Exercise 5 Building a COM Object using ATL

In this exercise you will be creating a COM object using the ATL object wizard. Your job is to add methods to the object and then test the object in Visual Basic using the test harnesses provided. Before you begin, you may find it helpful to run up the example solution to familiarise yourself with the test harness.

Your COM object will have one interface called ***ISimple*** with four methods defined:

```
Square
Multiply
Average
SquareRoot
```

Step 1 - Getting Started

Create a new project by selecting ***File/New/Projects/ATL COM AppWizard*** and choose a location of:

```
COM Programming\Exercises\Building a COM Object using ATL
```

and a project name of ***Wizard***. The wizard will ask if you want to implement your object as an EXE, DLL or service; select DLL. Press ***finish*** and the wizard will generate a set of source files.

Step 2 - Inspecting the Files Produced by the Wizard

Open up class view for you project. You should see that five global functions have been defined. One of these functions is the optional DLL entry function:

```
DllMain
```

The other 4 functions are the required exported functions from an in process COM object:

```
DllCanUnloadNow
DllGetClassObject
DllRegisterServer
DllUnregisterServer
```

Notice that you also get a global variable called ***_Module***. This is a C++ object that represents the COM libraries and registration object. The exported functions all delegate to ***_Module***.

Note that at this stage you do not have any code produced for your COM object or its class factory.

Step 3 - Using the Object Wizard

To produce code for your COM object and its class factory you should employ the services of the ***ATL Object Wizard***. Proceed as follows. Go to class view and highlight ***Wizard Classes***. Right click the mouse and select ***New ATL Object***. This

will run the object wizard. Choose ***Objects/Simple Object*** and at the next dialog choose a short name of ***Simple.*** Don't change any of the other options; just leave them with their default settings.

Now inspect the project again using class view. Note the changes. You now have a class called ***CSimple*** and an interface ***ISimple***. Spend some time familiarising yourself with the code produced by the wizard.

Build the project. Note that you project will be registered automatically as part of the custom build stage of this process. Inspect the ***Simple.rgs*** file. This file contains all the registration information.

Step 4 - Adding Methods to the ISimple Interface

Now you can add the 4 methods to the ***ISimple*** interface. To add methods select the ***ISimple*** interface and the click the right mouse button. Each time you do this you will be presented with a dialog. Choose ***Add Method*** and then enter the type information for each method in turn:

Method Name:	SquareRoot
Parameters:	[in] double number, [out, retval] double* pResult

Method Name:	Average
Parameters:	[in] double number1, [in] double number2, [out, retval] double* pResult

Method Name:	Multiply
Parameters:	[in] double number1, [in] double number2, [out, retval] double* pResult

Method Name:	Square
Parameters:	[in] double number, [out, retval] double* pResult

Working in ***Simple.cpp***, provide suitable implementations for these methods. Once you have completed the implementations of all of the methods, build again.

Step 5 - Testing in Visual Basic

The test harness has already been written for you. All you have to do is to change the project setting to link in to the test files. So select Project/Settings/Debug and then enter:

Executable:	C:\Program Files\Microsoft Visual Studio\VB98\Vb6.exe
Working directory:	../Visual Basic Files
Program arguments:	Test1.vbp

Press F5 to test out your COM object.