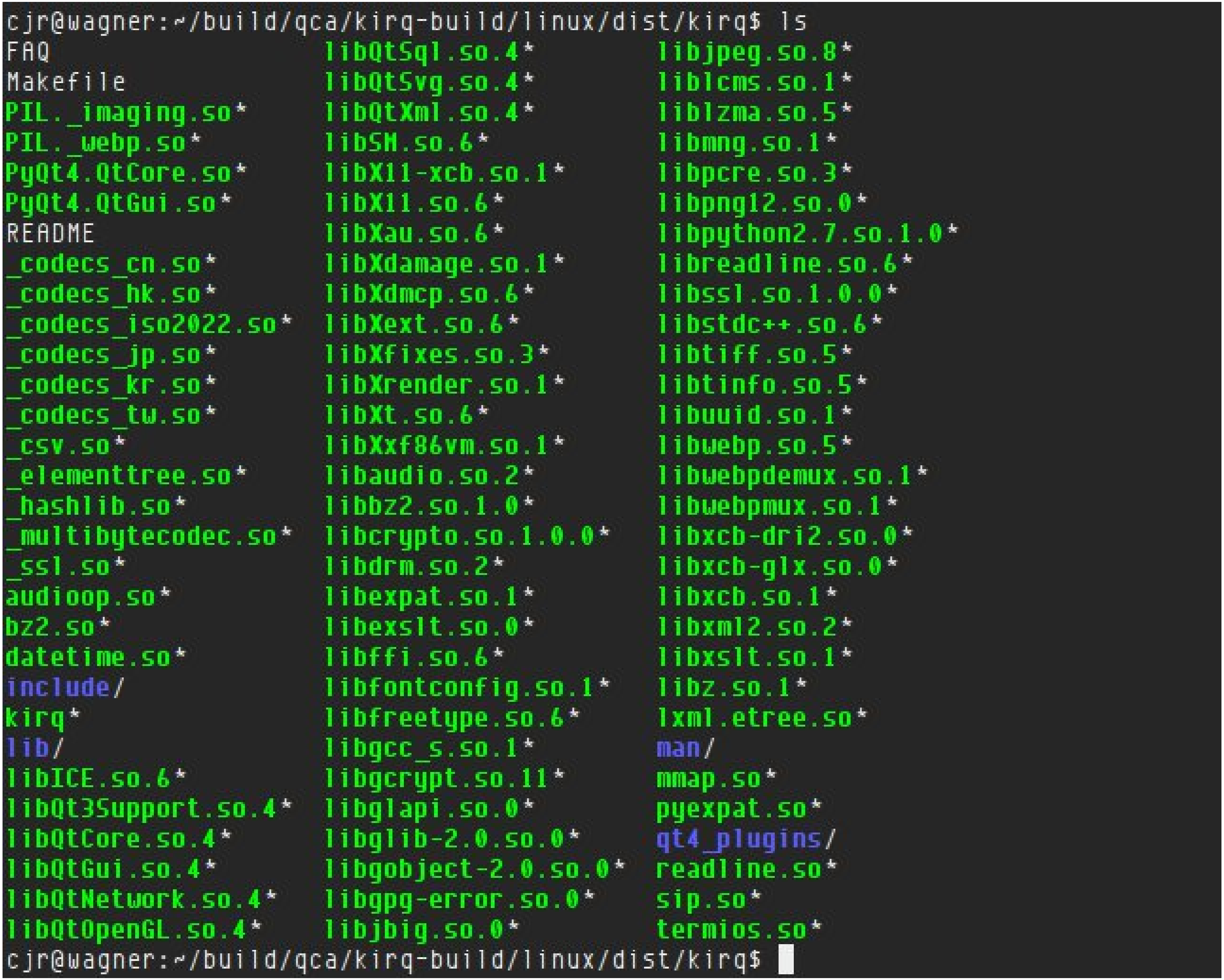
What does PyInstaller do?

- Creates a single directory (or file) that bundles together your Python script (converted to a native executable) with the Python binary and (hopefully!) all libraries/packages that your script depends on

How does PyInstaller do it?

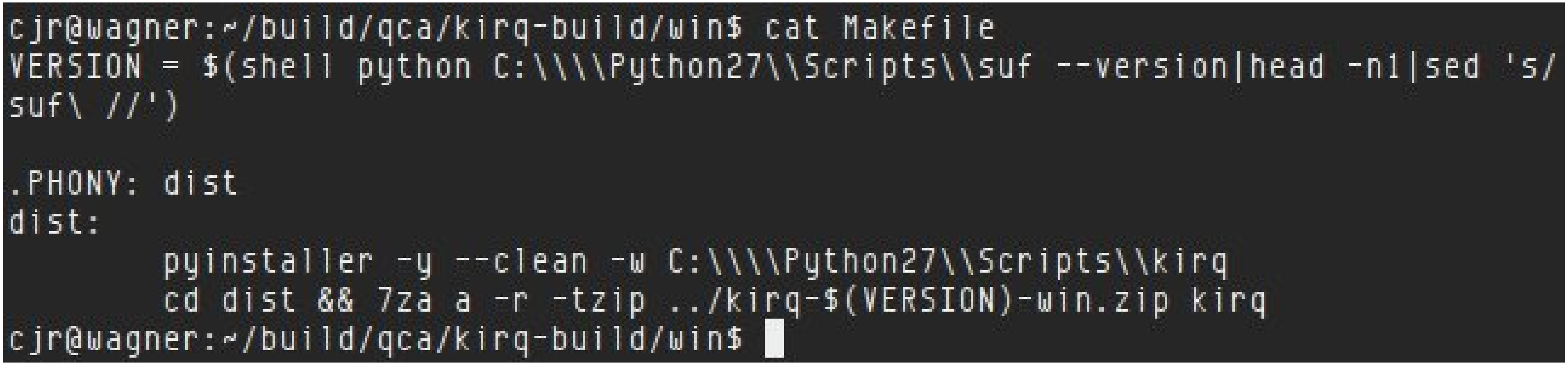
- Follows the `import` chain; includes explicit support for many 3rd-party libraries, and all SIP libraries
- Ideally, is as simple as:

```
   pyinstaller [-w] path/to/your/script
```



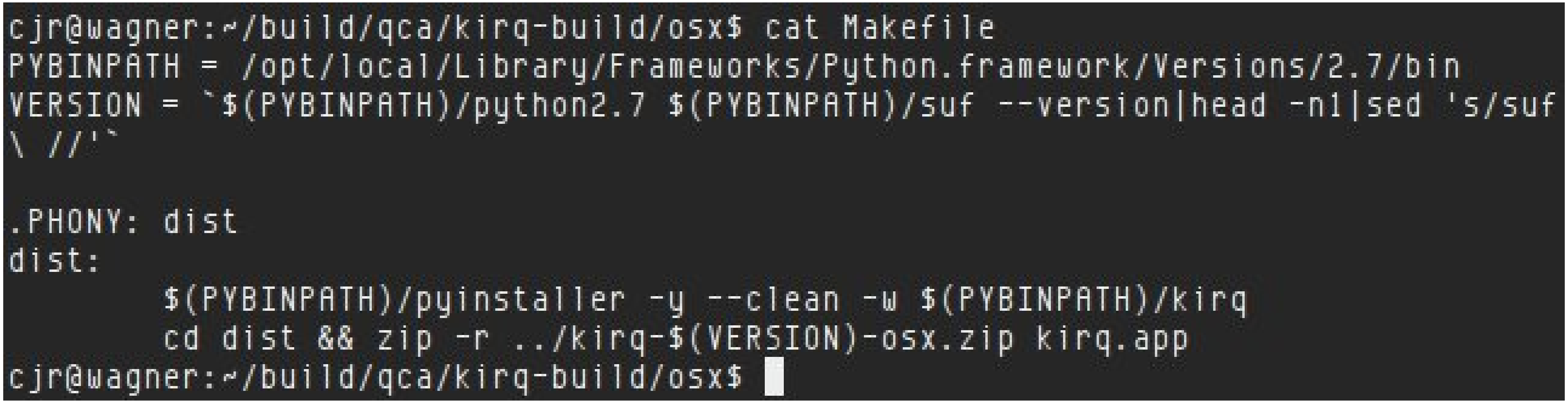
```
cjr@wagner:~/build/qca/kirq-build/linux/dist/kirq$ ls
FAQ                  libQtSql.so.4*        libjpeg.so.8*
Makefile             libQtSvg.so.4*        liblcms.so.1*
PIL._imaging.so*     libQtXml.so.4*        liblzma.so.5*
PIL._webp.so*        libSM.so.6*           libmng.so.1*
PyQt4.QtCore.so*     libX11-xcb.so.1*      libpcre.so.3*
PyQt4.QtGui.so*      libX11.so.6*          libpng12.so.0*
README               libXau.so.6*          libpython2.7.so.1.0*
_codecs_cn.so*       libXdamage.so.1*      libreadline.so.6*
_codecs_hk.so*       libXdmcp.so.6*        libssl.so.1.0.0*
_codecs_iso2022.so*  libXext.so.6*         libstdc++.so.6*
_codecs_jp.so*       libXfixes.so.3*       libtiff.so.5*
_codecs_kr.so*       libXrender.so.1*      libtinfo.so.5*
_codecs_tw.so*       libXt.so.6*           libuuid.so.1*
_csv.so*             libXxf86vm.so.1*      libwebp.so.5*
_elementtree.so*     libaudio.so.2*        libwebpdemux.so.1*
_hashlib.so*         libbz2.so.1.0*        libwebpmux.so.1*
_multibytecodec.so*  libcrypto.so.1.0.0*   libxcb-dri2.so.0*
_ssl.so*             libdrm.so.2*          libxcb-glx.so.0*
audioop.so*          libexpat.so.1*        libxcb.so.1*
bz2.so*              libexslt.so.0*        libxml2.so.2*
datetime.so*         libffi.so.6*          libxslt.so.1*
include/             libfontconfig.so.1*   libz.so.1*
kirq*                libfreetype.so.6*     lxml.etree.so*
lib/                 libgcc_s.so.1*        man/
libICE.so.6*         libgcrypt.so.11*      mmap.so*
libQt3Support.so.4*  libglapi.so.0*        pyexpat.so*
libQtCore.so.4*      libglib-2.0.so.0*     qt4_plugins/
libQtGui.so.4*       libgobject-2.0.so.0*  readline.so*
libQtNetwork.so.4*   libgpg-error.so.0*    sip.so*
libQtOpenGL.so.4*    libjbig.so.0*         termios.so*
cjr@wagner:~/build/qca/kirq-build/linux/dist/kirq$
```

Windows makefile:



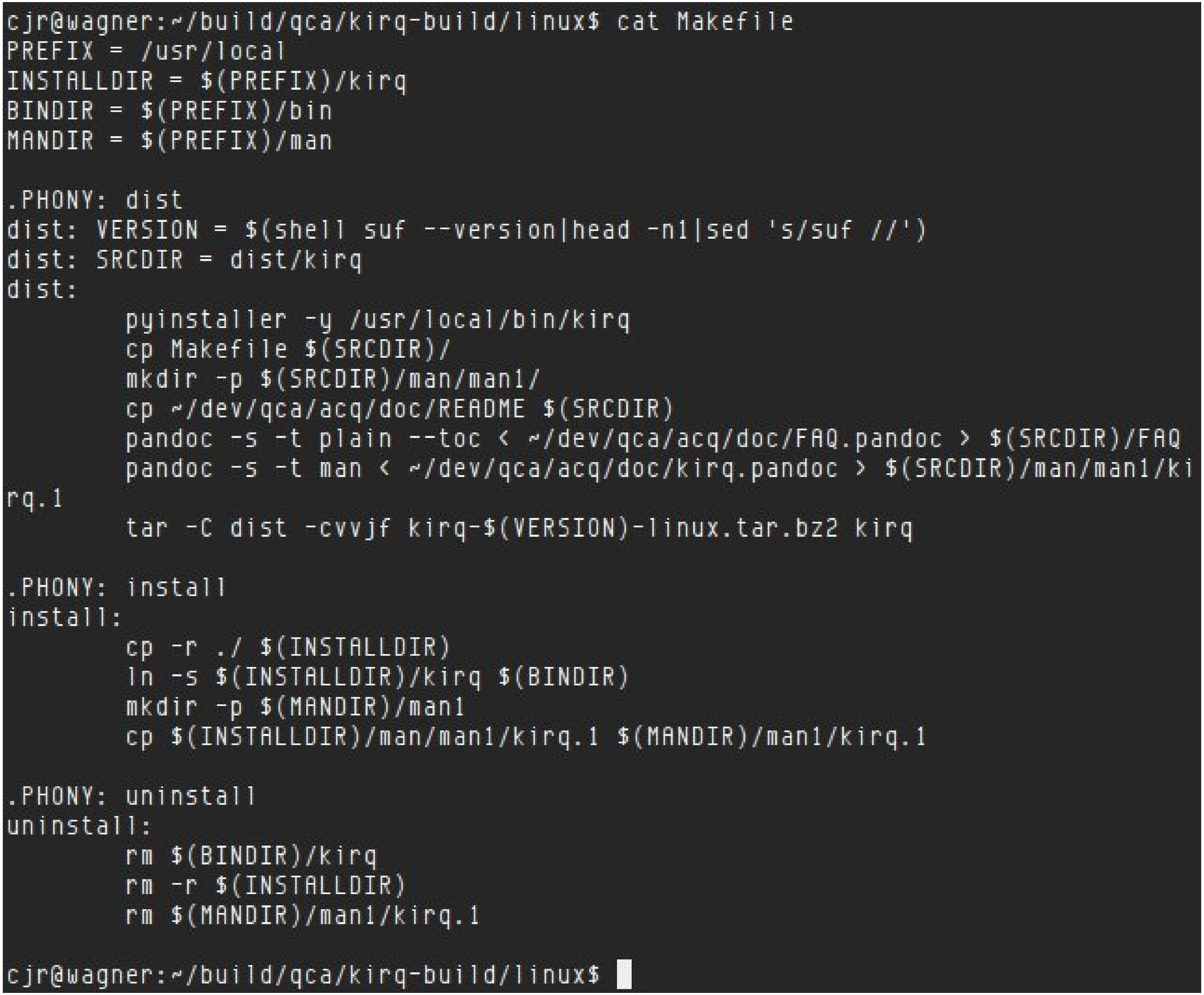
```
cjr@wagner:~/build/qca/kirq-build/win$ cat Makefile
VERSION = $(shell python C:\\\\Python27\\Scripts\\suf --version|head -n1|sed 's/
suf\ //')

.PHONY: dist
dist:
        pyinstaller -y --clean -w C:\\\\Python27\\Scripts\\kirq
        cd dist && 7za a -r -tzip ../kirq-$(VERSION)-win.zip kirq
cjr@wagner:~/build/qca/kirq-build/win$ 
```

OSX makefile:



```
cjr@wagner:~/build/qca/kirq-build/osx$ cat Makefile
PYBINPATH = /opt/local/Library/Frameworks/Python.framework/Versions/2.7/bin
VERSION = `$(PYBINPATH)/python2.7 $(PYBINPATH)/suf --version|head -n1|sed 's/suf
\ //'`

.PHONY: dist
dist:
        $(PYBINPATH)/pyinstaller -y --clean -w $(PYBINPATH)/kirq
        cd dist && zip -r ../kirq-$(VERSION)-osx.zip kirq.app
cjr@wagner:~/build/qca/kirq-build/osx$ 
```

Linux makefile:

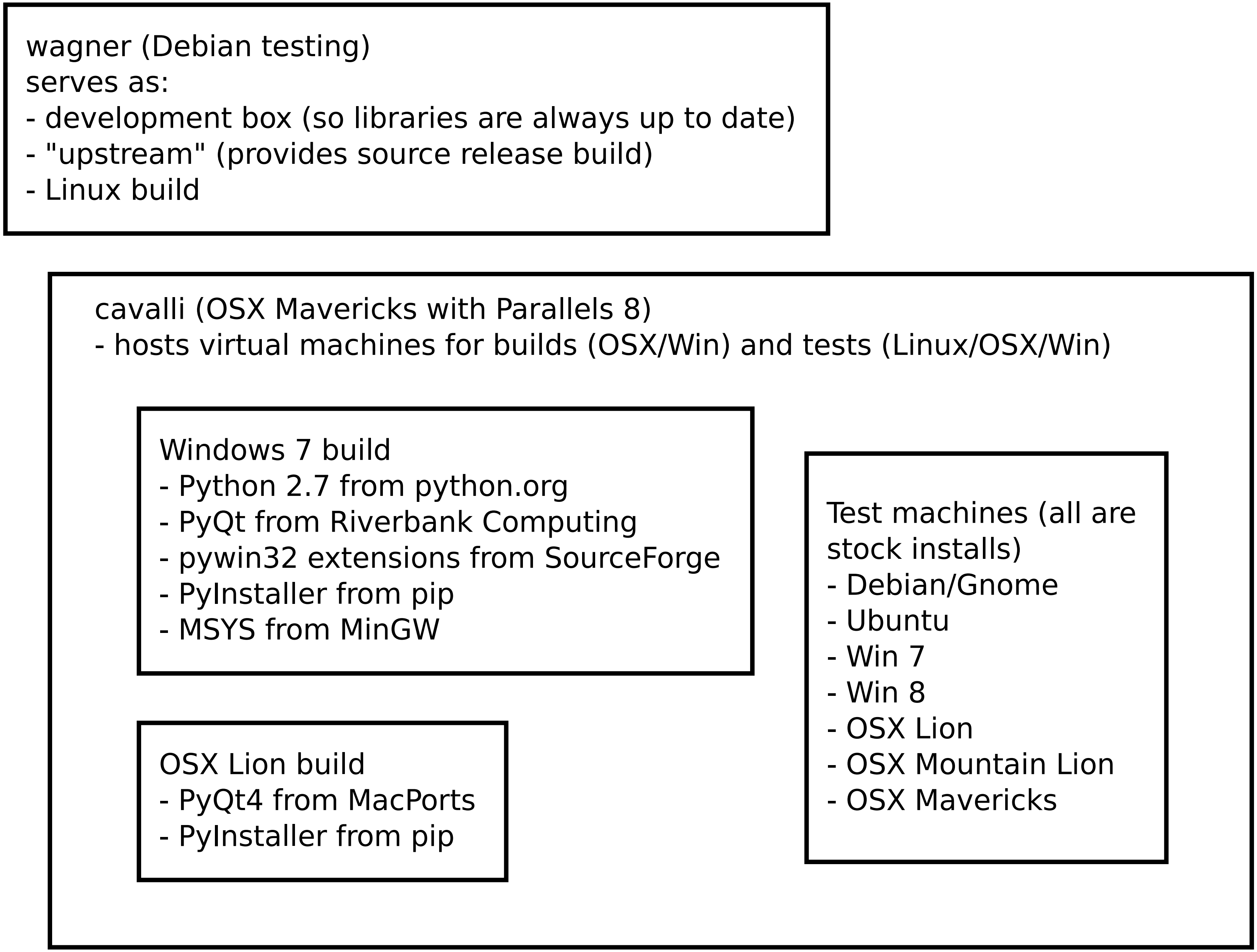


```
cjr@wagner:~/build/qca/kirq-build/linux$ cat Makefile
PREFIX = /usr/local
INSTALLDIR = $(PREFIX)/kirq
BINDIR = $(PREFIX)/bin
MANDIR = $(PREFIX)/man

.PHONY: dist
dist: VERSION = $(shell suf --version|head -n1|sed 's/suf //')
dist: SRCDIR = dist/kirq
dist:
        pyinstaller -y /usr/local/bin/kirq
        cp Makefile $(SRCDIR)/
        mkdir -p $(SRCDIR)/man/man1/
        cp ~/dev/qca/acq/doc/README $(SRCDIR)
        pandoc -s -t plain --toc < ~/dev/qca/acq/doc/FAQ.pandoc > $(SRCDIR)/FAQ
        pandoc -s -t man < ~/dev/qca/acq/doc/kirq.pandoc > $(SRCDIR)/man/man1/ki
rq.1
        tar -C dist -cvvjf kirq-$(VERSION)-linux.tar.bz2 kirq

.PHONY: install
install:
        cp -r ./ $(INSTALLDIR)
        ln -s $(INSTALLDIR)/kirq $(BINDIR)
        mkdir -p $(MANDIR)/man1
        cp $(INSTALLDIR)/man/man1/kirq.1 $(MANDIR)/man1/kirq.1

.PHONY: uninstall
uninstall:
        rm $(BINDIR)/kirq
        rm -r $(INSTALLDIR)
        rm $(MANDIR)/man1/kirq.1

cjr@wagner:~/build/qca/kirq-build/linux$
```

Installation

- Installing PyInstaller is straightforward
  - Available in pip
  - On Windows, depends on pywin32 extensions
- But, PyInstaller is *not* a cross-compiler; need to run a separate build for each operating system target
- Setting up the build/test/deployment system is non-trivial, partially negating the benefit of PyInstaller being cross-platform
- Most important: *Don't test a build on the build machine*

Build and Test Systems



wagner (Debian testing)
serves as:
- development box (so libraries are always up to date)
- "upstream" (provides source release build)
- Linux build

cavalli (OSX Mavericks with Parallels 8)
- hosts virtual machines for builds (OSX/Win) and tests (Linux/OSX/Win)

Windows 7 build
- Python 2.7 from python.org
- PyQt from Riverbank Computing
- pywin32 extensions from SourceForge
- PyInstaller from pip
- MSYS from MinGW

OSX Lion build
- PyQt4 from MacPorts
- PyInstaller from pip

Test machines (all are stock installs)
- Debian/Gnome
- Ubuntu
- Win 7
- Win 8
- OSX Lion
- OSX Mountain Lion
- OSX Mavericks

Snapshot of build files: http://grundrisse.org/cjr/pyhou-201403/

OS-specific Issues

- Linux
  - Must always test Ubuntu because of its popularity, but it's non-standard. Should also test GNOME, KDE, and no window manager.
- Windows
  - Most trouble-free; any Windows-build should run on XP, Vista, 7, or 8.
- OSX
  - Most problematic. Build on the oldest operating system that you want to support; test on all.

Limitations of PyInstaller

- Difficult to debug a failed build
- Must execute PyInstaller within build directory (spews build files)
- Development team is very small
  - bugs take time to be fixed
  - docs can be out of date