How to get the latest ScienceTools and build them using SCons

In the following, I compiled some useful information / links / problems that I encountered during my try to install the ScienceTools myself on a machine at CEA Saclay (i.e., external). No claim of completeness whatsoever, and a lot of these things might be obvious for most.

CVS : getting the latest ScienceTools

Look at the ScienceTools scripts online: CVS repository

You need a SLAC user account. If you don't have it, see Obtaining a SLAC account (2014), Setting up SLAC Accounts (2012), Getting a New SLAC Account (~ 2000)

Setting up CVS:

Follow the steps in Using the GLAST SLAC cvs Repository, then

--> write a mail to Thomas Glanzman to get CVS access for your account.

--> you will have to logout and login for the change to take effect.

--> you should have (if you have an external account):

```
CVSROOT -> :ext:YOURSLACUSERNAME@centaurusa.slac.stanford.edu:/nfs/slac/g/glast/ground/cvs
CVS_RSH -> ssh
```

Check the available versions of ScienceTools: New Release Manager web interface

How to get the ScienceTools (here, version LATEST-1-4167):

```
cvs co -r ScienceTools-LATEST-1-4167 ScienceTools-scons
```

Get the CVS status:

```
cvs status
```

Get the CVS update:

```
cvs up
```

SCons: Building the ScienceTools

SCons version SCons 1.3 required (SCons)

How to use SCons: SCons for Fermi/LAT: An Overview, Making Builds with SCons

Environment variable $GLAST_EXT:

> *"The environment variable GLAST_EXT should be suitably defined. (...)*
>
> *Note: The --with-GLAST-EXT option must always be supplied; others are optional under most circumstances."*

This is the variable defining the directory of external programs (e.g. python).

I pointed the GLAST_EXT variable to the ScienceTools RELEASE, which is **not** the place where I want to install the latest ScienceTools version.

```
$GLAST_EXT=/dsm/saplxglast/glast/sas/Binary/ScienceTools-RELEASE-10-01-01-sympy
```

How to build the ScienceTools:

(assuming you want to build a package called 'astro')

```
scons --with-GLAST-EXT=${GLAST_EXT} --compile-debug astro
```

I put some example output at the end of this page

Remark (python code):

python code, let's say the script foo.py example in

```
ScienceTools-scons/pointlike/python/uw/like/foo.py
```

will be copied to

```
ScienceTools-scons/python/uw/like/foo.py
```

when you build the respective package (here: pointlike. For the example output, see end of this page)

So do **not change the code in the python package directly**

Remark (dependencies):

SCons should handle dependencies correctly. However, at the moment there seems to be an issue when compiling the skymaps **before** astro package. Follow this thread:

> ⚠ STGEN-160 - Jira project doesn't exist or you don't have permission to view it.

Environment

Obviously, in order to use your brand new installation of the ScienceTools, you would need to adjust your environment.

You might consider changing the following environmental variables:

```
PATH
ST_INST
INST_DIR
LD_LIBRARY_PATH
TCL_LIBRARY
DYLD_LIBRARY_PATH
PYTHONPATH

## e.g.
export PATH=$MY_ScienceTools:${PATH}:.
```

E.g., one could create a second bashrc script that sets the environment so that you can chose between the 'standard' installation and your own one.

Example output of building with SCons:

```
jschmid@sappcfermi ScienceTools-scons$scons --with-GLAST-EXT=${GLAST_EXT} --compile-debug pointlike
scons: Reading SConscript files ...

This build is running on:  sappcfermi.extra.cea.fr

Argument list (one per line):
=>  /opt/core-3.1-amd64/scons/2.3.4/bin/scons
=>  --with-GLAST-EXT=/dsm/saplxglast/glast/sas/Binary/ScienceTools-RELEASE-10-01-01-sympy
=>  --compile-debug
=>  pointlike


Checking for C++ header file CLHEP/Vector/defs.h... (cached) yes
Checking for C++ library CLHEP... (cached) yes
( ................)
Install file: "pointlike/build/redhat5-x86_64-64bit-gcc41-Debug/python/uw/utilities/fitstools.py" as "python/uw
/utilities/fitstools.py"
scons: done building targets.
scons: printing failed nodes
scons: done printing failed nodes
```