

SCons Introduction

Introduction

We are planning to move away from CMT as our build tool in favor of a the more flexible SCons. What is SCons? "SCons is an Open Source software construction tool---that is, a next-generation build tool. Think of SCons as an improved, cross-platform substitute for the classic Make utility with integrated functionality similar to autoconf/automake and compiler caches such as ccache. In short, SCons is an easier, more reliable and faster way to build software." - <http://scons.org/>

Getting Started

The latest version of SCons installed at SLAC is version 1.2.0.

Please Note: As of March 10, 2009, ScienceTools is the only checkout package fully prepared to be built using SCons. GlastRelease is a work in progress, however 15-49-00 is fully set up to use SCons.

SLAC Linux

SCons is installed at SLAC in `/afs/slac/g/glast/applications/SCons`

Desktop Machines

Prerequisites:

Python 2.5.1-g11 available from the installer [see the workbook for details]

CVS access [see the workbook for details]

Obtain SCons from <http://scons.org>

Installer

There are two types of distributions: User Release which provide all the binaries and associated files required for running a pre-build version of the applications. Developer Releases will provide all source and associated files required for rebuilding and debugging the code as necessary.

User Releases are available, while Developer Releases are a work in progress. Please see the [RMViewer User Guide](#)

External Libraries

In our move to SCons, we have decided to also modify our directory structures for the external libraries to provide a directory structure which includes compiler in its path. For example, CLHEP's include and lib directories are in: `redhat4-x86_64bit/CLHEP/1.9.2.2/gcc34/`

The reorganized external libraries are installed at SLAC and are available in the V: drive on windows and on Linux in `/afs/slac/g/glast/ground/GLAST_EXT`
`redhat3-i686-32bit/ redhat5-i686-32bit/ redhat4-i686-32bit/ redhat4-x86_64-64bit/ tiger-i386-32bit/`

To set up the externals on a desktop, you may download them via the RMViewer. For more information concerning the RMViewer see: [RMViewer User Guide](#)

Tagging Convention

Under CMT our checkout packages, such as ScienceTools and GlastRelease are tagged as virjpk. In SCons, these same packages have tags of the form: `<packageName>-00-00-00 HEAD` versions of a checkout package are referred to as: `<packageName>-HEAD-1-1234` and similarly LATEST uses the form: `<packageName>-LATEST-1-1234`

GoGui

A GUI interface, the MRvcmt replacement is available and is named GoGui. Please see Joanne's lovely documentation:

[GoGui Status and Laundry List](#) and the [User Guide](#)

Command Line Use of SCons

Please see: [Using SCons at the Command Line](#)

SCons build output

When SCons is performing the build process it will put files in the following subdirectories that are located in the same directory as the SConstruct file:

- `include/[packageName]` - all the globally shared header files for [packageName]
- `bin/[variant]` - all wrapper scripts for the variant. A variant specifies the OS and compile options such as debug or optimized, such as `redhat4-x86_64-64bit`
- `exe/[variant]` - the binaries

- lib/[variant] - all the libraries for the variant. This variant string is the same as above
- pfiles/[packageName] - all the pfiles for [packageName]

Other such output directories will be created in the future and they will follow the same convention. If the contents of the directory is dependent on the OS or the compile options, it will include the variant sub directory. If it is independent of such changes it will not include that directory.

References

<https://confluence.slac.stanford.edu/display/SAS/SCons>

<http://www.slac.stanford.edu/exp/glast/ground/software/notes/GoGui/GoGui-use.shtml>