

# SCons on Windows - Status and Problems

## What Works

\* There is a script, [offlineObf-win.py](#), which will take as input a CMT-style obf external and [obf-version-specific text file](#) and create a reorganized one which SCons can understand.

\* One can check out GlastRelease-scons and run SCons with a null target

```
scons --with-GLAST-EXT=%GLAST_EXT% --vc7 NoTarget
```

or use GoGui to do the same thing and, if your external libraries are up to date, it will succeed. This doesn't build anything, but it does confirm that all the SConscript files are sensible and that external libraries and include paths are accessible and usable.

\* Some packages build successfully and their test programs run successfully: nearly all of ScienceTools and some of GlastRelease

\* GlastSvc can be built and its test program runs successfully as long as *it's not built debug*. Probably other Gaudi packages which don't depend on too much else exhibit the same behavior.

## What Doesn't Work

### Defining package environment variables

SCons is set up to install everything needed at run time and everything needed by package A from package B at build time (that is, B's public includes and libraries). Hence the environment variables MYPACKAGEROOT are not useful. They have been superseded with MYPACKAGEXMLPATH, MYPACKAGEDATAPATH, etc. For GlastRelease a new category was invented with members of the form MYPACKAGEJOBPTIONSPATH. The definitions are made by the program (Gleam, test\_xxx, etc.) itself at initialization time by calling a special routine in facilities::commonUtilities. This routine checks to see which install directories exist by trying to open them (e.g., starting from the installation directory, xml/xmlGeoDBs) and defines the corresponding environment variables (XMLGEODBSXMLPATH). The code works fine on Linux and, in a simple stand-alone program, works fine on Windows as well. But it does not work (any more, if it ever did - not clear) in the context of ScienceTools or GlastRelease.

### Linking (sometimes)

The GlastSvc problem described above is almost certainly due to a conflict of run-time libraries: debug for our code and non-debug for the gaudi external. This can cause problems at run-time if an entity built against one run-time library allocates memory which an entity built against a different run-time library tries to deallocate. CMT apparently manages by linking everything, even debug builds, against the non-debug run-time.

### More grunt work for externals

All the externals are now in good enough shape to get through the SCons initialization/configuration needed to build a null target. However they haven't all been exercised yet to the extent of actually building something. All the ones used in ScienceTools should be ok, but just about anything I haven't yet exercised is likely to need some tweaking. One such is the ldf external. There may be others. On the plus side, MySQL seems ok and gaudi is ok except for the problem described above.

### Environments

There are at least two kinds of problems:

1. failing to propagate everything that's needed
2. letting inappropriate settings "leak in"

The first is at the very least a problem for GoGui, though at one time it wasn't. I suspect the change from the two-step set-up process (SCons creates a vbscript; then local or remote user runs it in their installation to create a .bat file) to a single jscript either introduced a problem on its own or is not being used properly within GoGui.

2. is particularly an issue on machines with more than one version of Visual Studio. Unfortunately, in these turbulent times, that is the norm so we must have a system which works properly under these circumstances. It's evident from the SCons mailing lists that this is very difficult for them. I'm not convinced they're handling all cases correctly, and I *know* we're not.

### SCons bugs

(in addition to possible environment-handling bugs mentioned above). In our situation (using variant build directories so that builds of different types, made from the same source, can co-exist; using the --interactive option so that developers don't have to wait a minimum of several minutes for any sort of rebuild, no matter how trivial the change to source) SCons just doesn't get it right. It doesn't seem to keep track of the connection between a source file and the copy of it in the build directory. I've sent a bug report (months ago, plus a reminder a few days ago) and the SCons developers have acknowledged that it is a bug, but nothing has happened since. It's just not high on their list of priorities. They're putting a lot of effort into getting builds to work on Windows under a wide variety of circumstances, but I suspect they don't often encounter people who do their primary development on Windows; else bugs involving --interactive would have been fixed long ago.

## What Might Help

## **Bugs in SCons itself**

I don't know how to put more pressure on the SCons folk to fix their problems (particularly the ones associated with --interactive; they're actively working on environment-related problems). Since the problems I've seen a) don't exist on Linux b) don't exist on Windows when you invoke SCons fresh for each operation I think it's likely the fix is not that complicated and probably doesn't involve a lot of code. Maybe we could find it and submit it as a patch. But anyone who hopes to find it would have to spend some time familiarizing him/herself with a good bit of SCons. It would speed things up if such a volunteer already was quite comfortable with python.

## **Linking problems**

If I knew reliably what CMT was doing (e.g. literal compile and link commands) in all circumstances where it succeeds and SCons doesn't that would be a big help. It's not so difficult to get SCons to add particular options or libraries once I know what they should be. But I'm not at all adept at digging the information out of CMT.

## **Other problems with externals**

Mostly this just comes down to duplicating whatever obscure magic is in the CMT requirements file for that external or, in some cases, it might have to do with patterns in the \*Policy packages. Whoever handles the external for CMT probably has the special knowledge needed to get it to go on SCons.

## **Environment set-up**

This one is very messy and I have no good ideas how to break off a discrete piece of it for someone else to work on. If there is anyone willing to put some serious time and effort into learning about the problems\* and working on a solution, I'd welcome a partner.

\* I think there are at least two: coming up with a reliable, comprehensible mechanism for generating a set-up file and understanding what's involved in dealing with machines with multiple installations of Studio.