

# SCons-Windows Status October 22, 2008

## Windows Status

Can create a simple-minded project file and solution file but

- doesn't yet reference the right set of files. Simplistic use of SCons-provided FindSourceFiles function produces a list of files which is
  - too exclusive (leaves out .h's)
  - too inclusive (includes .exe's like linker)

However, in principal we have access to the full dependency tree, so, with some effort, we should be able to do a reasonable job with this

- The SCons-produced project file invokes SCons to do the building, not VS.  
This has value - in particular, it means the same logic is employed on all platforms to determine dependencies and build order - but is not likely to satisfy developers, especially since SCons start-up time on Windows is unpleasantly long.

## How to proceed

We (N & J) were thinking of starting with the SCons-produced file(s) and modifying it so that VS is invoked to do the build. This is likely to lead to some of the same kinds of maintenance problems we've had with cmt/MRvcmt. We need input from users on how project and solution files should be organized. Currently, when MRvcmt is asked to configure a package, in addition to doing cmt configure it also issues the command

```
cmt build vsnet
```

which

- makes project files per cmt constituent
- makes a solution file per package (<packagename>.sln) referencing only the project files

If you ask MRvcmt for a build, it constructs a comprehensive solution file <packagename>\_workspace.sln which references project files from \*all\* packages used by any constituent of the package being built, then invokes VS with this solution file to do the build.

Is this (one project file per build target, one solution file per project) what we should shoot for? What if we kept all the little project files but had only one installation-wide solution file?