SCons tools to-do

To be tackled by Navid and Joanne:

_setup

Make _setup.bat (for Windows) and _setup (for other OSes) a separate target. Eliminate _setup.vbs. Mostly affects the tool generateScripts.py.

June 9 Have made a separate target; also renamed _setup on Linux to _setup.sh for clarity. Mods were made to generateScripts (which could use some clean-up) and to SConstruct, to add the lines which invoke the Builder defined in SConstruct. Contents of _setup.bat are not yet complete.

June 15 Implemented a slightly different arrangement. _setup.vbs should be routinely generated as part of a build, but its function when run is not to set up the environment, but rather to create a (non-portable, containing absolute paths) file _setup.bat which, when run, will set environment variables. End users can create a suitable _setup.bat for their installation by running _setup.vbs from the command line. Developers can do the same or they can get SCons to create _setup.bat by building a special SCons target **setup**.

Externals handling

Reorganize externals.scons. Separate from code; keep common parts in SConsFiles (used by all containers). Individual containers should only need to specify which externals (and which versions) are wanted. Will probably take the form of

1. A file allExternals.scons in SConsFiles which will contain information (e.g. paths, library groups) about all supported versions of all available externals. See a couple drafts of how this might look: Navid's is in CVS as SConsFiles/allExternals.scons; Joanne's is attached.
2. One or more files in SConsFiles to do "code-like" things: e.g., set up swig, form lib set of externals (needed on Windows to make solution files), etc.
3. Replace per-container file externals.scons with something that will
    a. choose from among externals in allExternals, making a list called externalLibraries (as the current one is).
    b. invoke externals tools as needed. Will depend on precisely which externals the container is using.

June 11 (approx.) We have something which we believe is ready to go. See SConsFiles/allExternals.scons and SConsFiles/processExternals.scons in CVS.

Invocation in SConstruct will look like this:

```
allExternals = SConscript('allExternals.scons')
usedExternals = SConscript('externals.scons', exports = 'allExternals')
SConscript('processExternals.scons', exports = 'allExternals usedExternals')
```

The per-container externals.scons looks like this:

```
Import('allExternals')
usedExternals = [ ]
# several lines like the following, one for each external used by the container
usedExternals.append({'name' : 'cfitsio', 'iversion' : 'v3060'})
# .. and finally
Return('usedExternals')
```

Target names

Separate out target names, probably into master file, to be kept somewhere in SConsFiles, and optional per-container file. These files would be accessible both to SCons itself and to GoGui. Format will be two lists (one or both of which may be empty). First is list of global targets (e.g. **includes**); second is list of generic per-package target names to which package name should be prepended (e.g. **-includes**, resulting in a target for each package in the container like **facilities-includes**)