**Web Services – Debugging Tips and Tricks**

**General**

Use the link called ‘Creating an N-Tier Web Service’ in the WCF Web Services section on Moodle as a detailed description of the steps and approach you should use in Assignment 1. Although you will be using a web-based front end in the assignment, rather than a Windows Form Application described in those notes, the procedure is very similar. If you encounter errors, the notes below will help you to identify the likely cause and the steps you should take to remove them.

**Configuration**

You must have IIS set up correctly on your machine. Use the notes in the first two links in the Lab section of my Moodle page to configure IIS. If you receive a HTTP 404 error when running your web application, make sure that you have completed Step 3 in the link called ‘Additional IIS configuration notes’.

**Web Sites**

Make sure that you create the web site in IIS first before creating your web service and web client – create a folder for the web site (in the inetpub folder, not wwwroot), then add a web site and associate it to the physical path of the folder you created.

The web site needs to be associated with an application pool – which will be created for you when you add a new web site (IIS will use the same name as you web site – check this if problems occur during debugging, particularly if you are getting a Http 503 – Service Unavailable error). Also check that the physical path of the web service and web client applications (doesn’t matter for windows application clients) is within the web site folder listed in IIS and that they are also associated with the same application pool after you create them in Visual Studio. Clicking on the web site in IIS and selecting browse \*:80 should bring up the default web site. You usually cannot browse to the application(s) within the web site folder.

**Application Pools**

Stop all application pools associated with other websites (not including the ASP.NET application pools). However, do make sure that the application pool containing your website has been started otherwise you will get the Http 503 – Service Unavailable error.

The .NET version needs to be consistent with version used in the web service and client application you create in Visual Studio. Version 4 allows debug tracing and therefore is the preferred option.

You will need to set the Application pool Identity to **LocalSystem** – check this in Advanced Settings for the Application pool. This can cause Http 503 – Service Unavailable errors if not set correctly. It will also cause the Application Pool associated with your web site to stop automatically. If you find this happening whenever you run your application, check the Application pool Identity and reset it to LocalSystem if necessary. Setting the application pool identity to LocalSystem is also a step required to avoid login authentication problems if your website/windows application connects to a database (also refer to the **Database** section overleaf for additional steps)

**Web Service**

Prototype the web service as a simple windows application first.

Connection problems: Check the connection string – if necessary, add a data source to the windows application, connect to the database and copy the connection string details, which can then be pasted into your code.

Check the data types of parameters passed to the stored procedures – they must match the data types of the parameters in SQL Server. **Each** stored procedure needs to be ‘declared’ within its own operation contract in the Public Interface of the web service.

Once a stored procedure can be called from a simple windows application, create the web service. In the web.config file set includeExceptionDetailInFaults="true" to return more details from any errors that might occur during testing.

Run the web service and copy the service.svc URL from IE.

**Client Application**

Prototype the web client as a simple windows application first. Add a service reference and paste the service.svc URL. Make sure you can see all of the functions defined in your web service.

**Database**

A login authentication error can occur when running the client application if access to a database is required. A message will appear in the browser informing you of this problem (assuming you have modified the web.config file to return more detailed error messages, as described in the **Web Service** section above). In SQL Server, click on the **Security** node in the **Object Explorer** pane and then expand **Users** node. Double-click on the user **NT AUTHORITY\SYSTEM**. In the **Login Properties** dialogue box, click on **User Mappings** page. Use the scroll bar to locate the database that the web service interacts with and then click in the **Map** tick box to the left of its name. In the ‘database role membership’ pane of the Login Properties dialogue box, ensure that tick boxes for **db-owner** and **public** have been selected. You will also need to set the Application pool Identity to **LocalSystem,** asdescribed in the **Application Pools** section above.