

Scanning Enabler™

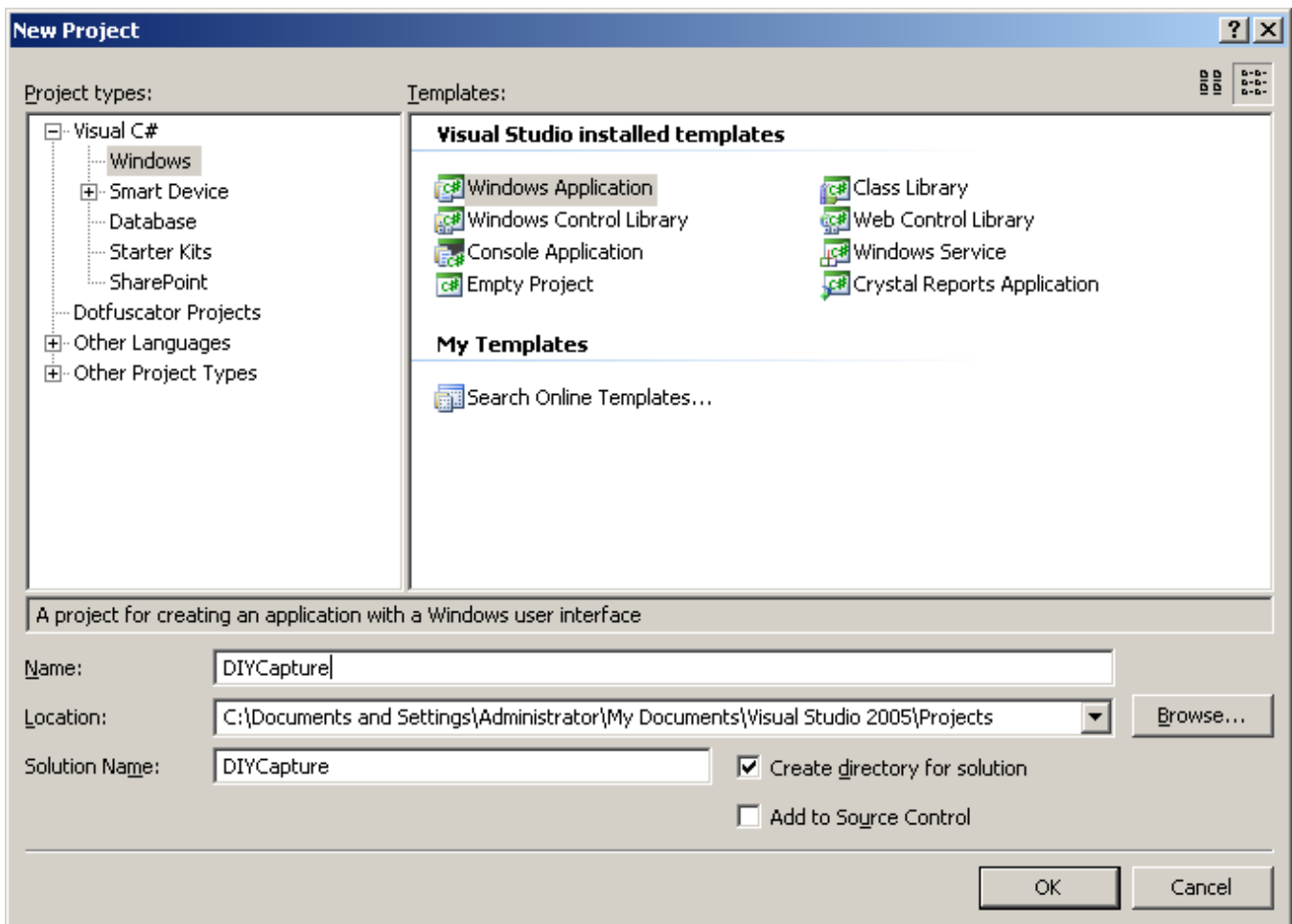
Software Development Kit

DIY Capture Application

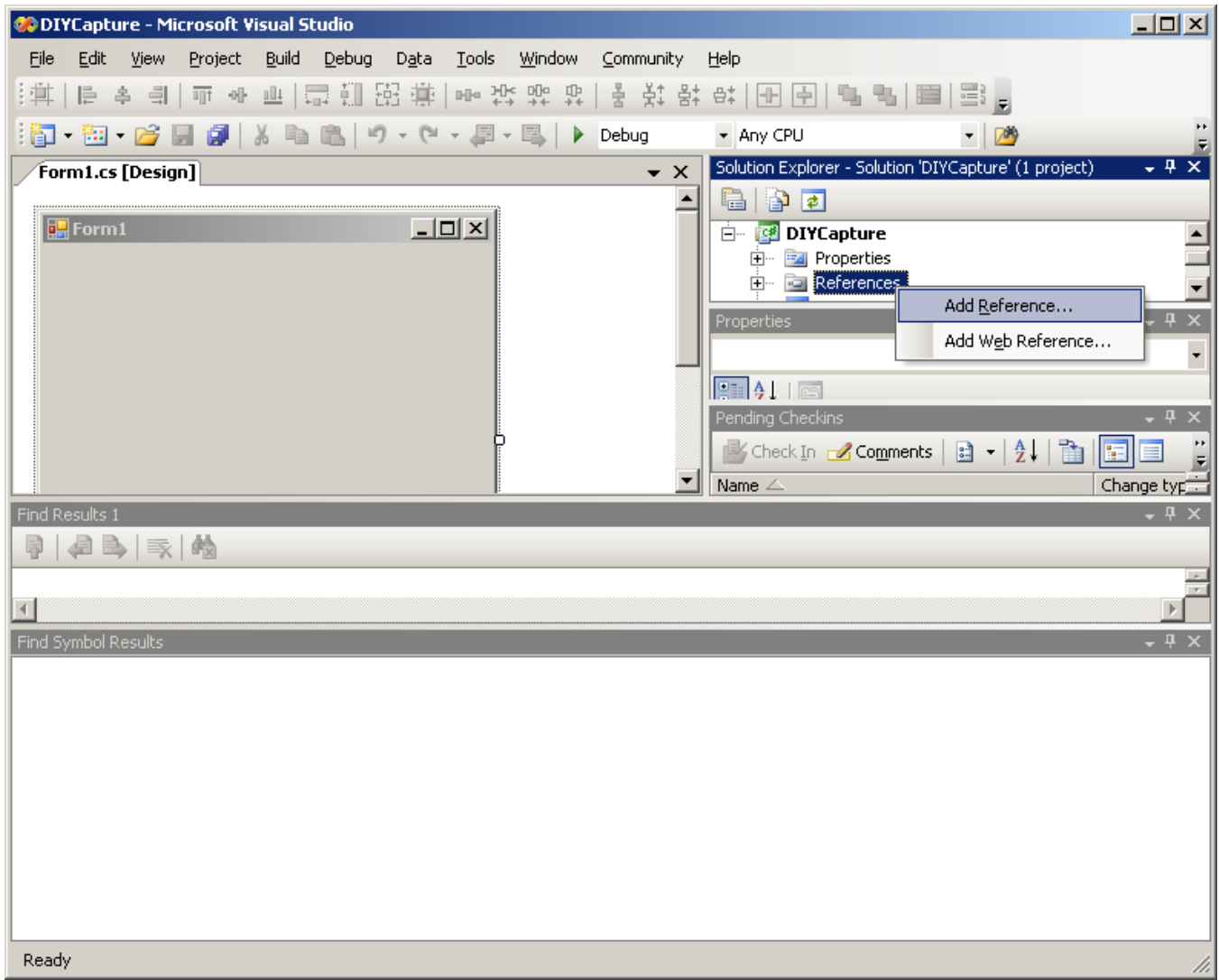
DIY Capture Application

This Guide demonstrates how the Dark Blue Duck Scanning Enabler components can be used to create a custom paper document Capture application. The key components used are the Scanning Control and the Dark Blue Duck Scanning Enabler Web Services.

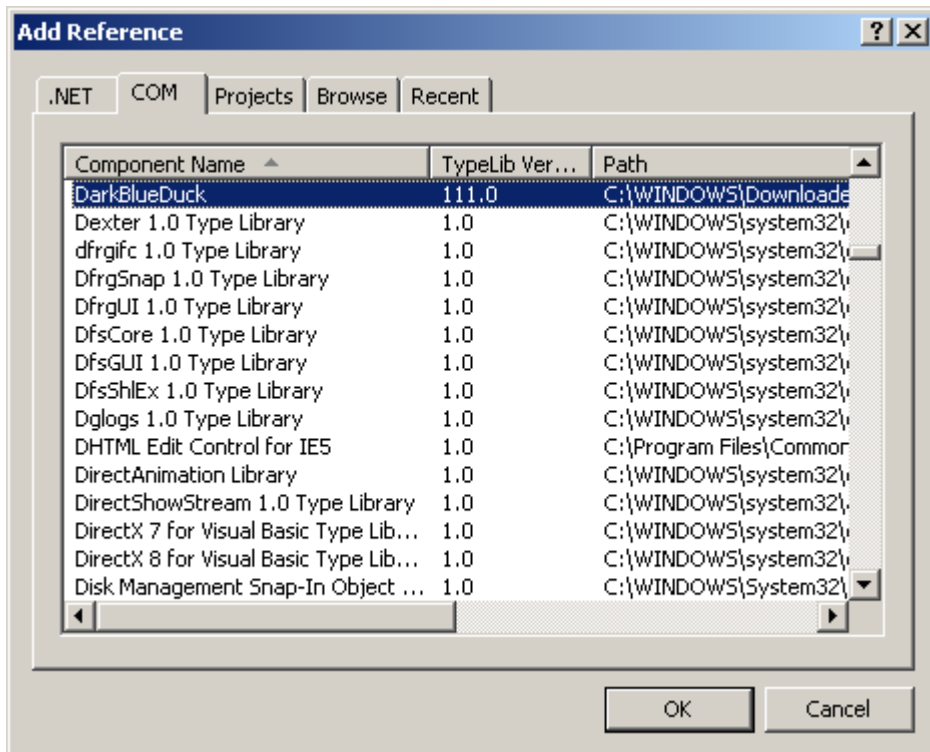
Start by creating a Windows Application using Microsoft Visual Studio 2005.



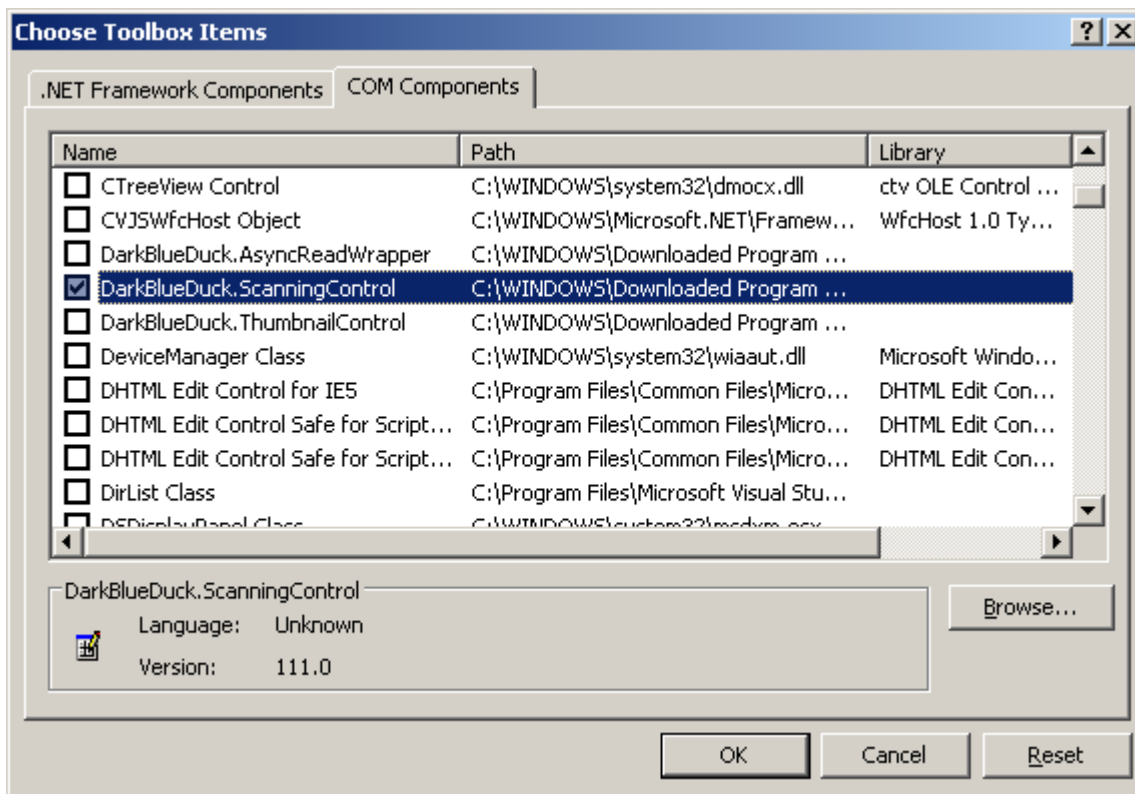
Add a reference to the Scanning Control.



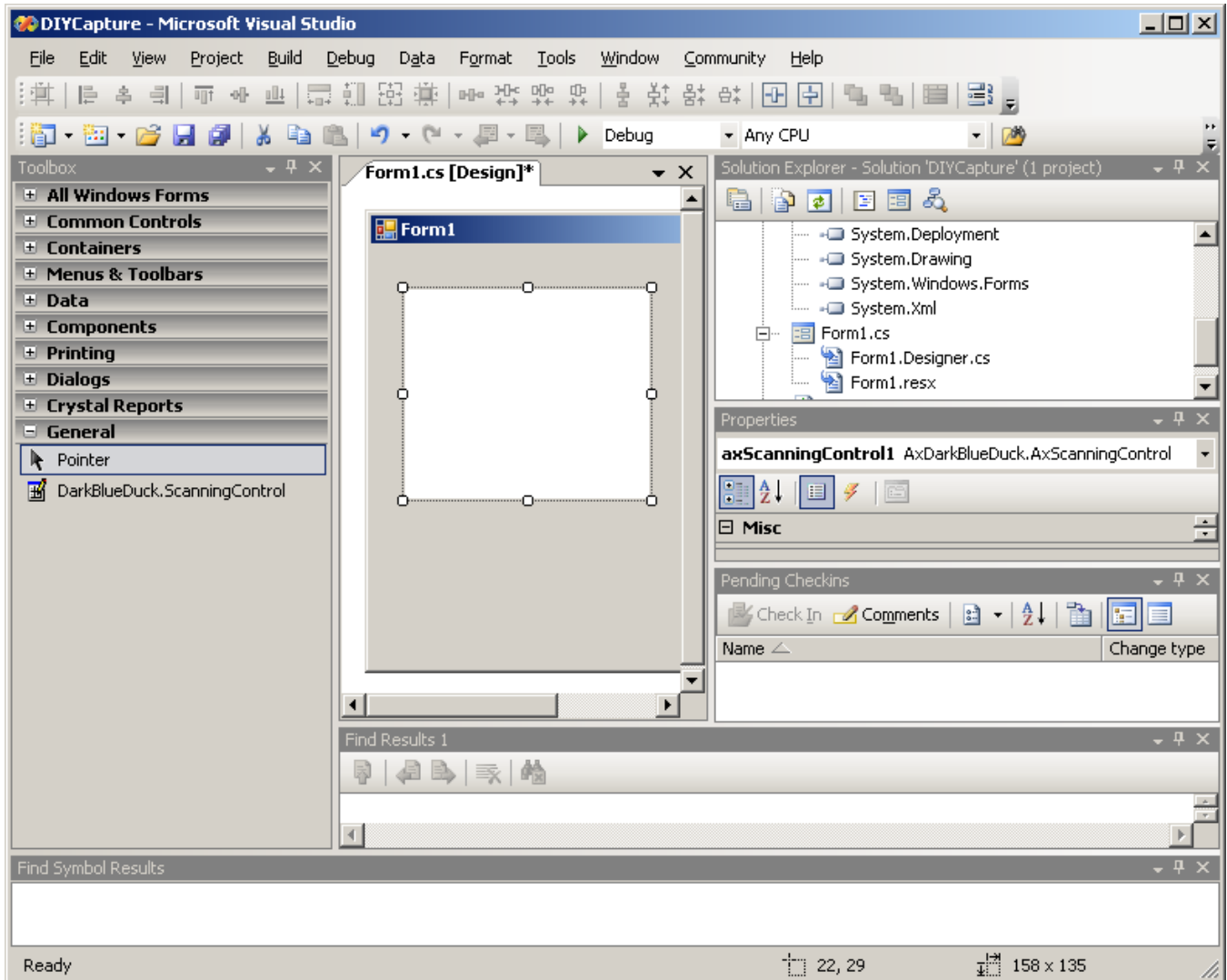
Use the "COM" tab to select the DarkBlueDuck COM Component



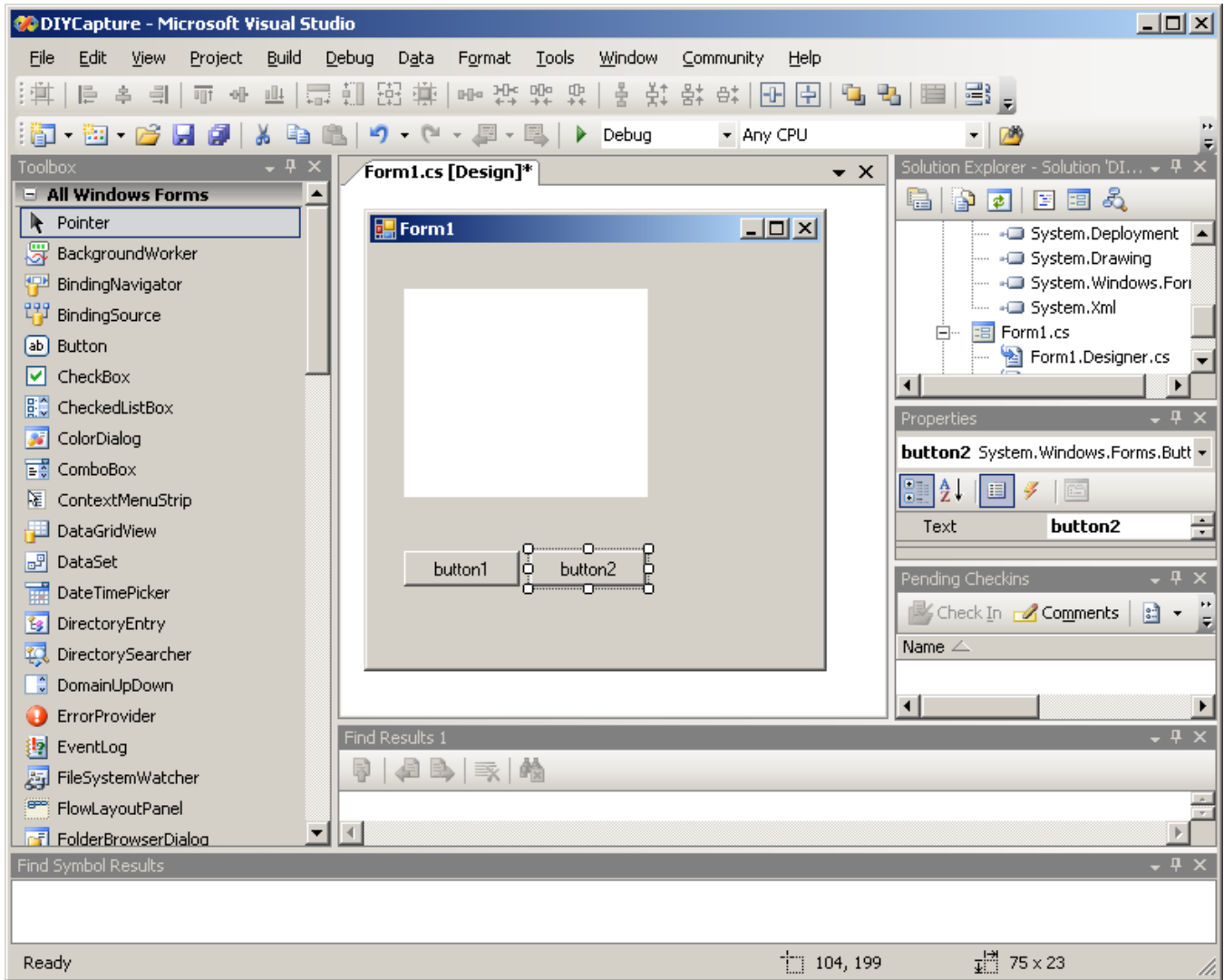
Add the Scanning Control to the Visual Studio toolbar using the “Choose items...” menu item.



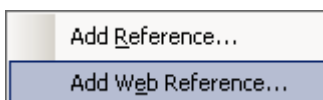
Drag and drop the Scanning Control from the Toolbox to the Windows Form. Then resize the control.



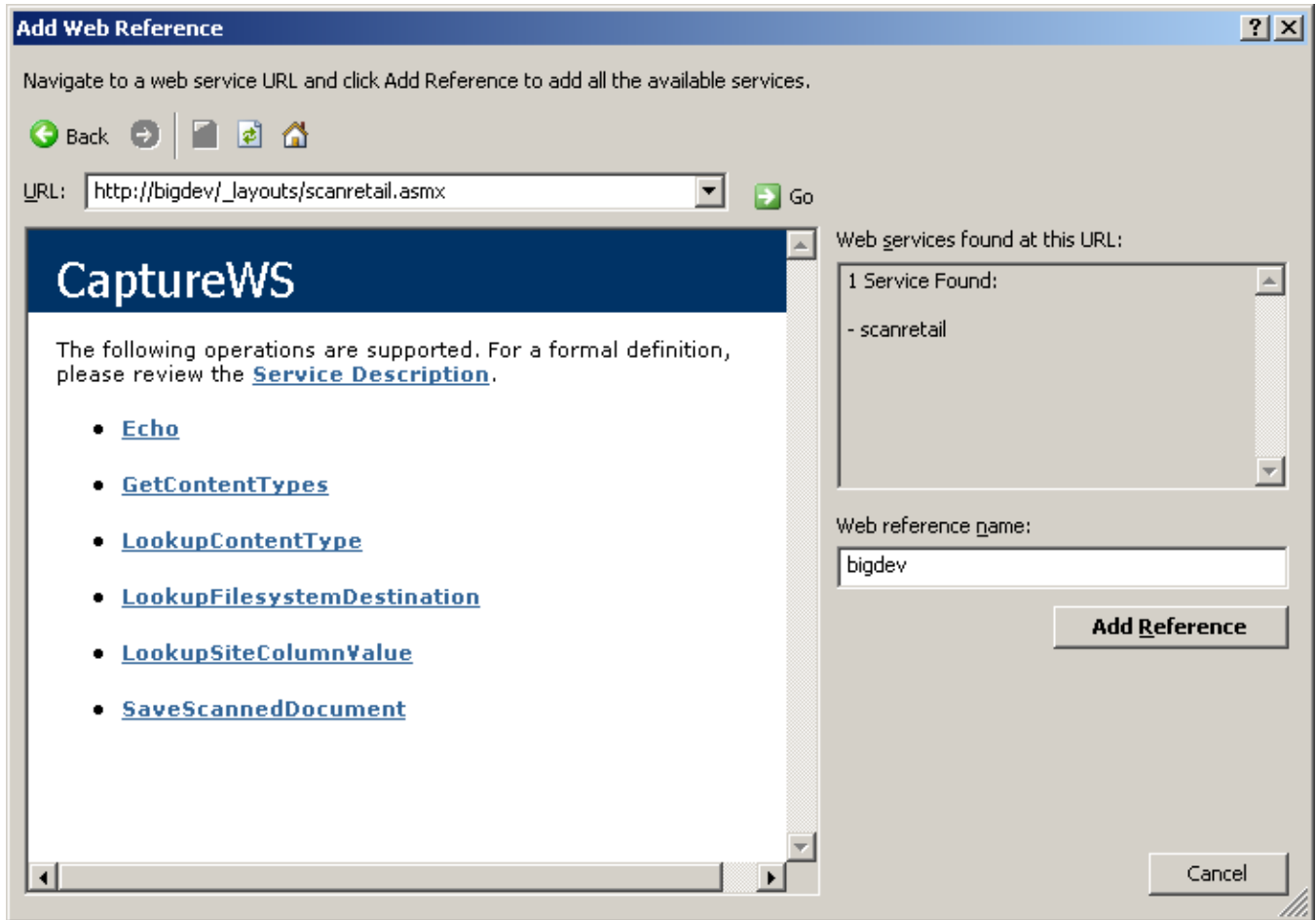
Add two buttons. One button will be used to initiate scanning. The second button will initiate the uploading of the scanned pages to your SharePoint Server.



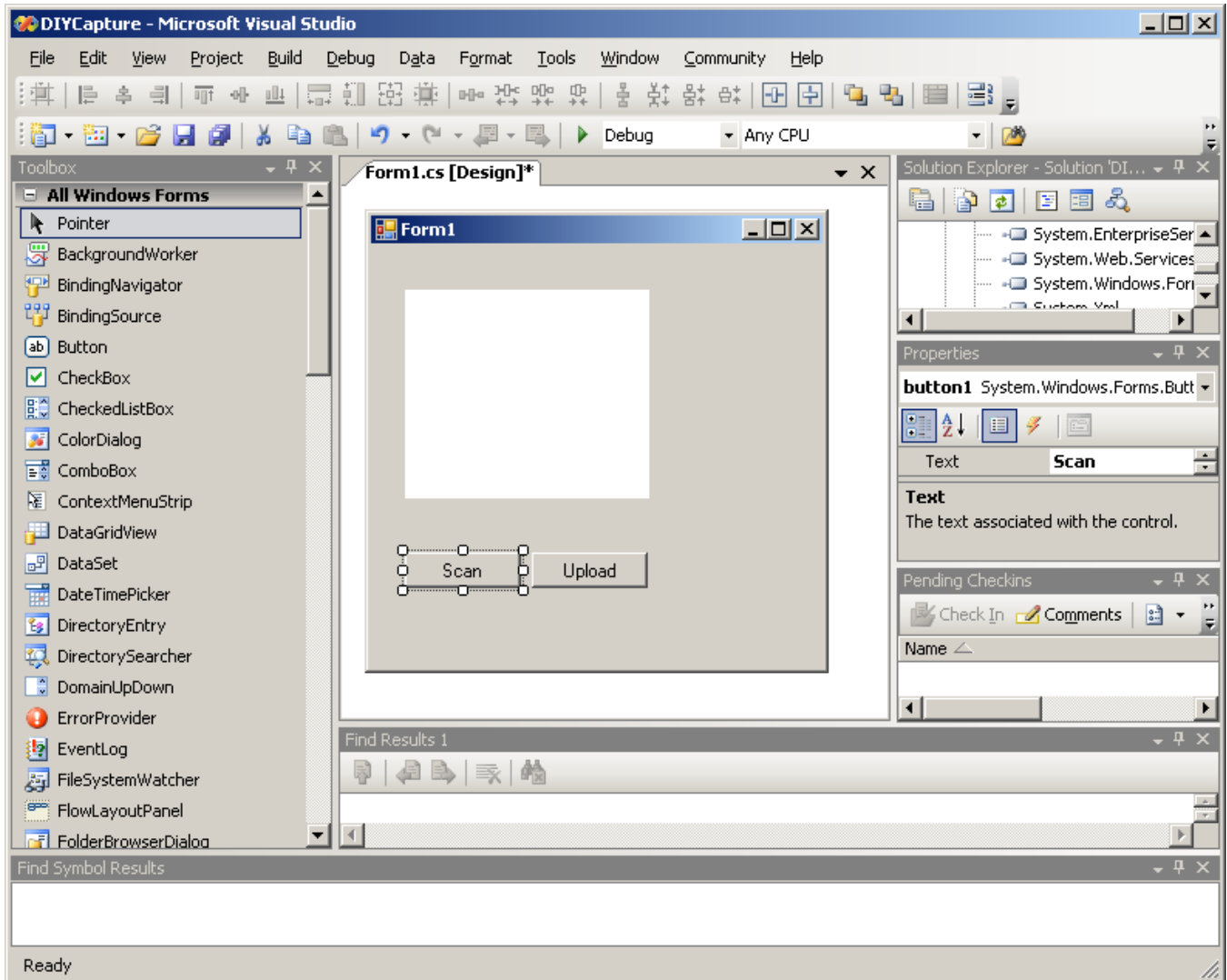
Next add a “web reference” to the Scanning Enabler web services installed on your SharePoint Server.



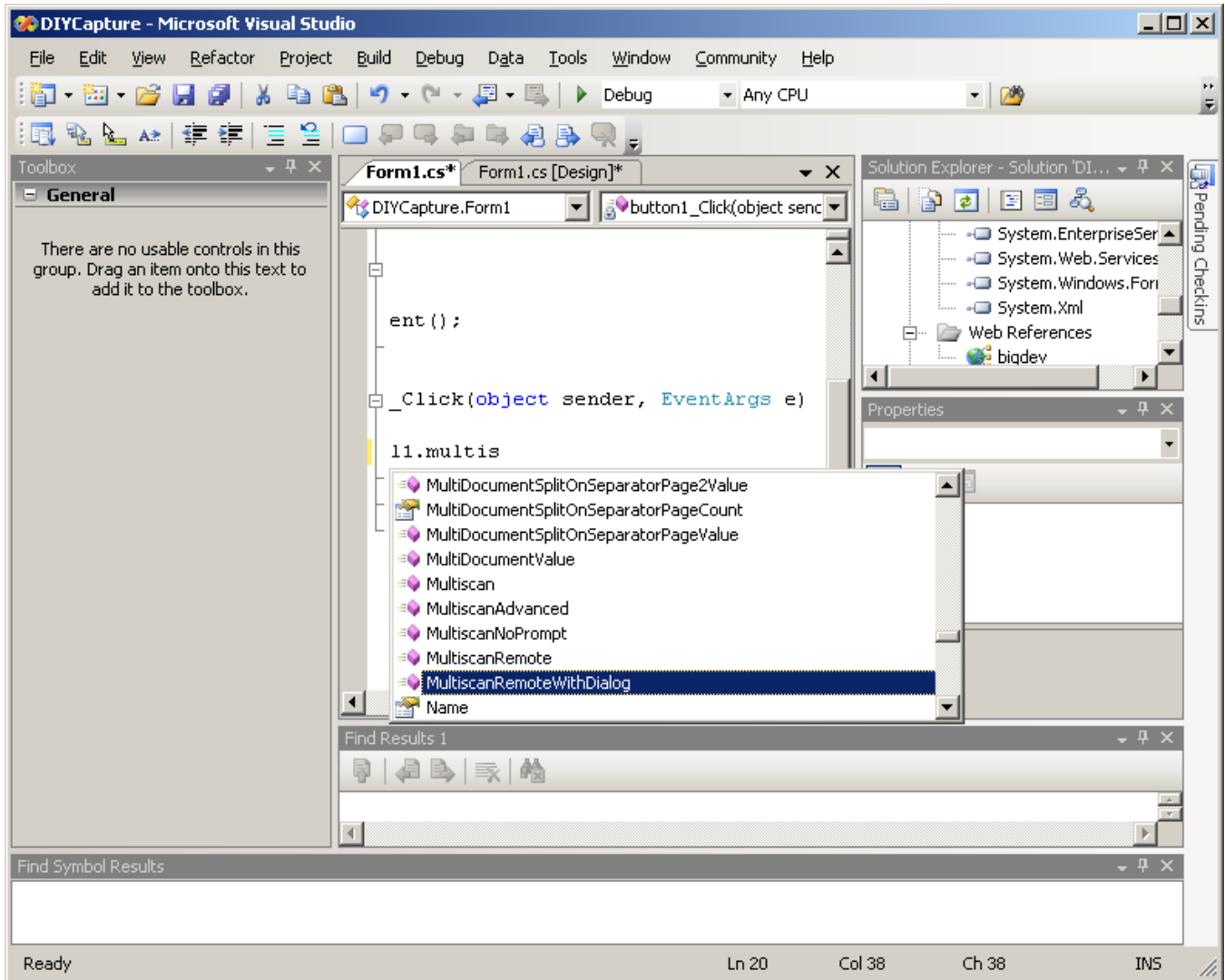
You will need to navigate to http://<your server name>/_layouts/scanretail.aspx



Update the "text" property of the two buttons.



Add the Scan button code.



Note that Visual Studio Intellisense will provide assistance as you type.

This code calls the `MultiscanRemoteWithDialog` method of the Scanning Control. This code will connect to an instance of the Scanner Sharing service running on a network connected machine.

```
private void button1_Click(object sender, EventArgs e)
{
    bool scanMorePagesPrompt = false;
    bool scanBothSides = true;
    short scanningArguments = -1;
    bool useDocumentFeeder = true;

    axScanningControl1.MultiscanRemoteWithDialog(ref scanMorePagesPrompt, ref scanBothSides, ref
scanningArguments, ref scanningArguments, ref scanningArguments, ref useDocumentFeeder);
}
```

To connect to a local Twain Scanner you would use the `MultiscanAdvanced` method.

Add the Upload button code.

Here we set the required document format

```
axScanningControl1.format = "pdf";
```

Use the ActiveX control to return a base64Encoded string representing the scanned document.

```
string base64Encoded = axScanningControl1.SingleDocument2Value.Replace("\0", "");
```

Upload the document to the SharePoint server

```
proxy.SaveScannedDocument(base64Encoded, "test.pdf", "pdf", "0x010104", properties, "http://bigdev", "Documents", "");
```

And reset the ActiveX control by deleting each of the Scanned Pages stored within it

```
private void button2_Click(object sender, EventArgs e)
{
    axScanningControl1.format = "pdf";
    string base64Encoded = axScanningControl1.SingleDocument2Value.Replace("\0", "");

    bigdev.CaptureWS proxy = new DIYCapture.bigdev.CaptureWS();
    proxy.Credentials = System.Net.CredentialCache.DefaultCredentials;

    //System.Net.NetworkCredential credentials = new System.Net.NetworkCredential("<username>", "<password>",
"<domain>");
    //proxy.Credentials = credentials;

    DIYCapture.bigdev.DBDNameValuePair[] properties = new DIYCapture.bigdev.DBDNameValuePair[0];
    proxy.SaveScannedDocument(base64Encoded, "test.pdf", "pdf", "0x010104", properties, "http://bigdev",
"Documents", "");

    while(axScanningControl1.pageCount>0)
        axScanningControl1.DeletePage();
}
```