

GlastRelease-scons build status

There are two needed external libraries that have yet to be incorporated into SCons (for rhel4 machines): **Obf** and **OmniOrb**. **As of today, Obf and OmniOrb have been installed for the rhel4 machines**, so what remains is them being incorporated into Scons. This will happen today.

All packages have been built with the exception of the following:

Package (failed to build)	Comment	Status
AnalysisNtuple	A linking error occurs when it tries to link against FluxSvc: Builds the shared library fine, but throws an error when AnalysisNtuple tries to link against it: /usr/local/lib/libf2c.a(main.o)(text+0x10c): In function `main': : undefined reference to `MAIN__' lib/redhat4-i686-32bit/libFluxSvc.so: undefined reference to `__ctype_b' collect2: ld returned 1 exit status Solution I had to make some changes to FluxSvc/SConscript. The FluxSvc shared library had to be built a certain way. See FluxSvc further down in this table.	Pass
CRflux	I need to resolve some undefined references, possibly via editing the SConscript and CRfluxLib.py files	Pass
OnboardFilterTools	depends on Obf	Pass
CalRecon	depends on (header files from) CalXtalResponse	Pass
RootConvert	depends on Obf	Pass
RootIo	depends on headers from RootConvert package	Pass
G4Generator	I Need to resolve some undefined references. This either has to do with the SConscript or directory structure of the package.	Pass
G4Propagator	depends on G4Propagator conforming to coding conventions. The header file missing here is located in G4Generator package, specifically G4Generator/src)	Pass
GCRCalib	depends on ITkrFilterTool.h from TkrRecon, but source files are not pointing to the correct location for ITkrFilterTool.h. Also, TkrRecon should have that header file "ITkrFilterTool.h" (used by external packages), put in the appropriate directory.	Pass
HepRepCorba	HepRepCorba/build/redhat4-i686-32bit/src/HepRep.hh:6:28: omniORB4/CORBA.h: No such file or directory (depends on OmniOrb)--	Pass
IdfReader	depends on Obf	Pass
digiRootData	depends on Obf	Pass
RootAnalysis	looks like this depends on digiRootData	Pass
calibGenCAL	calibGenCAL/build/redhat4-i686-32bit/src/Optical/fitMuonCalibTkr.o(text+0x392): In function `main': : undefined reference to `CalUtil::CalAsym::CalAsym()' collect2: ld returned 1 exit status (*)seems: *** [calibGenCAL/build/redhat4-i686-32bit/fitMuonCalibTkr] Error 1(*) I verified the following, with no success: (1) CalUtil was added to the list of libraries to link against and this still produces an error (2) #include "CalUtil/SimpleCalCalib/CalAsym.h" is properly included in fitMuonCalibTkr.oxx Solution The function in the SConscript file, used to copy the public headers of a package to the top-level includes/<package> directory, "listFiles," has a bug in it. Sometimes it does not copy everything. In the SConscript file, there is a call to a tool, 'registerObjects.' One of the parameters is, "includes = listFiles(...)," which specifies the public headers that need to be copied over. I manually copied the entire public directory of CalUtil to the top level includes/CalUtil directory via cp -r CalUtil/CalUtil includes/CalUtil Next, I deleted the CVS directories that I copied over.	Pass
calibGenACD	depends on digiRootData	Pass
calibGenTKR		Pass
OnboardFilter	depends on Obf	Pass
CalXtalResponse	I Need to resolve some undefined references. Strange linking errors when trying to build the test application: I get a series of errors similar to In function `test_CalCalibSvc::TestCalibSet::TestCalibSet()': : undefined reference to `CalUtil::CalAsym::CalAsym()' This error is the above error for calibGenCAL. My guess is that resolving the errors for calibGenCal will resolve the errors for this package.	Pass
Event	Need to resolve some undefined references	Pass
calibTkrUtil	In violation of package convention (GeoConstants.h is located in calibTkrUtil/src/test/), but should be in calibTkrUtil/calibTkrUtil. A header in the public directory needs this file, so its causing compiler errors:- calibTkrUtil/build/redhat4-i686-32bit/calibTkrUtil/TkrNoiseOcc.h:17:26: GeoConstants.h: No such file or directory In file included from calibTkrUtil/src/TkrNoiseOcc.oxx:1: calibTkrUtil/build/redhat4-i686-32bit/calibTkrUtil/TkrNoiseOcc.h:34: error: `g_nTower' was not declared in this scope calibTkrUtil/build/redhat4-i686-32bit/calibTkrUtil/TkrNoiseOcc.h:34: error: `g_nTkrLayer' was not declared in this scope calibTkrUtil/build/redhat4-i686-32bit/calibTkrUtil/TkrNoiseOcc.h:34: error: `g_nView' was not declared in this scope calibTkrUtil/build/redhat4-i686-32bit/calibTkrUtil/TkrNoiseOcc.h:35: error: `g_nTower' was not declared in this scope etc... Status I emailed the package owner, explained the problem, and requested the reorganization of the package. Waiting for response... Solution After the "ok" from Johann, I moved the calibTkrUtil/src/test/GeoConstants.h to the calibTkrUtil/calibTkrUtil.	Pass

FluxSvc (Passed
)

Builds the shared library fine, but throws an error when AnalysisNtuple tries to link against it:
/usr/local/lib/libf2c.a(main.o)(.text+0x10c): In function `main':
: undefined reference to `MAIN__'
lib/redhat4-i686-32bit/libFluxSvc.so: undefined reference to `__ctype_b'
collect2: ld returned 1 exit status

Solution

([here](#))

Basically I had to do two things

(1) Create a source file, which I arbitrarily named "flux_c_type_fix.cxx." The source file contains the source from here [here](#)

`ctype_fix=tempEnv1.SharedObject(source = ['src/flux_c_type_fix.cxx'])`

`FluxSvc = libEnv.SharedLibrary('FluxSvc', [ExposureAlg, FluxAlg_os, FluxSvc_os, FluxSvcRandom_os,
OrbitSvc_os, PointInfoAlg_os, PointInfo_os, FluxSvc_dll_os, FluxSvc_load_os] + [ctype_fix])`