

Notes on building the RM tools on supported platforms

For the most part, the build instructions given on the main SCons page are complete as I've tried to keep them up to date.

External Libraries Needed

Qt version

I don't remember exactly which version of Qt is used on each of the different operating systems but it varies from version 4.5 through 4.8.

The versions being used by OS are:

- 4.8 - Windows, OS X Mountain Lion
- 4.7 - OS X Snow Leopard
- 4.5 - RHEL 5, RHEL 6

In any case, the functionality we use from Qt is actually fairly minimal and doesn't change version to version. The version used was mainly selected to get a version that would compile without issue on the selected operating system.

Boost Spirit Library

This is only used on the Linux systems and the version used is spirit-1.6.4-miniboot. The source tarball can be found by logging on to fermilnx-v03 and moving to the externals/src directory.

stx-exparser Library

Again, this is only needed on the Linux systems. As far as I can tell this is a dead software project and is no longer supported. The version we use is 0.7 and the tarball can be found in the same place as the Boost Spirit Library tarball.

Building RM tools

Before building, make sure that Qt is properly build and installed and that the other libraries are installed as well if building on a Linux system. On the Linux systems, these are all installed into the externals directory (off the glastrm home directory). This directory is sym-linked to a directory on u35 based on the operating system of the machine you are logged on to using the value of @sys (to see this value for a particular Linux machine type, log in and issue the command `ls -lsysname`) You can follow the sym-link structure by looking at the directory listing for the glastrm home directory. If you are building on a new OS, you need to make sure this is set up properly. On Windows and Mac OS X Mountain Lion, these can be set to where ever you would like. Currently they are at ??? on Windows and /data/RM/externals on Mac OS X Mountain Lion.

Once the libraries are installed in the proper location, do the following:

1. check out the grits-cpp package from the CVS repository in the location of your choice (On the Linux systems this is just in the home directory of the glastrm account. On Mountain Lion it is in /data/RM/tools. For Windows the programs have to be build on a separate computer per the notes on the main page.)
2. move to the grits-cpp/src directory
3. run '`<path to externals>/bin/qmake -recursive`' to generate the makefiles
4. make
 - a. **Note:** on Linux might first need rerun qmake (regenerate Makefile(s)), then do make clean to get rid of objects built for another OS
 - b. **Note:** on Linux if this OS is not going to be handling workflow, consider only building programs under the releaseManager directory. Simply cd one level further down to releaseManager, then issue make command
5. make install (from same directory where you did the make)

This will install the programs in the grits-cpp/bin directory. On Linux this directory is sym-linked like the externals directory based on the OS you are logged in on. (Note: for a new Linux OS version, make a subdirectory named after appropriate sys name - e.g. amd64_70 for rhel 7 and then create a subdirectory bin of this.)

Updating RM Tools

If you are updating the RM tools due to a code change simply follow the instructions above for building but instead of checking out the package move to the location of the already checked out source code and do a cvs update. For the linux systems, since it uses the same code base across all the different OS flavors, after installing the code you should do a 'make clean' to remove the OS specific object, library, and binary files from the build tree. Then log into a computer of the next OS you want to build and repeat the process.