GMAT USING BUILD SYSTEM

How to Use the Build System

The Build process can be broken down into two phases:

Phase 1 : Dependency Configuration

This phase includes dependency configuration scripts in the form of batch files (.bat for Windows) and (.sh for Linux/Macintosh). If new dependencies are necessary these or existing dependencies need to be removed these scripts would be the place to make those changes.

The scripts can be found in the *gmatroot/depends/* directory.

Phase 2 : Automatic Build Script Generation & Compilation

This phase makes use of CMake, the cross-platform, open-source build system. (CMake is a family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and workspaces that can be used in the compiler environment of your choice.)

The CMake files for the GMAT build system are located in *gmatroot/src/* directory. Each CMake file is named CMakeLists.txt which is a required name by the CMake system.

There are currently four CMake build files. One for each project (base, gui, and console) as well as one main Script.

These scripts are located in the following directories:

- src – (CMakeLists.txt – Main Cmake Script)
    - base– (CMakeLists.txt – Base Library Cmake Script)
    - console– (CMakeLists.txt – Console Project Cmake Script)
    - gui – (CMakeLists.txt – GUI Project Cmake Script)

NOTE: You should not add/remove any files or dependencies to the Visual Studio or Eclipse files. CMake will now handle the generation of everything. All Compiler flags will also need to be set in the CMake files.

Adding/Removing Source Files

To add or remove source files to the projects you will need to modify the CMakeLists.txt file that corresponds with the project.

To do this find the “Source Files” section in the CmakeLists.txt file and add or remove the file path and name to the “Source Files” array.

Adding/Removing Include Directories

To add or remove include files to the projects you will need to modify the CMakeLists.txt file that corresponds with the project.

To do this find the “include directories for header files” section, in the CmakeLists.txt file and add a “INCLUDE_DIRECTORIES” line.

An Example would be:

```
INCLUDE_DIRECTORIES("/path/to/my/include/")
```

Additional CMake Functionality

To add additional functionality to the Cmake files (ie. Compiler flags, definitions, new linked libraries, etc.), please consult the CMake documentation.

Building with CMake

After CMakeLists.txt files have been modified you can build/rebuild your project by doing one of the following:

On Windows:

- Rerun Build.bat located in *gmatroot/build/* directory or
- Rebuild with Visual Studio IDE (Upon a rebuild the CMake Cache will be auto updated and the new source files and include files will appear in Visual Studio).

On Linux/Macintosh:

- Rerun Build.bat located in *gmatroot/build/* directory.

Build Output

GMAT Binaries and dependencies will be built to the *gmatroot/application/bin* directory. Intermediate builds will go to the *gmatroot/build/src/<configuration>/<project_name>.dir/* folder. The auto generated Visual studio solution files and make files will go to the *gmatroot/build/<os>/ folder*.

Additional Notes:

Developer

- wxWidgets need preconfigured setup.h file copied over to its Operating System specific folder before build. This preconfigured file is located in *gmatroot/depends/bin/wx/setup.h*. This file is automatically copied by configure.bat/configure.sh script. This will only be necessary if creating dependencies manually.

Windows

- Build process will not work with Visual Studio 2012.
  - This is due to wxWidgets 2.8.12 dependency not being compatible. VS complain about .pch files not working.
- Build will work on Windows 8 with VS 2012.
- Only Targeting builds with Visual Studio. NMake generated CMake files on windows have build problems.

Linux

- Only tested on Ubuntu 12.04 but should work on older versions as well (Still need to test)
  - Ubuntu 12.04 requires symlinks for gl, glu, and mesa libraries (Old versions were installed to */usr/lib/* new versions installed to architecture specific (e.g. i386-linux-gnu and x86_64-linux-gnu respectively).
  - Because of the symlink issue user will need to run configure script as sudo. This is an issue with wxWidgets 2.8.12. Ticket for this is at: *http://trac.wxwidgets.org/ticket/13376* and *http://trac.wxwidgets.org/ticket/13375*
    - Option to take out of configure script and have user run symlink command manually.
- The wxWidget dependency “NetUtils” *contrib/net* does not build on Linux32 or 64.
- Editor.cpp causes compiliation to fail on Linuxusing wxWidgets stc editor. Lin 747 is windows specific. There is no such thing as a Boolean in Linux. Change Boolean to bool and you can use stc editor on Linux.
- wxWidgets is hardcoded to use gtk2 libraries on Linux. The configure script installs these to the default folder of */usr/local/lib*. Requires sudo on configure script.
  - Option to take out of configure script and have user run make install from wxwidget dependency folder.
  - Option to use –prefix on wxWidgets make install to install to local dependency folder.
  - Option to try and included wx-Config in CMake which would allow support for more than just gtk2 libs.

Macintosh

- Only tested on OSX Lion.
- The wxWidget dependency “NetUtils” *contrib/net* does not build on Macintosh 32 or 64.
- Editor.cpp causes compilation to fail on Macintosh using wxWidgets stc editor. Lin 747 is windows specific. There is no such thing as a Boolean in Macintosh. Change Boolean to bool and you can use stc editor on Macintosh.
- wxWidgets is hardcoded to use carbon libraries on Macintosh. The configure script installs these to the default folder of */usr/local/lib*. Requires sudo on configure script.
    - Option to take out of configure script and have user run make install from wxwidget dependency folder.
    - Option to use –prefix on wxWidgets make install to install to local dependency folder.
    - Option to try and included wx-Config in CMake which would allow support for more than just carbon libs.