Subject: Re: Global variables in Upp
Posted by jerson on Tue, 14 Sep 2010 01:17:26 GMT
View Forum Message <> Reply to Message

A couple of things I learned

1. Declaring a global variable/struct in a H file that has header locks does not ensure that the variable will not be defined again. See this thread message 28688.

What I found a possible solution is to keep the declaration to the variable in the H file as extern var and declare the variable in any one of the C files that make my project.

the second alternative to this is to put the definition of the globals into the .icpp file. Now, the linker is happy and does not shout about duplicate definitions. This variable could well be the Image class you want your entire package to share.

in the H file, I put
extern T_Settings Settings;
extern T_Measure Measurements;

in the icpp file I put
T_Settings Settings;
T_Measure Measurements;

From the docs, this is what I got
.icpp
This file type is recognized as .cpp source file, but unlike regular .cpp, which might be placed into the library first and eventually eliminated by linker, .icpp one is always linked as object file. This forces the file to be always linked. The rationale of this is that you can put module initialization code into .icpp that is linked into the executable even when code is not referenced from other files.

2. I learned how to split my file into understandable modules like the s7dataarchiver code. Earlier, I had one BIG cpp file that had all the functions exposed by the class. Now, I have the functions grouped logically into separate CPP files.

3. Many utility functions are just _not documented_ for example hexstring, stringtohex etc.

4. Wav files support is lacking as yet. It is not a major game changer for me at the moment, but I need to code this part to output a siren/klaxon sound on fault conditions.

Overall, I think Upp has everything and more. It just needs a little work on the docs and I'm willing to help in any way.

Regards