

CMT

Current Issues

March, 2007

Using Python scripts to make up for CMT's deficiencies on Windows:

http://www-glast.slac.stanford.edu/software/core/minutes_02272007.htm

Purpose (as I understand it)

- Build VS2005 project files using python scripts rather than CMT, continue to use CMT for managing package lists
- Ran into trouble with packages building shareable libraries - due to issue with dumpbin (Tracy has a fix) - potentially will replace with python script or some other utility

Other Tools that are potentially impacted:

- MrStudio
- ReleaseManager
- Workbook documentation

Questions:

- Python scripts will be used exclusively on Windows while linux will continue to use CMT?
- Do more recent versions of CMT support VS2005 or are there any plans to do so? Does CMT-central have any wisdom to offer?
- Can the UW terminal server be set up to allow those who are interested to play with the new python scripts to see how it works and how easily it is used?

MRStudio/MRvcmt

For what concern these tools, they are both just wrappers on external system commands, mainly CMT and CVS .. at the moment on windows when you click on the *config* button they are calling CMT to generate also the vcproj files, while the solution file is generated internally by some Ruby code .. so, I don't see any problem if, instead of calling CMT, MRStudio/MRvcmt will call python scripts to generate vcproj and some other scripts to generate sln files (that btw will free me from the task of writing sln file generators, something I will be really happy), apart from the fact that python need to be installed on the user box (not a problem at all).

Status as of 11 March 2007 --Toby

Please see the child page that has my take on the above issues.

The scripts create sensible project and solution files for all the Glast packages that I have tried. There is a classic problem in this project: I am making two changes at the same time: the mechanism for writing the new files, and the compiler/studio. The latter is a major issue, in that many of the libraries have not been compiled: e.g. Gaudi, Geant4 and cppunit. (I have ROOT, CLHEP). Also there are apparent differences in creating/using shareables that I don't understand. Finally, whenever I have to consider modifying a CMT requirements file, I recoil in horror at the terrible system we (starting with me) have forced on the community.

I should drop this until I can get some help, but for now I'm trying to follow this philosophy: http://en.wikipedia.org/wiki/Eat_one's_own_dog_food

12 April 2007 --Toby

Not much has happened in the last month, but I feel the need to address two elements of skepticism about this project:

1. CMT or SCons can generate vcproj files equivalent to what we have now
2. The ROOT facility for creating dll's is not worth pursuing, since CMT, especially after the fix, is just fine.

and furthermore, we have discovered that cppunit has a shareable, so that is no longer an impediment for proceeding, at least with Science Tools.

Since irfs/irfInterface seems to be a good place to test this, I present the following demonstration. Note that I am **not** using the construction `__declspec (dllimport)`.

```

D:\Users\burnett\python\pycmt>python -m pycmt.main --rebuild irfLoader
starting build for package ['irfLoader']: please wait for output
building VS2005 files for package:
    irfLoader v0r6p3 d:\users\burnett\sensitivity\irfs, configuration vcc80dbg
constituents:
    library irfLoader ../src/*.h ../src/*.cxx ../irfLoader/*.h ../irfLoader/*.cxx
    application test_irfLoader test/*.cxx
----- Rebuild All started: Project: irfLoader/irfLoader, Configuration: Debug Win32 -----
Deleting intermediate and output files for project 'irfLoader/irfLoader', configuration 'Debug|Win32'
Compiling...
Loader.cxx
Performing Pre-Link Event...
bindexplib -o ../vcc80dbg\irfLoader\symbols.def irfLoader ../vcc80dbg\irfLoader*.obj
lib /machine:ix86 /def:../vcc80dbg\irfLoader\symbols.def&nbsp; /out:../vcc80dbg\irfLoader.lib
Creating library ../vcc80dbg\irfLoader.lib and object ../vcc80dbg\irfLoader.exp
Linking...
LINK : ../vcc80dbg\irfLoader.dll not found or not built by the last incremental link; performing full link
LINK : warning LNK4098: defaultlib 'MSVCRTD' conflicts with use of other libs; use /NODEFAULTLIB:library
Embedding manifest...
Build log was saved at "file:///d:\Users\burnett\sensitivity\irfs\irfLoader\v0r6p3\vcc80dbg\irfLoader\BuildLog.htm"
irfLoader/irfLoader - 0 error(s), 1 warning(s)
----- Rebuild All started: Project: irfLoader/test_irfLoader, Configuration: Debug Win32 -----
Deleting intermediate and output files for project 'irfLoader/test_irfLoader', configuration 'Debug|Win32'
Compiling...
test.cxx
Linking...
LINK : ../vcc80dbg\test_irfLoader.exe not found or not built by the last incremental link; performing full link
Embedding manifest...
Build log was saved at "file:///d:\Users\burnett\sensitivity\irfs\irfLoader\v0r6p3\vcc80dbg\test_irfLoader\BuildLog.htm"
irfLoader/test_irfLoader - 0 error(s), 0 warning(s)
===== Rebuild All: 2 succeeded, 0 failed, 0 skipped =====
done!
D:\Users\burnett\python\pycmt>python -m pycmt.main --execute=test_irfLoader irfLoader
running d:\users\burnett\sensitivity\irfs\irfLoader\v0r6p3\vcc80dbg\test_irfLoader.exe
....
OK (4 tests)

```

This shows issuing two commands: rebuilding the package, and running the test program.