

Office XP PIA from WinForms

You can use, extend and automate the hundreds of objects found in OfficeXP in your Windows forms projects



`system\cdrom\dotnet\oxpia.exe`

Writing a new .NET Windows Forms application? Need a spell-checker or a formula-driven grid control? You must have surely thought about MS Word and Excel now. Won't it be great if you can reuse the huge functionalities found in MS Office and also control them from your .NET application? That's exactly what Office XP Automation and Office XP PIA allows you to do.

Benefits

Using Office built-in objects has benefits for everyone. As a developer you don't need to buy any expensive third-party controls. Your end users are already familiar with Office Interface and there is no relearning for them.

Office COM interfaces

MS Office for years has provided hundreds of reusable COM (Microsoft Common Object Model) APIs. Developers used these objects to extend or automate Office in countless ways. But how do you use these COM objects in your .NET projects? Clearly, .NET compilers don't understand COM type libraries (like the common TLB files, found in VB 6.0). In fact, they only understand .NET metadata. Therefore, to make a COM component available in the .NET world, we need to create "metadata"

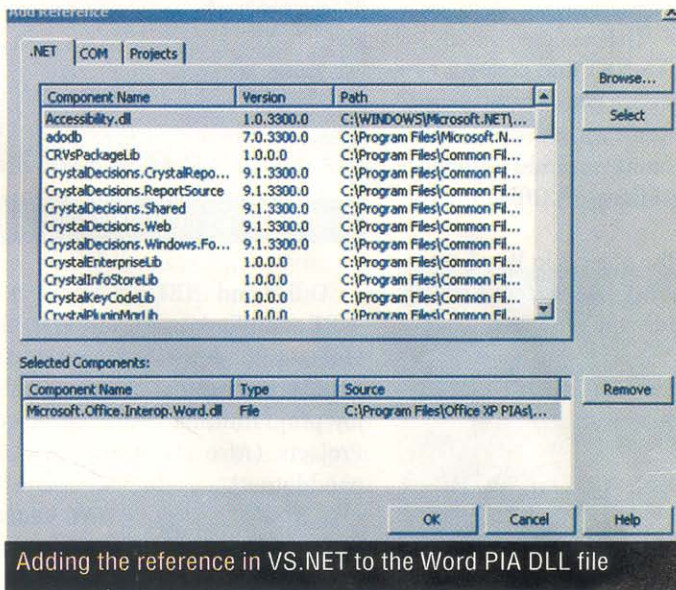
information. Generating metadata manually for the all Office XP objects is going to be a daunting task.

Interop assemblies

There are two ways to do this.

- The .NET Framework SDK provides a tool called the type library importer (TLBIMP.EXE). This utility accepts as input a COM type library file and outputs a .DLL file containing .NET Metadata.
- You use the Add Reference option to reference the COM component from Visual Studio .NET.

Regardless of which method you use, they each produce a new assembly with a .DLL extension. This assembly is used by your .NET application to automate Office. It is important to note that the interop assembly only contains definitions of COM types and does no modification on the actual binary code of Office. In other words, an interop assembly contains the needed metadata information that allows .NET-managed code to call unmanaged code in Microsoft Office. It does this by wrapping the original Microsoft Office COM-based object model APIs.

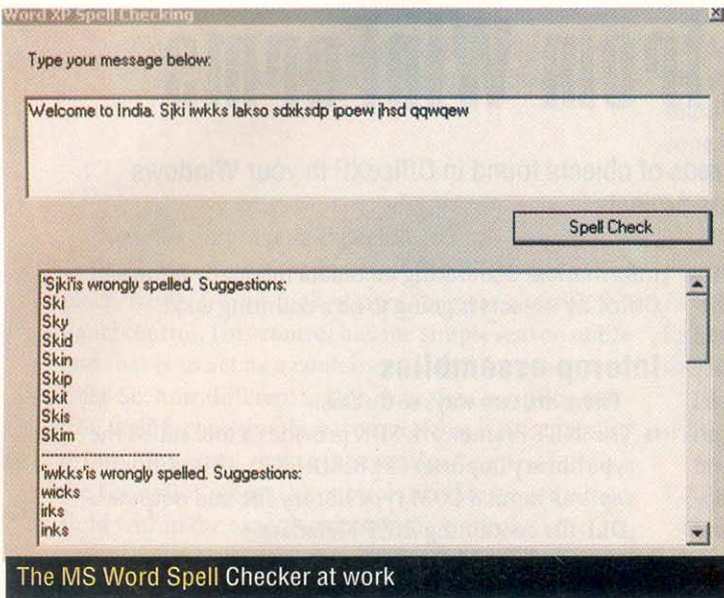


Adding the reference in VS.NET to the Word PIA DLL file

Primary Interop Assembly

All of this sounds like a lot of work, that's why Microsoft recently released Office XP PIA (Primary Interop Assembly). The Office XP PIA contain the official description (read as .NET Metadata) for type library files of all Office applications including Access, Excel, PowerPoint, Word and so on.

Is there a difference between interop assembly that you create with



The MS Word Spell Checker at work

TLBIMP and that of PIA? For most part, they are almost identical except that the PIA is the official interop assembly released by Microsoft and has been customized for ease of use from managed code.

Install Office XP PIA

You need to download the Office XP PIA from Microsoft MSDN website (also provided in the accompanying CD). After downloading, copy it to a folder and run the register.bat file.

Spell checker in your app

Let us plunge right into VS.NET and create a new Windows application that uses MS Words' built-in Spell Checker. Once you have your project, right-click on References (below your project name) and add a reference to the Word PIA DLL (Microsoft.Office.Interop.Word.dll).

Now you can use any of the objects in Word as you would use a built-in .NET object like System.Console, etc. You can see this in the code snippet below.

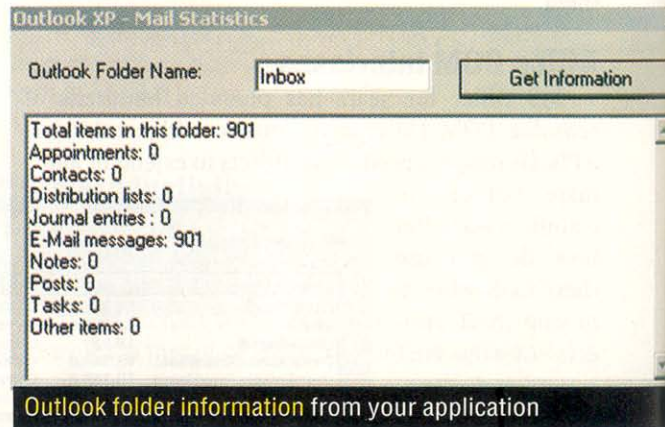
```
Dim oMSWord As New Microsoft.Office.Interop.Word.Application-
Class()
Dim oSuggestionsCollection As Microsoft.Office.Interop.Word.Spelli-
ngSuggestions
Dim oSuggestion As Microsoft.Of-
```

```
...
    fice.Interop.Word.SpellingSug-
    gestion
    ...
    oMSWord.Documents.Add()
    ...
    oSuggestionsCollection =
    oMSWord.GetSpellingSugges-
    tions(sWord)
    ...
```

This is all that it is needed to use the complete Word's Spell Checker from your application.

Excel, PowerPoint and Outlook

What we have seen with Word PIA is only tip of the iceberg. As the full source for this VS.NET solution (found in the accompanying CD) will show, there is a lot more that you can do with Office XP PIA. The Excel sample shows how you can create and populate a sheet from your Application. You will also see how to automate the creation of a slidedeck and run the slide show with PowerPoint PIA. Similarly, the Outlook PIA sample shows you how to get the statistics of your Outlook folders.



Outlook folder information from your application

Office and .NET are a perfect combination. While .NET and Windows Forms offers the best development tools for Windows Development, Office XP is the most popular end-user application in the world. Go and enjoy programming Office XP objects from your .NET Projects. (Also check out www.pcquest.com/officex-panddotnet).

TNC Venkata Rangan is CMD,
Vishwak Solutions and Regional Director,
MSDN