Building a Python Mac OS X distribution

The `build-install.py` script creates Python distributions, including certain third-party libraries as necessary. It builds a complete framework-based Python out-of-tree, installs it in a funny place with $DESTROOT, massages that installation to remove .pyc files and such, creates an Installer package from the installation plus other files in `resources` and `scripts` and placed that on a `.dmg` disk image. The installer package built on the dmg is a macOS bundle format installer package. This format is deprecated and is no longer supported by modern macOS systems; it is usable on macOS 10.6 and earlier systems. To be usable on newer versions of macOS, the bits in the bundle package must be assembled in a macOS flat installer package, using current versions of the pkgbuild and productbuild utilities. To pass macoS Gatekeeper download quarantine, the final package must be signed with a valid Apple Developer ID certificate using productsign. Starting with macOS 10.15 Catalina, Gatekeeper now also requires that installer packages are submitted to and pass Apple's automated notarization service using the altool command. To pass notarization, the binaries included in the package must be built with at least the macOS 10.9 SDK, must now be signed with the codesign utility, and executables must opt in to the hardened run time option with any necessary entitlements. Details of these processes are available in the on-line Apple Developer Documentation and man pages.

A goal of PSF-provided (python.org) Python binaries for macOS is to support a wide-range of operating system releases with one set of binaries. Currently, the oldest release supported by python.org binaries is macOS 10.9; it is still possible to build Python and Python installers on older versions of macOS but we not regularly test on those systems nor provide binaries for them.

Prior to Python 3.9.1, no Python releases supported building on a newer version of macOS that will run on older versions by setting MACOSX_DEPLOYMENT_TARGET. This is because the various Python C modules did not yet support runtime testing of macOS feature availability (for example, by using macOS AvailabilityMacros.h and weak-linking). To build a Python that is to be used on a range of macOS releases, it was necessary to always build on the oldest release to be supported; the necessary shared libraries for that release will normally also be available on later systems, with the occasional exception such as the removal of 32-bit libraries in macOS 10.15. For 3.9.x and recent earlier systems, PSF practice was to provide a "macOS 64-bit Intel installer" variant that was built on 10.9 that would run on macOS 10.9 and later.

Starting with 3.9.1, Python fully supports macOS "weaklinking", meaning it is now possible to build a Python on a current macOS version with a deployment target of an earlier macOS system. For 3.9.1 and later systems, we provide a "macOS 64-bit universal2 installer" variant, currently build on macOS 11 Big Sur with fat binaries natively supporting both Apple Silicon (arm64) and Intel-64 (x86_64) Macs running macOS 10.9 or later.

The legacy "macOS 64-bit Intel installer" variant is expected to be retired prior to the end of 3.9.x support.

build-installer.py requires Apple Developer tools, either from the Command Line Tools package or from a full Xcode installation. You should use the most recent version of either for the operating system version in use. (One notable exception: on macOS 10.6, Snow Leopard, use Xcode 3, not Xcode 4 which was released later in the 10.6 support cycle.) build-installer.py also must be run with recent versions of Python 3.x or 2.7. On older systems, due to changes in TLS practices, it may be easier to manually download and cache third-party source distributions used by build-installer.py rather than have it attempt to automatically download them.

1. universal2, arm64 and x86_64, for OS X 10.9 (and later):

   ```
   /path/to/bootstrap/python3 build-installer.py \
       --universal-archs=universal2 \
       --dep-target=10.9
   ```

   - builds the following third-party libraries

     - OpenSSL 1.1.1
     - Tcl/Tk 8.6
     - NCurses
     - SQLite
     - XZ
     - libffi

   - uses system-supplied versions of third-party libraries

     - readline module links with Apple BSD editline (libedit)
     - zlib
     - bz2

   - recommended build environment:

     - Mac OS X 11 or later
     - Xcode Command Line Tools 12.5 or later
     - current default macOS SDK
     - `MACOSX_DEPLOYMENT_TARGET=10.9`
     - Apple `clang`

2. legacy Intel 64-bit, x86_64, for OS X 10.9 (and later):

   ```
   /path/to/bootstrap/python3 build-installer.py \
       --universal-archs=intel-64 \
       --dep-target=10.9
   ```

   - builds the following third-party libraries

     - OpenSSL 1.1.1
     - Tcl/Tk 8.6
     - NCurses
     - SQLite
     - XZ
     - libffi

   - uses system-supplied versions of third-party libraries

     - readline module links with Apple BSD editline (libedit)
     - zlib
     - bz2

   - recommended build environment:

     - Mac OS X 10.9.5
     - Xcode Command Line Tools 6.2
     - `MacOSX10.9` SDK
     - `MACOSX_DEPLOYMENT_TARGET=10.9`
     - Apple `clang`

General Prerequisites

- No Fink (in `/sw`) or MacPorts (in `/opt/local`) or Homebrew or other local libraries or utilities (in `/usr/local`) as they could interfere with the build.
- It is safest to start each variant build with an empty source directory populated with a fresh copy of the untarred source or a source repo.
- It is recommended that you remove any existing installed version of the Python being built:

  ```
  sudo rm -rf /Library/Frameworks/Python.framework/Versions/n.n
  ```