

Porting from project_makefiles to pb_makefiles

Why port?

Well, aside from the fact that the pb_makefiles are new, and thus by definition cool, there are a couple of concrete reasons why you might want to switch over to using pb_makefiles.

- The new pb_makefiles will not change the time it takes to build a project from scratch, but they should dramatically decrease the time it takes to rebuild a project when only one or two files have been modified.
- You can build just a specific subproject by invoking "make all@MySubproject.subproj".
- You can examine the value of a variable in any of your subprojects (i.e., make sv@SecondLevel.subproj@FirstLevel.bproj V=ALL_CFLAGS)
- You can toggle optimization, profiling, and debugging independantly using OPTIMIZE=YES, PROFILE=YES, and DEBUG=YES.

How to port

Edit your `makefile.preamble` to include the following three lines at the top of the file:

```
MAKEFILEDIR = $(NEXT_ROOT)/NextDeveloper/Makefiles/pb_makefiles
OTHER_RECURSIVE_VARIABLES = MAKEFILEDIR
include $(MAKEFILEDIR)/platform.make
```

The first line specifies that you want to use the new makefiles. The second line ensures that you use the new makefiles for subprojects as well, and the third line includes the correct `platform.make` file (the only file included before the preamble).

Common Problems

In theory, it should all Just Work. In practice, here are some of the more common errors you may run into.

- target file ``XXX'` has both `:` and `::` entries.

This error probably means that you are overriding a rule that you should not be. If you want to extend the behaviour of a rule, define a rule which does your work and set up a dependency. (i.e. after-install: do-my-stuff)

- you get a compile error of the form "../foo.h not found" or "bar.h not found"
foo.h or bar.h should be project headers.
- you set a variable in the postamble, but it is not being used.
Ideally, your preamble should only contain new variables and your postamble should only contain rules. There are few situations where you should override existing variables. If you are in such a situation and the postamble is not working, then come talk to me.
- you deleted a .o file to force a rebuild, but the subproject was not rebuilt
"make remove-timestamps" will remove all optimization timestamps and force the next build to exhaustively check every subproject. Alternatively "make RECURSION=FULL" will disable recursion elimination.