Message Boxes

We use message boxes extensively throughout our applications so here are a couple of ways to make them better.

You can personalise your message boxes to the application name and the user name.

In this example the windows user name is displayed in the message and the title is the application's name –

```
strMessage = "Hi " & GetWindowsUserName() & _
    & ", are you sure you want to delete this record?"
If MsgBox(strMessage, vbYesNo, _
    CurrentDb.Properties("AppTitle")) = vbYes Then
    <Do stuff here>
end if
```

Personally I never delete records as it can cause all sorts of problems with orphaned records in other tables. I prefer to mark them as inactive and exclude then in the recordset for the form or report. That way you can recover them.

But if a client insists, then I find that people take more care with deleting records when they think that the application knows who it is that is deleting the record.

The GetWindowsUserName function is in a module with this code –

```
Option Compare Database
Option Explicit
' Access the GetUserNameA function in advapi32.dll and
' call the function GetUserName.
Declare Function GetUserName Lib "advapi32.dll" _
    Alias "GetUserNameA" (ByVal lpBuffer As String, _
    nSize As Long) As Long

Public Function GetWindowsUserName() As String
Dim lpBuff As String * 25
Dim ret As Long, UserName As String
' Get the user name minus any trailing spaces
ret = GetUserName(lpBuff, 25)
GetWindowsUserName = Left(lpBuff, InStr(lpBuff, _
          Chr(0)) - 1)
End Function
```

I didn't write this code but I use it a lot. This function is great for recording who edited records as well. You cannot of course record the name of the person who deleted the record in the record when you delete it. You would have to use a separate audit table. Another reason I use the inactive flag with a Modified_By and Modified_Date fields.

You can also see that VBA has intrinsic constants to format the message box buttons, in this case vbYesNo gives you a Yes and a No button.

There are plenty of other button configurations and they all self explanatory. You can select them from the autofill dropdown when you write the code. It is then easy to check the result with vbYes, vbNo or vbCancel and this also makes the code easy to read.

For more tips or to get help with Your Access Database go to Osel Access Development. Free Access Tips

For Help with your Access Database Osel Access Consulting