# Process of Building and Testing OpenStudio for Release

4/27/11

## Creation of Release Build Branch

A specific release branch is created for OpenStudio at the end of each iteration. A branch will also need to be created for any major version releases. The current naming convention for the iteration branch is as follows

* iter<iteration\_number>-build<build\_major>-<build\_minor>-<build\_patch>

So far we have been incrementing the <build\_patch> number by one for each iteration. The other numbers will be changed when it is deemed necessary.

Once the branch is created the version numbers need to be incremented in the following files:

* <Repo\_Path>/CMakeLists.txt
  + Increment CMAKE\_VERSION\_...
* <Repo\_Path>/resources/model/Openstudio.idd
  + Increment Version number at top of file

These version numbers will need to be incremented in both the branch and the trunk to keep them in sync. Alternatively you could increment the version numbers in trunk before branching.

## Building the Releases

I currently build the releases manually so I can keep an eye on the process. This is probably unnecessary and should be automated in the future. It should be possible to set up the continuous build machines to do the release builds, although one may need to kick off the builds manually because of the branch. One possible complication is that the Windows XP build machine is using Visual Studio Express, and the releases have not been fully tested with that configuration.

### Configuration Settings

The key part of building the releases is making sure that the CMake configuration settings are done properly. Settings common to all platforms are:

* BUILD\_PACKAGE should be turned on
* BUILD\_TESTING should be turned on
* BUILD\_WITH\_OPENSSL should be turned on
* CMAKE\_BUILD\_TYPE should be set to “Release”
* CMAKE\_VERSION\_BUILD should be set to the svn repository version. Ideally this should be the same number on all platforms. You will probably have to pick one to settle on.

Windows:

* BUILD\_CSHARP\_BINDINGS should be turned on
* On Windows it is critical to remember to click the Solution Configurations dropdown box and set it to Release in Visual Studio. This is an easy step to forget, and will result in a wasted build.
  + It is possible to make sure this setting gets used when building using a CTEST script.

Mac:

* CMAKE\_OSX\_ARCHITECTURES should be set to “i386;x86\_64”. This turns on universal binaries
* CMAKE\_OSX\_DEPLOYMENT\_TARGET should be set to “10.5”
* CMAKE\_OSX\_SYSROOT should be set to “/Developer/SDKs/MacOSX10.5.sdk”

Linux:

* There are no extra setting that need to be configured for Linux builds

## Sanity Testing the Release Builds

After the builds have been created, I install and test OpenStudio on clean machines, meaning computers that have not been set up with the developer tools for each platform. We do not currently have a Mac machine that does not have the developer tools installed. It is possible to do this testing in a Virtual Machine.

The process for performing the Sanity tests is documented in the repository in “<repo\_path>/doc/Tests/OpenStudio Installer Tests.docx”. If failures are found, it is up to the team to decide whether they are stoppages, or if the product can be released as is. The sanity test document will need to be updated from time to time as the software changes.

## Uploading the Release Builds

In order to distribute the Release builds we need to upload them to the OpenStudio website. At this time we are staging the files on Nick Long’s Mac server and then some scripts handle pushing the builds to the website.

The key directory on the Mac server is at /Users/nlong/staging/ and there are several subdirectories that relate to each of the different types of builds. The subdirectories and their uses are:

* …/staging/doxygen – Doxygen documentation for major releases
* …/staging/doxygen\_it – Doxygen documentation for iteration releases
* …/staging/iterations – The installers for iteration builds are placed here
* …/staging/nightly – The installers for nightly builds can be placed here (currently unused)
* …/staging/releases – The installers for major releases are placed here
* …/staging/ruby-docs – Ruby documentation for major releases
* …/staging/ruby-docs\_it – Ruby documentation for iteration releases

After the sanity test has been completed the installers for each iteration build are placed in the …/iterations/ directory. I currently copy them by hand using a flash drive, but it could be set up so that one could copy them over via ssh or something. The build manager would need to coordinate with Nick to set up this process.

### Documentation

At the end of each iteration I also build the documentation on one of the machines. It doesn’t really matter what platform is used to build the documentation. I then copy the Doxygen and Ruby (RDoc) documentation into the appropriate directory.