

Simple ASP Demonstration Application – Data Object Version

Quickstart Guide

This Guide is designed to help you build your own ASP database applications using Westfaro Corporation's ASP Scripts as the starting point.

This version is functionally equivalent to the **Demo ASP 01** project. The only difference is that database insertion, updating and deletion is handled by a class for each database table. By default, the class uses ADO recordsets to perform these operations. You can have the script generate SQL code instead by setting the script parameter Constant01 to the value SQL (see Figure 1).

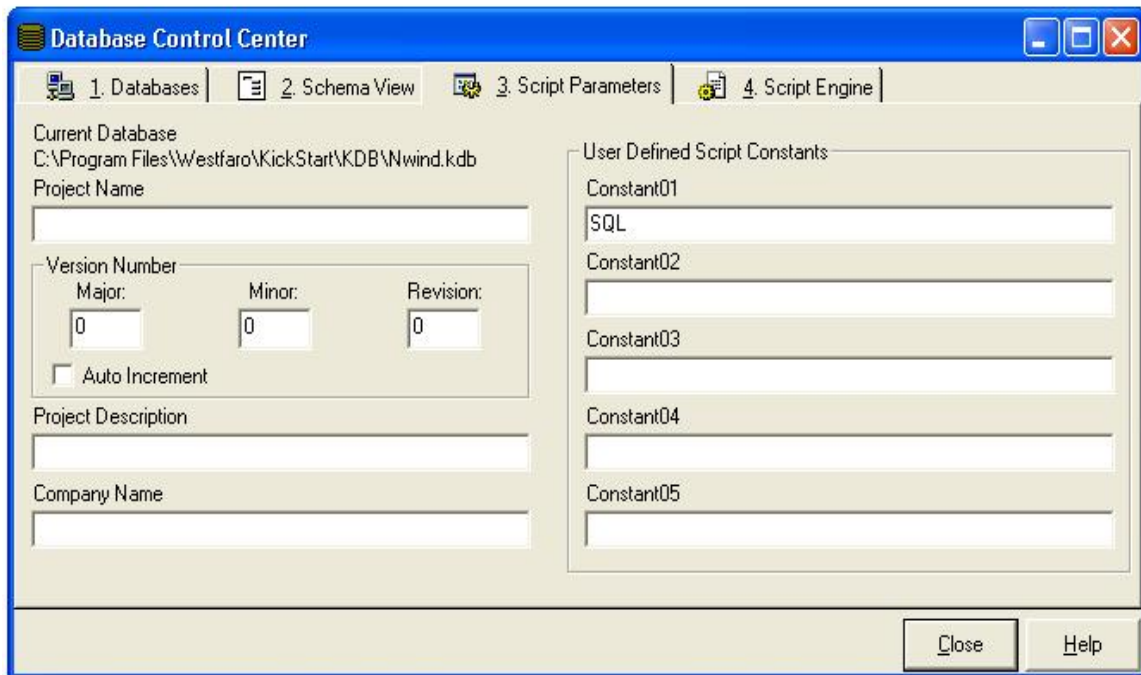


Figure 1 - Script Parameter Setting

A preview

To see what you can generate, navigate to:

<http://www.westfaro.com/software/kickstart/demo/asp/DemoASP01/default.asp>

This will show a page like Figure 2 with two menus. The body of the page is blank but you could include typical Home page information. This demo uses the Northwind demonstration database shipped with Microsoft Access so the LHS menu items list the tables from that Database.



Figure 2 - Default Page

Click on the Products link. This will show a page with a Search form for Products. At the bottom is a Key that explains the Icons used by the application. Enter **C*** as the **Productname** and **<30** as the **Unitprice** and then click the search icon. The search operation will return a page showing the products that satisfy the search conditions; Products names that start with C and cost less than 30. If the database has not been updated then the page will look like that shown in Figure 3.

The screenshot shows the "Maintain Products Table" page. At the top center, the title "Maintain Products Table" is displayed in a large blue font. Below it, there are two blue buttons labeled "Home" and "Westfaro". On the left side, there is a vertical menu with nine blue buttons: "Home", "Categories", "Customers", "Employees", "OrderDetails", "Orders", "Products", "Shippers", and "Suppliers".

The main content area features a search form with the following fields:

Action	Productname	Quantityperunit	Unitprice	Unitsinstock	Unitsonorder	Reorderlev
	C*		<30	0	0	0
	Chai	10 boxes x 20 bags	18	39	0	
	Chang	24 - 12 oz bottles	19	17	40	
	Chef Anton's Cajun Seasoning	48 - 6 oz jars	22	53	0	
	Chartreuse verte	750 cc per bottle	18	69	0	
	Chocolade	10 pkgs.	12.75	15	70	

Below the table is a "Key" section with the following information:

Key

- Search based on criteria. % or * are wild-cards. Fields may be preceded by <=, >=, <>, <, > and =.
- Insert a new row
- View current row
- Edit current row
- Delete current row

Figure 3 - Results of Products search

To see a detailed view for a particular item, click on the View icon in the action column. This will display a page like the one shown in Figure 4.

The screenshot shows a web application titled "Maintain Products Table". At the top, there are two buttons: "Home" and "Westfaro". Below this is a navigation menu on the left with buttons for "Home", "Categories", "Customers", "Employees", "OrderDetails", "Orders", "Products", "Shippers", and "Suppliers". The main content area is titled "View Products" and displays a form for editing a product record. The form fields are as follows:

View Products	
Productname	<input type="text" value="Chef Anton's Cajur"/>
Quantityperunit	<input type="text" value="48 - 6 oz jars"/>
Unitprice	<input type="text" value="22"/>
Unitsinstock	<input type="text" value="53"/>
Unitsonorder	<input type="text" value="0"/>
Reorderlevel	<input type="text" value="0"/>
Discontinued	<input type="checkbox"/>
CategoryID	<input type="text" value="2"/>
SupplierID	<input type="text" value="2"/>

At the bottom of the form is a "Back to list" button.

Figure 4 - Detail view of Product record 1

Creating the application

1. Download the SimpleASP01.zip file and extract the scripts (.ksc extension).
2. Connect to the Northwind database using Kickstart.
3. Open up the Utility.KSC script and change the database connection constants as required for your database type.
4. Select the ASP Script IncludeAll.ksc
5. Select a destination folder on your web server
6. Click the Generate button
7. Extract the Images folder from SimpleASP01.zip and create an Images subfolder in the destination folder to hold the images.
8. Navigate to Default.ASP using your browser
9. Repeat for your own databases

Next steps

The raw generated code could obviously do with some cleaning up. The biggest problem is the way it displays foreign keys (lookups). Since the Database doesn't know which columns in the **Categories** and

Suppliers tables should be displayed to the user, Kickstart uses the referenced column by default. However, you can use Kickstart to choose which columns should be displayed in a foreign key relationship. For **Categories**, you might choose **Categoryname** or **Description** or both.

A second problem is that the application shows every column in the Search form, the search results and the detailed view. In the first two cases, you will probably want to reduce the number of columns to those most useful for searching.

A third problem is that labels are taken directly from Database column names. You will probably want to change a few of them. If you use the convention of using meaningful column names with the component words separated by underscores, then Kickstart will display them better.

Customizing the generated code

Some application generators attempt to generate working applications but they are often difficult to customize. Others generate source code only a computer could like.

The thinking behind Kickstart is different. Lots of database and application code is repetitive. Programmers develop a template that works for one table and then cut and paste that code for the next table. Kickstart automates that grunt work. Write your best code for one table, convert that code into a Kickstart script and you've written the code for every table. When you have a large database with dozens or hundreds of tables, being able to replicate your best database code at the touch of a button is awesome.

Appendix 1 shows the ASP code that Kickstart created for one database table using the demonstration ASP script:

Appendix 1 – ASP Dataobject created using ASP script – SQL Version

```
<%
Class clsProducts
' Kickstart Generated
' Version: 0.0.0
Private m_cConnection
'
' Primary keys
Private m_lngProductID
'
' Primary key copies. Used when updating primary key values
Private m_lngProductIDCOPY
'
' Data columns
Private m_strProductName
Private m_lngSupplierID
Private m_lngCategoryID
Private m_strQuantityPerUnit
Private m_curUnitPrice
Private m_intUnitsInStock
Private m_intUnitsOnOrder
Private m_intReorderLevel
Private m_blnDiscontinued
'
' Concurrency checking fields

Private m_strSourceURL
Private m_intCurrentPage

Public Property Set Connection(ByVal cConn)
    Set m_cConnection = cConn
End Property

Public Property Get DBTableName()
    DBTableName = "[Products]"
End Property

Public Property Let ProductID(ByVal lngProductID)
    m_lngProductID = lngProductID
End Property
Public Property Get ProductID()
    ProductID = m_lngProductID
End Property

Public Property Let ProductName(ByVal strProductName)
    m_strProductName = strProductName
End Property
Public Property Get ProductName()
    ProductName = m_strProductName
End Property

Public Property Let SupplierID(ByVal lngSupplierID)
    m_lngSupplierID = lngSupplierID
End Property
Public Property Get SupplierID()
    SupplierID = m_lngSupplierID
End Property

Public Property Let CategoryID(ByVal lngCategoryID)
    m_lngCategoryID = lngCategoryID
End Property
Public Property Get CategoryID()
    CategoryID = m_lngCategoryID
End Property

Public Property Let QuantityPerUnit(ByVal strQuantityPerUnit)
    m_strQuantityPerUnit = strQuantityPerUnit
End Property
Public Property Get QuantityPerUnit()
```

```

    QuantityPerUnit = m_strQuantityPerUnit
End Property

Public Property Let UnitPrice(ByVal curUnitPrice)
    m_curUnitPrice = curUnitPrice
End Property
Public Property Get UnitPrice()
    UnitPrice = m_curUnitPrice
End Property

Public Property Let UnitsInStock(ByVal intUnitsInStock)
    m_intUnitsInStock = intUnitsInStock
End Property
Public Property Get UnitsInStock()
    UnitsInStock = m_intUnitsInStock
End Property

Public Property Let UnitsOnOrder(ByVal intUnitsOnOrder)
    m_intUnitsOnOrder = intUnitsOnOrder
End Property
Public Property Get UnitsOnOrder()
    UnitsOnOrder = m_intUnitsOnOrder
End Property

Public Property Let ReorderLevel(ByVal intReorderLevel)
    m_intReorderLevel = intReorderLevel
End Property
Public Property Get ReorderLevel()
    ReorderLevel = m_intReorderLevel
End Property

Public Property Let Discontinued(ByVal blnDiscontinued)
    if blnDiscontinued <> "" then
        m_blnDiscontinued = blnDiscontinued
    else
        m_blnDiscontinued = False
    end if
End Property
Public Property Get Discontinued()
    Discontinued = m_blnDiscontinued
End Property

Public Property Let SourceURL(ByVal strURL)
    m_strSourceURL = strURL
End Property

Public Property Let CurrentPage(ByVal intCurrentPage)
    m_intCurrentPage = intCurrentPage
End Property

Public Function ReadRecord
    dim rstSQL
    dim strSQL

    set rstSQL = CreateObject("ADODB.Recordset")

    with rstSQL
        strSQL = "SELECT * FROM [Products]"
        strSQL = strSQL & " WHERE ProductID = " & m_lngProductID

        .Open strSQL, m_cConnection, adOpenForwardOnly, adLockReadOnly
    if not .BOF and not .EOF then
        m_lngProductID = rstSQL("ProductID")
        m_lngProductIDCopy = rstSQL("ProductID")
        m_strProductName = rstSQL("ProductName")
        m_lngSupplierID = rstSQL("SupplierID")
        m_lngCategoryID = rstSQL("CategoryID")
        m_strQuantityPerUnit = rstSQL("QuantityPerUnit")
        m_curUnitPrice = rstSQL("UnitPrice")
        m_intUnitsInStock = rstSQL("UnitsInStock")
        m_intUnitsOnOrder = rstSQL("UnitsOnOrder")
    end if
end with

```

```

        m_intReorderLevel = rstSQL("ReorderLevel")
        m_blnDiscontinued = rstSQL("Discontinued")
    end if
    .close
end with
set rstSQL = nothing
End Function

Public Function InsertRecord
    dim strSQL
    Dim resExecuteResult
    Dim lngRowsAffected

    strSQL = "INSERT INTO [Products] ("
    if m_strProductName <> "" then
        strSQL = strSQL & "ProductName, "
    End If
    if m_lngSupplierID <> "" then
        strSQL = strSQL & "SupplierID, "
    End If
    if m_lngCategoryID <> "" then
        strSQL = strSQL & "CategoryID, "
    End If
    if m_strQuantityPerUnit <> "" then
        strSQL = strSQL & "QuantityPerUnit, "
    End If
    if m_curUnitPrice <> "" then
        strSQL = strSQL & "UnitPrice, "
    End If
    if m_intUnitsInStock <> "" then
        strSQL = strSQL & "UnitsInStock, "
    End If
    if m_intUnitsOnOrder <> "" then
        strSQL = strSQL & "UnitsOnOrder, "
    End If
    if m_intReorderLevel <> "" then
        strSQL = strSQL & "ReorderLevel, "
    End If
    if m_blnDiscontinued <> "" then
        strSQL = strSQL & "Discontinued, "
    End If

    strSQL = Left(strSQL, len(strSQL) - 2) & ") Values ("
    if m_strProductName <> "" then
        strSQL = strSQL & SQLVal(m_strProductName,"String") & ", "
    End If
    if m_lngSupplierID <> "" then
        strSQL = strSQL & SQLVal(m_lngSupplierID,"Long") & ", "
    End If
    if m_lngCategoryID <> "" then
        strSQL = strSQL & SQLVal(m_lngCategoryID,"Long") & ", "
    End If
    if m_strQuantityPerUnit <> "" then
        strSQL = strSQL & SQLVal(m_strQuantityPerUnit,"String") & ", "
    End If
    if m_curUnitPrice <> "" then
        strSQL = strSQL & SQLVal(m_curUnitPrice,"Currency") & ", "
    End If
    if m_intUnitsInStock <> "" then
        strSQL = strSQL & SQLVal(m_intUnitsInStock,"Long") & ", "
    End If
    if m_intUnitsOnOrder <> "" then
        strSQL = strSQL & SQLVal(m_intUnitsOnOrder,"Long") & ", "
    End If
    if m_intReorderLevel <> "" then
        strSQL = strSQL & SQLVal(m_intReorderLevel,"Long") & ", "
    End If
    if m_blnDiscontinued <> "" then
        strSQL = strSQL & SQLVal(m_blnDiscontinued,"Boolean") & ", "
    End If

```

```

strSQL = Left(strSQL, len(strSQL) - 2) & ")"
''' Response.write strSQL ' Debugging
Set resExecuteResult = Server.CreateObject("ADODB.Command")
With resExecuteResult
    Set .ActiveConnection = m_cConnection
    .CommandType = adCmdText
    .CommandText = strSQL
'
' Execute the Query
.Execute lngRowsAffected
Set .ActiveConnection = Nothing
End With

End Function

Public Function UpdateRecord
    dim strSQL
    dim strValue
    Dim resExecuteResult
    Dim lngRowsAffected

    strSQL = "UPDATE [Products] SET "
    if m_strProductName <> "" then
        strSQL = strSQL & "ProductName = " & SQLVal(m_strProductName,"String") & ", "
    End If
    if m_lngSupplierID <> "" then
        strSQL = strSQL & "SupplierID = " & SQLVal(m_lngSupplierID,"Long") & ", "
    End If
    if m_lngCategoryID <> "" then
        strSQL = strSQL & "CategoryID = " & SQLVal(m_lngCategoryID,"Long") & ", "
    End If
    if m_strQuantityPerUnit <> "" then
        strSQL = strSQL & "QuantityPerUnit = " & SQLVal(m_strQuantityPerUnit,"String") & ", "
    End If
    if m_curUnitPrice <> "" then
        strSQL = strSQL & "UnitPrice = " & SQLVal(m_curUnitPrice,"Currency") & ", "
    End If
    if m_intUnitsInStock <> "" then
        strSQL = strSQL & "UnitsInStock = " & SQLVal(m_intUnitsInStock,"Long") & ", "
    End If
    if m_intUnitsOnOrder <> "" then
        strSQL = strSQL & "UnitsOnOrder = " & SQLVal(m_intUnitsOnOrder,"Long") & ", "
    End If
    if m_intReorderLevel <> "" then
        strSQL = strSQL & "ReorderLevel = " & SQLVal(m_intReorderLevel,"Long") & ", "
    End If
    if m_blnDiscontinued <> "" then
        strSQL = strSQL & "Discontinued = " & SQLVal(m_blnDiscontinued,"Boolean") & ", "
    End If
    strSQL = Left(strSQL, len(strSQL) - 2) & " Where "
    strSQL = strSQL & "ProductID = " & SQLVal(m_lngProductID,"Long") & " and "
    strSQL = Left(strSQL, len(strSQL) - 5)
    ''' Response.write strSQL ' Debugging
    Set resExecuteResult = Server.CreateObject("ADODB.Command")
    With resExecuteResult
        Set .ActiveConnection = m_cConnection
        .CommandType = adCmdText
        .CommandText = strSQL
'
' Execute the Query
.Execute lngRowsAffected
Set .ActiveConnection = Nothing
    End With

End Function

Public Function DeleteRecord
    dim strSQL
    dim strValue
    Dim resExecuteResult
    Dim lngRowsAffected

```

```
strSQL = "Delete From [Products] Where "  
strSQL = strSQL & "ProductID = " & SQLVal(m_lngProductID,"Long") & " and "  
strSQL = Left(strSQL, len(strSQL) - 5)  
' ' Response.write strSQL ' Debugging  
Set resExecuteResult = Server.CreateObject("ADODB.Command")  
With resExecuteResult  
    Set .ActiveConnection = m_cConnection  
    .CommandType = adCmdText  
    .CommandText = strSQL  
'  
    ' Execute the Query  
    .Execute lngRowsAffected  
    Set .ActiveConnection = Nothing  
End With  
  
End Function  
  
End Class  
%>
```

Appendix 2 – ASP Dataobject created using ASP script – Recordset Version

```
<%
Class clsProducts
' Kickstart Generated
' Version: 0.0.0
Private m_cConnection
'
' Primary keys
Private m_lngProductID
'
' Primary key copies. Used when updating primary key values
Private m_lngProductIDCOPY
'
' Data columns
Private m_strProductName
Private m_lngSupplierID
Private m_lngCategoryID
Private m_strQuantityPerUnit
Private m_curUnitPrice
Private m_intUnitsInStock
Private m_intUnitsOnOrder
Private m_intReorderLevel
Private m_blnDiscontinued
'
' Concurrency checking fields

Private m_strSourceURL
Private m_intCurrentPage

Public Property Set Connection(ByVal cConn)
    Set m_cConnection = cConn
End Property

Public Property Get DBTableName()
    DBTableName = "[Products]"
End Property

Public Property Let ProductID(ByVal lngProductID)
    m_lngProductID = lngProductID
End Property
Public Property Get ProductID()
    ProductID = m_lngProductID
End Property

Public Property Let ProductName(ByVal strProductName)
    m_strProductName = strProductName
End Property
Public Property Get ProductName()
    ProductName = m_strProductName
End Property

Public Property Let SupplierID(ByVal lngSupplierID)
    m_lngSupplierID = lngSupplierID
End Property
Public Property Get SupplierID()
    SupplierID = m_lngSupplierID
End Property

Public Property Let CategoryID(ByVal lngCategoryID)
    m_lngCategoryID = lngCategoryID
End Property
Public Property Get CategoryID()
    CategoryID = m_lngCategoryID
End Property

Public Property Let QuantityPerUnit(ByVal strQuantityPerUnit)
    m_strQuantityPerUnit = strQuantityPerUnit
```

```

End Property
Public Property Get QuantityPerUnit()
    QuantityPerUnit = m_strQuantityPerUnit
End Property

Public Property Let UnitPrice(ByVal curUnitPrice)
    m_curUnitPrice = curUnitPrice
End Property
Public Property Get UnitPrice()
    UnitPrice = m_curUnitPrice
End Property

Public Property Let UnitsInStock(ByVal intUnitsInStock)
    m_intUnitsInStock = intUnitsInStock
End Property
Public Property Get UnitsInStock()
    UnitsInStock = m_intUnitsInStock
End Property

Public Property Let UnitsOnOrder(ByVal intUnitsOnOrder)
    m_intUnitsOnOrder = intUnitsOnOrder
End Property
Public Property Get UnitsOnOrder()
    UnitsOnOrder = m_intUnitsOnOrder
End Property

Public Property Let ReorderLevel(ByVal intReorderLevel)
    m_intReorderLevel = intReorderLevel
End Property
Public Property Get ReorderLevel()
    ReorderLevel = m_intReorderLevel
End Property

Public Property Let Discontinued(ByVal blnDiscontinued)
    if blnDiscontinued <> "" then
        m_blnDiscontinued = blnDiscontinued
    else
        m_blnDiscontinued = False
    end if
End Property
Public Property Get Discontinued()
    Discontinued = m_blnDiscontinued
End Property

Public Property Let SourceURL(ByVal strURL)
    m_strSourceURL = strURL
End Property

Public Property Let CurrentPage(ByVal intCurrentPage)
    m_intCurrentPage = intCurrentPage
End Property

Public Function ReadRecord
    dim rstSQL
    dim strSQL

    set rstSQL = CreateObject("ADODB.Recordset")

    with rstSQL
        strSQL = "SELECT * FROM [Products]"
        strSQL = strSQL & " WHERE ProductID = " & m_lngProductID

        .Open strSQL, m_cConnection, adOpenForwardOnly, adLockReadOnly
        if not .BOF and not .EOF then
            m_lngProductID = rstSQL("ProductID")
            m_lngProductIDCopy = rstSQL("ProductID")
            m_strProductName = rstSQL("ProductName")
            m_lngSupplierID = rstSQL("SupplierID")
            m_lngCategoryID = rstSQL("CategoryID")
            m_strQuantityPerUnit = rstSQL("QuantityPerUnit")
            m_curUnitPrice = rstSQL("UnitPrice")

```

```

        m_intUnitsInStock = rstSQL("UnitsInStock")
        m_intUnitsOnOrder = rstSQL("UnitsOnOrder")
        m_intReorderLevel = rstSQL("ReorderLevel")
        m_blnDiscontinued = rstSQL("Discontinued")
    end if
    .close
end with
set rstSQL = nothing
End Function

```

```

Public Function InsertRecord
    dim rstSQL
    dim intBookmark

    set rstSQL = CreateObject("ADODB.Recordset")

    with rstSQL
        .CursorType = adOpenKeyset
        .LockType = adLockOptimistic
        .Open "Products", m_cConnection, , , adCmdTable
        .AddNew
        rstSQL("ProductName") = m_strProductName
        rstSQL("SupplierID") = m_lngSupplierID
        rstSQL("CategoryID") = m_lngCategoryID
        rstSQL("QuantityPerUnit") = m_strQuantityPerUnit
        rstSQL("UnitPrice") = m_curUnitPrice
        rstSQL("UnitsInStock") = m_intUnitsInStock
        rstSQL("UnitsOnOrder") = m_intUnitsOnOrder
        rstSQL("ReorderLevel") = m_intReorderLevel
        rstSQL("Discontinued") = m_blnDiscontinued
        .Update
        .Close
    end with
    set rstSQL = nothing
End Function

```

```

Public Function UpdateRecord
    dim rstSQL
    dim strSQL

    set rstSQL = CreateObject("ADODB.Recordset")

    with rstSQL
        strSQL = "SELECT * FROM Products"
        strSQL = strSQL & " WHERE ProductID = " & m_lngProductIDCOPY

        .Open strSQL, m_cConnection, adOpenDynamic, adLockOptimistic
        rstSQL("ProductName") = m_strProductName
        rstSQL("SupplierID") = m_lngSupplierID
        rstSQL("CategoryID") = m_lngCategoryID
        rstSQL("QuantityPerUnit") = m_strQuantityPerUnit
        rstSQL("UnitPrice") = m_curUnitPrice
        rstSQL("UnitsInStock") = m_intUnitsInStock
        rstSQL("UnitsOnOrder") = m_intUnitsOnOrder
        rstSQL("ReorderLevel") = m_intReorderLevel
        rstSQL("Discontinued") = m_blnDiscontinued
        .Update
        .Close

        m_lngProductIDCOPY = m_lngProductID
    end with

    set rstSQL = nothing
End Function

```

```

Public Function DeleteRecord
    dim rstSQL
    dim strSQL

    set rstSQL = CreateObject("ADODB.Recordset")

```

```
with rstSQL
    strSQL = "SELECT * FROM Products"
    strSQL = strSQL & " WHERE ProductID = " & m_lngProductID
    .Open strSQL, m_cConnection, adOpenDynamic, adLockOptimistic
    .Delete
    .Close
end with

    set rstSQL = nothing
End Function
End Class
%>
```

Appendix 3 – ASP Table Maintenance Page created using ASP script

```
<SCRIPT LANGUAGE="VBScript" RUNAT="Server"></SCRIPT>
<%Option Explicit%>
<%Response.Buffer = true%>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Kickstart Script TableMaster.KSC">
<TITLE>Products</TITLE>
<link rel="stylesheet" type="text/css" href="PageStyle.css">
</HEAD>
<script language="JavaScript1.2" >
function validateForm(form)
{
    return (true);
}
</script>
<!-- #Include file="utility.asp" -->
<!-- #Include Virtual="clsProducts.asp" -->
<%
'
' ADO connection object
Dim m_cConnection
'
' URL action variable
Dim m_strAction
'
' Zero date
Dim m_dteUninitialized
'
' Table Columns & Search Relational Operator
Dim m_lngProductID
Dim m_strProductIDRelop
Dim m_strProductName
Dim m_strProductNameRelop
Dim m_lngSupplierID
Dim m_strSupplierIDRelop
Dim m_lngCategoryID
Dim m_strCategoryIDRelop
Dim m_strQuantityPerUnit
Dim m_strQuantityPerUnitRelop
Dim m_curUnitPrice
Dim m_strUnitPriceRelop
Dim m_intUnitsInStock
Dim m_strUnitsInStockRelop
Dim m_intUnitsOnOrder
Dim m_strUnitsOnOrderRelop
Dim m_intReorderLevel
Dim m_strReorderLevelRelop
Dim m_blnDiscontinued
Dim m_strDiscontinuedRelop
'
' Data object
Dim m_cProducts

Sub GetSearchCriteria
    Dim strField

    ParseField Request.Form("ctlProductID") & "", strField, m_strProductIDRelop
    m_lngProductID = NumVal(strField)
    ParseField Request.Form("ctlProductName") & "", strField, m_strProductNameRelop
    m_strProductName = strField
    ParseField Request.Form("ctlSupplierID") & "", strField, m_strSupplierIDRelop
    m_lngSupplierID = NumVal(strField)
    ParseField Request.Form("ctlCategoryID") & "", strField, m_strCategoryIDRelop
    m_lngCategoryID = NumVal(strField)
    ParseField Request.Form("ctlQuantityPerUnit") & "", strField, m_strQuantityPerUnitRelop
    m_strQuantityPerUnit = strField
```

```

ParseField Request.Form("ctlUnitPrice") & "", strField, m_strUnitPriceRelop
m_curUnitPrice = NumVal(strField)
ParseField Request.Form("ctlUnitsInStock") & "", strField, m_strUnitsInStockRelop
m_intUnitsInStock = NumVal(strField)
ParseField Request.Form("ctlUnitsOnOrder") & "", strField, m_strUnitsOnOrderRelop
m_intUnitsOnOrder = NumVal(strField)
ParseField Request.Form("ctlReorderLevel") & "", strField, m_strReorderLevelRelop
m_intReorderLevel = NumVal(strField)
ParseField Request.Form("ctlDiscontinued") & "", strField, m_strDiscontinuedRelop
m_blnDiscontinued = BoolVal(strField)

```

End Sub

Sub GetFormFields

```

m_strProductName = Request.Form("ctlProductName") & ""
m_lngSupplierID = NumVal(Request.Form("ctlSupplierID"))
m_lngCategoryID = NumVal(Request.Form("ctlCategoryID"))
m_strQuantityPerUnit = Request.Form("ctlQuantityPerUnit") & ""
m_curUnitPrice = NumVal(Request.Form("ctlUnitPrice"))
m_intUnitsInStock = NumVal(Request.Form("ctlUnitsInStock"))
m_intUnitsOnOrder = NumVal(Request.Form("ctlUnitsOnOrder"))
m_intReorderLevel = NumVal(Request.Form("ctlReorderLevel"))
m_blnDiscontinued = BoolVal(Request.Form("ctlDiscontinued"))

```

End Sub

Sub CopyFormFieldsToSqlObject(cSQL)

```

With cSQL
.AddColVal "ProductID", SQLVal(m_lngProductID,"Long"), True
.AddColVal "ProductName", SQLVal(m_strProductName,"String"), False
.AddColVal "SupplierID", SQLVal(m_lngSupplierID,"Long"), False
.AddColVal "CategoryID", SQLVal(m_lngCategoryID,"Long"), False
.AddColVal "QuantityPerUnit", SQLVal(m_strQuantityPerUnit,"String"), False
.AddColVal "UnitPrice", SQLVal(m_curUnitPrice,"Currency"), False
.AddColVal "UnitsInStock", SQLVal(m_intUnitsInStock,"Long"), False
.AddColVal "UnitsOnOrder", SQLVal(m_intUnitsOnOrder,"Long"), False
.AddColVal "ReorderLevel", SQLVal(m_intReorderLevel,"Long"), False
.AddColVal "Discontinued", SQLVal(m_blnDiscontinued,"Boolean"), False
.Table = "[Products]"
.AddWhere "[Products].ProductID = " & m_lngProductID
End With

```

End Sub

Sub SaveSearchCriteriaInCookies

Dim dteExpires

```

Response.Cookies("[Products]")( "ctlProductID" ) = m_lngProductID
Response.Cookies("[Products]")( "relProductID" ) = m_strProductIDRelop
Response.Cookies("[Products]")( "ctlProductName" ) = m_strProductName
Response.Cookies("[Products]")( "relProductName" ) = m_strProductNameRelop
Response.Cookies("[Products]")( "ctlSupplierID" ) = m_lngSupplierID
Response.Cookies("[Products]")( "relSupplierID" ) = m_strSupplierIDRelop
Response.Cookies("[Products]")( "ctlCategoryID" ) = m_lngCategoryID
Response.Cookies("[Products]")( "relCategoryID" ) = m_strCategoryIDRelop
Response.Cookies("[Products]")( "ctlQuantityPerUnit" ) = m_strQuantityPerUnit
Response.Cookies("[Products]")( "relQuantityPerUnit" ) = m_strQuantityPerUnitRelop
Response.Cookies("[Products]")( "ctlUnitPrice" ) = m_curUnitPrice
Response.Cookies("[Products]")( "relUnitPrice" ) = m_strUnitPriceRelop
Response.Cookies("[Products]")( "ctlUnitsInStock" ) = m_intUnitsInStock
Response.Cookies("[Products]")( "relUnitsInStock" ) = m_strUnitsInStockRelop
Response.Cookies("[Products]")( "ctlUnitsOnOrder" ) = m_intUnitsOnOrder
Response.Cookies("[Products]")( "relUnitsOnOrder" ) = m_strUnitsOnOrderRelop
Response.Cookies("[Products]")( "ctlReorderLevel" ) = m_intReorderLevel
Response.Cookies("[Products]")( "relReorderLevel" ) = m_strReorderLevelRelop
Response.Cookies("[Products]")( "ctlDiscontinued" ) = m_blnDiscontinued
Response.Cookies("[Products]")( "relDiscontinued" ) = m_strDiscontinuedRelop
Response.Cookies("[Products]")( "table" ) = "Products"
dteExpires = DateAdd("YYYY",1,Date)

```



```

        Discontinued
    </td>
    <td class="TableHead">
        Categoryid
    </td>
    <td class="TableHead">
        Supplierid
    </td>
</tr>
<tr bgcolor=#77aaff valign="top">
    <td class="TableHead" align="center">
        <a href="#"
            onclick="frmSearch.action='ProductsMaster.ASP?Action=<%=SHOW_NEW%>';
            frmSearch.submit();"></a>
        <a href="#"
            onclick="frmSearch.action='ProductsMaster.ASP?Action=<%=SHOW_SEARCH_RESULTS%>';
            frmSearch.submit();"></a>
    <td class="TableHead">
        <%=ShowTextBox("", "ctlProductName", m_strProductNameRelop, m_strProductName, 16,
40, False, False)%>
    </td>
    <td class="TableHead">
        <%=ShowTextBox("", "ctlQuantityPerUnit", m_strQuantityPerUnitRelop,
m_strQuantityPerUnit, 16, 20, False, False)%>
    </td>
    <td class="TableHead">
        <%=ShowTextBox("", "ctlUnitPrice", m_strUnitPriceRelop, m_curUnitPrice, 16, 19,
False, False)%>
    </td>
    <td class="TableHead">
        <%=ShowTextBox("", "ctlUnitsInStock", m_strUnitsInStockRelop, m_intUnitsInStock,
5, 5, False, False)%>
    </td>
    <td class="TableHead">
        <%=ShowTextBox("", "ctlUnitsOnOrder", m_strUnitsOnOrderRelop, m_intUnitsOnOrder,
5, 5, False, False)%>
    </td>
    <td class="TableHead">
        <%=ShowTextBox("", "ctlReorderLevel", m_strReorderLevelRelop, m_intReorderLevel,
5, 5, False, False)%>
    </td>
    <td class="TableHead">
        <%=ShowCheckBox("", "ctlDiscontinued", m_blnDiscontinued, False)%>
    </td>
    <td class="TableHead">
        <%=ShowDropDown("", "Categories", "CategoryID", "CategoryID", "ctlCategoryID",
SQLVal(m_lngCategoryID,"Supplierid"), CBO_ALL, false)%>
    </td>
    <td class="TableHead">
        <%=ShowDropDown("", "Suppliers", "SupplierID", "SupplierID", "ctlSupplierID",
SQLVal(m_lngSupplierID,"CategoryID"), CBO_ALL, false)%>
    </td>
</tr>
<%
End Sub

Sub ShowSearchFormBottom
%>

        </table>
    </form>
</td>
</tr>
</table>
<%
End Sub
'
' Add pagination to search results
Sub ShowSearchResults

```

```

Dim blnNoRecords
Dim cSQL
Dim blnShade
Dim strBGColor
Dim strReferenceColumnNames()
Dim strPKList
Dim lngCount
Dim lngFlushCount

lngFlushCount = 50

'
' Primary keys
Dim lngProductID
Redim strReferenceColumnNames(1)
'
' Show the search results
ShowSearchFormTop

set cSQL = new clsSQL
With cSQL
    set .Connection = m_cConnection
    .PageSize = 64
    .AddColumn "[Products].ProductID", ""
    .AddColumn "[Products].ProductName", ""
    .AddColumn "[Products].SupplierID", ""
    .AddColumn "[Products].CategoryID", ""
    .AddColumn "[Products].QuantityPerUnit", ""
    .AddColumn "[Products].UnitPrice", ""
    .AddColumn "[Products].UnitsInStock", ""
    .AddColumn "[Products].UnitsOnOrder", ""
    .AddColumn "[Products].ReorderLevel", ""
    .AddColumn "[Products].Discontinued", ""
    .AddColumn "Categories_1.CategoryID", "CategoryID_1"
    strReferenceColumnNames(0) = "Categories_1.CategoryID"
    .AddColumn "Suppliers_2.SupplierID", "SupplierID_2"
    strReferenceColumnNames(1) = "Suppliers_2.SupplierID"
    .AddFrom "[Products]", ""
    .AddFrom "Categories", "Categories_1"
    .AddFrom "Suppliers", "Suppliers_2"
    .AddWhere "[Products].CategoryID = " & strReferenceColumnNames(0)
    .AddWhere "[Products].SupplierID = " & strReferenceColumnNames(1)
    If m_lngProductID <> 0 then
        .AddWhere MakeWhereClause("[Products].ProductID", m_strProductIDRelop, NumVal(m_lngProductID))
    End If
    If m_strProductName <> vbNullString then
        .AddWhere MakeWhereClause("[Products].ProductName", m_strProductNameRelop,
CharVal(m_strProductName))
    End If
    If m_lngSupplierID <> 0 then
        .AddWhere MakeWhereClause("[Products].SupplierID", m_strSupplierIDRelop,
NumVal(m_lngSupplierID))
    End If
    If m_lngCategoryID <> 0 then
        .AddWhere MakeWhereClause("[Products].CategoryID", m_strCategoryIDRelop,
NumVal(m_lngCategoryID))
    End If
    If m_strQuantityPerUnit <> vbNullString then
        .AddWhere MakeWhereClause("[Products].QuantityPerUnit", m_strQuantityPerUnitRelop,
CharVal(m_strQuantityPerUnit))
    End If
    If m_curUnitPrice <> 0 then
        .AddWhere MakeWhereClause("[Products].UnitPrice", m_strUnitPriceRelop, NumVal(m_curUnitPrice))
    End If
    If m_intUnitsInStock <> 0 then
        .AddWhere MakeWhereClause("[Products].UnitsInStock", m_strUnitsInStockRelop,
NumVal(m_intUnitsInStock))
    End If
    If m_intUnitsOnOrder <> 0 then
        .AddWhere MakeWhereClause("[Products].UnitsOnOrder", m_strUnitsOnOrderRelop,
NumVal(m_intUnitsOnOrder))

```

```

End If
If m_intReorderLevel <> 0 then
    .AddWhere MakeWhereClause("[Products].ReorderLevel", m_strReorderLevelRelop,
NumVal(m_intReorderLevel))
End If
If m_blnDiscontinued <> False then
    .AddWhere MakeWhereClause("[Products].Discontinued", m_strDiscontinuedRelop,
BoolVal(m_blnDiscontinued))
End If
'.AddOrderBy "You decide!!!"
''' Response.write .buildsqlstatement ''' Debugging
.ExecuteQuery ""
If .Recordset is nothing then
    blnNorecords = True
Else
    If .Recordset.EOF and .Recordset.BOF then
        blnNorecords = True
    End If
End If
If blnNorecords Then
    Response.Write "<tr bgcolor=""#e8f8ff""><td colspan=3 align=""center"">No matching rows
found</td></tr>"
Else
    With .Recordset
        blnShade = False
        Do while not .eof
            lngCount = lngCount + 1
            If blnShade then
                strBGColor = "#e8f8ff"
            Else
                strBGColor = "#ffffff"
            End If
            blnShade = not blnShade
            strPKList = vbNullString
            strPKList = strPKList & "&ProductID=" & NumVal(.Fields("ProductID"))
%>
            <tr bgcolor=<%=strBGColor%>>
                <td Class ="TableBody">
                    <a href="#"
onclick="frmSearch.action='ProductsMaster.ASP?Action=<%=SHOW_FORM%><%=strPKList%>';
frmSearch.submit();"></a>
                    <a href="#"
onclick="frmSearch.action='ProductsMaster.ASP?Action=<%=SHOW_EDIT%><%=strPKList%>';
frmSearch.submit();"></a>
                    <a href="#"
onclick="frmSearch.action='ProductsMaster.ASP?Action=<%=DELETE_ROW%><%=strPKList%>';
frmSearch.submit();"></a>
                </td>
                <td class="TableBody" align="Left">
                    <%=.Fields("ProductName")%>
                </td>
                <td class="TableBody" align="Left">
                    <%=.Fields("QuantityPerUnit")%>
                </td>
                <td class="TableBody" align="right">
                    <%=.Fields("UnitPrice")%>
                </td>
                <td class="TableBody" align="right">
                    <%=.Fields("UnitsInStock")%>
                </td>
                <td class="TableBody" align="right">
                    <%=.Fields("UnitsOnOrder")%>
                </td>
                <td class="TableBody" align="right">

```

```

        <%=.Fields("ReorderLevel")%>
    </td>
    <td class="TableBody" align="center">
        <% if boolval(.Fields("Discontinued")) then %>
            
        <% else %>
            
        <% end If %>
    </td>
    <td class="TableBody">
        <%=.Fields("CategoryID_1")%>
    </td>
    <td class="TableBody">
        <%=.Fields("SupplierID_2")%>
    </td>
</tr>
<%
    .MoveNext
    If lngCount = lngFlushCount Then
        lngCount = 0
        Response.Flush
    end if
loop
End With
End If
ShowSearchFormBottom
End With

End Sub

Sub ShowForm(intMode)
    Dim strNextAction
    Dim strMode
    Dim blnReadOnly
    Dim strPK

    Select case intMode
        Case SHOW_FORM
            strNextAction = GO_BACK
            strMode = "View"
            blnReadOnly = True
        Case SHOW_EDIT
            strNextAction = UPDATE_ROW
            strMode = "Edit"
            blnReadOnly = False
        Case SHOW_NEW
            strNextAction = INSERT_ROW
            strMode = "New"
            blnReadOnly = False
    End select

    strPK = ""
    strPK = strPK & "&ProductID=" & m_lngProductID
%>
<table cellspacing="1" cellpadding="2" border="0" width="100%">
    <tr>
        <td>
            <form name="frmEdit"
                method="post"
                action="ProductsMaster.ASP?Action=<%=strNextAction%><%=strPK%>"
                onSubmit="return validateForm(frmEdit);">
                <table cellspacing="1" cellpadding="2" border="0" width="100%" bgcolor=#0033cc>
                    <tr bgcolor=#0033cc>
                        <td colspan=2 align="center" class="TableHead">
                            <%=strMode%> Products
                        </td>
                    </tr>
                    <tr bgcolor=#e8f8ff>
                        <td class="TableBody">
                            Productname
                        </td>
                    </tr>
                </table>
            </form>
        </td>
    </tr>
</table>

```

```

        <td class="TableBody">
            <%=ShowTextBox("", "ctlProductName", "", m_strProductName, 16, 40, False,
blnReadOnly)%>
        </td>
    </tr>
    <tr bgcolor=#e8f8ff>
        <td class="TableBody">
            Quantityperunit
        </td>
        <td class="TableBody">
            <%=ShowTextBox("", "ctlQuantityPerUnit", "", m_strQuantityPerUnit, 16, 20, False,
blnReadOnly)%>
        </td>
    </tr>
    <tr bgcolor=#e8f8ff>
        <td class="TableBody">
            Unitprice
        </td>
        <td class="TableBody">
            <%=ShowTextBox("", "ctlUnitPrice", "", m_curUnitPrice, 16, 19, False,
blnReadOnly)%>
        </td>
    </tr>
    <tr bgcolor=#e8f8ff>
        <td class="TableBody">
            Unitsinstock
        </td>
        <td class="TableBody">
            <%=ShowTextBox("", "ctlUnitsInStock", "", m_intUnitsInStock, 5, 5, False,
blnReadOnly)%>
        </td>
    </tr>
    <tr bgcolor=#e8f8ff>
        <td class="TableBody">
            Unitsonorder
        </td>
        <td class="TableBody">
            <%=ShowTextBox("", "ctlUnitsOnOrder", "", m_intUnitsOnOrder, 5, 5, False,
blnReadOnly)%>
        </td>
    </tr>
    <tr bgcolor=#e8f8ff>
        <td class="TableBody">
            Reorderlevel
        </td>
        <td class="TableBody">
            <%=ShowTextBox("", "ctlReorderLevel", "", m_intReorderLevel, 5, 5, False,
blnReadOnly)%>
        </td>
    </tr>
    <tr bgcolor=#e8f8ff>
        <td class="TableBody">
            Discontinued
        </td>
        <td class="TableBody">
            <%=ShowCheckBox("", "ctlDiscontinued", m_blnDiscontinued, blnReadOnly)%>
        </td>
    </tr>
    <tr bgcolor=#e8f8ff>
        <td class="TableBody">
            CategoryID
        </td>
        <td class="TableBody">
            <%=ShowDropDown("", "Categories", "CategoryID", "CategoryID", "ctlCategoryID",
m_lngCategoryID, "< Select one >", blnReadOnly)%>
        </td>
    </tr>
    <tr bgcolor=#e8f8ff>
        <td class="TableBody">
            SupplierID
        </td>

```

```

                <td class="TableBody">
                    <%=ShowDropDown("", "Suppliers", "SupplierID", "SupplierID", "ctlSupplierID",
m_lngSupplierID, "< Select one >", blnReadOnly)%>
                </td>
            </tr>
            <tr bgcolor=#77aaff>
                <td colspan=2 align="center">
<%
    Select case intMode
        Case SHOW_FORM
%>
                <input type="button" name="cmdBack" class="TableBody" value="Back to list"
onclick="history.back();">

<%
        Case SHOW_EDIT
%>
                <input type="submit" name="cmdSave" class="TableBody" value="Save">
                <input type="button" name="cmdCancel" class="TableBody" value="Cancel"
onclick="history.back();">
<%
        Case SHOW_NEW
%>
                <input type="submit" name="cmdSave" class="TableBody" value="Save">
                <input type="button" name="cmdCancel" class="TableBody" value="Cancel"
onclick="history.back();">
<%
    End select
%>
                </td>
            </tr>
        </table >
    </form>
</td>
</tr>
</table>
<%
End Sub

Sub ReadRecord
    Dim cSQL
    Dim blnNoRecord

    set cSQL = new clsSQL
    With cSQL
        set .Connection = m_cConnection
        .PageSize = 64
        .AddColumn "*", ""
        .AddFrom "[Products]", ""
        .AddWhere "[Products].ProductID = " & SQLVal(m_lngProductID, "Long")
        ''' response.write .buildsqlstatement ''' Debug
        .ExecuteQuery ""
        If .Recordset is nothing then
            blnNoRecord = True
        Else
            If .Recordset.EOF and .Recordset.BOF then
                blnNoRecord = True
            End If
        End If
        If blnNoRecord Then
            Response.Write "<tr bgcolor=""#e8f8ff""><td colspan=3 align=""center"">No matching row
found</td></tr>"
        Else
            With .Recordset
                m_lngProductID = .Fields("ProductID")
                m_strProductName = .Fields("ProductName")
                m_lngSupplierID = .Fields("SupplierID")
                m_lngCategoryID = .Fields("CategoryID")
                m_strQuantityPerUnit = .Fields("QuantityPerUnit")
                m_curUnitPrice = .Fields("UnitPrice")
            End With
        End If
    End With
End Sub

```

```

        m_intUnitsInStock = .Fields("UnitsInStock")
        m_intUnitsOnOrder = .Fields("UnitsOnOrder")
        m_intReorderLevel = .Fields("ReorderLevel")
        m_blnDiscontinued = .Fields("Discontinued")
    End With
End If
End With
set cSQL = Nothing

End Sub

Sub InsertRow

    InitializeClsProducts
    GetFormFields
    with m_cProducts
        .ProductID = m_lngProductID
        .ProductName = m_strProductName
        .SupplierID = m_lngSupplierID
        .CategoryID = m_lngCategoryID
        .QuantityPerUnit = m_strQuantityPerUnit
        .UnitPrice = m_curUnitPrice
        .UnitsInStock = m_intUnitsInStock
        .UnitsOnOrder = m_intUnitsOnOrder
        .ReorderLevel = m_intReorderLevel
        .Discontinued = m_blnDiscontinued
        .InsertRecord
    end with

End Sub

Sub UpdateRow

    InitializeClsProducts
    GetFormFields
    with m_cProducts
        .ProductID = m_lngProductID
        .ReadRecord
        .ProductID = m_lngProductID
        .ProductName = m_strProductName
        .SupplierID = m_lngSupplierID
        .CategoryID = m_lngCategoryID
        .QuantityPerUnit = m_strQuantityPerUnit
        .UnitPrice = m_curUnitPrice
        .UnitsInStock = m_intUnitsInStock
        .UnitsOnOrder = m_intUnitsOnOrder
        .ReorderLevel = m_intReorderLevel
        .Discontinued = m_blnDiscontinued
        .UpdateRecord
    end with

End Sub

Sub DeleteRow

    InitializeClsProducts
    GetFormFields
    with m_cProducts
        .ProductID = m_lngProductID
        .DeleteRecord
    end with

End Sub

Sub InitializeClsProducts

    set m_cProducts = new clsProducts

    with m_cProducts
        set .Connection = m_cConnection
        ' Set this parameter to the location of this file

```

```

        .SourceURL = "ProductsMaster.asp"
        .CurrentPage = 0
    end with

End Sub

%>
<BODY>
<table border=0>
    <tr>
        <td colspan=2>
            <%
ShowPageHeader "Products"
%>
        </td>
    </tr>
    <tr valign="top">
        <td>
            <%
ShowSideMenu "Products"
%>
        </td>
        <td>
            <%
'
' Mainline
Set m_cConnection = GetConnectedToAccess(DB_PATH_NAME)
m_dteUninitialized = DateSerial(0, 0, 0)
m_strAction = Request.QueryString("Action")
m_lngProductID = NumVal(Request.QueryString("ProductID"))
Select case m_strAction
    Case "", SHOW_SEARCH
        ShowSearchFormTop
        ShowSearchFormBottom
    Case SHOW_SEARCH_RESULTS
        GetSearchCriteria
        SaveSearchCriteriaInCookies
        ShowSearchResults
    Case SHOW_FORM, SHOW_EDIT
        ReadRecord
        ShowForm m_strAction
    Case SHOW_NEW
        ShowForm m_strAction
    Case UPDATE_ROW
        UpdateRow
        LoadSearchCriteriaFromCookies
        ShowSearchResults
    Case INSERT_ROW
        InsertRow
        LoadSearchCriteriaFromCookies
        ShowSearchResults
    Case DELETE_ROW
        DeleteRow
        LoadSearchCriteriaFromCookies
        ShowSearchResults
End select

CloseConnection(m_cConnection)
%>
        </td>
    </tr>
    <tr>
        <td>
            <!-- Under menu -->
            &nbsp;
        </td>
        <td>
            <%
ShowPageFooter "Products"
%>
        </td>
    </tr>
</table>

```

```
</tr>  
</table>  
</BODY>  
</HTML>
```