Lab 13.1: Building a COM Object Server

Objectives

After completing this lab, you will be able to:

- Use ATL COM AppWizard to create an out-of-process COM object server executable file.
- Add methods to ATL COM objects.

Prerequisites

Familiarity with the topics covered in this chapter.

Lab Setup

To see a demonstration for the solution to this lab, click this icon.

Estimated time to complete this lab: **25 minutes**.

Exercises

The following exercises provide practice working with the concepts and techniques covered in this chapter.

Exercise 1: Building an Application with the ATL COM AppWizard

In this exercise, you will implement a simple out-of-process (EXE) COM server and add two methods.

Copy the file \C13\Lab01\Baseline\ComServerBody.txt to your lab directory. This file contains the function-body source-code of the two methods you add to your ATL COM server.

If you have not completed Lab 9.1, use the 32-bit ODBC driver manager in Control Panel to add a data source using the Microsoft Access Driver. Name the data source **PERSONAL**. Use the Personnel.mdb database located in \Labs\C09\Lab01.

There is no other setup for this lab. Copy the baseline code for this lab from \Labs\C13\Lab01\Baseline. The completed code for these exercises is in \Labs\C13\Lab01\XXX, where XXX is the exercise number. If you are running on a Windows 95 system, you must install DCOM for Windows 95 from the Visual C++ installation disk before doing this lab.

Exercise 1: Building an Application with ATL COM AppWizard

In this exercise, you will implement a two-method COM Server. Your server exposes methods MDYfromBSTR and ValidateMDY to the client through an interface **Imdy**.

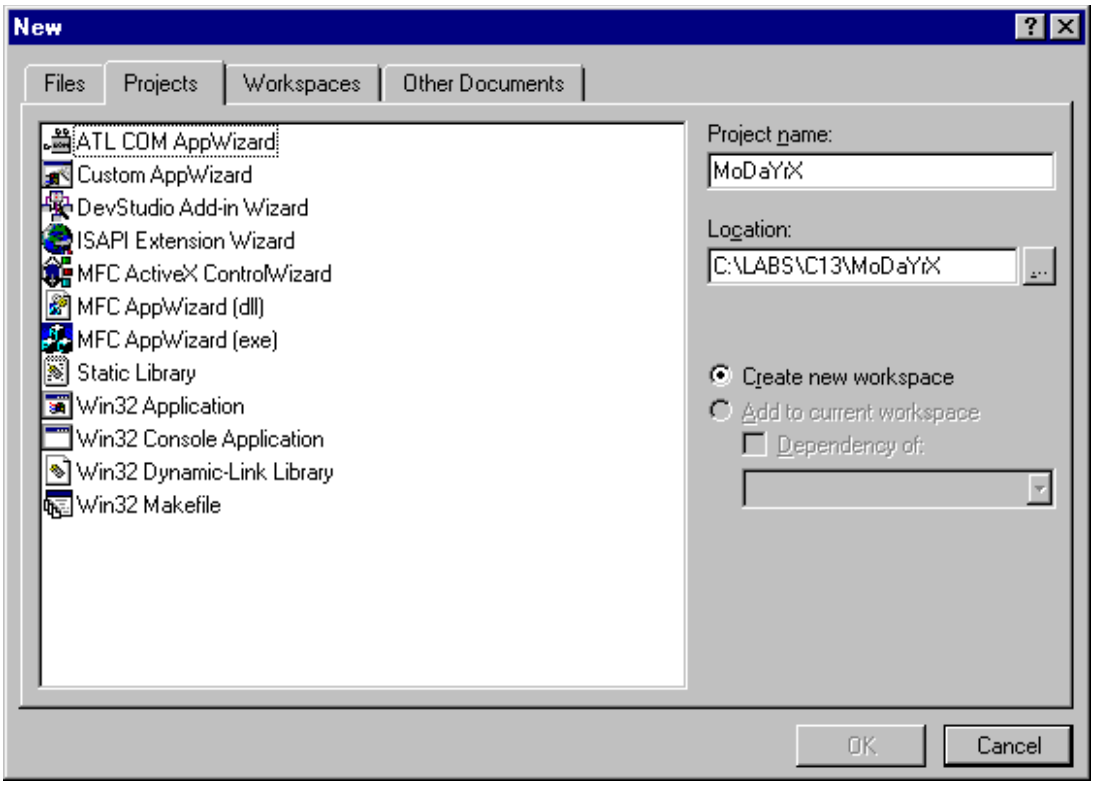
The method MDYfromBSTR receives a BSTR string that represents a date in the form “mm/dd/yy” when it is properly formatted. The function converts a properly formatted date string to three short integers representing month, day, and year. The method returns S_OK or S_FALSE, as appropriate.

The second method, ValidateMDY, receives three short integers from the client that represent the three parts of a date. ValidateMDY tests that the month value falls between one and twelve. Based on a valid month value, the method determines the day value’s validity. The method accounts for leap years. If the day or month value is invalid, ValidateMDY modifies an advisory string provided by the client. The method returns S_OK or S_FALSE, as appropriate.

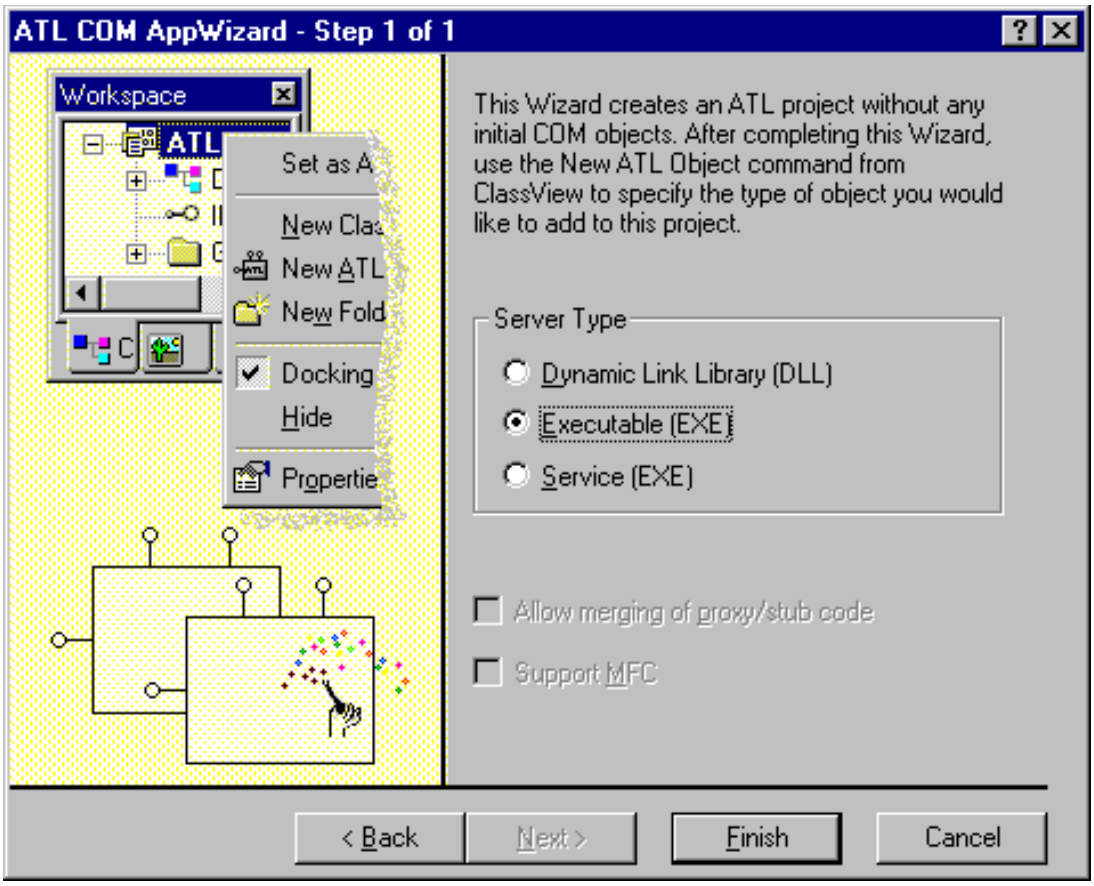
Your completed server has no graphical representation.

➢ **Create a new ATL COM AppWizard application**

1. From the File menu, choose New.
2. Select the Projects tab; choose ATL COM AppWizard, and type MoDaYrX in the Project Name box. Click OK.

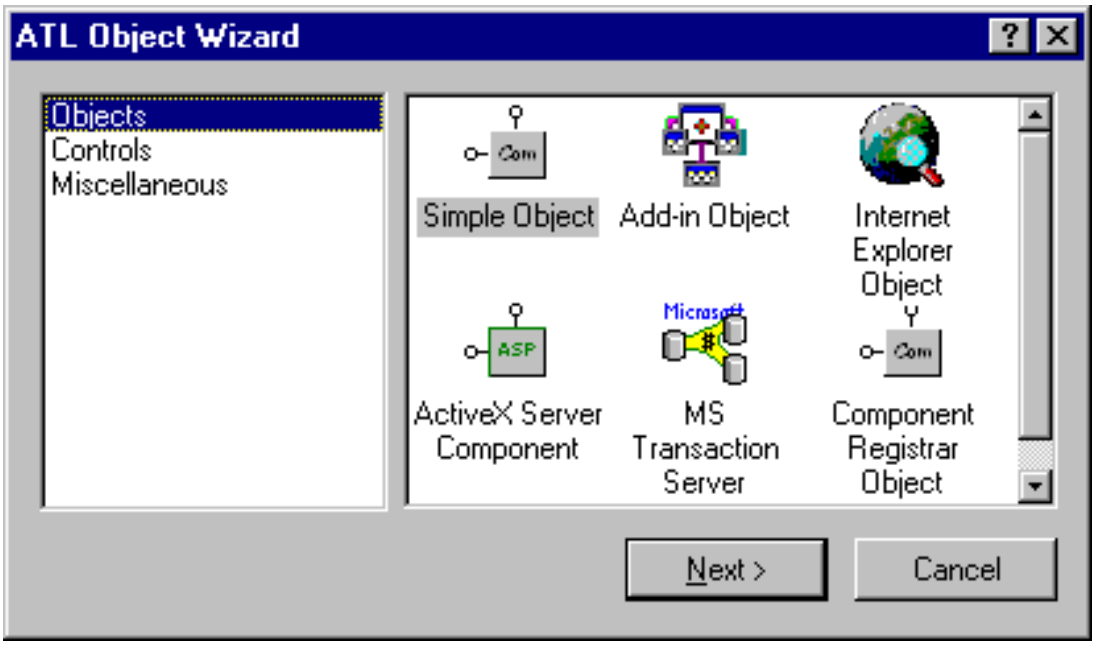


3. In Step 1 of 1, set the Server type to Executable (EXE). Click Finish. Take a moment to review the specifications, then click OK.



➢ **Create a new ATL object**

1. In the ClassView window, right-click the icon to the left of “MoDaYrX classes.” Choose New ATL Object from the context menu.
2. In the left pane of the ATL Object Wizard dialog, select Objects. In the right pane, select Simple Object. Click Next.



3. Select the Names tab of the ATL Object Wizard Properties dialog. Note that the C++ Short Name edit box has a raised edge. Enter your class “short name” in this box. A short name does not have the C prefix. The short name of your class is **mdy**. The wizard fills in the rest of the boxes. In the ClassView pane of the workspace window, you can see the class, **Cmdy**, and the interface, **Imdy.**

4. Click OK to create object.

➢ **Add the methods to Cmdy class**

1. In the left pane of the Workspace View, right-click either of the two **Imdy** interface glyphs. Select Add Method. In the Add Method to Interface dialog box, enter the first method’s name, **MDYfromBSTR**. The parameters are:

   BSTR bstrDate,

   short * pnM,

   short * pnD,

   short * pnY

2. Repeat Step 1 to add a method **ValidateMDY**. The method's parameters are:

   BSTR * pbstrMessage,

   short nM,

   short nD,

   short nY

➢ **Program the methods**

1. In the workspace’s ClassView, double-click the method icon **MDYfromBSTR** (under class Cmdy). This opens the code segment for the implementation header. Remove the statement “return S_OK,” from the implementation. Open the file, ComServerBody.Txt. (You copied this file to the local directory as part of the lab setup.) Copy the implementation code for **MDYfromBSTR** from this file. Paste the code into your method implementation.
2. Repeat the above process to implement method **ValidateMDY**.
3. At the top of the Mdy.Cpp file, include Stdio.H, which is needed for the **sscanf** function, and ComDef.H, which is needed for the _bstr_t type as follows:

```
#include <stdio.h>
#include "ComDef.h"
```

➢ **Modify project settings**

1. Because this project uses a C-run-time function, **sscanf**, remove the symbol _ATL_MIN_CRT from the project's Project Settings dialog box, C/C++ tab, Preprocessor definitions. Do this for all the release builds. _ATL_MIN_CRT is not defined for the debug builds.

➢ **Build your ATL COM Server**

1. Save your work.
2. Build the project.

The completed code for this exercise is in \Labs\C13\Lab01\Ex01.