




TMine External To-do and Status



Regularizing TMine

Now that the TMine external can be used with GlastRelease it would be good to keep the source somewhere in CVS more reputable than under users/jrb and to automate building and installation as much as possible. I propose the following:

New home in CVS

1.  Make a new top-level module, TMineExt-scons, in CVS
2.  Do a recursive copy of most of the contents of users/jrb/TMineDiv to the new module. This should preserve history.
3.  Make sym links for contents of SConsFiles (e.g. allExternals.scons, SConstruct, etc.) as is done for other containers

Possible gotchas

-  Doing file copies and the like directly in the CVS repository makes me a little queasy. Among other things, should check that resulting files and directories have the correct file permissions. *Seems ok. Checkout by other users, tagging work correctly.*
-  Versions of SConstruct, allExternals.scons, etc. used by TMine may have diverged somewhat from the official set in SConsFiles. May need to incorporate some elements from the former into the latter. *The few changes needed have been incorporated in SConsFiles-00-10-01.*

If we want to SCons to automatically build new tags it would be only a small amount of additional work. We would then have immediate feedback if the new tag failed to build properly for some OS. However note that the builds created by the SCons RM are not organized exactly as required for externals, and that there is additional hand work (new tag of IExternal/TMineExt for CMT; modifications to allExternal.scons) before a new tag of an external can actually be used by anything.

Automation

- Create two python scripts, one to check out and build a tag, another to install in a suitable location.

By default **buildTMine** checks out and builds in a subdirectory of current working directory named after the cvs tag checked out. By default **installTMine** will look for the build to install in the same directory buildTMine would have created, and by default installs under the directory specified by the GLAST_EXT environment variable (not directly under, but where you would expect given external name, version, etc.).

buildTMine.py

..has been committed to CVS (grits-tools/python/buildTMine.py), so far tested for rhel4 and Windows vc71 compiler. Here is the help output:

```
Usage: buildTMine.py [options] [major minor patch] where either args or TAG option is required

Options:
-h, --help            show this help message and exit
-t TAG, --tag=TAG      Tag to check out.  Alternatively specify major, minor,
                        patch arguments
-d, --debug            Compile debug [default: nodebug]
--nodebug              Compile debug [default: nodebug]
-g GLASTEXT, --glast_ext=GLASTEXT
                        Path to Glast externals [default: value of GLAST_EXT
                        environment variable]
--nooptimized          Compile optimized [default: optimized]
--optimized            Compile optimized [default: optimized]
-p PARENT, --parent_dir=PARENT
                        parent directory in which to checkout and build
                        TMineExt [default: cwd]
-r CVSROOT, --repository=CVSROOT
                        cvs repository root [default: value of CVSROOT env]
-s SCONSCMD, --scons_command=SCONSCMD
                        Path to scons command [default: scons]
-v VARIANT, --variant=VARIANT
                        Use as SCons variant directory [default: NONE]
--32bits               Ask for 32 bit compile [default: True; i.e., 32 bit compile]
--64bits               Ask for 64 bit compile [default: False; i.e., 32 bit compile]
--fake                Don't really do anything [default: False]
```

On Windows there is an additional option, --compiler, which may take on values vc7 (default) or vc9.

If --fake is specified, the script will print out the cvs checkout command and scons command it would have executed, e.g.

```
jrb@noric10 $ python buildTMine.py 3 2 0 --fake
cvs command would be
cvs -d /nfs/slac/g/glast/ground/cvs co -r TMineExt-03-02-00 -d TMineExt-03-02-00 TMineExt-scons
scons command is:
scons --rm --with-GLAST-EXT=/afs/slac/g/glast/ground/GLAST_EXT/redhat4-i686-32bit-gcc34/ --variant=NONE --
compile-opt --32bit all
```

installTMine.py

..has been committed to CVS (grits-tools/python/buildTMine.py), so far tested only for linux (rhel4) and Windows. Here is the help output:

```
[jrb@noric10 jrb]$ python SConsApps/grits-tools/python/installTMine.py --h
Usage: installTMine.py [options] major minor patch

Options:
  -h, --help                show this help message and exit
  -s SRCDIR, --srcdir=SRCDIR
                           Location of source build. Defaults to TMineExt-major-
                           minor-patch
  -c COMPILER, --compiler=COMPILER
                           Compiler used for compile. Possible values include
                           gcc34, gcc41, etc.. [default: derive from gcc version
                           in path]
  -g GLASTEXT, --glast_ext=GLASTEXT
                           Path to Glast externals [default: value of GLAST_EXT
                           environment variable]
  --destdir=DESTDIR         parent directory for installation. If non-empty path
                           is specified, must already exist. If defaulted, path
                           under GLAST_EXT will be created [default: suitable
                           directory under GLAST_EXT]
  -v VARIANT, --variant=VARIANT
                           SCons variant directory used in build [default: NONE]
  --fake                    Don't really do anything [default: False]
```

The description for --compiler is different on Windows. There possible values are vc7 and vc9; vc7 is the default.

Here is an example of use, where the GLAST_EXT environment variable has been set to /nfs/farm/g/glast/u54/jrb/myGLAST_EXT:

```
[jrb@noric10 jrb]$ python SConsApps/grits-tools/python/installTMine.py --fake 3 2 0
Would create /nfs/farm/g/glast/u54/jrb/myGLAST_EXT/TMineExt/3.2.0/gcc34
Fake Copy TMineExt-03-02-00/data to /nfs/farm/g/glast/u54/jrb/myGLAST_EXT/TMineExt/3.2.0/gcc34/data
Fake Copy TMineExt-03-02-00/include to /nfs/farm/g/glast/u54/jrb/myGLAST_EXT/TMineExt/3.2.0/gcc34/include
Fake Copy TMineExt-03-02-00/bin to /nfs/farm/g/glast/u54/jrb/myGLAST_EXT/TMineExt/3.2.0/gcc34/bin
Fake Copy TMineExt-03-02-00/exe to /nfs/farm/g/glast/u54/jrb/myGLAST_EXT/TMineExt/3.2.0/gcc34/exe
Fake Copy TMineExt-03-02-00/lib to /nfs/farm/g/glast/u54/jrb/myGLAST_EXT/TMineExt/3.2.0/gcc34/lib
```

Old Stuff

Windows scripts

I've written .bat files to do build and install for Windows (7.1 compiler only); python implementations should be able to handle all supported OS's and would be more flexible in other ways. The .bat files have this argument structure:

buildTMine maj min patch [debug-opt]

installTMine maj min patch [debug-opt] [destroot] [srcroot]

The first three arguments are required and specify major, minor and patch number for the cvs tag of TMine to be checked out. **debug-opt** defaults to **opt**. **buildTMine** checks out and builds in a subdirectory of current working directory named after the cvs tag checked out. By default **installTMine** will look for the build to install in the same directory buildTMine would have created, and by default installs under the directory specified by the GLAST_EXT environment variable (not directly under, but where you would expect given external name, version, etc.)

Original to-do list

This table was created Jan.27 and updated March 1.

Description	Status	Comments
Build TMine external for supported platforms	Exists for Windows vc71, vc90 and rhel4 32 bit. <i>And also mac</i>	This is all we need for current GR RM builds. Should be straightforward to build others (<i>mac actually took some doing</i>)
Install externals in appropriate spots	<i>Done</i> -	Need ability to write to AFS GLAST_EXT (<i>this got fixed</i>)
Test use in GR	For rhel4 GlastClassify app, which links to TMine, builds and runs	"Runs" is maybe an overstatement: it exits in orderly fashion because it wasn't called with proper args. <i>Leon has since used it more realistically within GR.</i>
Create interface package for CMT	See users/jrb/TMineExt-CMT-interface. <i>Better yet, see IExternal/TMineExt</i>	
Add entry in allExternals.scons	Done in private copy. <i>Done in production</i>	
Modify GR requirements: remove use of TMine pkg; add one for new interface pkg	<i>Done</i>	
Add line to externals.scons in GR-scons packageSettings subpkg to use TMineExt	Done in private copy. <i>Done in production</i>	