

© 2002 Visual Studio Magazine

Fawcette Technical Publications

Issue	October 2002
Section	Columns
Main file name	lowyt1.rtf
Listing file name	lowyl1.doc
Sidebar file name	
Table file name	
Screen capture file names	lowyf1.bmp, lowyf2.bmp
Infographic/illustration file names	
Photos or book scans	
Special instructions for Art dept.	
Editor	SP
Spellchecked (set Language to English U.S.)	
PM review	
Character count	
Package length	3 pages
ToC blurb	

Overline:

Q & A

Byline:

by Juval Löwy

Technology Toolbox:

[[Art: Check box for those highlighted below.]]

VB.NET

C#

SQL Server 2000

ASP.NET

XML

VB6

Q.

Generate a Skeletal Interface Implementation

In ATL, I could simply point at an interface, and have Visual C++ 6.0 implement that interface for me, by generating a skeletal interface implementation. Is there a similar wizard in Visual Studio .NET? I could not find one.

A.

You can use VS.NET to generate a skeletal implementation of an interface on your class, except this process isn't in the open as it was with Visual C++ 6.0. In addition, this service is available to both C# and Visual Basic .NET.

First, you need to add the interface you want implemented to the class derivation chain. Next, go to the Class View window. Expand the Classes tree item, and find your class. Expand the class's Bases and Interfaces tree item, go to the Interfaces tree item, and find the interface you wish to implement (the one you just added). Right-click on the interface, and select Add | Implement Interface from the pop-up context menu (see Figure 1). VS.NET then creates a skeletal implementation of the interface on your class, and scopes it with a collapsible #region directive (see Listing 1 for an interface definition and the code VS.NET generated). The skeleton returns default values from methods that have a return value, so you can compile the skeleton.

Q.

Link Labels to a Web Site

I would like to put a link on my Windows Forms form, that users can click on and go to a Web site. I was happy to discover that the Windows Forms toolbox has a LinkLabel control, which looks like what I need. I tried to use it, but how do I launch the Internet browser?

A.

All the LinkLabel control gives you is the look and feel of a link: The label looks like an HTML link according to your system configuration, usually underlined with a hand pointer when you hover over it. You still need to launch the browser yourself, and change the label color to reflect that it was clicked. For example, to add a link to a Windows Forms form that points to the *VSM* Web site, drop a link label control on your form. Set its Text property to “Visual Studio Magazine”—only the text is displayed, not the URL itself (see Figure 2). Provide a method to handle the link label click event, using the Events Properties tab, and name it OnLinkClicked. The Wizard generates the correct signature for you automatically:

```
private void OnLinkClicked(object sender,  
    LinkLabelLinkClickedEventArgs e)  
{}
```

Launch the browser by using the Process class’s static Start() method, defined in the System.Diagnostics namespace. By passing in a URL (such as “<http://www.visualstudiomagazine.com/>”), Start() launches the default browser, and has it navigate to the URL. You also need to set the LinkLabel control’s LinkVisited property to true, to have it appear as a visited link. The OnLinkClicked method should now look like this:

```
private void OnLinkClicked(object sender,  
    LinkLabelLinkClickedEventArgs e)  
{  
    m_LinkLabel.LinkVisited = true;  
    Process.Start  
        ("http://www.visualstudiomagazine.com/");  
}
```

Once I got the link label working, I wanted to take the mechanism one step further. Often, you have an image that behaves like a link and launches the browser when you click on it. You can achieve a link-image by dropping a picture box control on the form (such as the *VSM* logo). Using the image Properties window, set the picture box’s Cursor property to a pointing hand (see Figure 2). Provide an event handling method for the Click event, and redirect it to the OnLinkClicked method:

```
private void OnLogoClicked(object sender,EventArgs e)
{
    OnLinkClicked(sender,null);
}
```

Now, the user can click on either the link label or the image, and launch a browser automatically to display the Web site.

About the Author:

Juval Löwy is a software architect and the principal of IDesign, a consulting and training company focused on .NET design and .NET migration. Juval is currently working on a comprehensive .NET component-programming book. He speaks at software development conferences. Juval is a Microsoft Regional Director and he chairs the .NET California Bay Area User Group's program committee. Contact him at www.idesign.net

Captions:

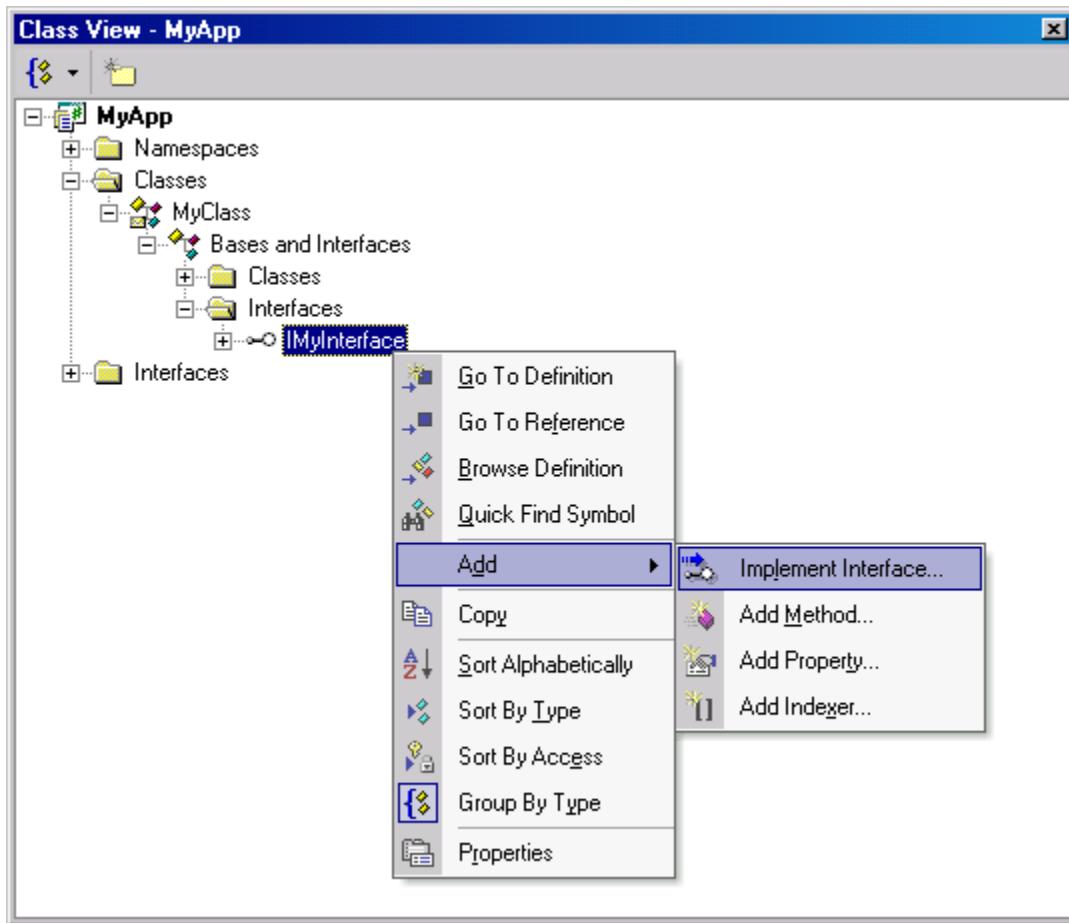


Figure 1

Use VS.NET to Provide a Skeletal Interface Implementation. In the Class View window, find your class and the interfaces it derives from. The Interface context-menu lets you add skeletal implementation of the interface.



Figure 2

Point to a Web Site. The Links and Clicks demo uses the out-of-the-box LinkLabel control. You can also have an image or a picture box behave like a link with a few lines of code.

C# Implement Interfaces Automatically

Listing 1 VS.NET generates skeletal implementation, enclosed in a code region. The wizard-generated code compile, because the skeleton returns default values from the methods. It even provides a comment, visible as a tool tip when hovering over the collapsed region. [[Please add another couple descriptive sentences to this caption, talking about what's going on in the code. Thanks!]]

```
public interface IMyInterface
{
    void Method1();
    int Method2(int num);
    string Method3();
}

//Visual Studio.NET will generate this skeletal
//implementation:

public class MyClass : IMyInterface
{
    #region Implementation of IMyInterface
    public void Method1()
    {
    }

    public int Method2(int num)
    {
        return 0;
    }

    public string Method3()
    {
        return null;
    }
    #endregion
}
```