

<%

Dim Login**Set** Login = **New** LoginSession

```

'*****
' The LoginSession class is created for each page click and manages session
' state for the user. It encapsulates cookie handling, database user-
' tracking, login/logout, and user permissions for various modules.
'*****

```

Class LoginSession**Public** Theme**Public** Name**Public** Role**Public** ID**Public** Email**Public** IsRemembered**Public** Permanent

```

'*****
' When the class is initialized it checks for previous login and session
' settings. If none exist it tracks user logins and sets session state for
' this session/login.
'*****

```

Private Sub Class_Initialize()

'check the theme setting

If Len(Request.Cookies("fw_theme")) = 0 **Then**

Me.Theme = "Primary"

Response.Cookies("fw_theme") = Me.Theme

Response.Cookies("fw_theme").Expires = **Date** + 365**Else**

'hack to work around the old site

If Request.Cookies("fw_theme") = "default" **Then** Response.Cookies("fw_theme") = ""

Me.Theme = Request.Cookies("fw_theme")

End If

'check remember login

If Len(Request.Cookies("userName")) > 0 **And** Len(Session("userName")) = 0 **Then**Me.IsRemembered = **True**

Me.Name = Request.Cookies("userName")

Me.Role = 0 ' this will get set when they are logged back in.

Me.ID = Request.Cookies("userId")

Me.Email = Request.Cookies("email")

ElseMe.IsRemembered = **False**

Me.Name = Session("userName")

Me.Role = **CInt**("0" & Session("role"))

Me.ID = Session("userId")

Me.Email = Session("email")

End If

'handle blocked users

```

If Me.Role = 6 Then Response.End

'track the session in the db
'!!! There is SQL code here because the data objects haven't been created yet!
Dim strSQL
If Session("IP") = "" Then Exit Sub 'no use in tracking what we can't see
strSQL = "SELECT COUNT(*) FROM Sessions WHERE sIP=" & ToSQL(Session("IP"), "Text")
If OpenRst(strSQL).Fields(0) = 0 Then 'add the session
    strSQL = "INSERT INTO Sessions(sLogin,sIP,sDateFirst,sDateLast,sReferer) " & _
        "VALUES(" & ToSQL(Me.Name, "Text") & ", " & _
        ToSQL(Session("IP"), "Text") & ", " & _
        ToSQL(Now(), "DateTime") & ", " & _
        ToSQL(Now(), "DateTime") & ", " & _
        ToSQL(Request.ServerVariables("HTTP_REFERER"), "Text") & ")"
Else 'update the session
    strSQL = "UPDATE Sessions SET sLogin=" & ToSQL(Me.Name, "Text") & _
        ",sDateLast=" & ToSQL(Now(), "DateTime") & _
        " WHERE sIP =" & ToSQL(Session("IP"), "Text")
End If
ExecuteSQL strSQL

'hack to work around the old site; delete old sessions every 10 logins
'!!! There is SQL code here because the data objects haven't been created yet!
Application.Lock
Application("Clear_Sessions_CountDown") = Application("Clear_Sessions_CountDown") + 1
If Application("Clear_Sessions_CountDown") > 10 Then
    ExecuteSQL "DELETE FROM Sessions WHERE sDateLast < " & ToSQL(DateAdd("n",-30,Now()),
"DateTime")
    Application("Clear_Sessions_CountDown") = 0
End If
Application.Unlock
End Sub

'*****
' Start() accepts a user object and populates the session and cookie from
' the user info
'*****
Public Default Function Start(objUser)
    Response.Cookies("userId") = objUser.uID
    If Permanent = True Then Response.Cookies("userId").Expires = Date + 365
    Response.Cookies("userName") = objUser.uLogin
    If Permanent = True Then Response.Cookies("userName").Expires = Date + 365
    Response.Cookies("email") = objUser.uEmail
    If Permanent = True Then Response.Cookies("email").Expires = Date + 365
    Session("userName") = objUser.uLogin
    Session("role") = objUser.uRole
    Session("userId") = objUser.uID
    Session("email") = objUser.uEmail
    Name = objUser.uLogin
    Role = objUser.uRole
    ID = objUser.uID
    Email = objUser.uEmail
    objUser.UpdateLastLogin

```

End Function

```
'*****
' CanEdit() accepts a table name and returns a bool whether the user can
' edit rows from table. It loads "actions/load_permissions.asp" to
' populate the application variable with a CSV of roles and privileges. It
' then compares the user's current role to the list.
'*****
```

Function CanEdit(p_TableName)

CanEdit = **False**

If Name = "" **Then Exit Function**

If Len(Application("Permissions_CanEdit")) = 0 **Then Include**

"actions/load_permissions.asp"

If Instr(1,Application("Permissions_CanEdit"),p_TableName & ":" & Role,1) > 1 **Then**

CanEdit = **True**

End Function

```
'*****
' CanAddNew() accepts a table name and returns a bool whether the user can
' add to that table.
'*****
```

Function CanAddNew(p_TableName)

CanAddNew = **False**

If Name = "" **Then Exit Function**

If Len(Application("Permissions_CanAddNew")) = 0 **Then Include**

"actions/load_permissions.asp"

If Instr(1,Application("Permissions_CanAddNew"),p_TableName & ":" & Role,1) > 1 **Then**

CanAddNew = **True**

End Function

```
'*****
' CanDelete() accepts a table name and returns a bool whether the user can
' delete from that table.
'*****
```

Function CanDelete(p_TableName)

CanDelete = **False**

If Name = "" **Then Exit Function**

If Len(Application("Permissions_CanDelete")) = 0 **Then Include**

"actions/load_permissions.asp"

If Instr(1,Application("Permissions_CanDelete"),p_TableName & ":" & Role,1) > 1 **Then**

CanDelete = **True**

End Function

```
'*****
' SetTheme() This function supports functionality from the previous
' version of the site. It gathers the theme name from the querystring and
' if it exists, sets it as the theme in the cookies.
'*****
```

Public Function SetTheme()

Dim objFSO 'As Scripting.FileSystemObject

Set objFSO = Server.CreateObject("Scripting.FileSystemobject")

```

If objFSO.FolderExists (Server.MapPath (THEMES_VIRTUAL_PATH & Request.QueryString ("uTheme"
))) Then
    Response.Cookies ("fw_theme") = Request.QueryString ("uTheme")
    Response.Cookies ("fw_theme").Expires = Date + 365
    Me.Theme = Request.QueryString ("uTheme")
End If

```

```

Set objFSO = Nothing
End Function

```

```

'*****
'  Expire() empties out the session variables and cookies and resets the
'  login state.
'*****

```

```

Public Function Expire ()
    'empty everything out and start over
    Session ("userName") = ""
    Session ("role") = 0
    Session ("email") = ""
    Session ("userId") = 0
    Response.Cookies ("userId") = ""
    Response.Cookies ("userName") = ""
    Response.Cookies ("email") = ""
    Response.Cookies ("userId").Expires = Date - 1
    Response.Cookies ("userName").Expires = Date - 1
    Response.Cookies ("email").Expires = Date - 1

    Me.Role = 0
    Me.Name = ""
    Me.ID = ""
    Me.Email = ""

```

```

End Function

```

```

'*****
'  GetOnlineMembersList() returns a CSV list of usernames currently online.
'  This info is tracked in the sessions table of the database
'*****

```

```

Function GetOnlineMembersList ()
    Dim rst
    Dim strSQL

    '!!! There is SQL code here because the data objects haven't been created yet!
    strSQL = "SELECT sLogin FROM sessions WHERE sLogin IS NOT NULL"
    GetOnlineMembersList = OpenRst (strSQL).GetString (,,, ",")

```

```

End Function

```

```

End Class

```

```

'*****

```

```

%>

```