

## Preprocessor and conditional compilation

During development you may want to generate lots of debugging info as your program runs. But you definitely do not want to have this in the final release (extra code, performance degradation, aids in reverse engineering). Use #define and the preprocessor for this.

In the following if the \_\_DEBUG symbol is defined the loglotsofdebuginfo() code is inserted into the final executable, if not the loglotsofdebuginfo() function is not included in the file, the compiler never sees it.

```
//Define the debug symbol, can make this foolproof by defining the symbol
//as part of the build process this way you can not accidently leave
// __DEBUG defined in a release build,
#define __DEBUG
:
//sometime later..
:
#ifdef __DEBUG
loglotsofdebuginfo(exception);
#endif
```

You can name the symbol anything you want it does not have to be \_\_DEBUG . In fact many companies will make custom builds of their products for their customers using this technique. Additionally the first person to ask me how to do this in class will get 5 extra points on a single project