Home | SQL Server | ASP.Net | LINQ | Excel | Web Services & iOS
XCode | Other | Contact

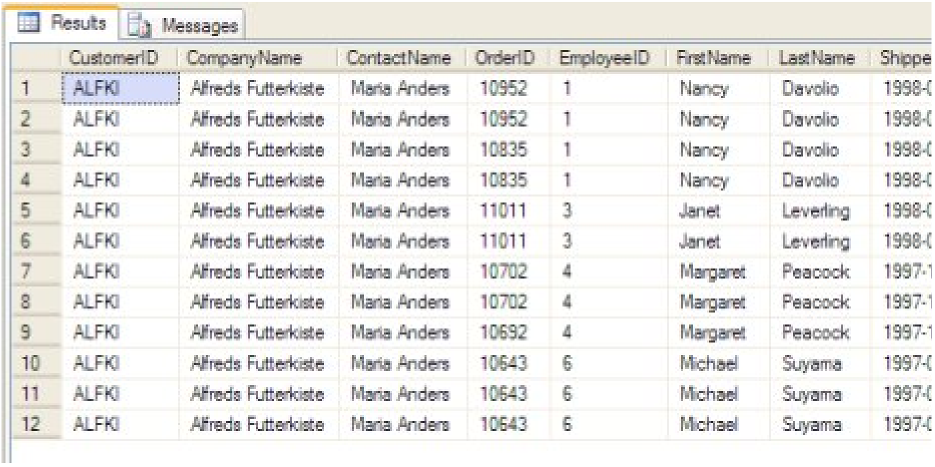
olutions for software developers.

Running a *SQL Server* stored procedure from Excel

By Mike Gledhill

In this walkthrough, we'll see how to call a Stored Procedure directly from Excel.

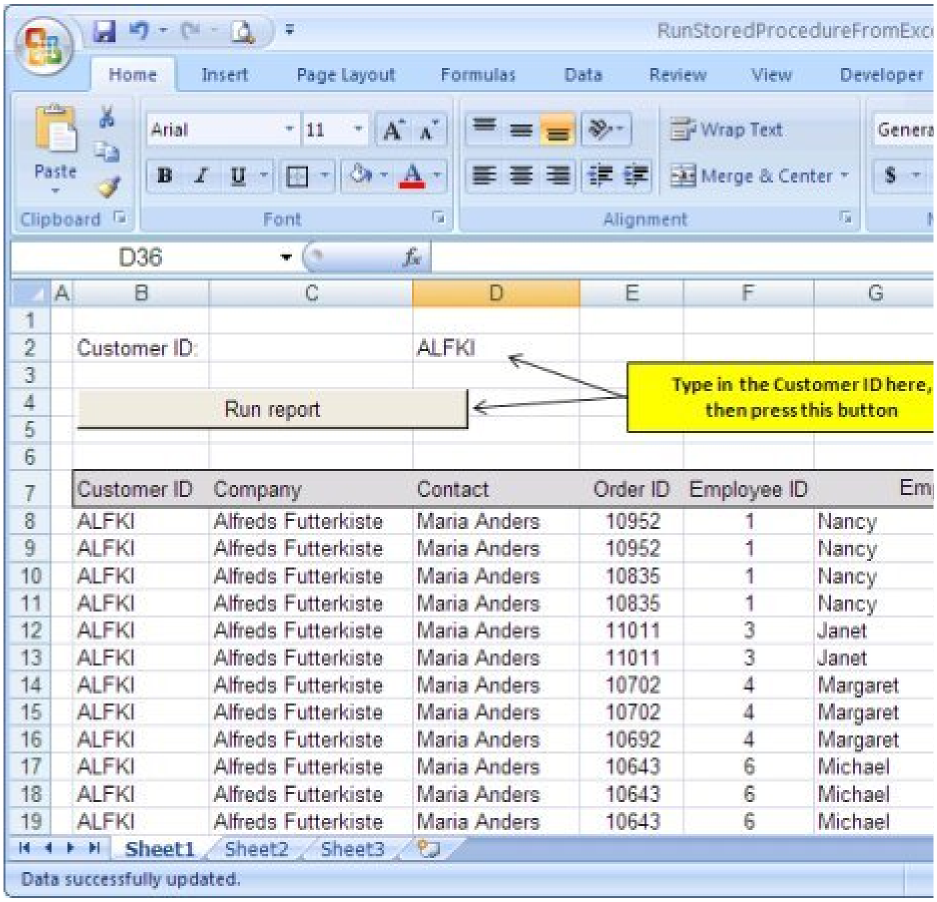
We'll add a Stored Procedure to Microsoft's sample *Northwind* database, get Excel to call it, passing it a parameter, then we'll display its results in the Excel worksheet. Here's what the stored procedure's results look like:



Results | Messages

	CustomerID	CompanyName	ContactName	OrderID	EmployeeID	FirstName	LastName	Shippe
1	ALFKI	Alfreds Futterkiste	Maria Anders	10952	1	Nancy	Davolio	1998-
2	ALFKI	Alfreds Futterkiste	Maria Anders	10952	1	Nancy	Davolio	1998-
3	ALFKI	Alfreds Futterkiste	Maria Anders	10835	1	Nancy	Davolio	1998-
4	ALFKI	Alfreds Futterkiste	Maria Anders	10835	1	Nancy	Davolio	1998-
5	ALFKI	Alfreds Futterkiste	Maria Anders	11011	3	Janet	Leverling	1998-
6	ALFKI	Alfreds Futterkiste	Maria Anders	11011	3	Janet	Leverling	1998-
7	ALFKI	Alfreds Futterkiste	Maria Anders	10702	4	Margaret	Peacock	1997-1
8	ALFKI	Alfreds Futterkiste	Maria Anders	10702	4	Margaret	Peacock	1997-1
9	ALFKI	Alfreds Futterkiste	Maria Anders	10692	4	Margaret	Peacock	1997-1
10	ALFKI	Alfreds Futterkiste	Maria Anders	10643	6	Michael	Suyama	1997-
11	ALFKI	Alfreds Futterkiste	Maria Anders	10643	6	Michael	Suyama	1997-
12	ALFKI	Alfreds Futterkiste	Maria Anders	10643	6	Michael	Suyama	1997-

.. and this is how the users will see it in Excel (after manually adding a header row, and some formating on the figures):



Customer ID: ALFKI

Type in the Customer ID here, then press this button

Customer ID	Company	Contact	Order ID	Employee ID	Em
ALFKI	Alfreds Futterkiste	Maria Anders	10952	1	Nancy
ALFKI	Alfreds Futterkiste	Maria Anders	10952	1	Nancy
ALFKI	Alfreds Futterkiste	Maria Anders	10835	1	Nancy
ALFKI	Alfreds Futterkiste	Maria Anders	10835	1	Nancy
ALFKI	Alfreds Futterkiste	Maria Anders	11011	3	Janet
ALFKI	Alfreds Futterkiste	Maria Anders	11011	3	Janet
ALFKI	Alfreds Futterkiste	Maria Anders	10702	4	Margaret
ALFKI	Alfreds Futterkiste	Maria Anders	10702	4	Margaret
ALFKI	Alfreds Futterkiste	Maria Anders	10692	4	Margaret
ALFKI	Alfreds Futterkiste	Maria Anders	10643	6	Michael
ALFKI	Alfreds Futterkiste	Maria Anders	10643	6	Michael
ALFKI	Alfreds Futterkiste	Maria Anders	10643	6	Michael

Data successfully updated.

Running a Stored Procedure from Excel 2007

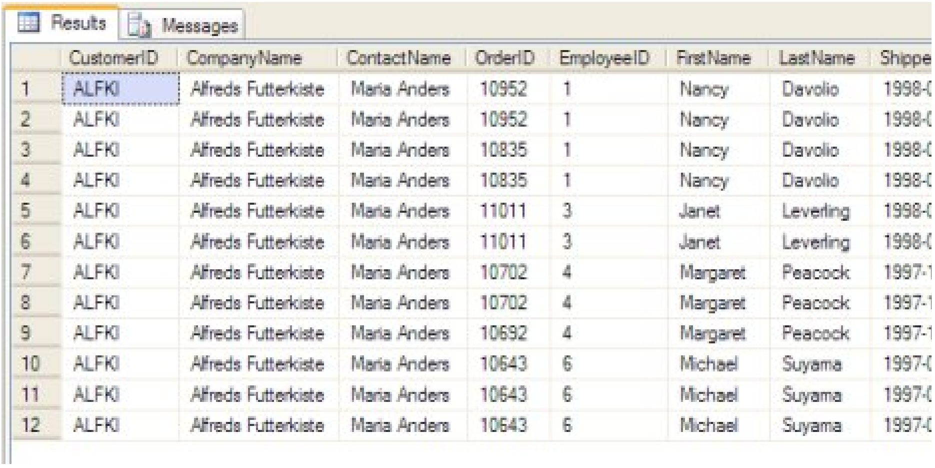
First, let's create a simple Stored Procedure in the *Northwind* database, which takes a parameter (Customer ID) and returns some details of orders that this customer has placed.

```
CREATE PROCEDURE [dbo].[SP_GetOrdersForCustomer]
  @CustomerID nchar(5)
AS
BEGIN
  SELECT cst.[CustomerID],
     cst.[CompanyName],
     cst.[ContactName],
     ord.[OrderID],
     ord.[EmployeeID],
     emp.[FirstName],
     emp.[LastName],
     ord.[ShippedDate],
     prd.[ProductName],
     od.[UnitPrice],
     od.[Quantity]
  FROM [Customers] cst,
    [Orders] ord,
    [Order Details] od,
    [Employees] emp,
    [Products] prd
  WHERE ord.[CustomerID] = cst.[CustomerID]
  AND emp.[EmployeeID] = ord.[EmployeeID]
  AND od.[OrderID] = ord.[OrderID]
```

```
        AND prd.[ProductID] = od.[ProductID]
        AND cst.[CustomerID] = @CustomerID
        ORDER BY cst.[CustomerID], emp.[EmployeeID], ord.[ShippedDa
    END
```

In *SQL Server Management Studio* if I run this Stored Procedure using the following command, it returns the results that we saw earlier:

```
    exec [dbo].[SP_GetOrdersForCustomer] 'ALFKI'
```



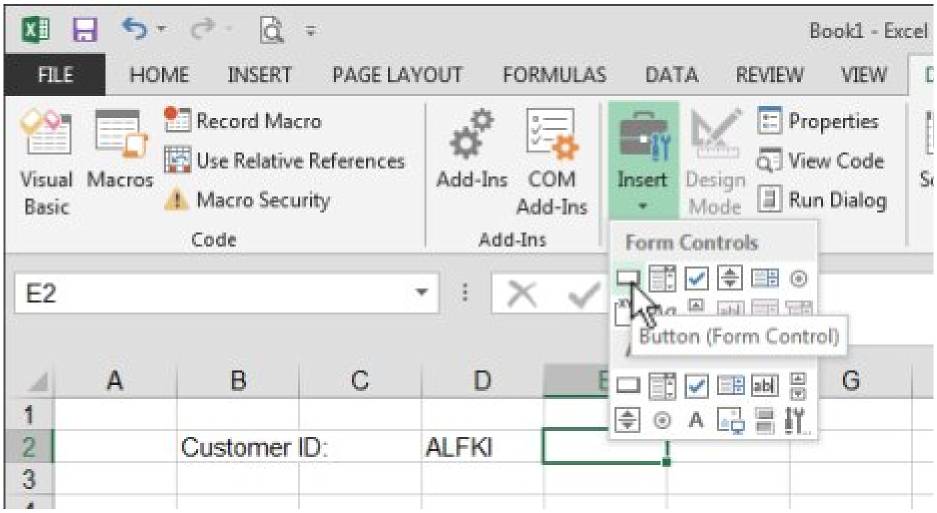
	CustomerID	CompanyName	ContactName	OrderID	EmployeeID	FirstName	LastName	Shippe
1	ALFKI	Alfreds Futterkiste	Maria Anders	10952	1	Nancy	Davolio	1998-0
2	ALFKI	Alfreds Futterkiste	Maria Anders	10952	1	Nancy	Davolio	1998-0
3	ALFKI	Alfreds Futterkiste	Maria Anders	10835	1	Nancy	Davolio	1998-0
4	ALFKI	Alfreds Futterkiste	Maria Anders	10835	1	Nancy	Davolio	1998-0
5	ALFKI	Alfreds Futterkiste	Maria Anders	11011	3	Janet	Leverling	1998-0
6	ALFKI	Alfreds Futterkiste	Maria Anders	11011	3	Janet	Leverling	1998-0
7	ALFKI	Alfreds Futterkiste	Maria Anders	10702	4	Margaret	Peacock	1997-1
8	ALFKI	Alfreds Futterkiste	Maria Anders	10702	4	Margaret	Peacock	1997-1
9	ALFKI	Alfreds Futterkiste	Maria Anders	10692	4	Margaret	Peacock	1997-1
10	ALFKI	Alfreds Futterkiste	Maria Anders	10643	6	Michael	Suyama	1997-0
11	ALFKI	Alfreds Futterkiste	Maria Anders	10643	6	Michael	Suyama	1997-0
12	ALFKI	Alfreds Futterkiste	Maria Anders	10643	6	Michael	Suyama	1997-0

But, how do we get Excel to run this Stored Procedure ?

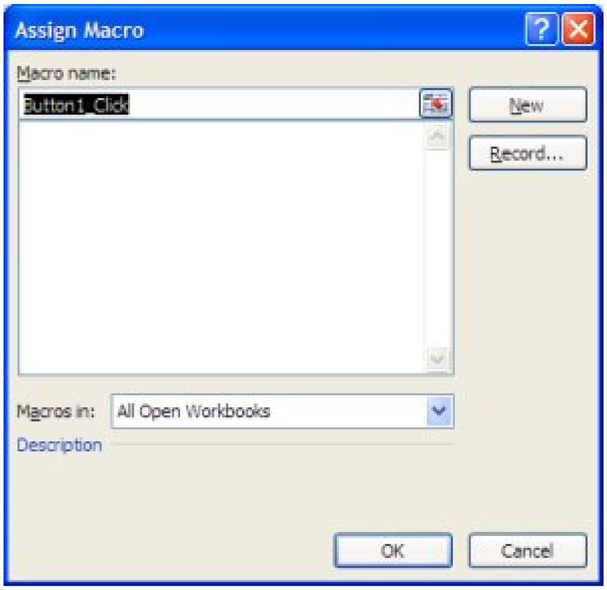
First, run Excel, and create a blank workbook.

Now, in cell B2, type in "Customer ID:", and in cell D2, type in "ALFKI" (both without the quotes). ALFKI is the customer code of one of Northwind's example customers. We will get Excel to read in the value of cell D2 and pass it to the Stored Procedure as the *CustomerID* parameter.

Next, we need to add a Button to our spreadsheet. Click on the **Developer** tab, then the **Insert** button, then click on the icon which looks like a button.



Now, hold down the left mouse button, and drag the outline of a button, somewhere around row 4 below our "Customer ID" label.
When you *release* the left mouse button, you'll see this dialog.

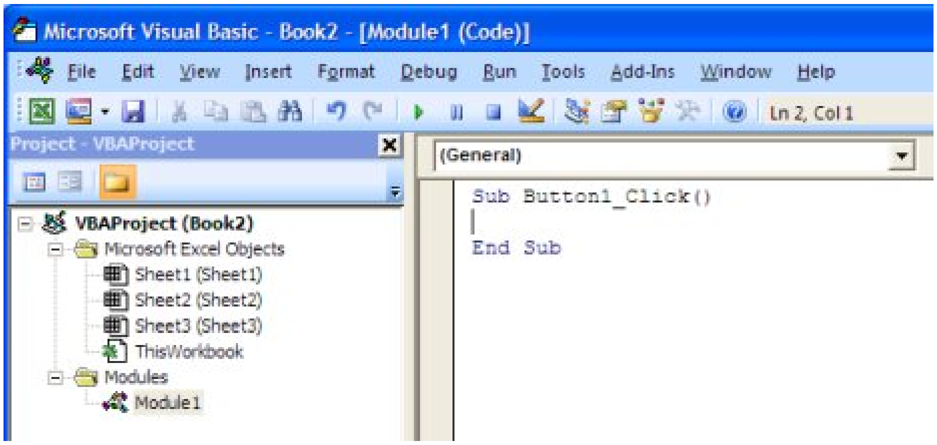


Macro name: Button1_Click

Macros in: All Open Workbooks

Description

Click on the **New** button. You are then taken into Excel's beloved VBA scripting window.



```
Sub Button1_Click()

End Sub
```

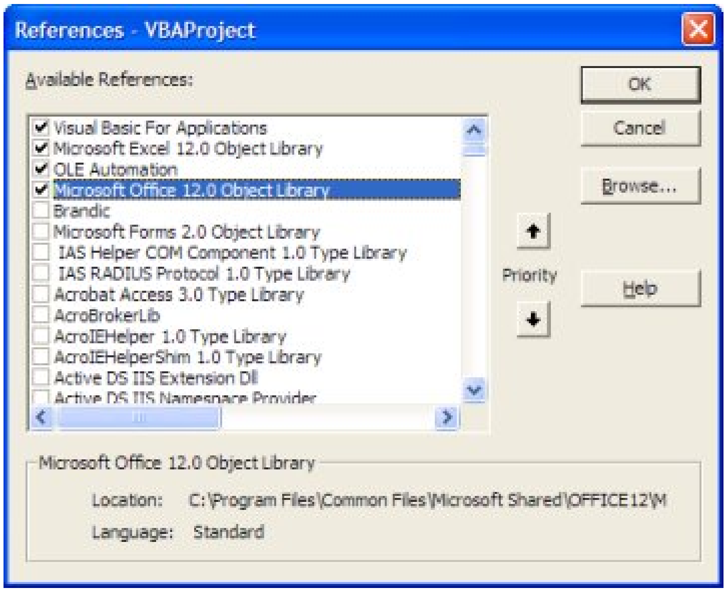
Paste the following code between the *Sub Button1_Click* and *End Sub* lines.

```
    Dim con As ADODB.Connection
    Dim cmd As ADODB.Command
    Dim rs As ADODB.Recordset
    Dim WSP1 As Worksheet
    Set con = New ADODB.Connection
    Set cmd = New ADODB.Command
    Set rs = New ADODB.Recordset

    Application.DisplayStatusBar = True
    Application.StatusBar = "Contacting SQL Server..."

    ' Remove any values in the cells where we want to put our Sto
    Dim rngRange As Range
    Set rngRange = Range(Cells(8, 2), Cells(Rows.Count, 1)).Entir
    rngRange.ClearContents

    ' Log into our SQL Server, and run the Stored Procedure
    con.Open "Provider=SQLOLEDB;Data Source=localhost;Initial Cat
    cmd.ActiveConnection = con

    ' Set up the parameter for our Stored Procedure
    ' (Parameter types can be adVarChar,adDate,adInteger)
    cmd.Parameters.Append cmd.CreateParameter("CustomerID", adVar
    Application.StatusBar = "Running stored procedure..."
    cmd.CommandText = "SP_GetOrdersForCustomer"
    Set rs = cmd.Execute(, , adCmdStoredProc)

    ' Copy the results to cell B7 on the first Worksheet
    Set WSP1 = Worksheets(1)
    WSP1.Activate
    If rs.EOF = False Then WSP1.Cells(8, 2).CopyFromRecordset rs

    rs.Close
    Set rs = Nothing
    Set cmd = Nothing

    con.Close
    Set con = Nothing

    Application.StatusBar = "Data successfully updated."
```

This code does three things.

- Clears the "results area" of our worksheet, namely from cell B8 to the bottom of our worksheet.
- Creates a connection to our database, and calls our "*SP_GetOrdersForCustomer*" Stored Procedure.
- Pastes the results into our worksheet, from cell B8.

This script uses the *Microsoft ADO* libraries to connect to SQL Server, so we need to tell Excel to include these libraries.
To do this, click on **Tools** then **References**.



References - VBAProject

Available References:

- ☑ Visual Basic For Applications
- ☑ Microsoft Excel 12.0 Object Library
- ☑ OLE Automation
- ☑ Microsoft Office 12.0 Object Library
- □ Brandic
- □ Microsoft Forms 2.0 Object Library
- □ IAS Helper COM Component 1.0 Type Library
- □ IAS RADIUS Protocol 1.0 Type Library
- □ Acrobat Access 3.0 Type Library
- □ AcroBrokerLib
- □ AcroIEHelper 1.0 Type Library
- □ AcroIEHelperShim 1.0 Type Library
- □ Active DS IIS Extension Dll
- □ Active DS IIS Namespace Provider

Microsoft Office 12.0 Object Library

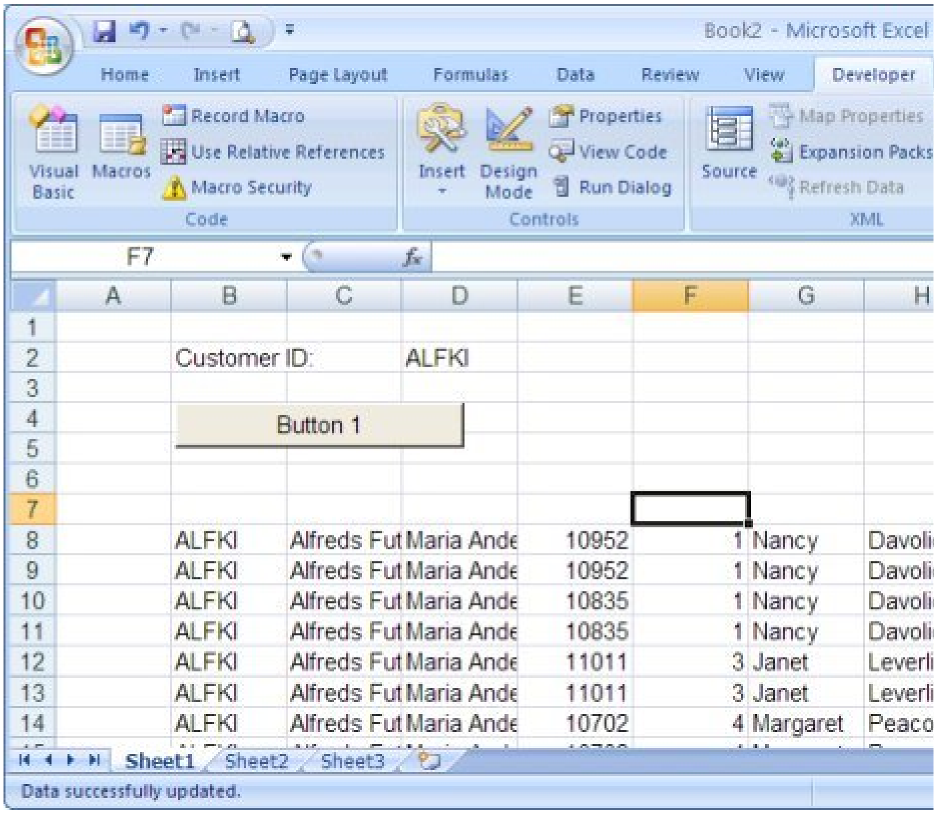
Location: C:\Program Files\Common Files\Microsoft Shared\OFFICE12\M

Language: Standard

Scroll down until you find the item "Microsoft ActiveX Data Objects 2.8 Libary" (or whichever version is most recent on your PC), tick the checkbox next to it, then click on the **OK** button.

You can now click on **File**, then **"Close and return to Microsoft Excel"**.

If all goes well, now, when you click on your button, Excel will run the Stored Procedure, and display the results in cell B8 onwards:



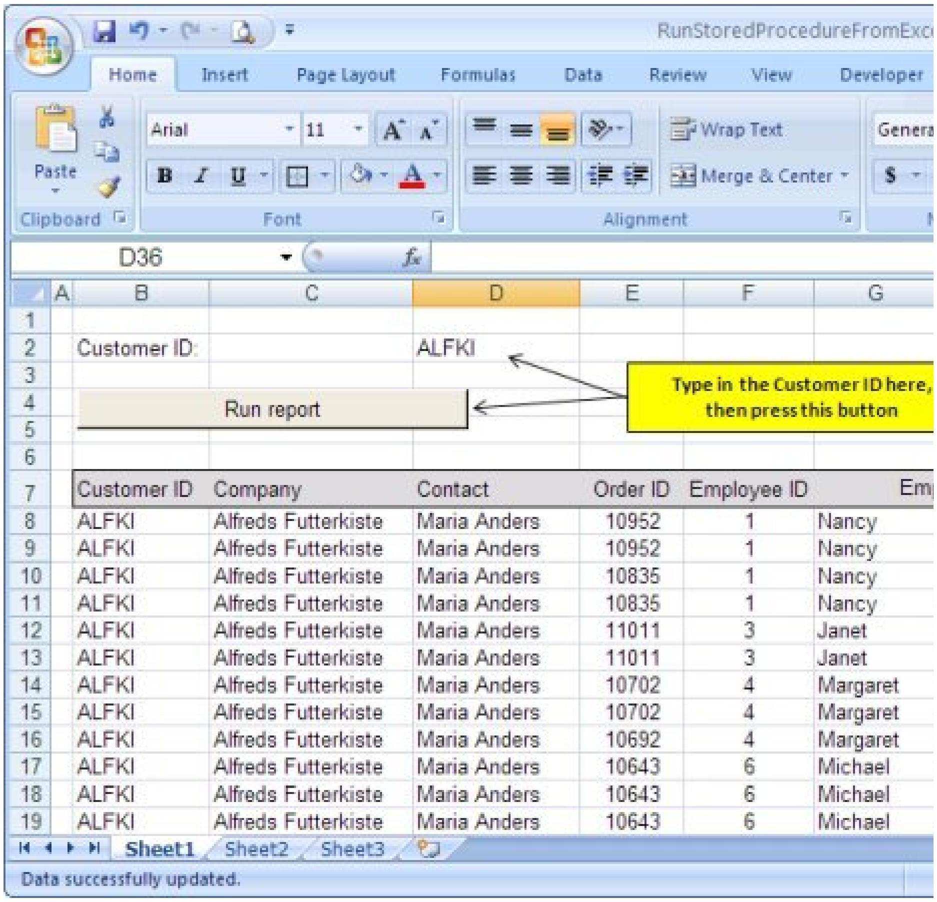
	A	B	C	D	E	F	G	H
1								
2		Customer ID:		ALFKI				
3								
4								
5								
6								
7								
8		ALFKI	Alfreds Fut	Maria Ande	10952	1	Nancy	Davoli
9		ALFKI	Alfreds Fut	Maria Ande	10952	1	Nancy	Davoli
10		ALFKI	Alfreds Fut	Maria Ande	10835	1	Nancy	Davoli
11		ALFKI	Alfreds Fut	Maria Ande	10835	1	Nancy	Davoli
12		ALFKI	Alfreds Fut	Maria Ande	11011	3	Janet	Leverli
13		ALFKI	Alfreds Fut	Maria Ande	11011	3	Janet	Leverli
14		ALFKI	Alfreds Fut	Maria Ande	10702	4	Margaret	Peaco

Data successfully updated.

How cool is that !

You can now add formatting and a header row to the worksheet to make it look more professional. You could start by giving "Button 1" a friendlier caption, by right-clicking on the button, select **Edit text**, and type in a friendlier caption.

With a bit of TLC, you can make your report look professional:



Customer ID: ALFKI

Run report

Type in the Customer ID here, then press this button

Customer ID	Company	Contact	Order ID	Employee ID	Em
ALFKI	Alfreds Futterkiste	Maria Anders	10952	1	Nancy
ALFKI	Alfreds Futterkiste	Maria Anders	10952	1	Nancy
ALFKI	Alfreds Futterkiste	Maria Anders	10835	1	Nancy
ALFKI	Alfreds Futterkiste	Maria Anders	10835	1	Nancy
ALFKI	Alfreds Futterkiste	Maria Anders	11011	3	Janet
ALFKI	Alfreds Futterkiste	Maria Anders	11011	3	Janet
ALFKI	Alfreds Futterkiste	Maria Anders	10702	4	Margaret
ALFKI	Alfreds Futterkiste	Maria Anders	10702	4	Margaret
ALFKI	Alfreds Futterkiste	Maria Anders	10692	4	Margaret
ALFKI	Alfreds Futterkiste	Maria Anders	10643	6	Michael
ALFKI	Alfreds Futterkiste	Maria Anders	10643	6	Michael
ALFKI	Alfreds Futterkiste	Maria Anders	10643	6	Michael

Data successfully updated.

RunStoredProcedureFromExcel.xlsm

When you Save this Excel file, remember to save it as a "Excel Macro-Enabled Workbook (*.xlsm)" file, as it contains Excel VBA script.

Closing thoughts

So, do I recommend that you throw away your Silverlight and WPF books, and start writing applications using Excel and Stored Procedures from now on ?

Of course not. But I have found this trick particularly useful with major applications which have been tested and deployed to live environments, but then the user wants to see a *little extra* information from the database.

I don't really want to add extra screens to my application, then suffer having to go through the testing and deployment phase again.

Giving the users one-off reports, which they can run themselves like this, gives me a "quick win", without needing to make changes to my original app.

The other big advantage is that *users love Excel* (I'm sorry... but you know it's true).
And this gives us a really easy way to get raw data from *SQL Server* directly into Excel, ready for them to play with.

Useful links

Sample "Northwind" database for SQL Server 2008
http://msdn.microsoft.com/en-us/library/ms143221.aspx

Sample "Northwind" database for SQL Server 2000/2005
http://www.microsoft.com/download/en/details.aspx?id=23654
This installs the .mdf and .ldf database files into the directory:
 C:\SQL Server 2000 Sample Databases

Comments

Comments Community 1 Login ▾

♥ Recommend 3 Share Sort by Best ▾

Join the discussion…

RSP • a month ago
Thanks Mike !!!, its very useful.
⌃ | ⌄ • Reply • Share ›

Edgardo M Salinas L • 3 months ago
Thanks a lot. Well explained.
⌃ | ⌄ • Reply • Share ›

Rajeshwari Uniyal • 8 months ago
hi please tell me code of vba to execute stored procedure of oracle in which i have 3 parameter input andwhich vl be on excel cell and i want to execute the same please its urgent
⌃ | ⌄ • Reply • Share ›

Andres Avila • 10 months ago
Perfect fit to what I needed, you save me a lot of time! Thanks a lot.
⌃ | ⌄ • Reply • Share ›

Diian Indrani • a year ago
is it possible to execute 2 stored procedures in one sheet?
⌃ | ⌄ • Reply • Share ›

Mike Gledhill ➦ Diian Indrani • a year ago
Yes, you can execute as many as you want.
⌃ | ⌄ • Reply • Share ›

Kevin • a year ago

Great example, but the following line:

Dim prmCustomerID As ADODB.Parameter

is unnecessary.

• Reply • Share ›

Mike Gledhill ➦ Kevin • a year ago

It is, sorry. I need to remove this line !

• Reply • Share ›

Sweety • a year ago

i want parameters to be pass to sql query also..so that i can use append the parameter to my sql query and excute

• Reply • Share ›

Mike Gledhill ➦ Sweety • a year ago

Then simply build a string, containing your SQL and the parameter values, and Execute it. The very first Google match in the link I sent you shows you how to run a piece of SQL, and all you need to do is to create the SQL string yourself.

• Reply • Share ›

Sweety • a year ago

Dear sir, is it possible to create vba code using sql query same as above instead of using stored procedure..

• Reply • Share ›

Mike Gledhill ➦ Sweety • a year ago

Yes, this is quite possible. Just follow the steps here: http://bfy.tw/84V

• Reply • Share ›

Graeme Anderson • a year ago

Just ensure the values you pass are valid i.e. numbers do not have quotes around the value

• Reply • Share ›

Graeme Anderson • a year ago

Hi Jorge,

to add extra parameters is easy:

1st Parameter
cmd.Parameters.Append cmd.CreateParameter("StateId", adInteger, adParamInput, 10, 5)

2nd Parameter
cmd.Parameters.Append cmd.CreateParameter("ClientCode", adVarChar, adParamInput, 10, "M7243")

⌃ | ⌄ • Reply • Share ›

jorge jimenez • a year ago

A question, how do you put two input parameters? I I can't put more than one. When Im compiling always debug in the second one and says@Arguments are of the wrong type, are out of acceptable range, or are in conflict with another

⌃ | ⌄ • Reply • Share ›

jorge jimenez • a year ago

Really good ;) , you deserve a high five!

⌃ | ⌄ • Reply • Share ›

Mike Gledhill ➦ jorge jimenez • a year ago

Thanks ! If this helped you, please consider clicking on the Donate button at the top of the screen.
100% of all donations made on my website get forwarded to www.water.org (and I'll send you the receipt).
Merci !