Exercise 3 COM Fundamentals (Server)

In this exercise you will be creating a COM object from first principles. You will be asked to write most of the object's methods and write the code for the class factory. If you get really stuck take a look at the example solution code.

The code is relatively straightforward, but a good understanding of how to build a COM object from scratch is essential. All COM objects conform to the basic structure defined in this example. Later in the course we will create COM objects using the ***ATL COM Wizard***. The code generated by this wizard will be incomprehensible if you can't follow this example.

You are given a test harness. You need to modify the code in 3 source files:

ClassFactory.cpp	- implements the class factory
Component.cpp	- implements the component (the COM object)
Exports.cpp	- implements the exported functions required by COM

and then enter registry information in the file

Component.reg

Step 1 - Getting Started

Open the workspace file:

COM Programming\Exercises\COM Fundamentals (Server)\COM Fundamentals (Server).dsw

This workspace has 2 projects defined. The ***Test*** project is complete, but feel free to modify it. The ***Radio*** project will house the code for your COM object. You will modify the files in this project.

Step 2 - Implementing the Class Factory

Study the file ***ClassFactory.h***: this contains the class definition of your class factory. Implement the methods of this class in ***ClassFactory.cpp*** using the detailed instructions contained in the file.

Step 3 - Implementing the Component

Study the file ***Radio.h***: this contains the class definition of your object. Implement the methods of this class in ***Radio.cpp*** using the detailed instructions contained in the file.

Step 4 - Implementing the Exported Functions

Study the file ***Exports.def***: this contains the list of functions exported by the DLL. COM will call these functions when the DLL is loaded and unloaded. Implement these functions in ***Exports.cpp*** using the detailed instructions contained in the file.

Step 5 - Building the Radio DLL

You should now be able to build your DLL. Note that the DLL will not be useable until it is registered in the system registry (see below).

Have you got a clean compile? If not, check you have corrected the deliberate mistake at the beginning of ***Interface.h***!

Step 6 - Registering your Component Object

Study the file ***Radio.reg***: this should contain the required registry entries for your COM object. You will need to update this file with the correct GUIDs for your COM object and interface. This registry information will be added to the registry as part of the custom build step of your project. Select ***rebuild all*** - the new entries will be added to the registry. Use ***Regedit*** to check the entries in the registry.

Step 7 - Testing your Component Object

Use the test harness provided to test your component. It is best to single step through the code in the debugger to check everything happens as expected. If you encounter problems, remember that COM will call ***DllGetClassObject()*** initially. If you put a hard-coded breakpoint at the beginning of this function:

```
asm int 3
```

and you reach the breakpoint, you know that the registry information is correct. You can then single step through your code to find the error. If you don't even get to the breakpoint then you must have supplied incorrect information in the registration file (***Radio.reg***).

As an alternative, you can always debug the DLL, but set the debug target to be the test executable. Then you can actually set breakpoints in the DLL without having to resort to hard coded asm statements.