

Accessing MySQL from Visual Basic

Contents

1. Introduction	2
2. Prerequisites	2
3. Setup	2
3.1. Populate MySQL Database.....	2
3.2. Set up ODBC Access	3
4. Discovery.....	4
4.1. Web Service Creation using SOA Gateway	4
4.2. Accessing the WSDL	6
5. Accessing Web Service with Visual Basic.....	8
5.1. Initial Setup	8
5.2. Designing the Form	13
5.3 Writing the Code.....	15
General Declarations.....	15
Button1_Click.....	15
Button2_Click.....	17
5.4 Running the code	19
6. Conclusion.....	20
7. Appendix	20
Form1.vb for VB 2005	20
Form1.vb for VB 2008	20

1. Introduction

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

2. Prerequisites

It is assumed that you are running the 3 components, MySQL, 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

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*.

You will also need a MySQL database. Again, the Open Source version (known as the *MySQL Community Server*) can be freely downloaded from the MySQL website. See [this link](#) for download, and [here](#) to step you through the installation and configuration.

3.1. Populate MySQL Database

Now that you've got MySQL installed and configured, you will need to populate it with some demo data. For this we use the Risaribank sample. This is available [here](#).

Save this file to "C:\Temp\Risaribank.sql"

- Connect to the MySQL Server using the **mysql** command.

```
E.g shell> mysql -u root -p
```

This command connects to the server using the MySQL `root` account to make sure that you'll have permission to create the `Risaribank` database. The `-p` option tells **mysql** to prompt you for the `root` password. Enter the password when prompted. (Remember that the MySQL `root` account is not the same as the operating system `root` account and probably will have a different password.)

- Create the `Risaribank` database.

```
mysql> CREATE DATABASE Risaribank;
```

```
mysql> use Risaribank;
```

- Load the contents of `Risaribank.sql` into the `Risaribank` database. E.g.

```
mysql> SOURCE c:\Temp\Risaribank.sql
```

- After the SOURCE command finishes, you can view your new tables.

```
mysql> SHOW TABLES;
```

```
mysql> DESCRIBE CustomerInformation;
```

```
mysql> DESCRIBE Branch;
```

etc ...

3.2.Set up ODBC Access

The final thing to do with your MySQL 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 "MySQL ODBC 3.51 Driver".

If you do not see this driver in the list, you need to install the MySQL Connector. See [here](#) for more information. We recommend installing v3.51.

Click Finish, and a window will appear allowing you to enter the DSN information. Add the following:

Data Source Name: RisarisBank

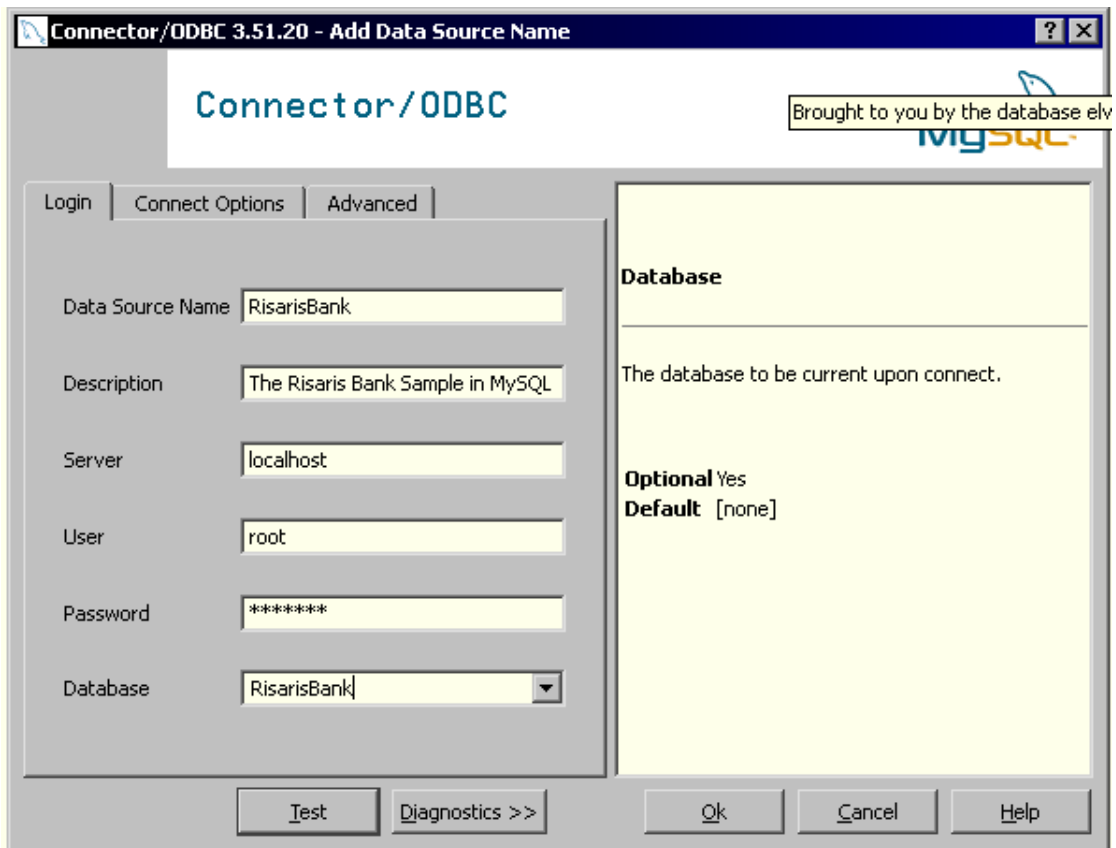
Description: The Risaris Bank Sample in MySQL

Server: localhost

User: root

Password: **** your MySQL root password ****

Database: RisarisBank (*select from the drop down list*)



All other options can be left as-is. Click OK.

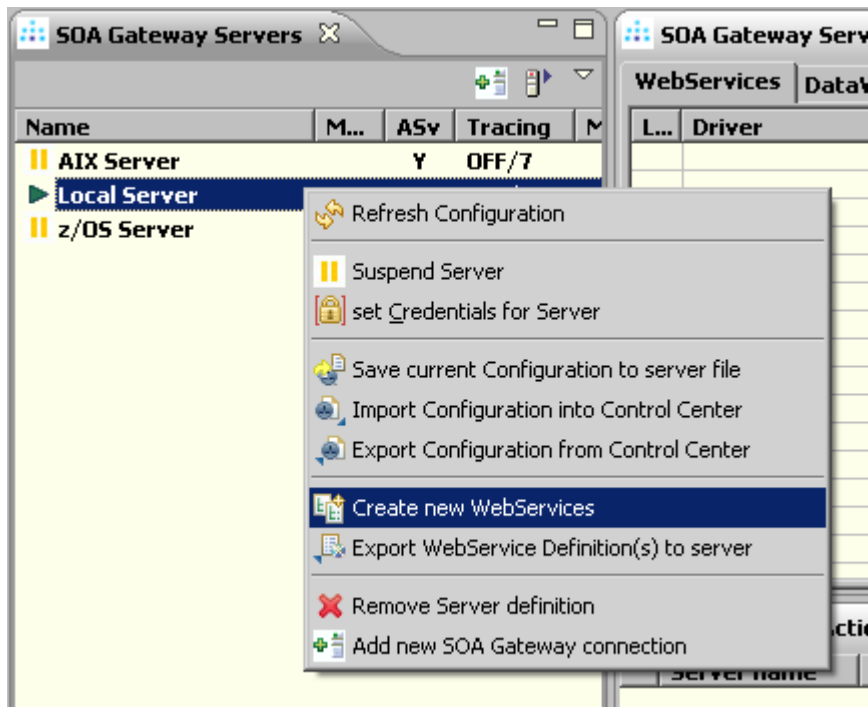
4. Discovery

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

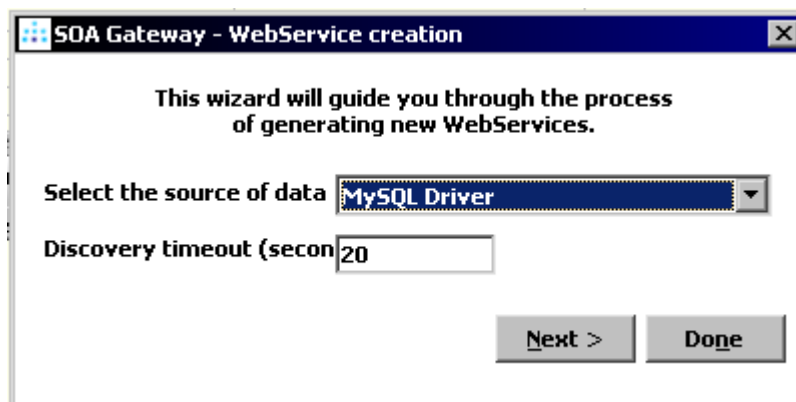
4.1. Web Service Creation using SOA Gateway

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".

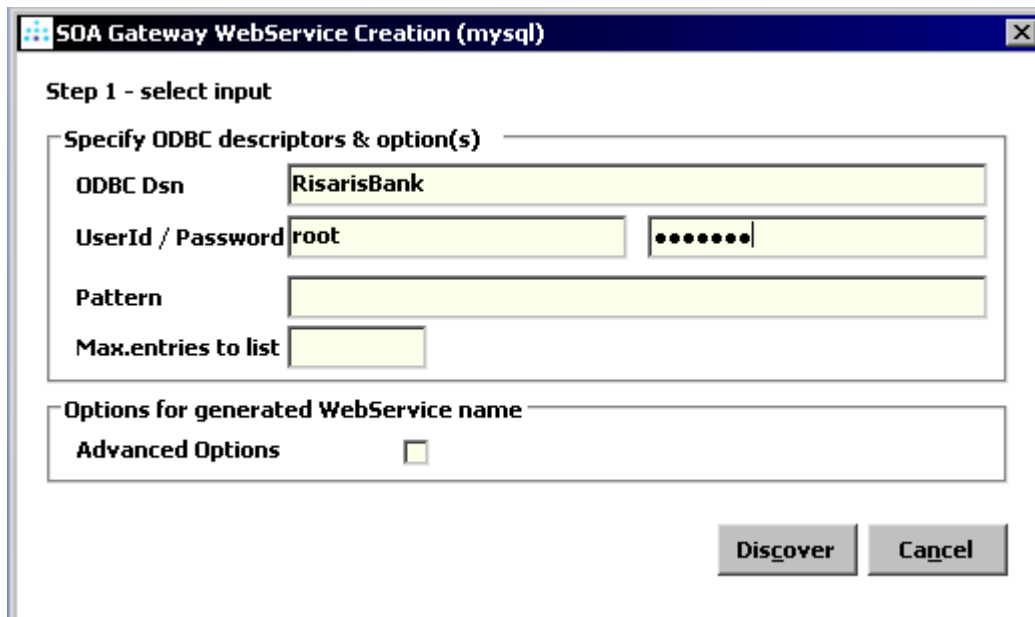


From the next dialog, choose “MySQL Driver”. If you do not see have a MySQL Driver in the list, see how to create one [here](#).



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.

SOA Gateway Servers			SOA Gateway Server Configuration - Local Server				
			WebServices / XSDs / XSLs				
Name	M...	ASv	Mod	Driver	WebService	DataSource Id	DataView
AIX Server	Y		MySQL	MySQL Driver	accountsmovements	odbcDsn=RisarisBank, tableName=accountsmovements	accountsmovements
DMZ	Y		MySQL	MySQL Driver	audit	odbcDsn=RisarisBank, tableName=audit	audit
dublin dev	Y		MySQL	MySQL Driver	branch	odbcDsn=RisarisBank, tableName=branch	branch
jk server	Y		MySQL	MySQL Driver	currentaccount	odbcDsn=RisarisBank, tableName=currentaccount	currentaccount
jk server linux	Y		MySQL	MySQL Driver	customeraccountxref	odbcDsn=RisarisBank, tableName=customeraccountxref	customeraccountxref
jom server	Y		MySQL	MySQL Driver	customerinformation	odbcDsn=RisarisBank, tableName=customerinformation	customerinformation
Local Server	Y		MySQL	MySQL Driver	depositaccount	odbcDsn=RisarisBank, tableName=depositaccount	depositaccount
lxbre server	Y		MySQL	MySQL Driver	tellertable	odbcDsn=RisarisBank, tableName=tellertable	tellertable
PCRJW9	Y						
risaris.com server	Y						
vse	Y						
z/OS Server	Y						
z/vse	Y						

You've just created 8 Web Services from your 8 MySQL 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.

The screenshot displays the SOA Gateway Server Configuration interface. The top window, titled "SOA Gateway Server Configuration - Local Server", has two tabs: "WebServices" and "DataViews / XSDs / XSLs". The "WebServices" tab is active, showing a table with the following data:

Mod	Driver	WebService	DataSource Id
	MySQL Driver	accountsmovements	odbcDsn=RisarisBank, tableN
	MySQL Driver	audit	odbcDsn=RisarisBank, tableN
	MySQL Driver	branch	odbcDsn=RisarisBank, tableN
	MySQL Driver	currentaccount	odbcDsn=RisarisBank, tableN
	MySQL Driver	customeraccountxref	odbcDsn=RisarisBank, tableN
	MySQL Driver	customerinformation	odbcDsn=RisarisBank, tableN
	MySQL Driver	depositaccount	odbcDsn=RisarisBank, tableN
	MySQL Driver	tellertable	odbcDsn=RisarisBank, tableN

Below this table is the "SOA Gateway Action Log" window, which shows two messages: "Local Server ... discovery completed, 8 WebService(s) generated" and "Local Server ... configuration autosaved due to published WebService modification(s)".

The bottom window, titled "Properties", shows the "WebService properties" for the "branch" service. The "Resource" is "WebService". The "Name" is "branch", the "DataView" is "branch", and the "Driver" is "MySQL Driver". The "WSDL URL is" is <http://localhost:56000/branch?WSDL>. Under "WebService Identification and options", the "odbcDsn" is "RisarisBank" and the "tableName" is "branch". A green arrow points to the "WSDL URL" field.

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

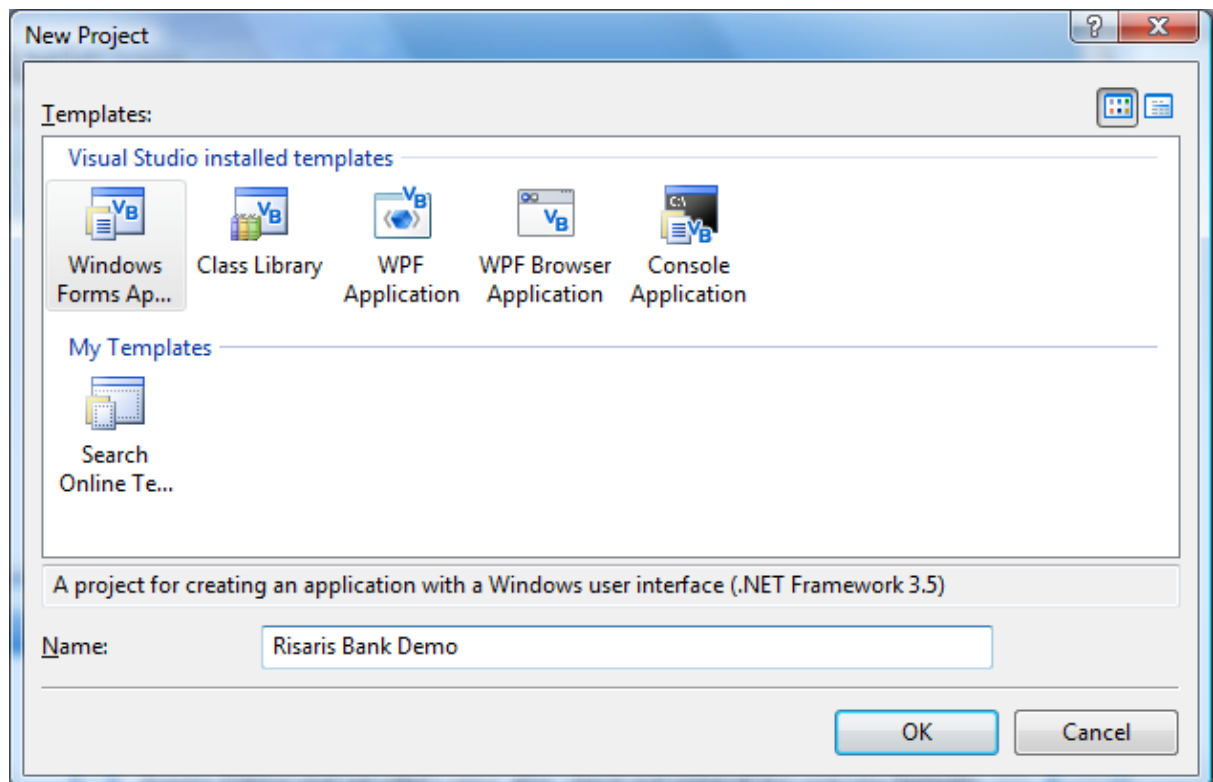
This WSDL is the starting point for using Web Services, and can be used time and again by different web service clients.

5. Accessing Web Service with Visual Basic

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

5.1. Initial Setup

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



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

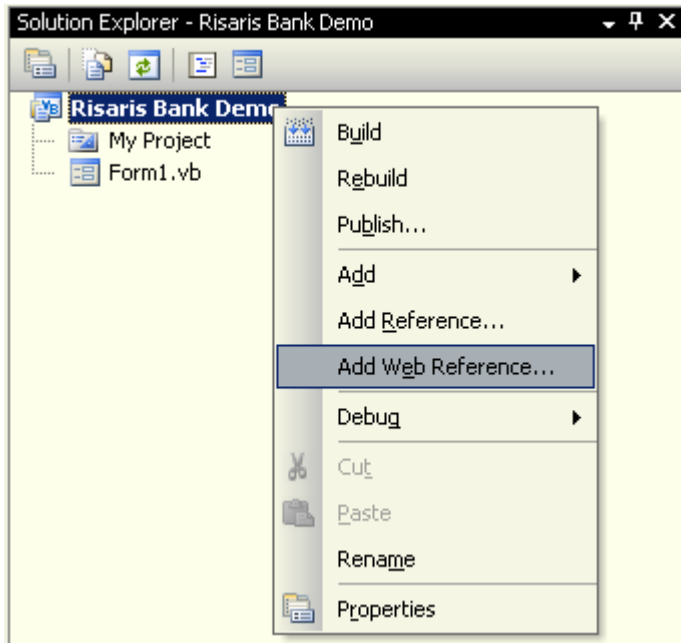


Figure 1: VB 2005

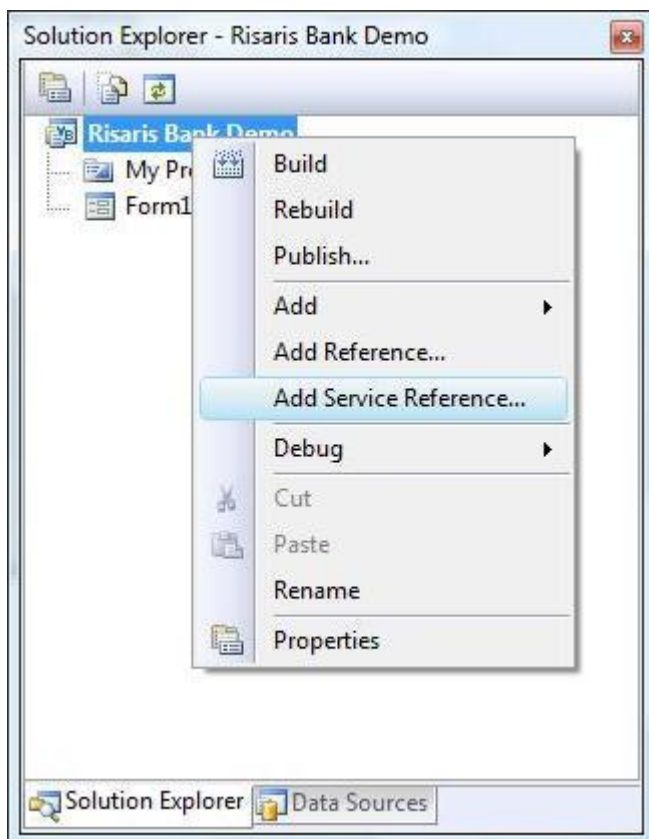


Figure 2: VB 2008

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

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

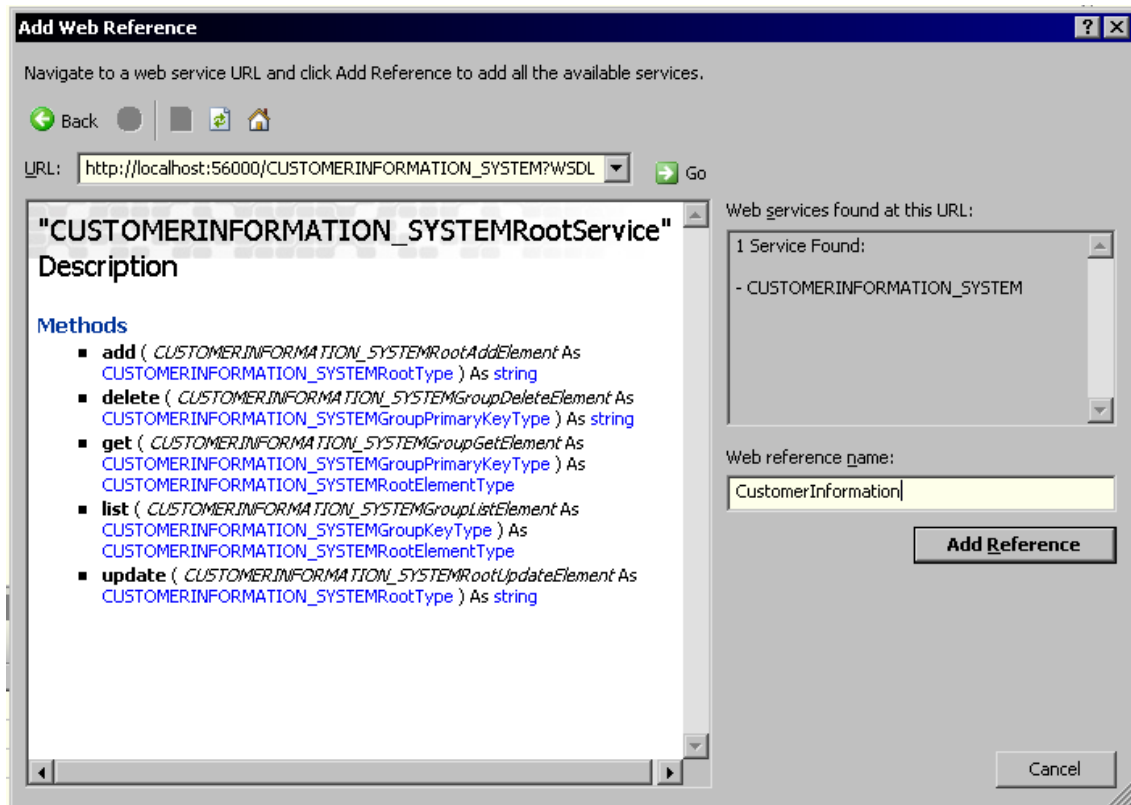


Figure 3: VB 2005

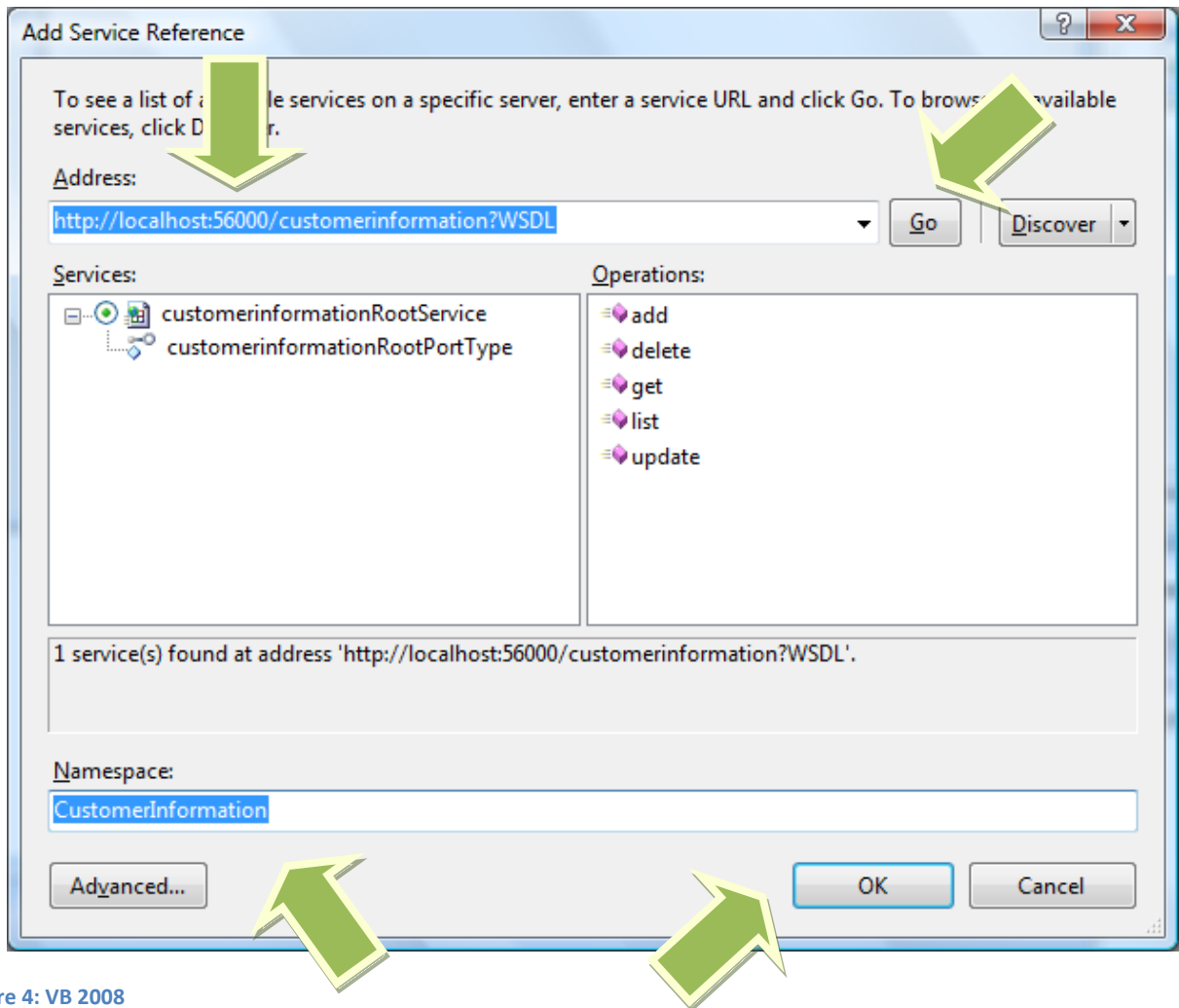
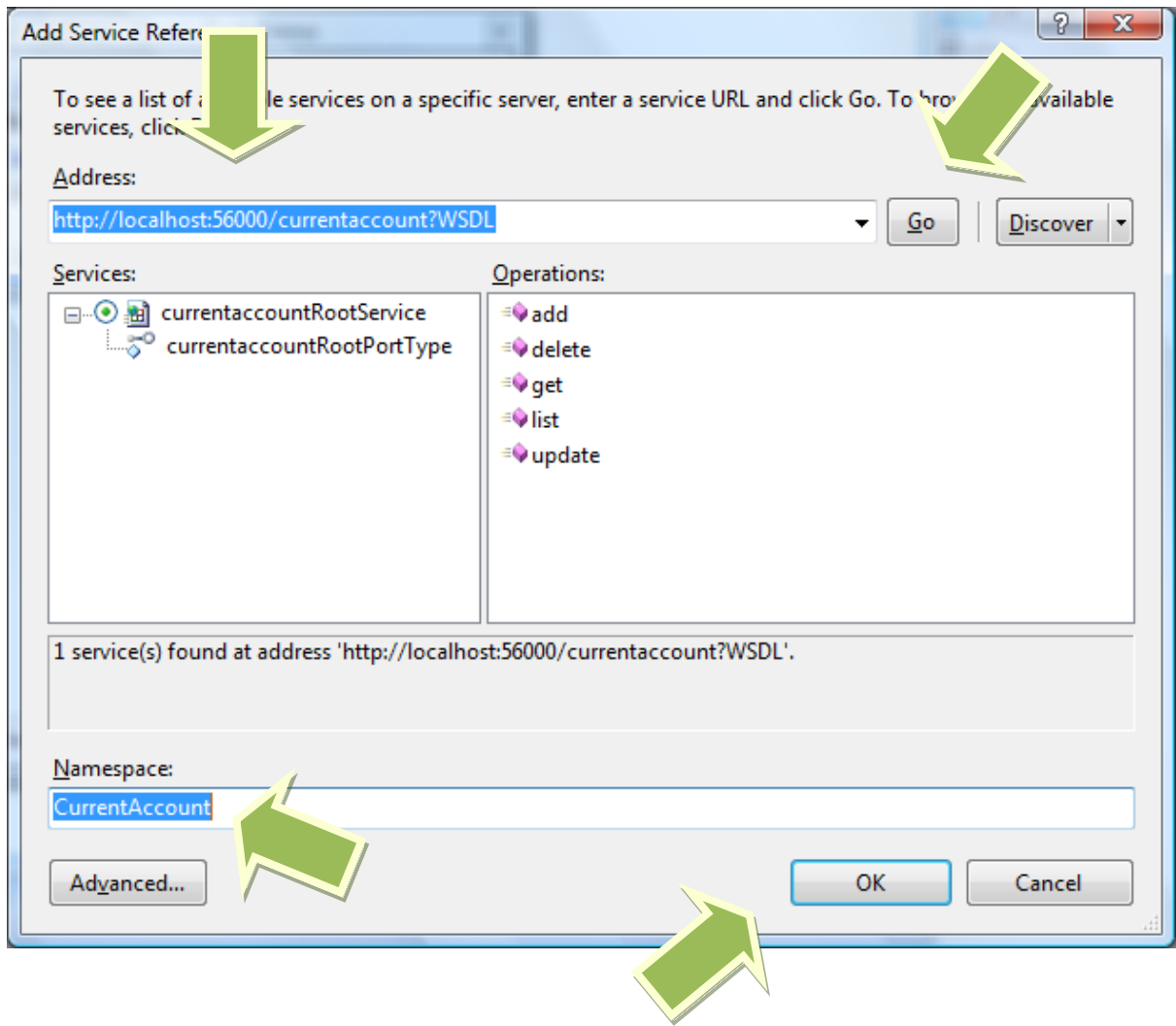


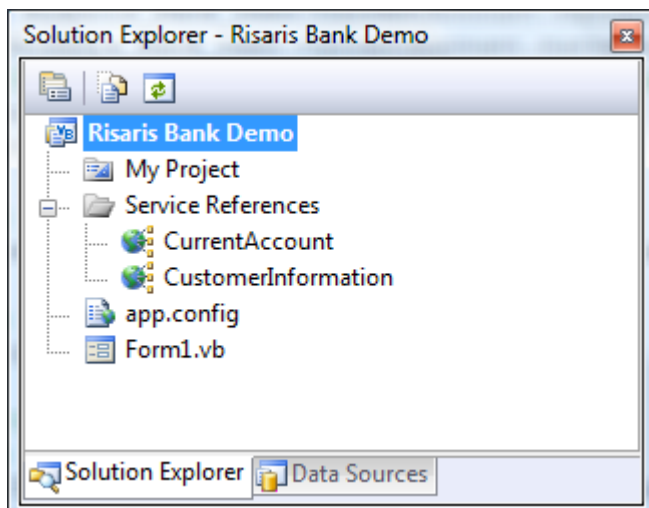
Figure 4: VB 2008

Do the same for the currentaccount WSDL, <http://localhost:56000/currentaccount?WSDL>, except change the Namespace to **CurrentAccount**

N.B. Obviously, depending on your particular setup, localhost:56000, may have to change appropriately.

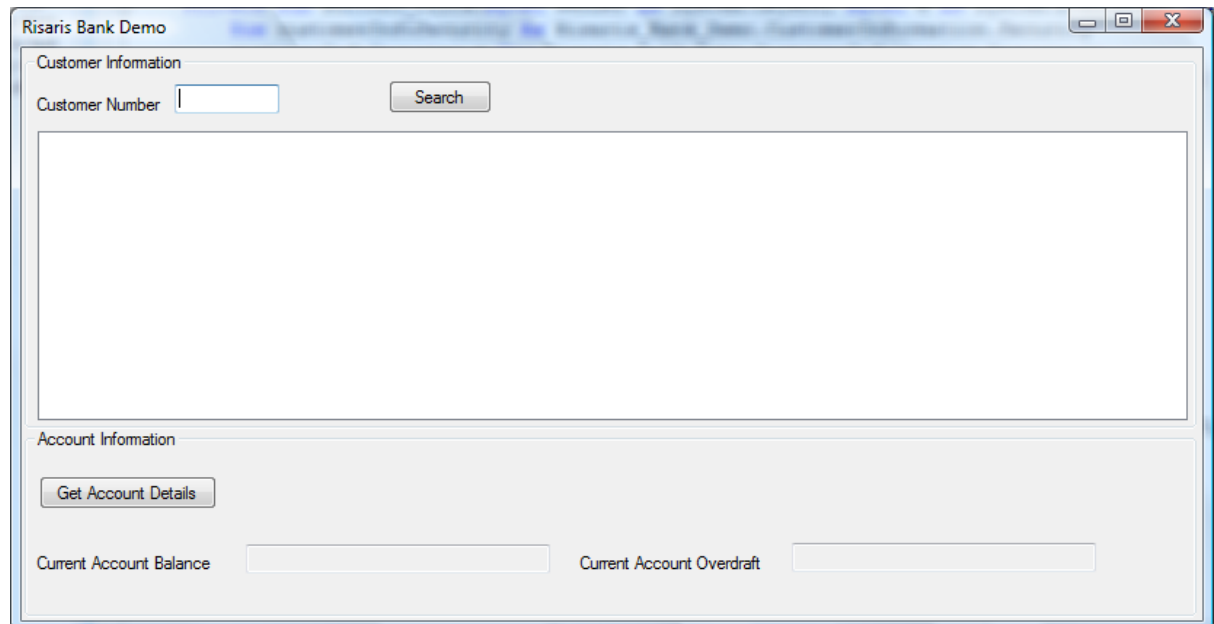


You should now have 2 new Service / Web References loaded into your Solution Explorer



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

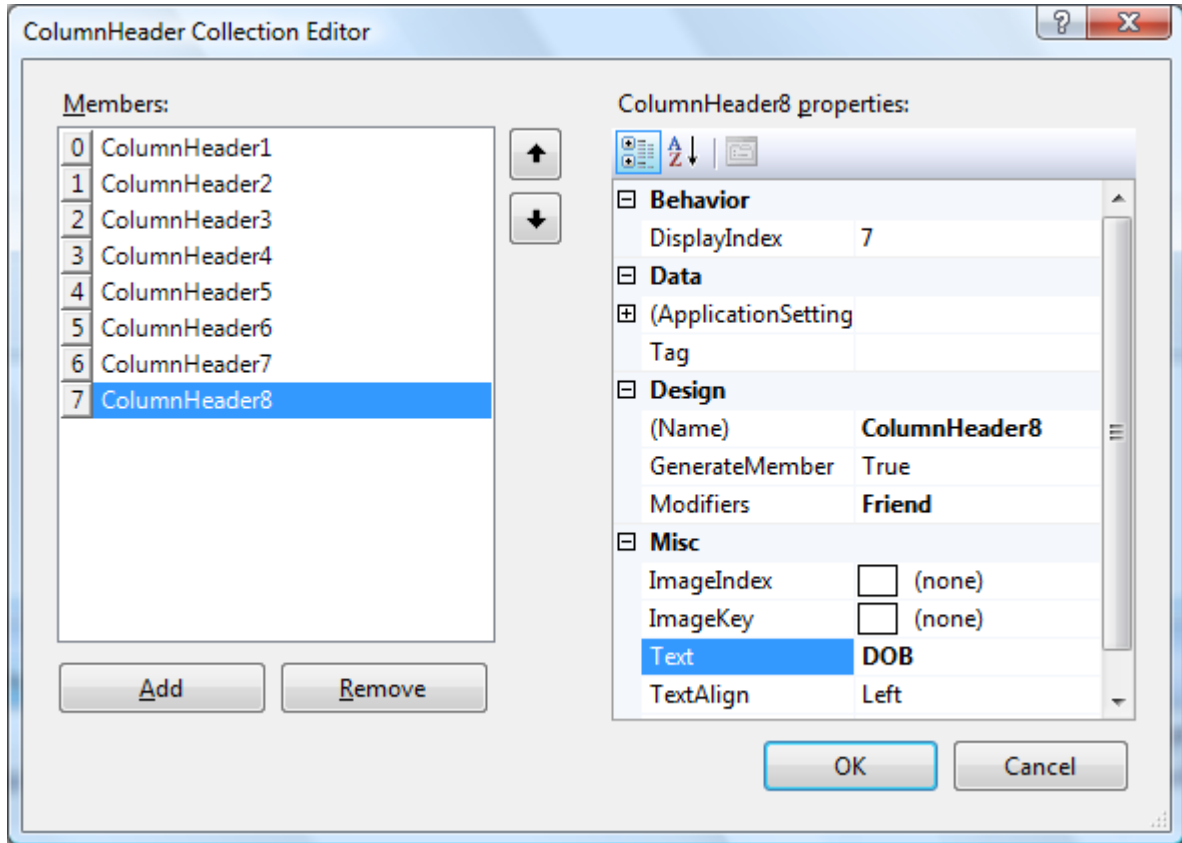
- GroupBox with Text property set to Customer Information.
- GroupBox with Text property set to Account Information.
- Label (Customer Number, Current Account Balance, Current Account Overdraft).
- TextBox 1 (Customer Number).
- TextBox 2 (Current Account Balance).
- TextBox 3 (Current Account Overdraft).
- 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 Rissaris_Bank_Demo.CustomerInformation
Imports Rissaris_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, Button1_Click, 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.

N.B. Please note the security values for username and password, which in our case, are root and "" respectively. Change to those for your MySQL .

The code is as follows:

VB 2005 Code

```
Dim customerInfoSecurity As Rissaris_Bank_Demo.CustomerInformation.Security
    customerInfoSecurity = New
Rissaris_Bank_Demo.CustomerInformation.Security()
    customerInfoSecurity.UsernameToken = New
CustomerInformation.SecurityUsernameToken()
    customerInfoSecurity.UsernameToken.Username = "root"
    customerInfoSecurity.UsernameToken.Password = ""

    Dim service As customerinformationRootService
    service = New customerinformationRootService()

    ' create a new key value that we send to the web service
    Dim key As
Rissaris_Bank_Demo.CustomerInformation.customerinformationGroupKeyType
    key = New
Rissaris_Bank_Demo.CustomerInformation.customerinformationGroupKeyType
    ' 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.customerinformationRootElementType

        ' call the "list" operation of web service!
        results = service.list(key)
        ListView1.Items.Clear()
        Dim customer As customerinformationGroupType
        For Each customer In results.customerinformationRoot()
            Dim lv As ListViewItem
            lv = New ListViewItem(customer.CustomerNumber)

            ' add the rest of the items in the row
            lv.SubItems.Add(customer.FirstName)
            lv.SubItems.Add(customer.Surname)
            lv.SubItems.Add(customer.AddressLine1)
            lv.SubItems.Add(customer.AddressLine2)
            lv.SubItems.Add(customer.City)
            lv.SubItems.Add(customer.Postcode)
            lv.SubItems.Add(customer.DateOfBirth)

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

```

VB 2008 Code

```

        Dim customerInfoSecurity As
Risaris_Bank_Demo.CustomerInformation.Security
        customerInfoSecurity = New
Risaris_Bank_Demo.CustomerInformation.Security()
        customerInfoSecurity.UsernameToken = New
CustomerInformation.SecurityUsernameToken()
        customerInfoSecurity.UsernameToken.Username = "root"
        customerInfoSecurity.UsernameToken.Password = ""

        Dim service As
CustomerInformation.customerinformationRootPortTypeClient
        service = New
CustomerInformation.customerinformationRootPortTypeClient()

        ' create a new key value that we send to the web service
        Dim key As
Risaris_Bank_Demo.CustomerInformation.customerinformationGroupKeyType
        key = New
Risaris_Bank_Demo.CustomerInformation.customerinformationGroupKeyType
        ' 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.customerinformationRootElementType

        ' call the "list" operation of web service!

```

```

        results = service.list(customerInfoSecurity, Nothing, key)
        ListView1.Items.Clear()
        For Each customerinformationGroupType In
results.customerinformationRoot()
            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 Code

```

If ListView1.SelectedIndices.Count = 1 Then
    ' create a new instance of the CurrentAccount web service
    Dim currentAccountSecurity As
Risaris_Bank_Demo.CurrentAccount.Security
    currentAccountSecurity = New
Risaris_Bank_Demo.CurrentAccount.Security()
    currentAccountSecurity.UsernameToken = New
CurrentAccount.SecurityUsernameToken()
    currentAccountSecurity.UsernameToken.Username = "root"
    currentAccountSecurity.UsernameToken.Password = ""
    Dim currentAccountService As New currentaccountRootService()

    Dim currentCustomerId As String
    ' get the currently 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.currentaccountGroupKeyType
    key = New
Risaris_Bank_Demo.CurrentAccount.currentaccountGroupKeyType

    key.AccountNumber = ""
    key.CustomerNumber = currentCustomerId.ToString

```

```

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

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

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

```

VB 2008 Code

```

    If ListView1.SelectedIndices.Count = 1 Then
        ' create a new instance of the CurrentAccount web service
        Dim currentAccountSecurity As
Risaris_Bank_Demo.CurrentAccount.Security
        currentAccountSecurity = New
Risaris_Bank_Demo.CurrentAccount.Security()
        currentAccountSecurity.UsernameToken = New
CurrentAccount.SecurityUsernameToken()
        currentAccountSecurity.UsernameToken.Username = "root"
        currentAccountSecurity.UsernameToken.Password = ""
        Dim currentAccountService As
CurrentAccount.currentaccountRootPortTypeClient
        currentAccountService = New
CurrentAccount.currentaccountRootPortTypeClient()

        Dim currentCustomerId As String
        ' get the currently 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.currentaccountGroupKeyType
        key = New
Risaris_Bank_Demo.CurrentAccount.currentaccountGroupKeyType

        key.AccountNumber = ""
        key.CustomerNumber = currentCustomerId.ToString

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

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

        ' Now put the results of the web service
        ' into the Balance and Overdraft text boxes
        TextBox2.Text =
FormatCurrency(results.currentaccountRoot(0).Balance / 100, 2)

```

```

        TextBox3.Text =
FormatCurrency(results.currentaccountRoot(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.

The screenshot shows a window titled "Risis Bank Demo" with a "Customer Information" section. It features a search box with an asterisk and a "Search" button. Below is a table with 10 rows of customer data. The "Account Information" section below the table contains a "Get Account Details" button and two empty text boxes for "Current Account Balance" and "Current Account Overdraft".

Customer Number	First Name	Surname	Address Line 1	Address Line 2	City	Post Co...	Date of Birth
1	Casper	Ankergren	34 Green Street	Crosses Street	Mansfield	MA5 9AJ	30/05/1978
2	Peter	Richardson	West Gate Aven...	Frankfield	Manchester	MA9 4AA	18/07/1948
3	Donald	Smith	32 Parsons Gree...	Fulham	London	SW4 3JA	05/09/1988
4	Lillian	Marques	1 Father Delahu...	Bordesley	Bimminham	BO4 4LW	14/01/1979
5	Arthur	Holman	12 Padbury Close	Griffin Close	Newcastle upon...	NW8 2KK	16/05/1975
6	Bartholomew	McCarthy	22 Hillview	Griffin Close	York	SW9 1...	24/03/1974
7	Neville	Lee	72 Thoresby Cre...	Sinfin	Leeds	LS9 2DS	21/03/1980
8	Brian	Colman	14 Gilliver Garde...	West Field	Birmingham	BA9 3KK	20/04/1983
9	Ian	Burton	144 Allestree Lane	Lime Kiln Lane	York	YR0 2NT	06/09/1971
10	Grenville	Dekker	22 Uttotter New...	Chaddesden	Leeds	LS4 9WB	21/10/1974

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:

The screenshot shows the same application window, but now the 4th row of the customer list is highlighted in blue. The "Account Information" section shows the "Get Account Details" button and the text boxes are now populated with values: "Current Account Balance" is £1,829.68 and "Current Account Overdraft" is £1,000.00.

Customer Number	First Name	Surname	Address Line 1	Address Line 2	City	Post Co...	Date of Birth
1	Casper	Ankergren	34 Green Street	Crosses Street	Mansfield	MA5 9AJ	30/05/1978
2	Peter	Richardson	West Gate Aven...	Frankfield	Manchester	MA9 4AA	18/07/1948
3	Donald	Smith	32 Parsons Gree...	Fulham	London	SW4 3JA	05/09/1988
4	Lillian	Marques	1 Father Delahu...	Bordesley	Bimminham	BO4 4LW	14/01/1979
5	Arthur	Holman	12 Padbury Close	Griffin Close	Newcastle upon...	NW8 2KK	16/05/1975
6	Bartholomew	McCarthy	22 Hillview	Griffin Close	York	SW9 1...	24/03/1974
7	Neville	Lee	72 Thoresby Cre...	Sinfin	Leeds	LS9 2DS	21/03/1980
8	Brian	Colman	14 Gilliver Garde...	West Field	Birmingham	BA9 3KK	20/04/1983
9	Ian	Burton	144 Allestree Lane	Lime Kiln Lane	York	YR0 2NT	06/09/1971
10	Grenville	Dekker	22 Uttotter New...	Chaddesden	Leeds	LS4 9WB	21/10/1974

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 MySQL 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 for VB 2005

[Code available here](#)

Form1.vb for VB 2008

[Code available here](#)