Subject: multiple classes include-problem
Posted by michael on Mon, 25 Feb 2008 21:35:12 GMT
View Forum Message <> Reply to Message

This seems to be a beginners-problem in understanding classes...

Lets say i have two classes and two header-files:

class1.cpp

```
#include "class1.h"

class1::class1()
{
//some code
}
```

class1.h

```
#ifndef _testapp_class1_h
#define _testapp_class1_h

using namespace Upp;

#include "class2.h"

class1
{
//some code
};

#endif
```

class2.cpp

```
#include "class1.h"

class2::class2()
{
//some code
}
```

class2.h

```
#ifndef _testapp_class2_h
#define _testapp_class2_h
```

```
using namespace Upp;

class2
{
//some code
};

#endif
```

Using MinGW compiling and linking went well, no errors, when using MSVC9 i get several "already defined" errors.

I'm sure this is no good class-design, but this was my only successful attempt to get the application running.

I need to access members of both classes in both classes.

How should a better class-design look like?

Thanks for helping.

Subject: Re: multiple classes include-problem
Posted by michael on Tue, 26 Feb 2008 09:22:24 GMT

View Forum Message <> Reply to Message

Ok, it seems to be a problem with my IMAGECLASS:

```
Linking...
editdialog.obj : error LNK2005: "public: static class Upp::Iml & __cdecl MyImg::Iml(void)" (?Iml@MyImg@@SAAAV0Upp@@XZ) already defined in
 prohibisZA.obj
```

I include Class1.h in both Class1 and Class2, so this define in Class1 seems to be the problem:

```
#define IMAGECLASS MyImg
#define IMAGEFILE <prohibisZA/prohibisZA.iml>
```

How can i use #ifndef in this case?

Subject: Re: multiple classes include-problem
Posted by mrjt on Tue, 26 Feb 2008 10:14:46 GMT

View Forum Message <> Reply to Message

It's diffcult to tell without seeing more source, but there is a common reasons for this error.

Have you included <Draw/iml.h> in the header file? You should rather use <Draw/iml_header.h> in the header, and <Draw/iml_source.h> in ONE source file (with the same #define IMAGECLASS). See CtrlCore for an example of this.

Subject: Re: multiple classes include-problem
Posted by bytefield on Tue, 26 Feb 2008 13:03:09 GMT

View Forum Message <> Reply to Message

I don't know if i do good but I was having problems with Images, so i simply use different classes for the same *.iml file(maybe it duplicate the images stored in executable, don't know). Here is an example (maybe bad one   )

```
#ifndef _ImageEx_ImageEx_h
#define _ImageEx_ImageEx_h

#include <CtrlLib/CtrlLib.h>

using namespace Upp;

#define LAYOUTFILE <ImageEx/ImageEx.lay>
#include <CtrlCore/lay.h>

class ImageEx : public WithImageExLayout<TopWindow> {
public:
 ImageEx();
};

class Second: public TopWindow
{
public:
 Second();
};
#endif
```

```
#include "ImageEx.h"

#define IMAGECLASS FirstImg
#define IMAGEFILE <ImageEx/ImageEx.iml>
#include <Draw/iml.h>

ImageEx::ImageEx()
{
 CtrlLayout(*this, "Window title");
 Icon(FirstImg::MyImage());
}

GUI_APP_MAIN
{
 ImageEx().Run();
}
```

```
#include "ImageEx.h"

#define IMAGECLASS SecondImg
#define IMAGEFILE <ImageEx/ImageEx.iml>
#include <Draw/iml.h>

Second::Second()
{
 Icon(SecondImg::MyImage());
}
```

Subject: Re: multiple classes include-problem
Posted by michael on Tue, 26 Feb 2008 15:14:20 GMT

View Forum Message <> Reply to Message

Hmm... good point. I now use a new IMAGECLASS for every class.

Looks not very nice to me, but it works. Maybe someone has a better solution...

Subject: Re: multiple classes include-problem
Posted by mrjt on Tue, 26 Feb 2008 15:39:53 GMT

View Forum Message <> Reply to Message

Maybe I wasn't clear enough, I believe the correct solution is:

In .h file:

```
#define IMAGECLASS SomeImg
#define IMAGEFILE <SomeImg/SomeImg.iml>
#include <Draw/iml_header.h>
```

In ONE .cpp file:

```
#define IMAGECLASS SomeImg
#define IMAGEFILE <SomeImg/SomeImg.iml>
```

```
#include <Draw/iml_source.h>
```

The way the image stuff works is that the macros create a class with functions taht return the image, so naturally it is best to have the header and source for this class in .h/.cpp files.

Subject: Re: multiple classes include-problem
Posted by bytefield on Tue, 26 Feb 2008 15:54:56 GMT

View Forum Message <> Reply to Message

mrjt wrote on Tue, 26 February 2008 17:39Maybe I wasn't clear enough, I believe the correct solution is:

In .h file:

```
#define IMAGECLASS SomeImg
#define IMAGEFILE <SomeImg/SomeImg.iml>
#include <Draw/iml_header.h>
```

In ONE .cpp file:

```
#define IMAGECLASS SomeImg
#define IMAGEFILE <SomeImg/SomeImg.iml>
#include <Draw/iml_source.h>
```

The way the image stuff works is that the macros create a class with functions taht return the image, so naturally it is best to have the header and source for this class in .h/.cpp files.

Yeah, know that but if we have more sources and still want to use same *.iml file, what we do? I cannot put all my code in 2 files (header and source).

An example:
main.h - contain the generally includes, some global functions declarations, etc.
MyApp.h - MyApp class which implement main windows (it use some images from *.iml file)
MyApp.cpp - MyApp implementation and GUI_APP_MAIN

MyDialog.h - a dialog (complex) or something
MyDialog.cpp - dialog implementation

ConfigDlg.h - ...
ConfigDlg.cpp - ...

So i need images in every class (in every source). Is there another solution or mine is not so dirty?
Or perhaps others use just 2 files (*.h/*.cpp) to write an application?
I don't want 10000 lines of code or more to share the same file. Should i implement the code which use images in a single source then, and other keep separated?

Subject: Re: multiple classes include-problem
Posted by mrjt on Tue, 26 Feb 2008 16:12:05 GMT
View Forum Message <> Reply to Message

It's just like a a normal class declaration.

What would probably work best for you is:
MyApp.h: iml_header.h
MyApp.cpp: iml_source.cpp

The images will be accessible from anywhere that #includes MyApp.h

Subject: Re: multiple classes include-problem
Posted by bytefield on Tue, 26 Feb 2008 16:25:11 GMT
View Forum Message <> Reply to Message

Well, thanks. It solve the problem. I was trying the same thing with header and source, but because in source I wasn't define IMAGECLASS (guessing it is defined once in header) I've got errors.
Now, looking on iml_header.h i understand why it wasn't work, IMAGECLASS get undefined at the end of file.