

WebServices with .Net and ASP.Net Web Matrix

Making Web-based Web Services Easy

Developed by: **transentia**

Unit 1, 40 Skew St.
Sherwood
Queensland 4075
Australia

Mobile: 0416 078 813
Phone/FAX: (07) 3278 5890
Email: contact@transentia.com.au
WWW: www.transentia.com.au

ACN: 091-142-880
ABN: 680-911-428-80

WebServices with .Net and ASP.Net Web Matrix

Making Web-based Web Services Easy

Introduction

To lower the barrier to entry into the .Net world, Microsoft makes a simple IDE available free of charge at <http://www.asp.net>. According to Microsoft: *"Microsoft ASP.NET Web Matrix Project is a free, easy to use, light-weight community-supported web development tool for quickly building ASP.NET Web applications....Web Matrix wasn't built by a formal team at Microsoft - but rather by a group of people across the ASP.NET team who worked on it in their spare time (mainly nights and weekends)."*

While not as capable as the full Visual Studio.NET product, you can still use it to do useful work, as this session will show.

Simple ASP.Net Server-Side Application

In this session you will use Microsoft's free ASP.Net Web Matrix development tool to create a C# ASP.Net web application that serves data from a database to standard web and WebServices clients.

In this session you will need:

- Microsoft's .Net Framework
- Microsoft's Web Matrix IDE
- Microsoft's SQL Server 2000 Desktop Engine (MSDE)

These will all need to be installed prior to undertaking this session. Your instructor will give you the necessary guidance regarding how this can be done.

Create and Populate a Project Home Directory

You should create a directory solely for this session's activities.

Open a new Command Prompt window and issue the following command to copy the directory `Z:\Exercises\7 Webmatrix\Webmatrix` to `C:\Webmatrix`:

```
C:\> xcopy /ieq "Z:\Exercises\7 Webmatrix\Webmatrix"  
C:\Webmatrix  
C:\> cd /d C:\Webmatrix
```

Notes:

- This should be typed as a single line
- Do not retype the prompt (shown in grey); it is shown for your guidance only
- This assumes that your CD-ROM device is Z:, you should use your real device letter instead
- Do not simply drag&drop the folder from the CD-ROM; if you do, you will end up with a read-only directory structure and this will complicate things later on

Create a Database

The first step is to create a simple database.

This can be done fairly simply using one of the tools supplied with Microsoft's MSDE database.

Issue the following command into your existing Command Prompt window:

```
C:\Webmatrix> C:\Program Files\Microsoft SQL Server\
80\Tools\Binn\OSQL.EXE -U sa -P sa -i .\webmatrix.sql
```

Note:

- This command should be typed as a single line

Create and Exercise an ASP.Net WebService with the Web Matrix IDE

WebMatrix has some built-in tools to make it easy to develop simple two-tier database-driven Web Services. You will look at these in this section.

Start the Web Matrix IDE (look for the appropriate entry on the Start menu of your computer).

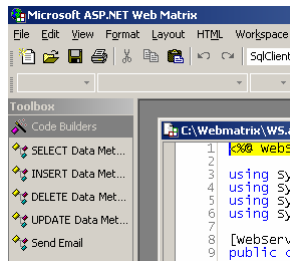
Use Web Matrix to look for and open the C# ASP.Net file **C:\Webmatrix\WS.asmx**. Portions of this file have been created for you, but you still have to finish up the task of creating the Web Service.

Look for the comment:

```
/*
 * TODO: add a FindArtistByInstrument WebMethod here
 */
```

You may delete this comment...it exists simply to show you what part of the file needs to be worked upon.

On the left-hand side of the Web Matrix screen you should see a "Code Builders" task bar:



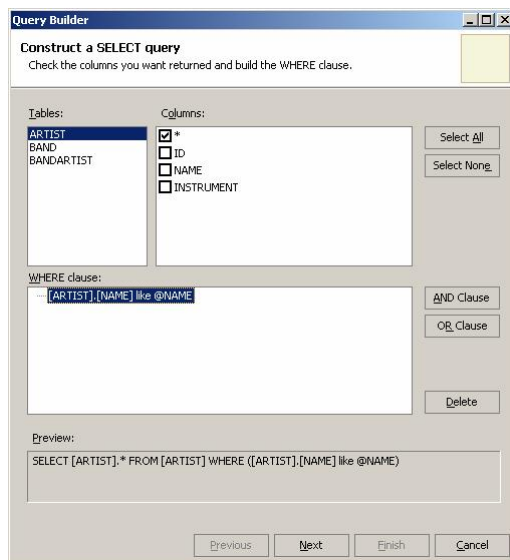
Double-click the **SELECT Data Met...** entry. The “Connect to Database” Dialog will appear and should be filled in as shown here:



Note:

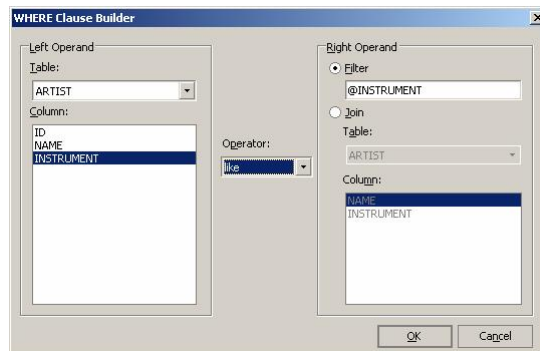
- For your server name, use the *actual hostname of your computer* (not 'phoenix'). The name 'localhost' does *not* work.

You will then be presented with the following “Query Builder” dialog, which should be completed so that it resembles the dialog shown here:

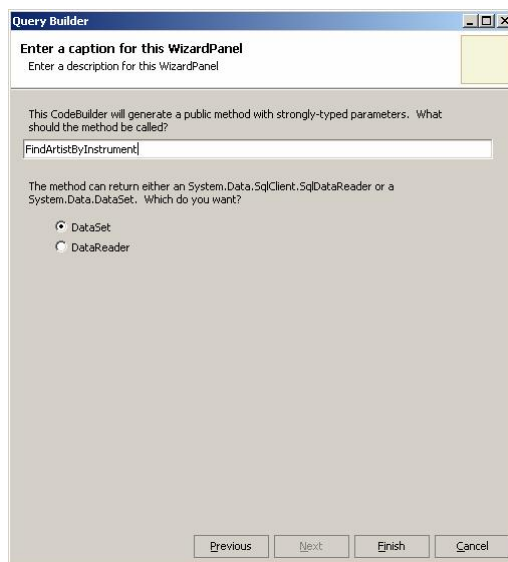


Notes:

- You will need to build the appropriate WHERE clause for the query by using the dialog that will appear when the button labelled **WHERE** is clicked (once a query has been constructed, the WHERE button is replaced by the “AND Clause” button as was shown in the above screenshot)



Once you have constructed the appropriate query, click the **OK** button, then the **Next** button. You will be presented with a “Query Preview” screen (you may use this [try the query string ‘%guitar%’], but it is not required for this session). Press **Next** and you will finish up at the Final screen, which should be completed as shown below:



When you click the **Finish** button, code to perform this query will be generated and inserted into the file under edit.

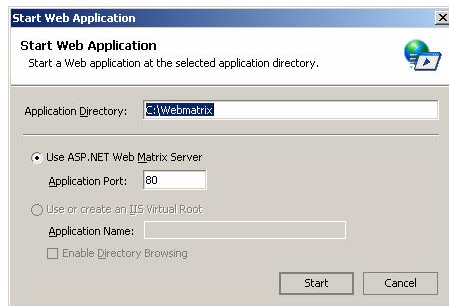
To complete the procedure, look for the first line of the inserted code and edit it. The line:

```
System.Data.DataSet FindArtistByInstrument(string INSTRUMENT) {
```

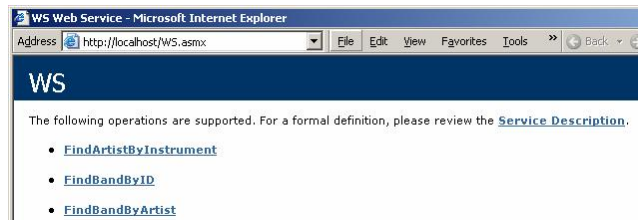
Should be augmented to become:

```
[WebMethod]
public System.Data.DataSet FindArtistByInstrument
    (string INSTRUMENT) {
```

Execute the completed WebService by pressing the **F5** function key. You will be presented with the “Start Web Application” dialog, as shown here:

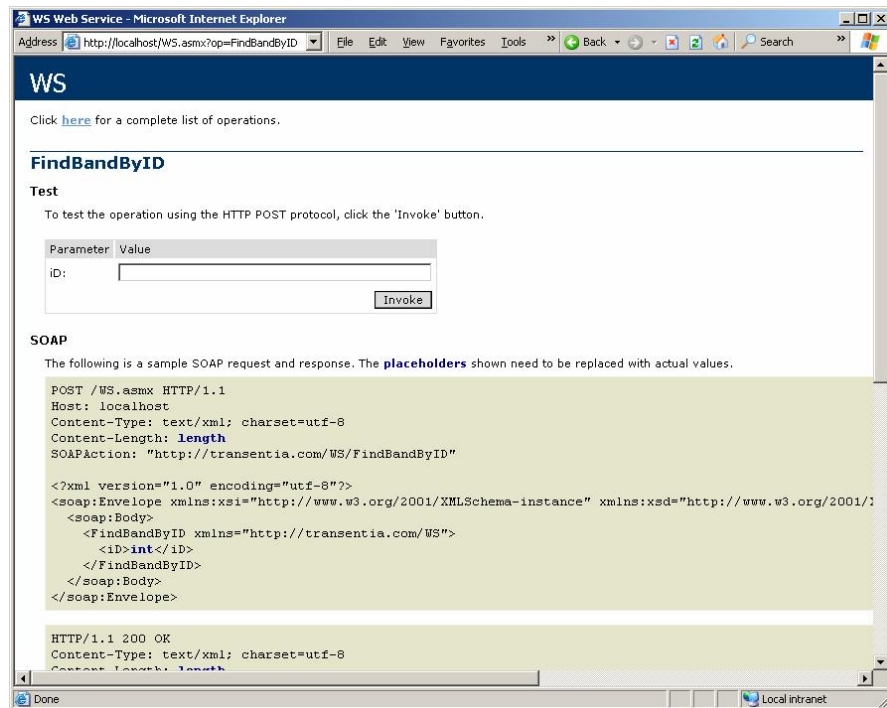


Simply press the Start button. If the ASP.Net Web Matrix server started correctly, you will see its icon (🔧) appear in your taskbar notification area and a web browser will be started to let you explore the service you just created and deployed:

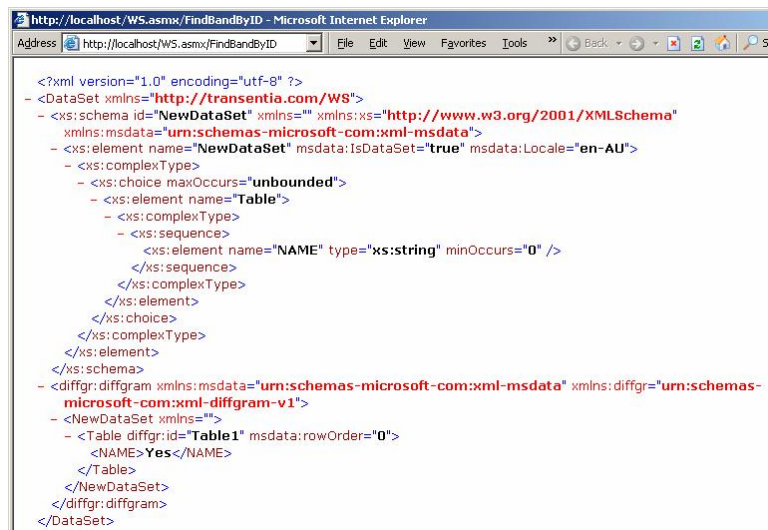


Click on the “Service Description” link and you will be presented with the WSDL file that defines the service (try it, but don’t forget to return to this page once you have examined the generated XML file).

Click on the ‘FindBandByID’ link and you will be taken to a simple client page that describes the SOAP invocation sequence required for correct interaction with the service and which also allows you to exercise the remote method, as shown here:



Enter 1 into the **ID** field and press the **Invoke** button and you will be rewarded with a new window displaying the XML-formatted response SOAP message containing the appropriate data retrieved from the database by the service, as shown here:



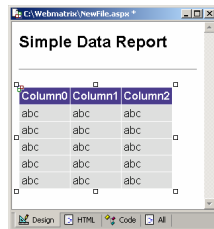
Congratulations! You have created, executed and utilised a WebServices Web Application.

Neither the service nor the automatically-generated default browser view really do very much useful work, however...In the next sections you will create some custom-built client applications that utilise the services provided by this service.

Create a Simple ASP.Net Data-Bound Data Report with Web Matrix

This section should illustrate how easy it is to create more useful web-based client applications using Web Matrix.

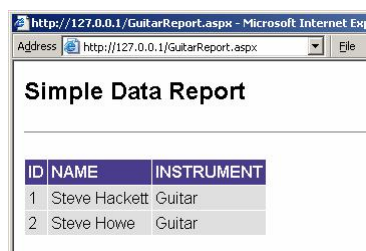
Using Web Matrix, select **File→New...** then choose the “Data Pages/Simple Data Report” template. You will be presented with a sample screen:



Switch to the ‘Code’ tab and edit the template code section to be as shown here:

```
void Page_Load(object sender, EventArgs e)
{
    DataSet
        gds = new transentia.WS().FindArtistByInstrument("Guitar");
    DataGrid1.DataSource = gds.Tables[0].DefaultView;
    DataGrid1.DataBind();
}
```

Once you have done this, you should use **File→Save As...** to save the file under the name **C:\Webmatrix\GuitarReport.aspx**. You should then access the report by using your external browser (a bug in Web Matrix requires you to use an external browser “by hand”) to access the URL <http://127.0.0.1/GuitarReport.aspx>. You will be rewarded with the following:



As you can see, it is simple to get “up and running” with Web Matrix’s tools and templates.

Some customisation of the report format is possible. Return to the code and edit it so that it becomes as shown here:

```
void Page_Load(object sender, EventArgs e)
{
    DataSet
```



```

        gds = new transentia.WS().FindArtistByInstrument("Guitar");

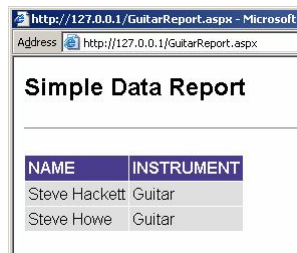
        DataTable gdst = gds.Tables[0];
        DataColumnCollection gdsc = gdst.Columns;
        gdsc.Remove("ID");

        DataGrid1.DataSource = gds.Tables[0].DefaultView;
        DataGrid1.DataBind();

    }

```

When you revisit the page, you will be presented with a filtered view of the data:



The screenshot shows a web browser window with the address bar displaying 'http://127.0.0.1/GuitarReport.aspx - Microsoft'. The page title is 'Simple Data Report'. Below the title is a table with two columns: 'NAME' and 'INSTRUMENT'. The table contains two rows of data: 'Steve Hackett' with 'Guitar' and 'Steve Howe' with 'Guitar'.

NAME	INSTRUMENT
Steve Hackett	Guitar
Steve Howe	Guitar

Create and Execute a Standalone C# Application

Of course, you do not need to access the Web Service you created only from Web Matrix; the service is universally available, as this section will show.

Create the following in the file **C:\Webmatrix.Artist.cs**:

```

namespace transentia
{
    using System;
    using System.Data;

    public class Artist
    {
        public static void Main()
        {
            WS ws = new WS();
            DataSet data = ws.FindArtistByInstrument("Guitar");
            DataTable dt = data.Tables[0];
            DataRowCollection dr = dt.Rows;
            Console.WriteLine("Found {0} guitarists:", dr.Count);
            foreach (DataRow r in dr)
            {
                int artistID = (int) r["ID"];
                string artistName = (string) r["NAME"];
                DataSet artistData = ws.FindBandByArtist(artistID);
                DataTable artistDt = artistData.Tables[0];
                DataRowCollection artistDr = artistDt.Rows;
                foreach (DataRow ar in artistDr)
                {
                    int bandID = (int) ar["BAND"];
                    DataSet bandData = ws.FindBandByID(bandID);
                    DataTable bandDt = bandData.Tables[0];
                    DataRowCollection bandDr = bandDt.Rows;
                    foreach (DataRow br in bandDr)

```

```
{  
    string bandName = (string) br["NAME"];  
    Console.WriteLine("{0} ({1})",  
                        artistName, bandName);  
}  
  
}  
  
}  
  
}
```

Once you have created this, compilation and execution is very simple. Execute the following sequence of commands into the same Command Prompt window that you have used earlier in this session:

```
C:\Webmatrix>
C:\Program Files\Microsoft.NET\SDK\v1.1\Bin\sdkvars.bat
C:\Webmatrix> csc *.cs
C:\Webmatrix> Artist
```

Note:

- There are only three lines here; the first line has been ‘wrapped’ for readability

After compilation and execution, you should be greeted with the following:

```
C:\WINDOWS\System32\cmd.exe

C:\Webmatrix>Artist
Found 2 guitarists:
Steve Hackett (Yes)
Steve Howe (Genesis)

C:\Webmatrix>
```

You can see the versatility of WebServices...

Wrapup

While not as fully-featured as the full Visual Studio.NET product suite, the combination of Web Matrix and the standard SDK makes it possible to do useful work very easily.