

Basic Installation Instructions: OS X

Original Reference: <https://www.ettus.com/kb/detail/basic-installation-instructions-os-x>

UHD on OS X

UHD has been compiled and installed on Mac OS X 10.6 ("Snow Leopard") through OSX 10.10 ("Yosemite") running any compatible version of Xcode on all recent and many older Macs. Primary support is for 64-bit Intel-based Macs, running 10.6 or newer. There is very little support for getting the background libraries and applications installed on Mac OS X 10.5 or earlier, for 32-bit installs, or for PowerPC based Macs, though any or all of these might be possible.

X11/XQuartz or Terminal

For the purposes of building and using UHD, you can use Apple's Terminal.app if you so choose, no matter how you install UHD.

That said, running almost any graphical interface (GUI) will require downloading and installing X11/XQuartz first. Through OSX 10.8, Apple provided a means to install X11.app, but XQuartz has always been more up to date. Starting in 10.9, Apple no longer provides a full working version of X11.app. Hence, just use XQuartz from the get-go. Note that unless you experiment with using the Quartz interface to various graphical toolkits (e.g., GTK), you must use X11 as the terminal interface for any GUI applications.

Xcode

Apple provides a fully integrated development environment via their Xcode toolkit, which can be downloaded either via the App store or directly from Apple's Developer area depending on the version of OSX in use. Xcode provides the compilers and related development tools needed to build or execute UHD and its dependencies.

Once Xcode is installed, you must still install the Command Line Tools, which can be accomplished by running Xcode.app, then going to Preferences... -> Downloads and making sure Command Line Tools is selected/enabled [feel free to select other downloads too]. You might be able to install the Command Line Tools in a terminal using

```
xcode-select --install
```

but this command will not work with every OSX / Xcode combination (e.g., does not work with OSX 10.8 and Xcode 5, but does work with OSX 10.9 and Xcode 5).

Once the Command Line Tools are installed, UHD and other projects can be installed either from source or, preferably, via MacPorts.

UHD Software Installation on OS X

UHD can be installed via a package manager or from source.

Via MacPorts

We recommend using MacPorts and actively support UHD development on OS X via this method. Using MacPorts, the correct hardware device images are automatically installed for you, and there are no post install tasks. With a single command, you can be up and running in short order.

If you do not already have MacPorts installed, you will need to install it first. Make sure to follow the MacPorts shell environment changes needed such that MacPorts installed executables are found before all others. These are the only changes to the shell environment needed to execute any MacPorts-installed executable. Setting any DYLD environment variable (e.g., DYLD_LIBRARY_PATH) for general use is highly discouraged, because doing so, in our experience, leads to problems down the road that are very difficult to diagnose. OSX provides robust means for correcting DYLD-based issues even after dependencies are installed.

Once MacPorts is installed, UHD and all of its dependencies can be installed by executing

```
sudo port install uhd
```

This method of installing UHD is actively kept up to date by Michael Dickens, and hence is the recommended way to install GNU Radio on OS X.

The latest developments in UHD can be installed via

```
sudo port install uhd-devel
```

Please note that the uhd-devel port, while tested for basic compilation and functionality, is not a formal release and hence should be considered beta software which might contain bugs or major issues.

From Source

Background Dependencies

There are a number of background libraries and applications that must be installed from source or binary in order to compile or execute UHD; a full list is provided in the UHD build guide. These can be obtained by using MacPorts, Fink, HomeBrew, and/or from source / scratch. MacPorts tends to be more up-to-date with respect to new releases, which can be both a blessing and a curse since sometimes new releases are untested and result in build or runtime errors. MacPorts, HomeBrew, and Fink offer thousands of ready-to-install libraries and applications, and hence they are highly recommended to use instead of installing from source / scratch.

NOTE: We highly recommend that you install all dependencies via the same package manager! When issues arise, they are much easier to track down, and your updating to newer versions is also much easier.

Many UHD developers first install UHD using MacPorts in order to get all of the necessary background dependencies installed, then remove just UHD via

```
sudo port uninstall uhd
```

Compiling UHD from Source

Installing UHD from source follows the standard cmake method as found in many places, with a few arguments to make sure cmake always finds the correct version of Python, and uses the desired compiler. First, download the source code either via a release or via GIT.

For example, on OSX 10.8 or 10.9 using Xcode 5's Apple GCC, MacPorts installed into /opt/local (the default), and for Python 2.7 (as installed by MacPorts), issue the following commands from within the UHD source directory:

```
$ mkdir build
```

```
$ cd build $ CC=/usr/bin/llvm-gcc CXX=/usr/bin/llvm-g++ cmake -DPYTHON_EXECUTABLE=/opt/local/bin/python2.7 \
-DPYTHON_INCLUDE_DIR=/opt/local/Library/Frameworks/Python.framework/Versions/2.7/Headers \
-DPYTHON_LIBRARY=/opt/local/Library/Frameworks/Python.framework/Versions/2.7/Python ../host
```

```
$ make
```

If make succeeds, then you can test the build for errors via

```
$ make test
```

To install the build, issue

```
$ sudo make install
```

Selecting another compiler is as simple as changing the CC and CXX pre-arguments to the cmake command. Note that all of the PYTHON defines must point to the same install of Python, otherwise runtime errors are likely to occur.