```
<%
class cPlacement
    private m_CustomerCode
    private m_CustomerName
    private m_Reference
    private m_EmployeeNumber
    private m_StartDate
    private m_PStopDate
    private m_PlacementId
    private m_LastnameFirst
    private m_JobNumber
    private m_WorkCode
    private m_RegPayRate
    private m_RegBillRate
    private m_TotalExpenses
    private m_TotalTime
    private m_ExpenseSummary
    private m_TimeSummary
    Private m_Status
    private m_NeedFinalTime
    private m_WCDescription

    public property get CustomerCode ()
        CustomerCode = m_CustomerCode
    end property

    public property let CustomerCode(p_CustomerCode)
        m_CustomerCode = p_CustomerCode
    end property

    public property get CustomerName()
        CustomerName = m_CustomerName
    end property

    public property let CustomerName(p_CustomerName)
        m_CustomerName = p_CustomerName
    end property

    public property get Reference()
        Reference = m_Reference
    end property

    public property let Reference(p_Reference)
        m_Reference = p_Reference
    end property

    public property get EmployeeNumber()
        EmployeeNumber = m_EmployeeNumber
    end property

    public property let EmployeeNumber(p_EmployeeNumber)
        m_EmployeeNumber = p_EmployeeNumber
    end property

    public property get StartDate()
        StartDate = m_StartDate
    end property

    public property let StartDate(p_StartDate)
        m_StartDate = p_StartDate
    end property
    public property get PStopDate()
        PStopDate = m_PStopDate
    end property

    public property let PStopDate(p_PStopDate)
        m_PStopDate = p_PStopDate
    end property

    public property get PlacementId()
        PlacementId = m_PlacementId
    end property

    public property let PlacementId(p_PlacementId)
        m_PlacementId = p_PlacementId
    end property

    public property get LastnameFirst()
        LastnameFirst = m_LastnameFirst
    end property

    public property let LastnameFirst(p_LastnameFirst)
        m_LastnameFirst = p_LastnameFirst
    end property

    public property get JobNumber()
        JobNumber = m_JobNumber
    end property

    public property let JobNumber(p_JobNumber)
        m_JobNumber = p_JobNumber
    end property

    public property get WorkCode()
        WorkCode = m_WorkCode
    end property

    public property let WorkCode(p_WorkCode)
        m_WorkCode = p_WorkCode
    end property

    public property get RegPayRate()
        RegPayRate = m_RegPayRate
    end property

    public property let RegPayRate(p_RegPayRate)
        m_RegPayRate = p_RegPayRate
    end property

    public property get RegBillRate()
        RegBillRate = m_RegBillRate
    end property

    public property let RegBillRate(p_RegBillRate)
        m_RegBillRate = p_RegBillRate
    end property


    public property get ExpenseSummary
        'need placement id and company site
        dim placementId
        dim company
        if m_TotalExpenses = 0 then
```

```
            ExpenseSummary = SummaryObj("Expense", "Grey", me.PlacementId, company, m_TotalExepnses)
        else
            ExpenseSummary = SummaryObj("Expense", "Expecting", me.PlacementId, company, 0)
        end if

    end property

    private function SummaryObj(summarytype, style, placementid, companysite, total)
        ' summarytype - 'Expense'
        '             - 'Time'
        '
        ' style       - 'Expecting'
        '             - 'Grey'

        SummaryObj = "" &_
            "<p class=""" & summarytype & style & """>" &_
                "<span id=""" & lcase(summarytype) & "summary" &_
                    placementid & """ " &_
                "onclick=""" & lcase(summarytype) & "summary.open('" &_
                    placementid & "', '" & companysite & "')"">" &_
            "</span>$" &_
                TwoDecimals(total) &_
            "</p>"

    end function



    public property get TimeSummary
    end property









    public property get Status()
        Status = m_Status
    end property

    public property let Status(p_Status)
        m_Status = p_Status
    end property

    public property get NeedFinalTime()
        NeedFinalTime = m_NeedFinalTime
    end property

    public property let NeedFinalTime(p_NeedFinalTime)
        m_NeedFinalTime = p_NeedFinalTime
    end property

    public property get WCDescription()
        WCDescription = m_WCDescription
    end property

    public property let WCDescription(p_WCDescription)
        m_WCDescription = p_WCDescription
    end property

end class %>
<%
class cPlacements
'Private, class member variable
private m_Placements
private m_Company
private m_Customer
private m_Order
private m_Applicant
private m_FromDate
private m_ToDate

private m_ReportWhen

private m_NumberOfPages
private m_ItemsPerPage
private m_PageCount
private m_Page

Sub Class_Initialize()
    set m_Placements = Server.CreateObject ("Scripting.Dictionary")
End Sub
Sub Class_Terminate()
    set m_Placements = Nothing
End Sub

%>
<%
'Read the current placements
Public Property Get Placements()
    Set Placements = m_Placements
End Property

public property get Company()
    Company = m_Company
end property
public property let Company(p_Company)
    if len(p_Company) = 0 then
        p_Company = request.form("whichCompany")
        if len(p_Company) = 0 then
            p_Company = session("location")
        end if
    end if
    m_Company = p_Company
end property

public property get Customer()
    Customer = m_Customer
end property
public property let Customer(p_Customer)
    if len(p_Customer) = 0 then
        p_Customer = "@ALL" 'default
    end if
    m_Customer = Replace(p_Customer, "'", "''")
end property

public property get Order()
    Order = m_Order
end property
public property let Order(p_Order)
```

```
    if len(p_Order) = 0 then
        p_Order = request.form("whichOrder")
    end if
    m_Order = p_Order
end property

public property get Applicant()
    Applicant = m_Applicant
end property
public property let Applicant(p_Applicant)
    if len(p_Applicant) = 0 then
        p_Applicant = request.form("whichApplicant")
    end if
    m_Applicant = p_Applicant
end property

public property get FromDate()
    FromDate = m_FromDate
end property
public property let FromDate(p_FromDate)
    if isDate(p_FromDate) = false then
        p_FromDate = request.form("fromDate")
        if isDate(p_FromDate) = false then
            p_FromDate = CStr(Date() - 4)
        end if
    end if
    m_FromDate = p_FromDate
end property

public property get ToDate()
    ToDate = m_ToDate
end property
public property let ToDate(p_ToDate)
    if isDate(p_ToDate) = false then
        p_ToDate = request.form("toDate")
        if isDate(p_ToDate) = false then
            toDate = CStr(Date() + 1)
        end if
    end if
    m_ToDate = p_ToDate
end property
%>
<%

%>
<%
Public Property get NumberOfPages()
    NumberOfPages = m_NumberOfPages
end property'set page size

Public Property let NumberOfPages(p_NumberOfPages)
    m_NumberOfPages = p_NumberOfPages
end property

'Items Per Page
Public Property get ItemsPerPage()
    ItemsPerPage = m_ItemsPerPage
end property

Public Property let ItemsPerPage(p_ItemsPerPage)
    m_ItemsPerPage = p_ItemsPerPage
end property
```

```
'Read the current Customers
Public Property get PageCount()
    PageCount = m_PageCount
End Property

Public Property let PageCount(p_PageCount)
    m_PageCount = p_PageCount
End Property

'Page Number

public property get Page()
    Page = m_Page
end property

public property let Page(p_Page)
    p_Page = Trim(Replace(p_Page, ",", ""))
    if len(p_Page) = 0 then
        p_Page = request.form("WhichPage")
        if len(p_Page) = 0 then
            p_Page = "1" 'default
        end if
    end if
    m_Page = p_Page
end property
%>
<%

'#############  Public Functions ##############
    public function PageSelection()
        const StartSlide = 32 ' when to start sliding
        const StopSlide = 112 'when to stop sliding and show the smallests amount
        const SlideRange = 8 'the most pages to show minus this = smallest number to show aka the slide
        const TopPages = 25 'the most records to show

        dim maxPages, slidePages

        if m_Page <= StartSlide then
            maxPages = TopPages
        elseif m_Page > StartSlide and m_Page < StopSlide then
            maxPages = TopPages - (SlideRange - Cint(SlideRange * ((StopSlide - m_Page)/(StopSlide -
            StartSlide))))
        else
            maxPages = TopPages - SlideRange
        end if
        slidePages = cint((maxPages/2)+0.5)

        'check if we need to slide page navigation "window"
        if global_debug then
            output_debug("* navRecordsByPage(): nPageCount: " & m_PageCount & " *")
            output_debug("* navRecordsByPage(): nPage: " & m_Page & " *")
        end if

        dim startPage, stopPage
        if m_PageCount > maxPages then
            startPage = m_Page - slidePages
            stopPage = m_Page + slidePages

            'check if startPages is less than 1
            if startPage < 1 then
                startPage = 1
                stopPage = maxPages
            end if
            'check if stopPages is greater than total pages
            if stopPage > m_PageCount then
                stopPage = m_PageCount
                startPage = m_PageCount - slidePages
            end if
        else
            startPage = 1
            stopPage = m_PageCount
        end if

        rsQuery = request.serverVariables("QUERY_STRING")

        queryPageNumber = whichPage
        if queryPageNumber then
            rsQuery = Replace(rsQuery, "WhichPage=" & queryPageNumber & "&", "")
            rsQuery = Replace(rsQuery, "WhichPage=" & queryPageNumber, "")
            rsQuery = Replace(rsQuery, "WhichPage=", "")
        end if

        dim holdNavRecords : holdNavRecords = ""

        holdNavRecords = "<div id=""topPageRecords"" class=""navPageRecords"">" &_
                "<input name=""WhichPage"" id=""WhichPage"" type=""hidden"" value="""" />"

        holdNavRecords = holdNavRecords &_
        "<A HREF=""#"" onclick=""etc_refresh_page('1');"">First</A>"

        For i = startPage to stopPage
            holdNavRecords = holdNavRecords &_
                "<A HREF=""#"" onclick=""etc_refresh_page('" & i & "');"">&nbsp;"
            if i = m_Page then
                holdNavRecords = holdNavRecords &_
                    "<span style=""color:red"">" & i & "</span>"
            Else
                if (i = stopPage and i < m_PageCount) or (i = startPage and i > 1) then
                    holdNavRecords = holdNavRecords & "..."
                else
                    holdNavRecords = holdNavRecords & i
                end if
            end if
                holdNavRecords = holdNavRecords &_
                    "&nbsp;</A>"
        Next
        holdNavRecords = holdNavRecords &_
            "<A HREF=""#"" onclick=""etc_refresh_page('" & m_PageCount & "');"">Last</A>" &_
            "</div>"

        if len(holdNavRecords) > 0 then PageSelection = holdNavRecords

    end function

    public function CustomerSelection
        if len(me.Company & "") > 0 then
            sqlWhichCustomer = "SELECT DISTINCT Orders.Customer, Customers.CustomerName " &_
                "FROM (((Placements Placements LEFT OUTER JOIN Orders Orders ON
                Placements.Reference=Orders.Reference) " &_
                "LEFT OUTER JOIN WorkCodes WorkCodes ON Placements.WorkCode=WorkCodes.WorkCode) " &_
                "LEFT OUTER JOIN Applicants Applicants ON
                Placements.EmployeeNumber=Applicants.EmployeeNumber) " &_
                "LEFT OUTER JOIN Customers Customers ON Placements.Customer=Customers.Customer " &_
                "WHERE  (Placements.PlacementStatus=3 AND Placements.NeedFinalTime=TRUE OR
                Placements.PlacementStatus=0) " &_
                " ORDER BY Orders.Customer"

            set WhichCustomer = Server.CreateObject("ADODB.RecordSet")
            with WhichCustomer
                .CursorLocation = 3 ' adUseClient
                .Open sqlWhichCustomer, dsnLessTemps(getTempsDSN(me.Company))
            end with

            dim CurrentCustomer, strDisplayText, strBufferRepsponse
                CurrentCustomer = "@ALL"
                strBufferRepsponse = "" &_
                    "<div id=""topPageRecordsByCust"" class=""altNavPageRecords
                    navPageRecords""><strong>Select Customer: </strong>" &_
                    "<input name=""WhichCustomer"" id=""WhichCustomer"" type=""hidden"" value=""" &
                    thisCustomer & """>" &_
                    "<input name=""enteredby"" id=""enteredby"" type=""hidden"" value=""" &
                    showEnteredBy & """>" &_
                    "<input name=""assignedto"" id=""assignedto"" type=""hidden"" value=""" &
                    showAssignedTo & """>" &_
                    "<br><div id=""scrollCustomers"">" &_
                    "<A HREF=""#"" onclick=""etc_refresh_customer('" & CurrentCustomer & "')"">&nbsp;"

                if thisCustomer = CurrentCustomer then
                    strBufferRepsponse = strBufferRepsponse & "<span style=""color:red"">" &
                    CurrentCustomer & "</span>"
                Else
                    strBufferRepsponse = strBufferRepsponse &  CurrentCustomer
                end if
                strBufferRepsponse = strBufferRepsponse & "&nbsp;</A>"

            do while not WhichCustomer.Eof
                CurrentCustomer = WhichCustomer("Customer")

                strDisplayText = Replace(WhichCustomer("CustomerName"), "&", "&amp;")
                strDisplayText = Replace(strDisplayText, " ", "&nbsp;")

                strBufferRepsponse = strBufferRepsponse & "<A HREF=""#""
                onclick=""etc_refresh_customer('" & CurrentCustomer & "')"">&nbsp;"
                if thisCustomer = CurrentCustomer then
                    strBufferRepsponse = strBufferRepsponse & "<span style=""color:red"">" &
                    strDisplayText & "</span>"
                Else
                    strBufferRepsponse = strBufferRepsponse & strDisplayText
                end if
                strBufferRepsponse = strBufferRepsponse & "&nbsp;</A>"
                WhichCustomer.MoveNext

                linkNumber = linkNumber + 1
                if linkNumber > 10 and Not WhichCustomer.Eof then
                    linkNumber = 0
                    strBufferRepsponse = strBufferRepsponse & "<br>"
                end if
            loop
            strBufferRepsponse = strBufferRepsponse & "</div></div>"

            WhichCustomer.Close
            Set WhichCustomer = Nothing
            CustomerSelection = strBufferRepsponse
        end if
    end function
    public function ChooseJobOrder
        dim strDisplayText, strResponseBuffer

        if len(me.Company & "") > 0 then
            thisConnection = dsnLessTemps(getTempsDSN(me.Company))

            sqlWhichOrder = "SELECT Orders.Reference, Orders.JobDescription " &_
                    "FROM (((Placements Placements LEFT OUTER JOIN Orders Orders ON
                    Placements.Reference=Orders.Reference) " &_
                    "LEFT OUTER JOIN WorkCodes WorkCodes ON Placements.WorkCode=WorkCodes.WorkCode) " &_
                    "LEFT OUTER JOIN Applicants Applicants ON
                    Placements.EmployeeNumber=Applicants.EmployeeNumber) " &_
                    "LEFT OUTER JOIN Customers Customers ON Placements.Customer=Customers.Customer " &_
                    "WHERE  (Placements.PlacementStatus=3 AND Placements.NeedFinalTime=TRUE OR
                    Placements.PlacementStatus=0) " &_
                    "AND (Orders.Customer='" & thisCustomer & "') " &_
                    "ORDER BY Orders.Customer, Applicants.LastnameFirst"

            Set rsWhichOrder = Server.CreateObject("ADODB.RecordSet")
            with rsWhichOrder
                .CursorLocation = 3 ' adUseClient
                .Open sqlWhichOrder, thisConnection
            end with

            dim CurrentOrder
                CurrentOrder = "@ALL"

            strResponseBuffer = "" &_
                "<div id=""topPageRecordsByOrder"" class=""altNavPageRecords
                navPageRecords""><strong>Job Orders: </strong>" &_
                "<input name=""WhichOrder"" id=""WhichOrder"" type=""hidden"" value=""" & thisOrder &
                """>" &_
                "<br><div id=""scrollCustomers"">" &_
                "<A HREF=""#"" onclick=""etc_refresh_order('" & CurrentOrder & "')"">&nbsp;"

                if thisOrder = CurrentOrder or thisOrder = "" then
                    strResponseBuffer = strResponseBuffer & "<span style=""color:red"">" & CurrentOrder
                    & "</span>"
                Else
                    strResponseBuffer = strResponseBuffer & CurrentOrder
                end if
                strResponseBuffer = strResponseBuffer & "&nbsp;</A>"

            do while not rsWhichOrder.Eof
                CurrentOrder = rsWhichOrder("Reference")
                strDisplayText = Replace(rsWhichOrder("JobDescription"), "&", "&amp;")
                strDisplayText = Replace(strDisplayText, " ", "&nbsp;")

                strResponseBuffer = strResponseBuffer & "<A HREF=""#"" onclick=""etc_refresh_order('" &
                CurrentOrder & "')"">&nbsp;"

                if trim(thisOrder) = trim(CurrentOrder) then
                    strResponseBuffer = strResponseBuffer & "<span style=""color:red"">" &
                    strDisplayText & "</span>"
                Else
                    strResponseBuffer = strResponseBuffer & strDisplayText
                end if
                strResponseBuffer = strResponseBuffer & "&nbsp;</A>"
                rsWhichOrder.MoveNext

                linkNumber = linkNumber + 1
                if linkNumber > 10 and Not rsWhichOrder.Eof then
                    linkNumber = 0
                    strResponseBuffer = strResponseBuffer & "<br>"
                end if
            loop
            strResponseBuffer = strResponseBuffer & "</div></div>"

            rsWhichOrder.Close
            Set rsWhichOrder = Nothing
            ChooseJobOrder = strResponseBuffer
        end if
    end function

    public function GetActivePlacements()
        dim strSql
        strSql = "SELECT Orders.Customer, Orders.Reference, Placements.EmployeeNumber,
        Placements.StartDate, " &_
                "Placements.PStopDate, Customers.CustomerName, Applicants.LastnameFirst,
                Orders.JobNumber, " &_
                "Placements.WorkCode, Placements.RegPayRate, Placements.RegBillRate,
                Placements.PlacementStatus, " &_
                "Placements.PlacementID, Placements.NeedFinalTime, WorkCodes.Description " &_
                "FROM (((Placements Placements LEFT OUTER JOIN Orders Orders ON
                Placements.Reference=Orders.Reference) " &_
                "LEFT OUTER JOIN WorkCodes WorkCodes ON Placements.WorkCode=WorkCodes.WorkCode) " &_
                "LEFT OUTER JOIN Applicants Applicants ON
                Placements.EmployeeNumber=Applicants.EmployeeNumber) " &_
                "LEFT OUTER JOIN Customers Customers ON Placements.Customer=Customers.Customer " &_
                "WHERE  (Placements.PlacementStatus=3 AND Placements.NeedFinalTime=TRUE OR
                Placements.PlacementStatus=0) " &_
                " ORDER BY Orders.Customer, Applicants.LastnameFirst"


        GetAllCustomers = LoadData (strSQL)
    end function



'#############  Private Functions ##############

    'Takes a recordset
    'Fills the object's properties using the recordset
    private function FillFromRS(p_RS)
        p_RS.PageSize = m_ItemsPerPage
        m_PageCount = p_RS.PageCount

        if m_Page < 1 Or m_Page > m_PageCount then
            m_Page = 1
        end if

        if not p_RS.eof then p_RS.AbsolutePage = m_Page

        dim thisPlacement
        do while not ( p_RS.eof Or p_RS.AbsolutePage <> m_Page )
            set thisPlacement = New cPlacement
            with thisPlacement
                .CustomerCode    = p_RS.fields("Customer").Value 'Customer
                .CustomerName    = p_RS.fields("CustomerName").Value 'CustomerName
                .Reference       = p_RS.fields("Reference").Value
                .EmployeeNumber  = p_RS.fields("EmployeeNumber").Value
                .StartDate       = p_RS.fields("StartDate").Value 'Placement.StartDate
                .PStopDate       = p_RS.fields("PStopDate").Value 'Placements.PStopDate
                .PlacementId      = p_RS.fields("PlacementID")
                .LastnameFirst    = p_RS.fields("LastnameFirst").Value 'Applicants..LastnameFirst
                .JobNumber        = p_RS.fields("JobNumber").Value 'Orders.JobNumber
                .WorkCode         = p_RS.fields("WorkCode").Value 'WorkCode
                .RegPayRate       = TwoDecimals(p_RS.fields("RegPayRate").Value)
                .RegBillRate      = TwoDecimals(p_RS.fields("RegBillRate").Value)
                .Status           = p_RS.fields("PlacementStatus").Value
                .NeedFinalTime    = p_RS.fields("NeedFinalTime").Value
                .WCDescription    = p_RS.fields("Description").Value 'WorkCodes.Description
            end with
            m_Placements.Add thisPlacement.PlacementId, thisPlacement

            p_RS.movenext
        loop
    End Function

    Private Function LoadData(p_strSQL)
        dim rs
        set rs = GetRSfromDB(p_strSQL, dsnLessTemps(getTempsDSN(me.Company)))
        FillFromRS(rs)
        LoadData = rs.recordcount
        rs. close
        set rs = nothing
    End Function

end class
%>
```