# Accessing DB2 from Visual Basic

## Contents


1. Introduction ..... 2
2. Prerequisites ..... 2
3. Setup ..... 2
   - 3.1. Visual Basic ..... 2
   - 3.2. DB2 ..... 2
   - Populate DB2 Database ..... 2
   - Set up ODBC Access ..... 3
4. Discovery ..... 3
   - 4.1. Web Service Creation using SOA Gateway ..... 3
   - 4.2. Accessing the WSDL ..... 5
5. Accessing Web Service with Visual Basic ..... 7
   - 5.1. Initial Setup ..... 7
   - 5.2. Designing the Form ..... 13
   - 5.3 Writing the Code ..... 15
     - General Declarations ..... 15
     - Button1_Click ..... 15
     - VB 2005 ..... 15
     - VB 2008 ..... 16
     - Button2_Click ..... 17
   - 5.4 Running the code ..... 18
6. Conclusion ..... 19
7. Appendix ..... 19
   - Form1.vb ..... 19

# 1. Introduction

In this tutorial we will show you how to build a Visual Basic application to access DB2 via the SOA Gateway.

# 2. Prerequisites

It is assumed that you are running the 3 components, DB2, Visual Basic and the SOA Gateway on Windows.

It is assumed you already have a SOA Gateway server and Control Centre installed. See here for more info about installing the SOA Gateway.

# 3. Setup

## 3.1. Visual Basic

To build and run Visual Basic applications, you will need a Visual Studio IDE. If you do not already have it installed, we recommend using the *Microsoft Visual Studio Express* range of products. They can be downloaded freely from Microsoft website, packaged for a number of languages. See here for more information about downloading, installing, and configuring *Visual Basic Express*.

## 3.2. DB2

You will also need a DB2 database. IBM provides a downloadable edition of DB2, called "DB2 Express-C". See this link for the DB2 Express-C homepage. Download and install a version of DB2 Express.

### Populate DB2 Database

Now that you've got DB2 installed, we need to populate it with some demo data. For this we'll use the Risaris Bank Demo, which is available here. Save this file to "c:\Temp\RisarisBank_db2.sql".

- Open the DB2 Control Centre under "IBM DB2", "General Admin Tools" in the Start Menu.
- Right click "All databases" and select "Create Database", "Standard".
- Name your new database "RISBANK". All other options can be left as default, so click "Finish".
- Open a DB2 command shell by typing "db2cmd" in a DOS box.
- From the db2 command shell, change directory to where you downloaded the RisarisBank_db2.sql. E.g "cd \temp".
- Populate the RISBANK database by running the command "db2 -f RisarisBank_db2.sql". Note you may see errors about "SYSTEM.CUSTOMERINFORMATION is an undefined name". These occur because the RisarisBank_db2.sql attempts to drop any existing tables before creating new ones. To prove this, you can run the same command again, and the errors will disappear.
- You can now return to the DB2 Control Centre and view the newly created tables in the RISBANK database.

## Set up ODBC Access

The final thing to do with your DB2 Database is to set up an ODBC DSN which will be used by the SOA Gateway to access this database.

Click Start, Control Panel, Administrative Tools, Data Sources (ODBC)

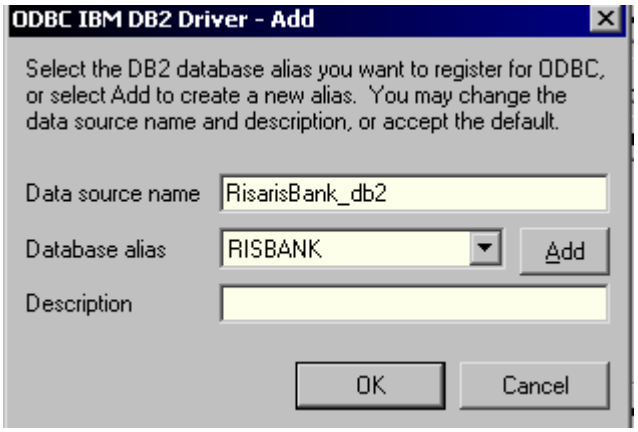
From the resulting screen, choose the "System DSN" Tab.

Click Add

From the list of data source drivers, select "IBM DB2 ODBC DRIVER", and click "Finish".

Enter "RisarisBank_db2" as the Data source Name.

Ensure that the Database Alias is RISBANK, and click "OK".



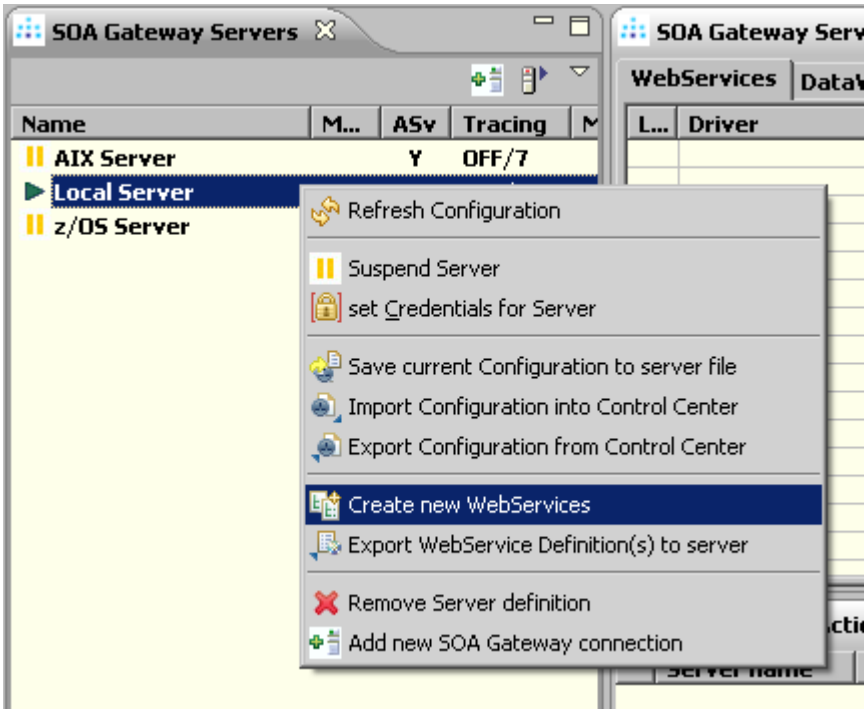

# 4. Discovery

At this stage you've got a Visual Basic IDE, and a DB2 database with some sample data in it. In this section we'll show you how to create web services from each of the DB2 tables. These web services can be used by the Visual Basic language (and many others) to give you direct real-time access to your DB2 Data.

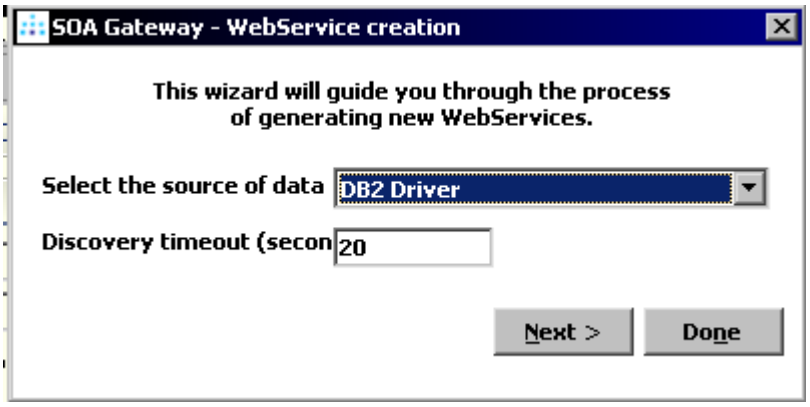
## 4.1.Web Service Creation using SOA Gateway

Start your SOA Gateway Control Centre. See here for an introduction to the Control Centre.

In your servers view, right click the entry which represents your local SOA Gateway Server. Select "Create New Web Services".

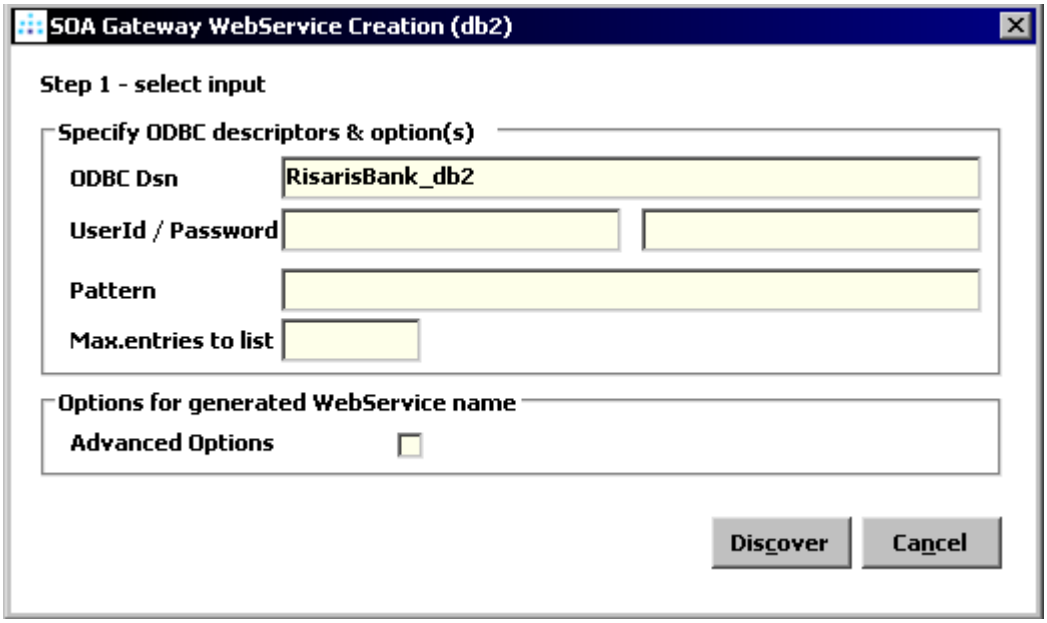



From the next dialog, choose “DB2 Driver”. If you do not see have a DB2 Driver in the list, see how to create one here.




Click Next.

The next screen gives you the ability to add information about your DSN

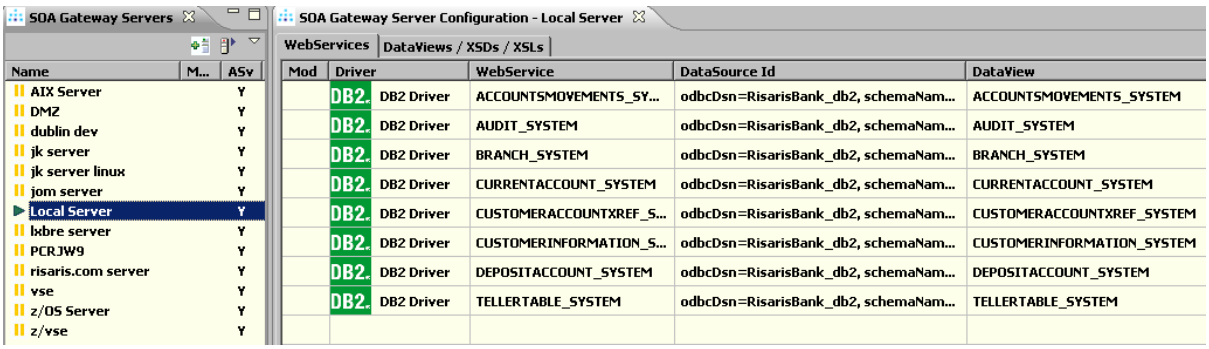



Enter the above information and click Discover.

The wizard will display all the tables it finds at this (RisarisBank) DSN.

Click “Select All”, and click “Import”.

The wizard will create web services from each one of these tables.




You’ve just created 8 Web Services from your 8 DB2 Tables!

## 4.2.Accessing the WSDL

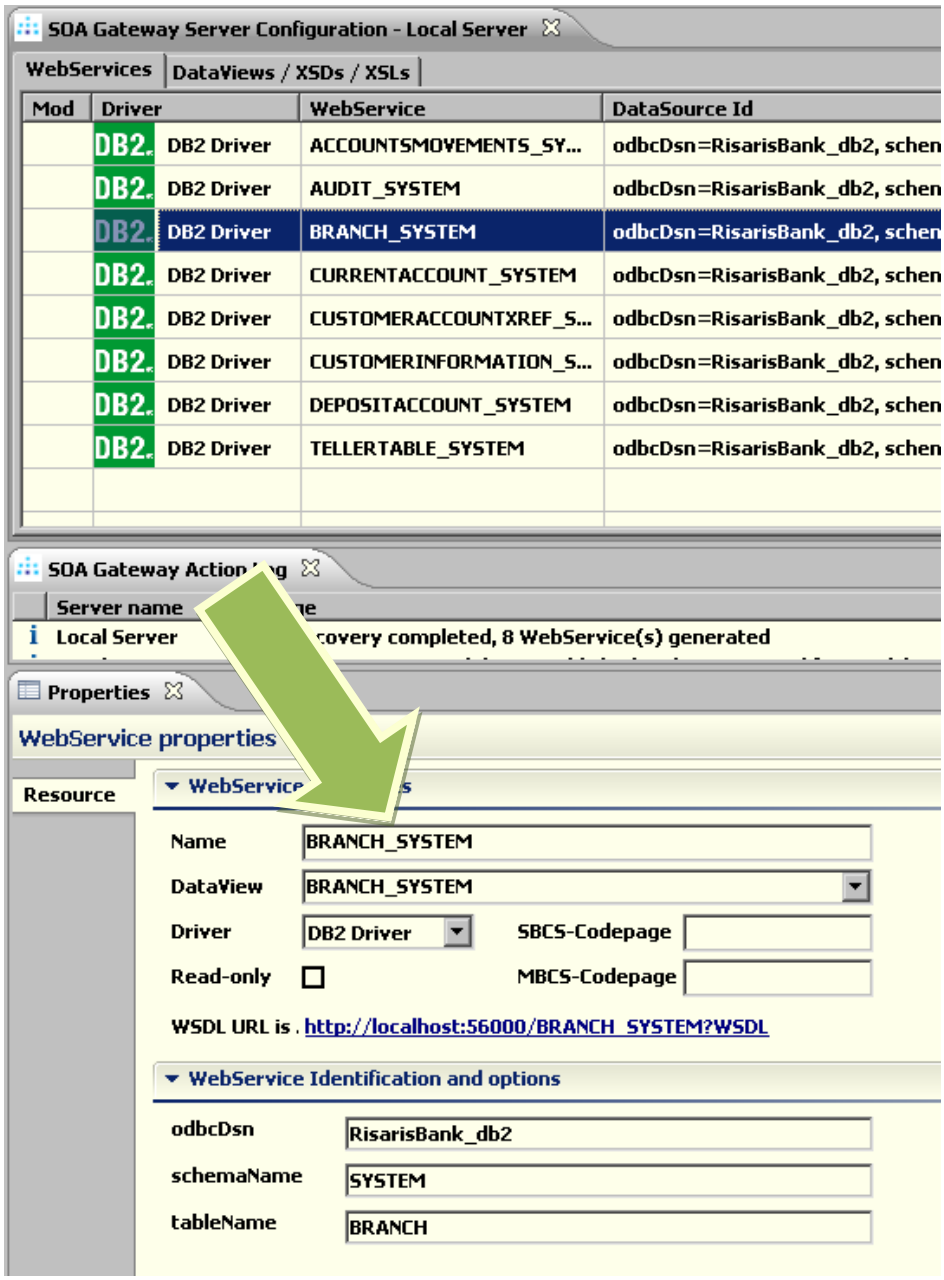
Web Service Description Language (WSDL) is a standard, XML-based language that is used to describe a Web Service.

For each of the 8 web services you’ve created in the previous section, the SOA Gateway provides you with a WSDL to describe the Web Service. The WSDL itself is usually interpreted by a web

service client, such as Visual Basic, but it is useful to know where to find the WSDL for each of your Web Services.

As WSDL is XML-based, it will open in your browser of choice. To see the WSDL for one of your Risaris Bank web services, do the following in your SOA Gateway Control Centre:

- Click on the web service you are interested in, for example the branch service.
- The properties for this web service should appear in your Properties View. If you do not see the Properties view, select Window -> Show View -> Other -> General -> Properties and click OK.
- In the properties view, there is a link to your WSDL. Click it to open the WSDL in a browser.



You can view the WSDL for the other web services by clicking the link from their properties view.

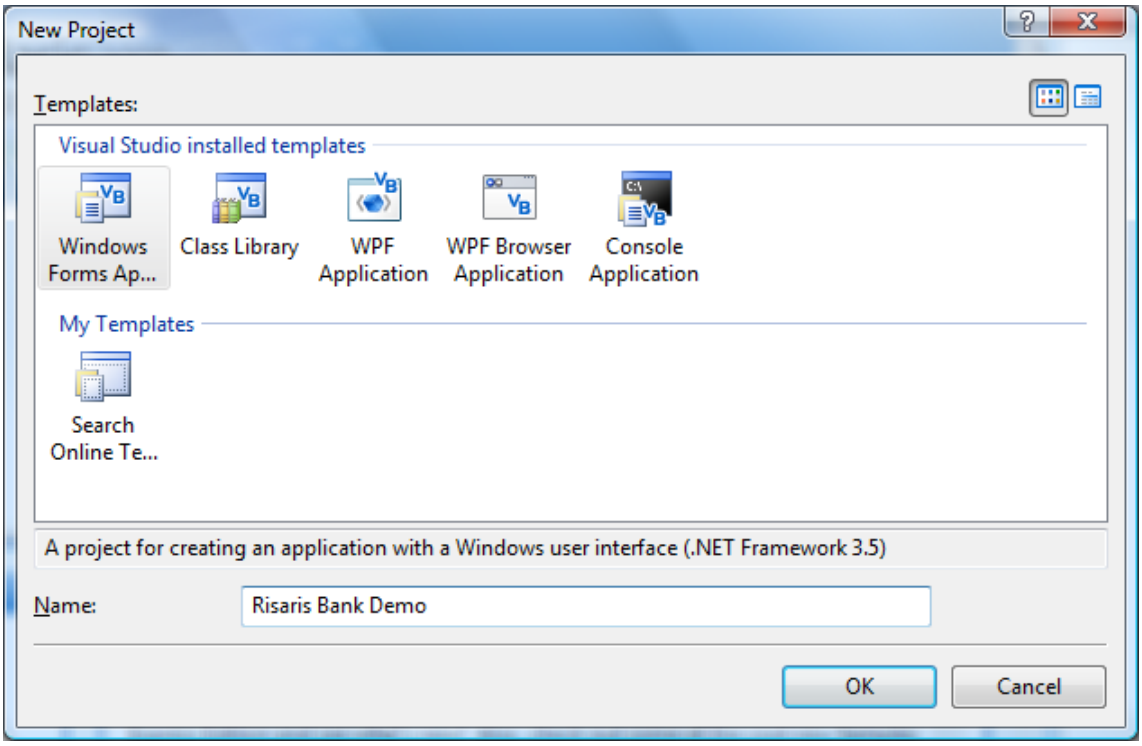
This WSDL is the starting point for using Web Services, and can be used time and again by different web service clients.

# 5. Accessing Web Service with Visual Basic

We will use Visual Basic to build an application which accesses our new Risaris Bank Web Services via the WSDL.

## 5.1.Initial Setup

Start *Microsoft Visual Basic Express* and create a New Windows Forms Application Project named **Risaris Bank Demo**.




In the *Solution Explorer*, right click the solution name**,** then **Add Web Reference (VC #** 2005) or select **Add Service Reference** (VB 2008)

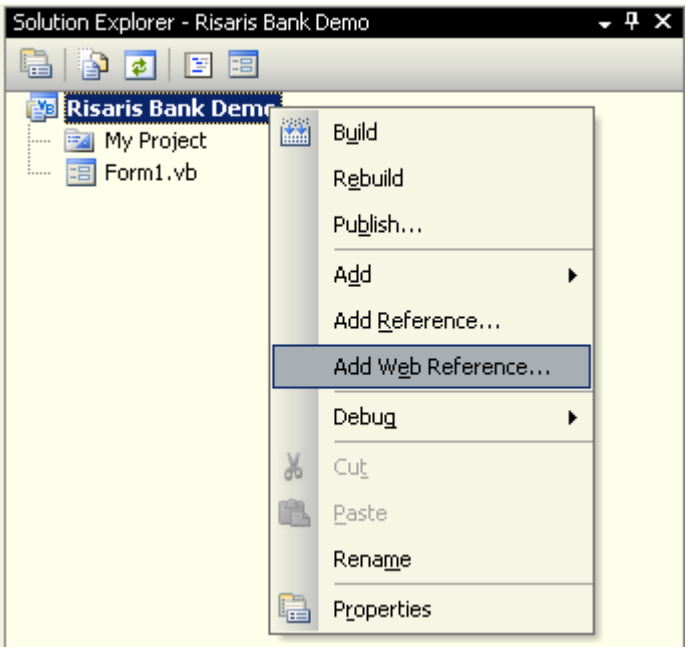



Figure 1: VB 2005

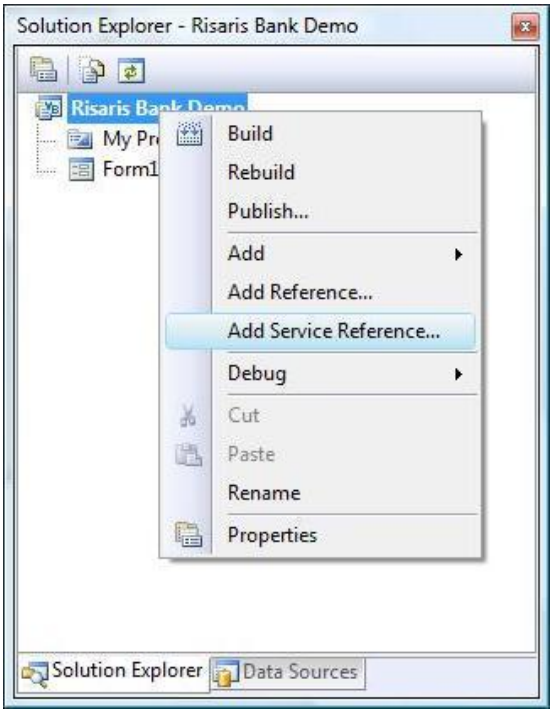



Figure 2: VB 2008

We want to use 2 of the Web Services we've created, the CUSTOMERINFORMATION_SYSTEM and the CURRENTACCOUNT_SYSTEM web services.

- Copy the URL of your web service WSDL into the URL box e.g. http://localhost:56000/CUSTOMERINFORMATION_SYSTEM?WSDL
- Click **Go**.
- Once the WSDL has been loaded, change Web Reference / Namespace to **CustomerInformation**

- Click **Add Reference / OK**

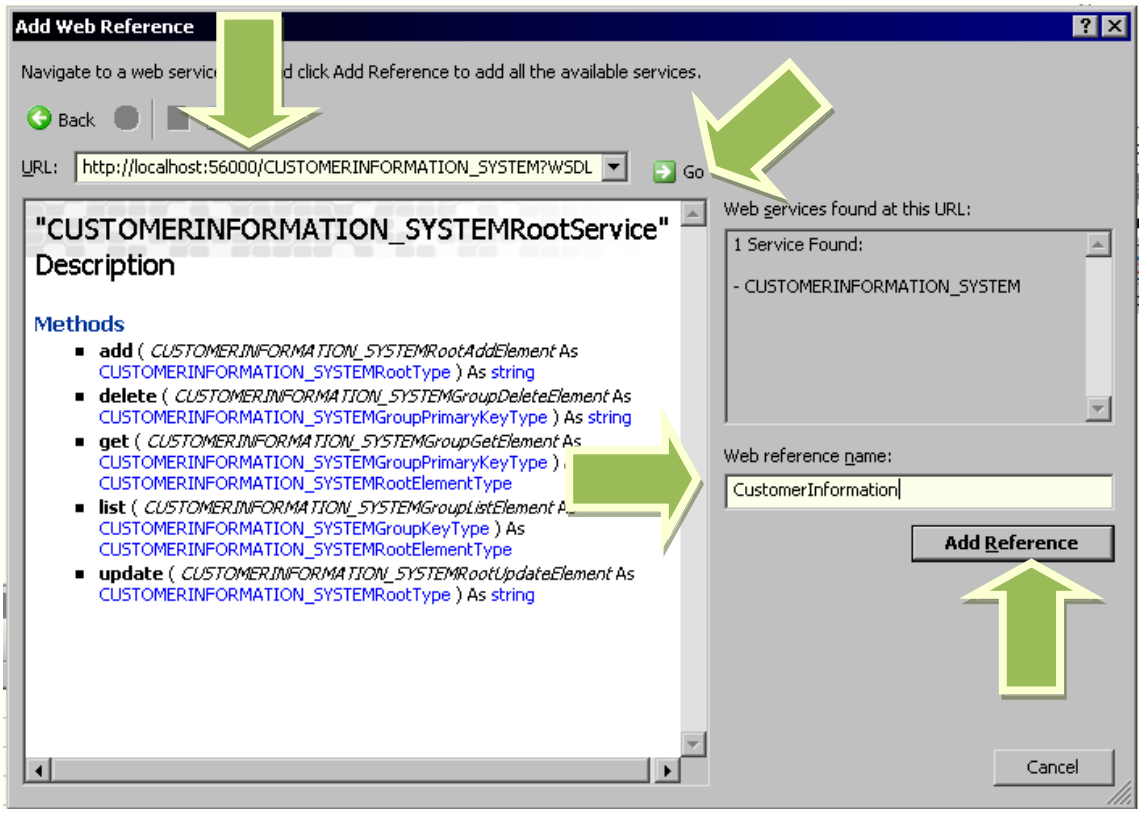



Figure 3: VB 2005

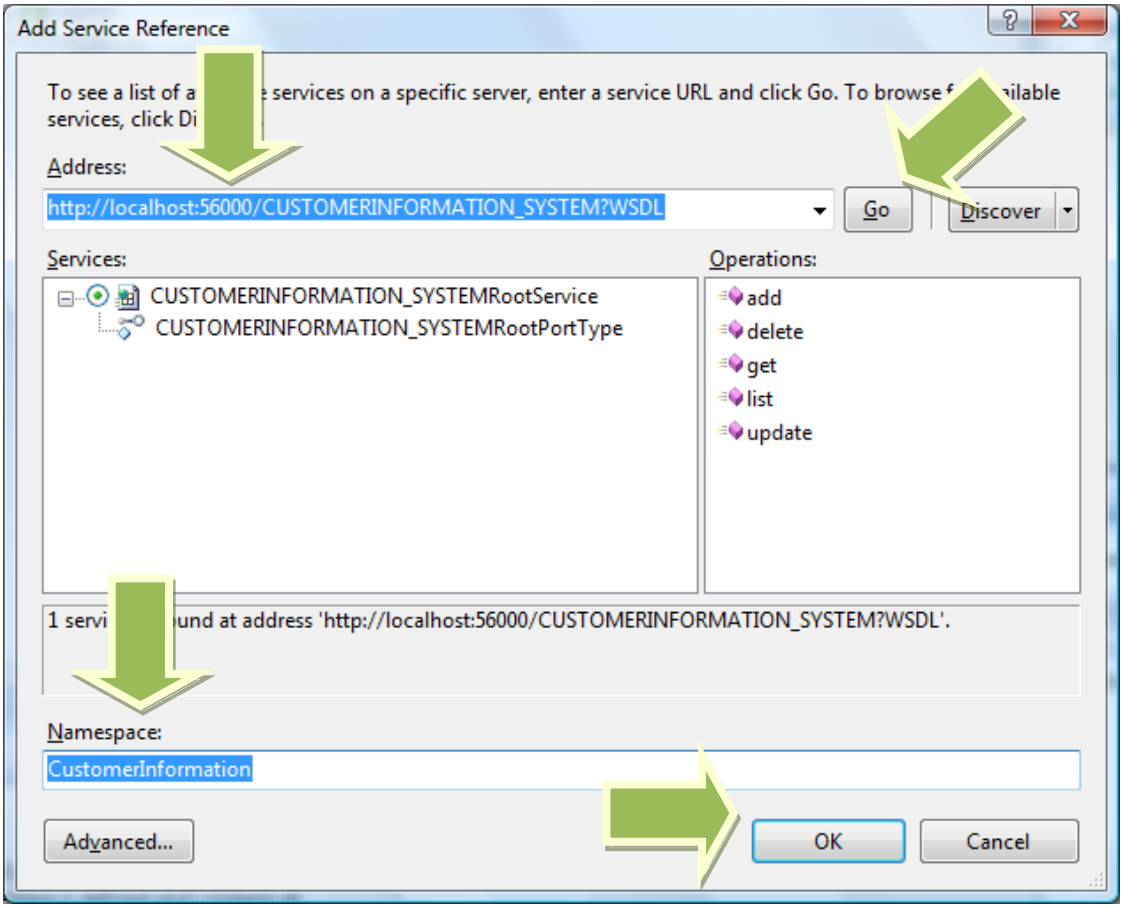



Figure 4: VB 2008

Do the same for the CURRENTACCOUNT_SYSTEM WSDL, http://localhost:56000/CURRENTACCOUNT_SYSTEM?WSDL, except change the Web Reference / Namespace to **CurrentAccount**

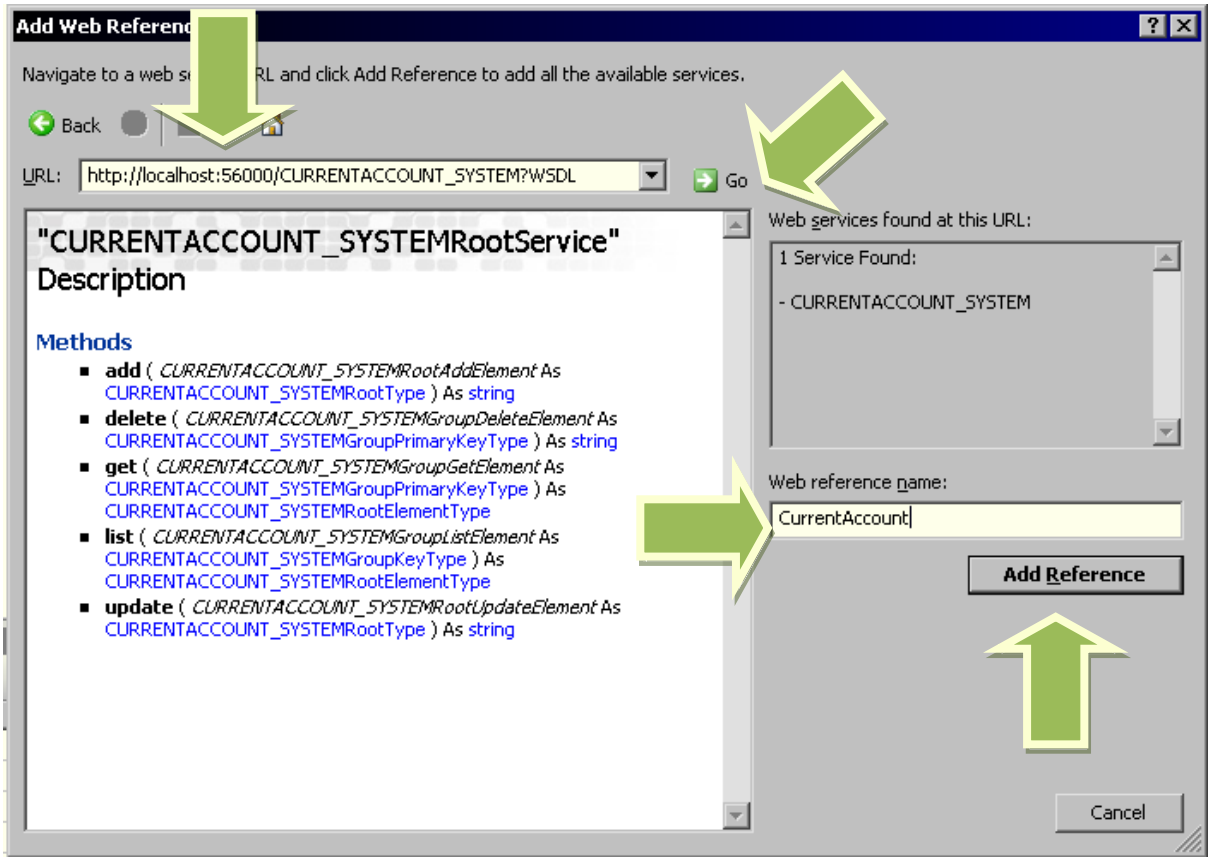



Figure 5: VB 2005

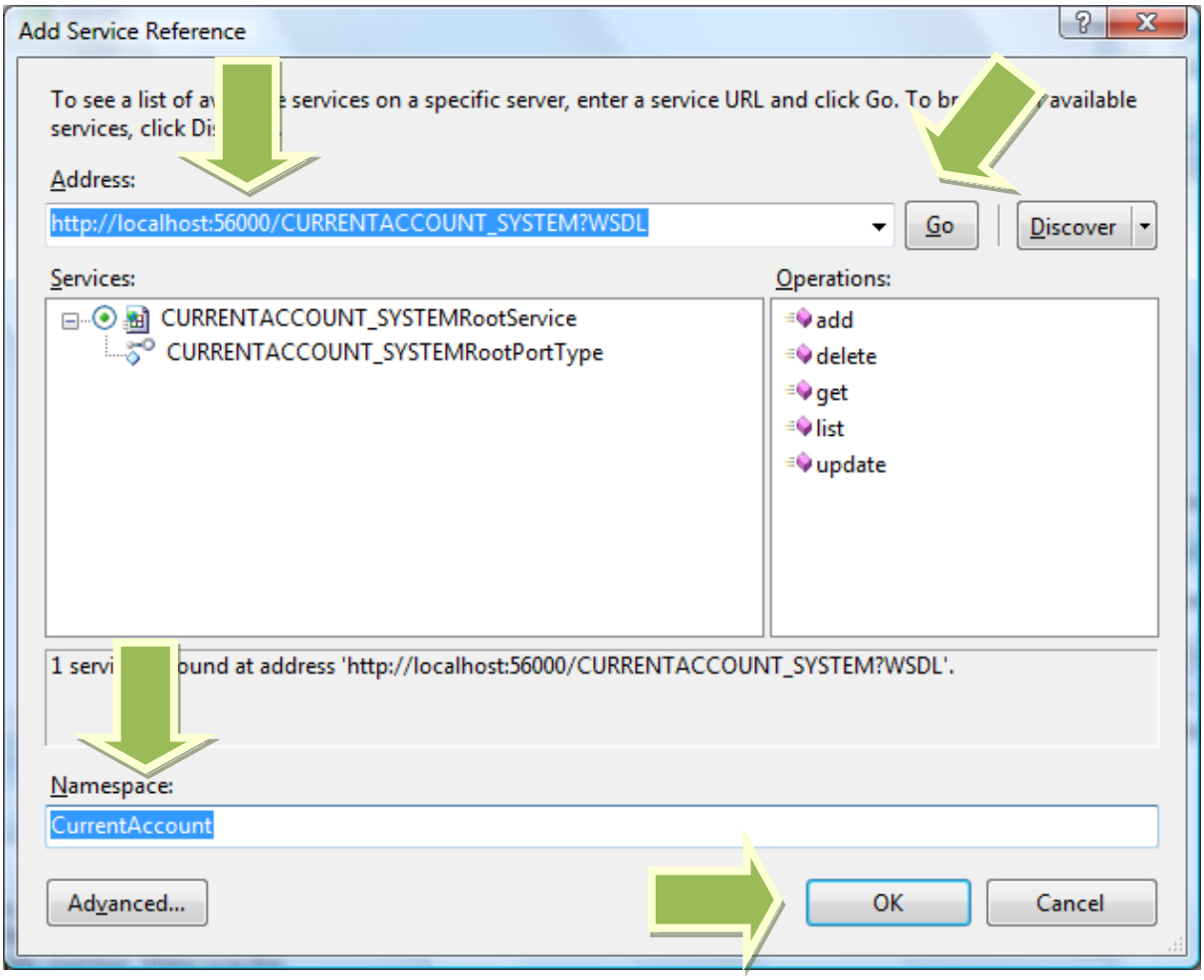
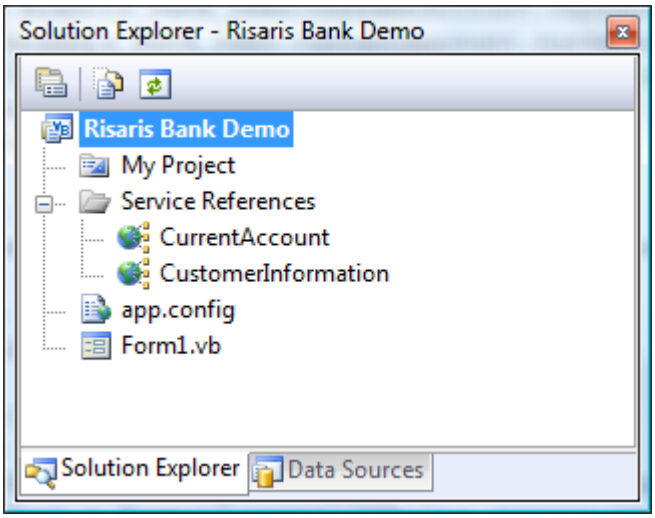


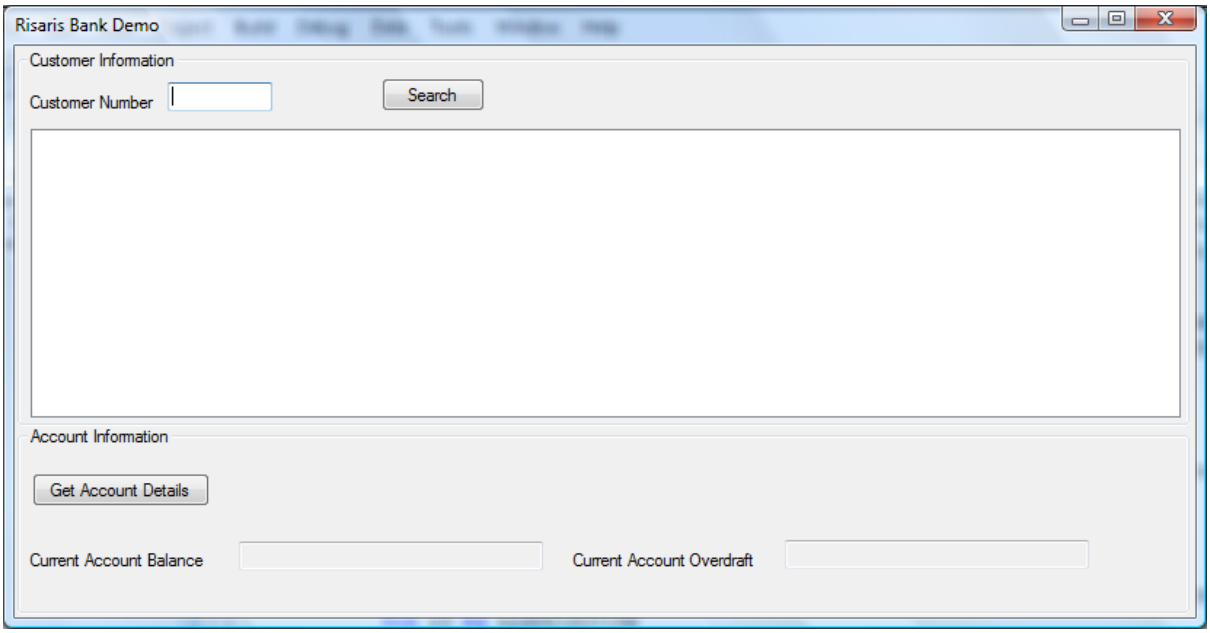

Figure 6: VB 2008

You should now have 2 new Service / Web References loaded into your Solution Explorer



## 5.2.Designing the Form

In this section we'll add the necessary controls to our Form. Before you start ensure, that you are in the Designer View (View -> Designer), and that you have the control Toolbox available (View->Toolbox)




I've used the following controls in this Form

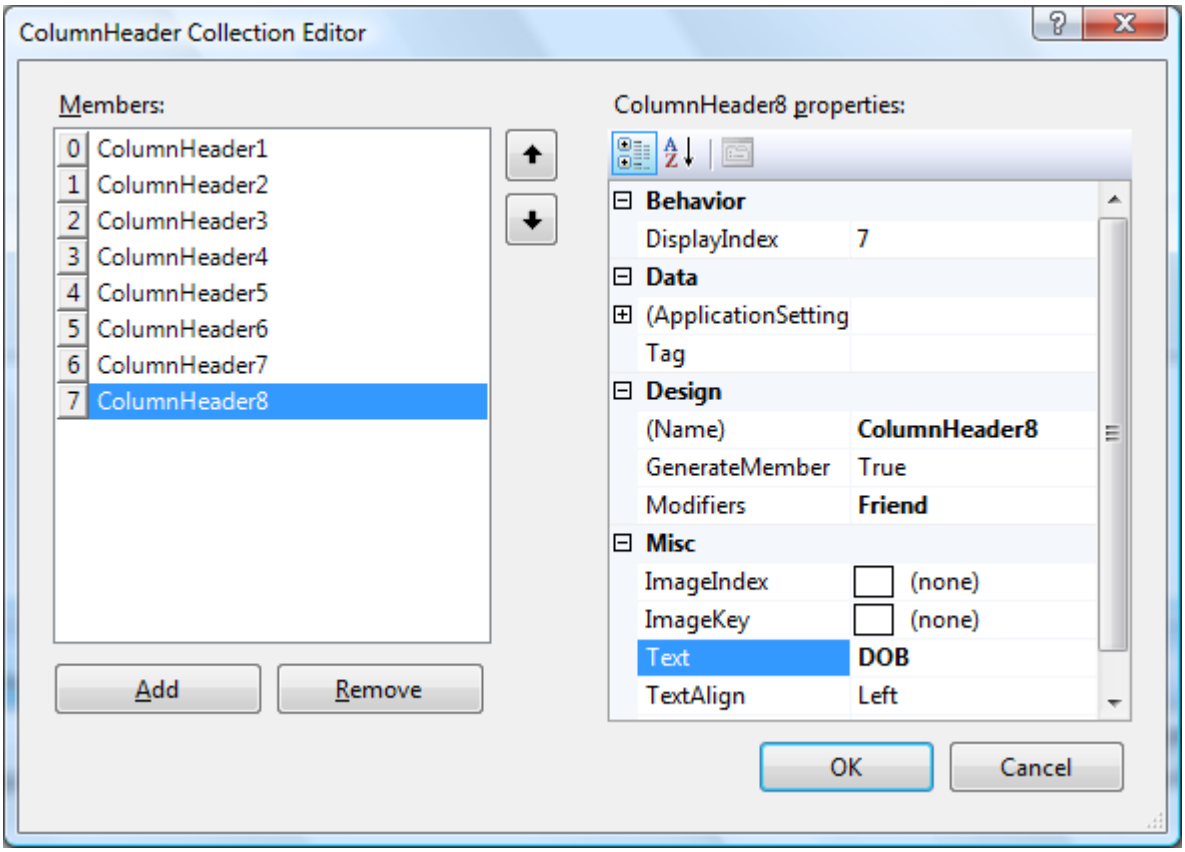
- GroupBox with Text property set to Customer Information.
- GroupBox with Text property set to Account Information.
- Label (Customer Number, Current Account Balance, Current Account Overdraft).
- TextBox 1 (Customer Number).
- TextBox 2 (Current Account Balance).
- TextBox 3 (Current Account Overdraft).
- Button 1 (Text property set to Search)
- Button 2 (Text property set to Get Account Details)
- ListView1 (MultiSelect property set to False).

The ListView is the only control that needs additional setup.

When you add this control, right-click on it and select Edit Column from the pop-up menu. Add 8 members, each with the following Text property as this is the order of the columns passed back by the web service.

- ✓ Customer Number
- ✓ First Name
- ✓ Surname
- ✓ Address Line 1
- ✓ Address Line 2
- ✓ City

- ✓ Post Code
- ✓ DOB




Note that I haven't changed any of the default design names that the VB designer has given me.

You may change these to whatever you wish, but be aware your code in the next section will have to be cognisant of this!

## 5.3 Writing the Code

Now that the Form controls have been added, we need to write the code to call our Web Services when the buttons are clicked.
The entire code for Form1.vb is listed in the Appendix.

Switch to your Code view, by clicking View->Code

### General Declarations

In the General Declarations section at the top of the form add the following:

```
Imports Risaris_Bank_Demo.CustomerInformation
Imports Risaris_Bank_Demo.CurrentAccount
```

These statements include your 2 Web References you added earlier.

### Button1_Click

Switch back to your Designer view, and double-click the "Search" button in your Form. Your IDE will switch over to the code view, and a new member function to handle the button click will be created.

When this button is clicked, we want to take the contents of TextBox1 (which is the Customer ID), and send this to our Customer Information web service. The web service should return the required customer information for that ID.

We break that customer information down into its respective parts, and then add that to our listView.

The code is as follows:

### VB 2005

```
      Dim service As New CUSTOMERINFORMATION_SYSTEMRootService()

      ' create a new key value that we send to the web service
      Dim key As
Risaris_Bank_Demo.CustomerInformation.CUSTOMERINFORMATION_SYSTEMGroupKeyTyp
e
      key = New
Risaris_Bank_Demo.CustomerInformation.CUSTOMERINFORMATION_SYSTEMGroupKeyTyp
e
      ' set the CustomerNumber to the contents of textBox1
      key.CUSTOMERNUMBER = TextBox1.Text

        ' set up a variable to store the result
      Dim results As
Risaris_Bank_Demo.CustomerInformation.CUSTOMERINFORMATION_SYSTEMRootElement
Type

      ' call the "list" operation of web service!
      results = service.list(key)

      ListView1.Items.Clear()

      Dim customerinformationGroupType As
CUSTOMERINFORMATION_SYSTEMGroupType
```

```
      For Each customerinformationGroupType In
results.CUSTOMERINFORMATION_SYSTEMRoot()
            Dim lv As ListViewItem
            lv = New
      ListViewItem(customerinformationGroupType.CUSTOMERNUMBER)

            ' add the rest of the items in the row
            lv.SubItems.Add(customerinformationGroupType.FIRSTNAME)
            lv.SubItems.Add(customerinformationGroupType.SURNAME)
            lv.SubItems.Add(customerinformationGroupType.ADDRESSLINE1)
            lv.SubItems.Add(customerinformationGroupType.ADDRESSLINE2)
            lv.SubItems.Add(customerinformationGroupType.CITY)
            lv.SubItems.Add(customerinformationGroupType.POSTCODE)
            lv.SubItems.Add(customerinformationGroupType.DATEOFBIRTH)

            ' add the row to the listView
            ListView1.Items.Add(lv)
            ListView1.View = View.Details
            ListView1.FullRowSelect = True
        Next
```

### VB 2008

```
        Dim service As
CustomerInformation.CUSTOMERINFORMATION_SYSTEMRootPortTypeClient
        service = New
CustomerInformation.CUSTOMERINFORMATION_SYSTEMRootPortTypeClient()

        ' create a new key value that we send to the web service
        Dim key As
Risaris_Bank_Demo.CustomerInformation.CUSTOMERINFORMATION_SYSTEMGroupKeyTyp
e
        key = New
Risaris_Bank_Demo.CustomerInformation.CUSTOMERINFORMATION_SYSTEMGroupKeyTyp
e
        ' set the CustomerNumber to the contents of TextBox1
        key.CUSTOMERNUMBER = TextBox1.Text

        ' set up a variable to store the result
        Dim results As
Risaris_Bank_Demo.CustomerInformation.CUSTOMERINFORMATION_SYSTEMRootElement
Type

        ' call the "list" operation of web service!
        results = service.list(Nothing, Nothing, key)
        ListView1.Items.Clear()
        For Each customerinformationGroupType In
results.CUSTOMERINFORMATION_SYSTEMRoot()
            Dim lv As ListViewItem
            lv = New
ListViewItem(customerinformationGroupType.CUSTOMERNUMBER)

            ' add the rest of the items in the row
            lv.SubItems.Add(customerinformationGroupType.FIRSTNAME)
            lv.SubItems.Add(customerinformationGroupType.SURNAME)
            lv.SubItems.Add(customerinformationGroupType.ADDRESSLINE1)
            lv.SubItems.Add(customerinformationGroupType.ADDRESSLINE2)
```

```
            lv.SubItems.Add(customerinformationGroupType.CITY)
            lv.SubItems.Add(customerinformationGroupType.POSTCODE)
            lv.SubItems.Add(customerinformationGroupType.DATEOFBIRTH)

            ' add the row to the listView
            ListView1.Items.Add(lv)
            ListView1.View = View.Details
            ListView1.FullRowSelect = True
        Next
```

### Button2_Click

Switch back to Design view and double-click on the Get Account Details button which will add the Button2_Click handler. Within this subroutine we will call the current account web service. Set up security details. Call the service with customer number as key.

### VB 2005

```
If ListView1.SelectedIndices.Count = 1 Then
            ' create a new instance of the Current Account web service
            Dim service As New CURRENTACCOUNT_SYSTEMRootService()

            Dim currentCustomerId As String
            ' get the currrently selected Customer ID
            currentCustomerId =
ListView1.SelectedItems.Item(0).SubItems.Item(0).Text
            ' create a new key value that we send to the web service
            Dim key As
Risaris_Bank_Demo.CurrentAccount.CURRENTACCOUNT_SYSTEMGroupKeyType
            key = New
Risaris_Bank_Demo.CurrentAccount.CURRENTACCOUNT_SYSTEMGroupKeyType
            key.ACCOUNTNUMBER = ""
            key.CUSTOMERNUMBER = currentCustomerId.ToString

            ' set up a variable to store the result
            Dim results As
Risaris_Bank_Demo.CurrentAccount.CURRENTACCOUNT_SYSTEMRootElementType

            ' call the "list" operation of web service!
            results = service.list(key)

            ' Now put the results of the web service
            ' into the Balance and Overdraft text boxes
            TextBox2.Text =
FormatCurrency(results.CURRENTACCOUNT_SYSTEMRoot(0).BALANCE / 100, 2)
            TextBox3.Text =
FormatCurrency(results.CURRENTACCOUNT_SYSTEMRoot(0).OVERDRAFTLIMIT / 100, )
        End If
```

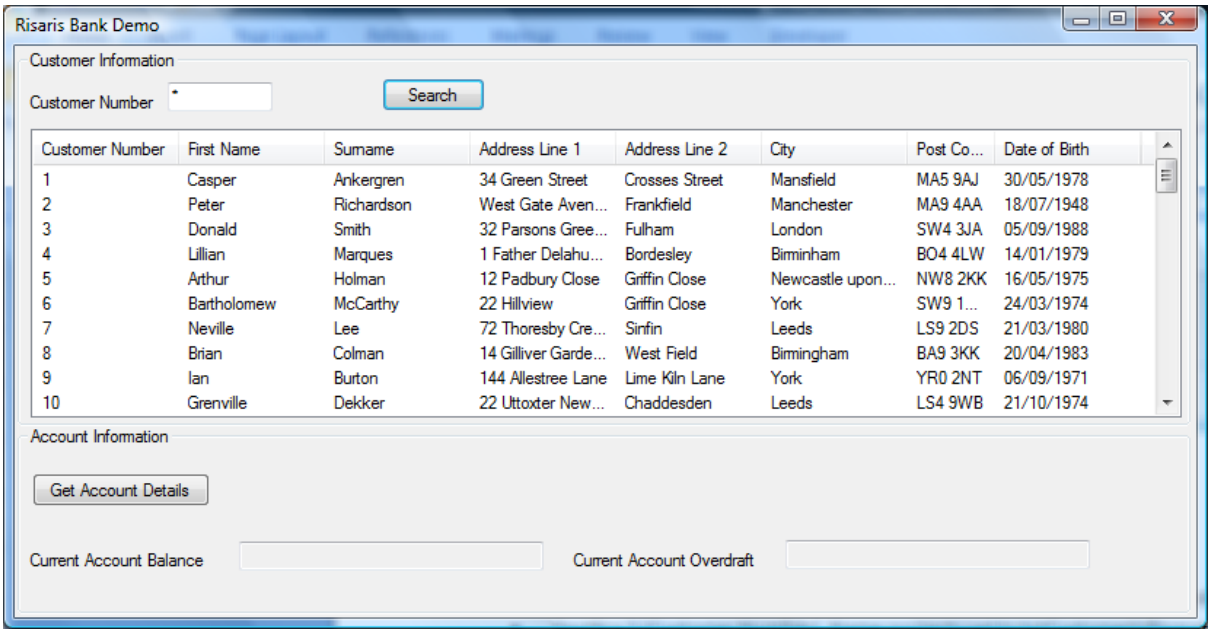
### VB 2008

```
        If ListView1.SelectedIndices.Count = 1 Then
            ' create a new instance of the Current Account web service
```

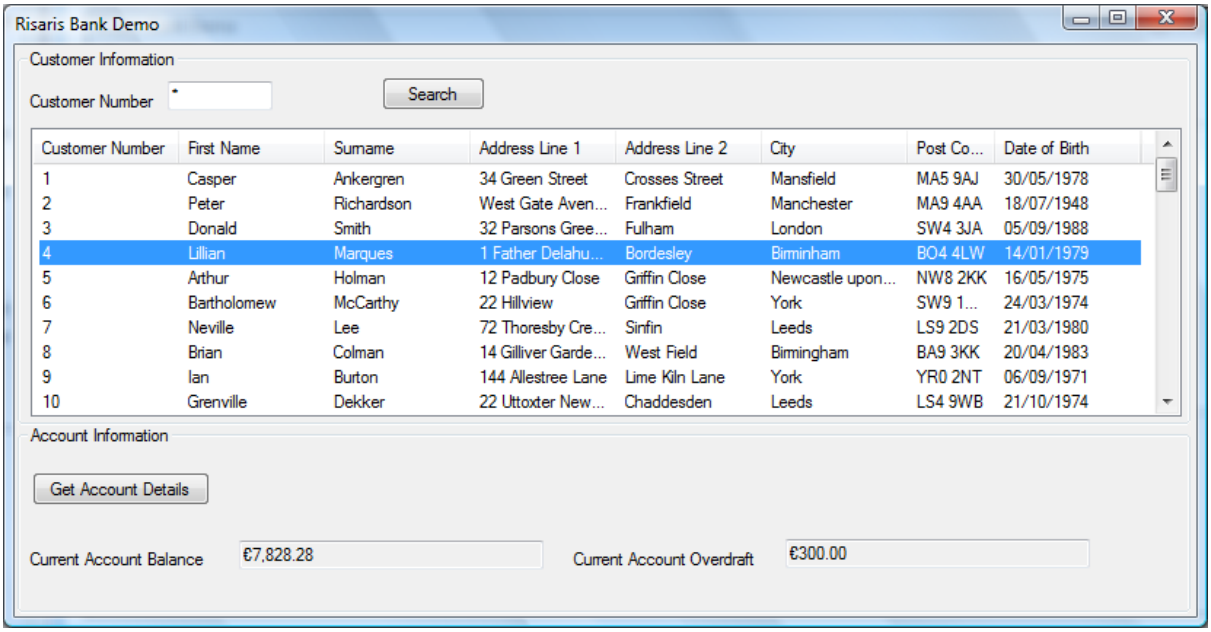
```
            Dim currentAccountService As
CurrentAccount.CURRENTACCOUNT_SYSTEMRootPortTypeClient
            currentAccountService = New
CurrentAccount.CURRENTACCOUNT_SYSTEMRootPortTypeClient()

            Dim currentCustomerId As String
            ' get the currrently selected Customer ID
            currentCustomerId =
ListView1.SelectedItems.Item(0).SubItems.Item(0).Text
            ' create a new key value that we send to the web service
            Dim key As
Risaris_Bank_Demo.CurrentAccount.CURRENTACCOUNT_SYSTEMGroupKeyType
            key = New
Risaris_Bank_Demo.CurrentAccount.CURRENTACCOUNT_SYSTEMGroupKeyType
            key.ACCOUNTNUMBER = ""
            key.CUSTOMERNUMBER = currentCustomerId.ToString

            ' set up a variable to store the result
            Dim results As
Risaris_Bank_Demo.CurrentAccount.CURRENTACCOUNT_SYSTEMRootElementType

            ' call the "list" operation of web service!
            results = currentAccountService.list(Nothing, Nothing, key)

            ' Now put the results of the web service
            ' into the Balance and Overdraft text boxes
            TextBox2.Text =
FormatCurrency(results.CURRENTACCOUNT_SYSTEMRoot(0).BALANCE / 100, 2)
            TextBox3.Text =
FormatCurrency(results.CURRENTACCOUNT_SYSTEMRoot(0).OVERDRAFTLIMIT / 100, )
        End If
```

## 5.4 Running the code

By hitting F5 or Debug -> Start Debugging, you can run your code. In the Customer Number text box, you may enter * or a known customer number and hit the Search button to call the Customer Information web service and get a list of all the customers in the CustomerInformation table.



From the resultant list, select the record you are interested in. Then click on the Get Account Details button which will call the CurrentAccount web service to retrieve the current account balance and overdraft limit for this customer. Example show results for Customer Number 4:




If you hit problems, you may wish to debug your code by adding breakpoints in your code. See the IDE documentation for further information.

## 6. Conclusion

This tutorial shows how to access DB2 from Visual Basic using the SOA Gateway. As you can see, you have built a powerful application that uses Web Services to retrieve information in real-time.

## 7. Appendix

### Form1.vb VB 2005

Code available here

### Form1.vb VB 2008

Code available here