```
'----------------------------------------------------------------------
'
' Copyright (c) Jedat China Software Inc. All rights reserved.
'
' Abstract:
' weekly_report.vbs
'
' Usage:
'
' Examples:
'
'----------------------------------------------------------------------
Option Explicit

Const ForReading = 1
Const ForWriting = 2
Const ForAppending = 8

Const xlCenter = -4108
Const xlLeft = -4131
Const xlRight = -4152

' configurable via "..\config\parameter.txt"
Const IDX_PrjNum = 0
Const IDX_PrjName = 1
Const IDX_CurWeekPrjTime = 2
Const IDX_TotalPrjTime = 3
Const PRJ_MaxIdx = 3
Const MaxPrjSuffixNum = 93015
Const MyEmail = "weeklyreport@jedat-soft.com.cn"

Dim StartWeekName : StartWeekName = GetConfigValueOf("StartWeekName")
Dim WeekSumName : WeekSumName = GetConfigValueOf("WeekSumName")
Dim ForceGetWeekSum : ForceGetWeekSum = GetConfigValueOf("ForceGetWeekSum")
Dim AttachmentWild : AttachmentWild = GetConfigValueOf("AttachmentWild")
Dim StartWeekCol
Dim DataHome, ModuleHome, LogHome, UserListHome, RecipientListHome
Dim ThisDate
'On Error Resume Next

WScript.Quit(VBSMain)

Function VBSMain()
    Dim argc
    Dim argv
    argc = WScript.Arguments.Count
    Set argv = WScript.Arguments
    Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystemObject")
    DataHome = objFSO.GetParentFolderName(objFSO.GetParentFolderName(WScript.ScriptFullname))
    LogHome = DataHome & "\logs"
    ModuleHome = DataHome & "\modules"
    UserListHome = DataHome & "\users"
    RecipientListHome = DataHome & "\recipient"

    Dim arrPrjInfoList()
    ThisDate = Now
    ThisDate = CDate("2013-9-17")
    Call GetLatestProjectInfo(GetProjectCodeFile(GetConfigValueOf("ProjectCodeFileDir")),
    arrPrjInfoList)

    Dim prevfile: prevfile = GetPrevStatisticDataFilePath(ThisDate)
    Dim prevweekname
    Dim prevflg: prevflg = GetPrevProjectTimeInfo(prevfile,  prevweekname, arrPrjInfoList)
    Call CalculateProjectTime(GetXlsDataPath(ThisDate), prevflg, prevweekname, arrPrjInfoList)

    Dim reportfile: reportfile = GetStatisticDataFilePath(ThisDate)
    Call SavePrjectInfoToFile(arrPrjInfoList, reportfile)
    'Call SendReportFile(reportfile, GetRecipientList)
    VBSMain = Err.Number
End Function

'Sub-Routines
Function GetYYYYIW(ByVal pDate)
    Dim iw: iw = DatePart("ww", pDate)
    If CInt(iw) < 10 Then
        iw = "0" & iw
    End If
    GetYYYYIW = Year(pDate) & iw
End Function

Function GetYYYYIWRev(ByVal pDate)
    Dim yyyyiw: yyyyiw = GetYYYYIW(pDate)
    'Set to last week if date is (or earlier than) Wed.
    If Weekday(pDate) <= vbWednesday Then
        yyyyiw = CStr(CLng(yyyyiw) - 1)
    End If
    GetYYYYIWRev = yyyyiw
End Function

Function GetXlsDataPath(ByVal pDate)
    Dim strTargetPath: strTargetPath = DataHome
    Dim objFSO: Set objFSO = CreateObject( "Scripting.FileSystemObject" )
    Dim objFolder: Set objFolder = objFSO.GetFolder(strTargetPath)
    GetXlsDataPath = objFolder.Path & "\received\" & GetYYYYIWRev(pDate)
End Function

Function SetPrjectTime(ByRef pArrPrjInfoList, ByVal pPrjCodeNum, ByVal pTotalPrjTime, ByVal
pCurWeekPrjTime)
    Dim dblTotalHours, dblCurWeekHours
    Dim i
    For i = LBound(pArrPrjInfoList) To UBound(pArrPrjInfoList)
        If pArrPrjInfoList(i, IDX_PrjNum) = pPrjCodeNum Then
            dblCurWeekHours = CDbl(pArrPrjInfoList(i, IDX_CurWeekPrjTime))
            dblTotalHours = CDbl(pArrPrjInfoList(i, IDX_TotalPrjTime))
            pArrPrjInfoList(i, IDX_CurWeekPrjTime) = dblCurWeekHours + CDbl(pCurWeekPrjTime)
            pArrPrjInfoList(i, IDX_TotalPrjTime) = dblTotalHours + CDbl(pTotalPrjTime)
            Exit For
        End If
    Next
End Function

Sub SetCurrentWeekName(ByVal pCurWeekName, ByRef pArrPrjInfoList)
    Dim i
    For i = LBound(pArrPrjInfoList) To UBound(pArrPrjInfoList)
        If pArrPrjInfoList(i, IDX_CurWeekPrjTime) = "工数" Then
            pArrPrjInfoList(i, IDX_CurWeekPrjTime) = pArrPrjInfoList(i, IDX_CurWeekPrjTime) &
             _
                vbCrlf & "(" & pCurWeekName & ")"
        End If
    Next
End Sub

Sub CalculateProjectTime(ByVal pXlsDataPath,  ByVal pPrevFlg, ByVal pPrevWeekname, ByRef
pArrPrjInfoList)
On Error Resume Next
Const WeeklySheet = "周统计表"
Const ProjectNumCol = 4 ' Column "D"
Const ProjectNumName = "工程编号"
Const ProjectCodeName = "工程代码"
Const ProjectCostName = "成本代码"

Const MaxRowNum = 50

    Dim strPrjNum, strStartWeekName, strWeekSumName
    Dim lPrjNum, lPrjNumStartAtRow, lTotalPrjHours, lCurWeekPrjHours
    Dim lWeekNameRow, lWeekNameStartCol, lWeekSumCol
    Dim lRowNum, lColNum
    Dim lLastWeekCol
    Dim lMinPrjNum
    Dim lOneRowCol
    Dim lCurWeekCol, szCurWeekName
    Dim objXlsApp: Set objXlsApp = CreateExcelObject
    Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystemObject")
    Dim objXlsFolder: Set objXlsFolder = objFSO.GetFolder(pXlsDataPath)
    'Dim lWeekColOffset: lWeekColOffset = (GetWeekCountByFolders - 1)
    'Dim lEndProjectNum

    '201303 \ 100 = 2013
    lMinPrjNum = (CLng(objXlsFolder.Name) \ 100 mod 100) * 1000000
    'lEndProjectNum = lMinPrjNum + MaxPrjSuffixNum

    Dim objOneFile, objWorkBook, objWeeklySheet, strFileName
    For Each objOneFile In objXlsFolder.Files
        strFileName = UCase(objOneFile.Path)
        If InStr(strFileName, AttachmentWild) And Right(objOneFile.Path, 4) = ".XLS" Then
            msgbox strFileName
            Set objWorkBook = objXlsApp.Workbooks.Open(objOneFile.Path)
            Set ObjWeeklySheet = objWorkBook.Sheets(WeeklySheet)

            For lRowNum = 1 To 10
                strPrjNum = ObjWeeklySheet.Cells(lRowNum, ProjectNumCol)
                If strPrjNum = ProjectNumName Or strPrjNum = ProjectCodeName Or strPrjNum =
                ProjectCostName Then
                    lWeekNameRow = lRowNum
                    lPrjNumStartAtRow = lRowNum + 1
                    Exit For
                End If
            Next

            For lColNum = 1 To 50
                strStartWeekName = Trim(ObjWeeklySheet.Cells(lWeekNameRow, lColNum))
                If strStartWeekName = StartWeekName Then
                    StartWeekCol = lColNum
                    Exit For
                End If
            Next

            For lColNum = StartWeekCol To (StartWeekCol + 50)
                strWeekSumName = Trim(ObjWeeklySheet.Cells(lWeekNameRow, lColNum))
                If strWeekSumName = WeekSumName Then
                    lWeekSumCol = lColNum
                    Exit For
                End If
            Next

            'lCurWeekCol = StartWeekCol + GetWeekColOffset2
            If pPrevFlg = True And Len(pPrevWeekname) > 0 Then
                lCurWeekCol = StartWeekCol + GetWeekColOffsetByWeekName(StartWeekName,
                pPrevWeekname)
            Else
                lCurWeekCol = StartWeekCol + GetWeekColOffset
            End If
            If IsEmpty(szCurWeekName) Then
                szCurWeekName = ObjWeeklySheet.Cells(lWeekNameRow, lCurWeekCol)
                Call SetCurrentWeekName(szCurWeekName, pArrPrjInfoList)
            End If

            For lRowNum = lPrjNumStartAtRow To MaxRowNum
                strPrjNum = ObjWeeklySheet.Cells(lRowNum, ProjectNumCol)
                If CLng(strPrjNum) > lMinPrjNum Then
                    'BEGIN
    '               lTotalPrjHours = ObjWeeklySheet.Cells(lRowNum, StartWeekCol)
    '               If lWeekColOffset > 0 Then
    '                   lCurWeekPrjHours = ObjWeeklySheet.Cells(lRowNum, (StartWeekCol +
    lWeekColOffset))
    '                   For lOneRowCol = (StartWeekCol + 1) To (StartWeekCol +
    lWeekColOffset - 1)
    '                       lTotalPrjHours = CDbl(lTotalPrjHours) +
    ObjWeeklySheet.Cells(lRowNum, lOneRowCol)
    '                   Next
    '                   lTotalPrjHours = CDbl(lTotalPrjHours) + lCurWeekPrjHours
    '               End If
                    'END
                    lCurWeekPrjHours = ObjWeeklySheet.Cells(lRowNum, lCurWeekCol)
                    If (Not ForceGetWeekSum) And pPrevFlg = True And Len(pPrevWeekname) > 0
                    Then
                        lTotalPrjHours = lCurWeekPrjHours
                    Else
                        lTotalPrjHours = ObjWeeklySheet.Cells(lRowNum, lWeekSumCol)
                    End If

                    Call SetPrjectTime(pArrPrjInfoList, CLng(strPrjNum), CDbl(lTotalPrjHours
                    ), CDbl(lCurWeekPrjHours))

                    'If CLng(strPrjNum) = lEndProjectNum Then
                    '   Exit For
                    'End If
```

```
                    If (CLng(strPrjNum)  Mod 100000) = MaxPrjSuffixNum Then
                        Exit For
                    End If
                End If
            Next ' lRowNum

            Set objWorkBook = Nothing
            Set objWeeklySheet = Nothing
        End If
    Next ' objOneFile

    'strPrjNum = objWorkBook.Sheets("周统计表").Range("D4")
    'msgbox strPrjNum
'    Sheets("周统计表").Select
'    Range("D5").Select
'    ActiveCell.FormulaR1C1 = "a"
'    With ActiveCell.Characters(Start:=1, Length:=1).Font
'        .Name = "宋体"
'        .FontStyle = "常规"
'        .Size = 10
'        .Strikethrough = False
'        .Superscript = False
'        .Subscript = False
'        .OutlineFont = False
'        .Shadow = False
'        .Underline = xlUnderlineStyleNone
'        .ColorIndex = xlAutomatic
'    End With
'    Range("D6").Select

    Call FreeExcelObject(objXlsApp)
End Sub

Function GetWeekCountByFolders()
    Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystemObject")
    Dim objHomeFolder, objRecvFolder
    Set objHomeFolder = objFSO.GetFolder(DataHome)
    Set objRecvFolder = objFSO.GetFolder(objHomeFolder.Path & "\received")

    Dim objRecvSubFolders: Set objRecvSubFolders = objRecvFolder.SubFolders
    GetWeekCountByFolders = objRecvSubFolders.Count
End Function

Function GetProjectCodeFile(ByVal pPrjCodeFilePath)
    Dim strTargetFile
    Dim dMod: dMod = 0
    Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystemObject")
    Dim objFolder: Set objFolder = objFSO.GetFolder(pPrjCodeFilePath)
    Dim objFile
    For Each objFile in objFolder.Files
        If InStr(LCase(objFile.Name), "projectcodes") _
        And LCase(objFSO.GetExtensionName(objFile.Name)) = "xls" Then
            If dMod = 0 Then
                dMod = objFile.DateLastModified
                strTargetFile = objFile.Path
            Else
                If objFile.DateLastModified > dMod Then
```

```
                    dMod = objFile.DateLastModified
                    strTargetFile = objFile.Path
                End If
            End If
        End If
    Next
    GetProjectCodeFile = strTargetFile
End Function

Function GetLatestProjectInfo(ByVal pPrjCodeFile, ByRef pArrPrjInfoList)
    Const MaxRow = 100
    Const MaxBlankRows = 3
    Const PrjCodeCol = 2
    Const PrjNameCol = 17

    Dim iRet: iRet = 0

    Dim lCurRow, lBlankRows
    Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystemObject")
    If Not objFSO.FileExists(pPrjCodeFile) Then
        GetLatestProjectInfo = 0
        Exit Function
    End If

    Dim objXlsApp: Set objXlsApp = CreateExcelObject
    Dim objWorkBook: Set objWorkBook = objXlsApp.Workbooks.Open(pPrjCodeFile)
    Dim lPrjCnt: lPrjCnt = 0
    Dim objWorkSheet

    'Get total project count including each sheet name
    For Each objWorkSheet In objWorkBook.Sheets
    With objWorkSheet
        lBlankRows = 0
        If .Cells(2,PrjCodeCol) = "编码" Then
            lPrjCnt = lPrjCnt + 1 ' sheet name
            For lCurRow = 4 To MaxRow
                If CLng(.Cells(lCurRow, PrjCodeCol)) = 0 Then
                    lBlankRows = lBlankRows + 1
                    If lBlankRows = MaxBlankRows Then Exit For
                Else
                    lBlankRows = 0
                    'msgbox .Cells(lCurRow, PrjCodeCol) & .Cells(lCurRow,PrjNameCol)
                    lPrjCnt = lPrjCnt + 1
                End If
            Next
        End If
    End With
    Next

    'Resize 2D-array and insert values
    ReDim pArrPrjInfoList(lPrjCnt-1, PRJ_MaxIdx)
    lPrjCnt = 0
    For Each objWorkSheet In objWorkBook.Sheets
    With objWorkSheet
        lBlankRows = 0
        If .Cells(2,PrjCodeCol) = "编码" Then
            'pArrPrjInfoList(lPrjCnt, IDX_PrjNum) = Replace(.Name, "编码", "")
            pArrPrjInfoList(lPrjCnt, IDX_PrjNum) = .Name
            pArrPrjInfoList(lPrjCnt, IDX_PrjName) = "工程名"
            pArrPrjInfoList(lPrjCnt, IDX_CurWeekPrjTime) = "工数"
            pArrPrjInfoList(lPrjCnt, IDX_TotalPrjTime) = "工数" & vbCrlf & "(累计)"
            lPrjCnt = lPrjCnt + 1
            For lCurRow = 4 To MaxRow
                If CLng(.Cells(lCurRow, PrjCodeCol)) = 0 Then
                    lBlankRows = lBlankRows + 1
                    If lBlankRows = MaxBlankRows Then Exit For
                Else
                    lBlankRows = 0
                    pArrPrjInfoList(lPrjCnt, IDX_PrjNum) = .Cells(lCurRow, PrjCodeCol)
                    pArrPrjInfoList(lPrjCnt, IDX_PrjName) = .Cells(lCurRow, PrjNameCol)
                    pArrPrjInfoList(lPrjCnt, IDX_CurWeekPrjTime) = 0
                    pArrPrjInfoList(lPrjCnt, IDX_TotalPrjTime) = 0
                    lPrjCnt = lPrjCnt + 1
                End If
            Next
        End If
    End With
    Next
    Call FreeExcelObject(objXlsApp)
    GetLatestProjectInfo = 1
End Function

Function GetAllUsers()
    Dim allusers
    Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystemObject")
    Dim objFolder: Set objFolder = objFSO.GetFolder(UserListHome)
    Dim objOneList, objListFile
    For Each objOneList In objFolder.Files
        If Not Left(LCase(objOneList.Name), 3) = "bak" Then
            Set objListFile = objFSO.OpenTextFile(objOneList.Path, 1, False)
            If IsEmpty(allusers) Then
                allusers = objListFile.ReadAll
            Else
                allusers = allusers & objListFile.ReadAll
            End If
        End If
    Next

    Set objFSO = Nothing
    Set objFolder = Nothing
    GetAllUsers = allusers
End Function

Function FindOneFile(ByVal pPath, ByVal pName)
    Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystemObject")
    Dim objFolder: Set objFolder = objFSO.GetFolder(pPath)
    Dim objOneFile

    For Each objOneFile In objFolder.Files
        If InStr(LCase(objOneFile.Name), pName) Then
            FindOneFile = objOneFile.Path
            Exit For
        End If
    Next
    Set objFSO = Nothing
    Set objFolder = Nothing
End Function

Function GetAbsentUserList(ByVal pDate, ByRef pUserList)
    Dim allusers: allusers = GetAllUsers
    Dim xlspath: xlspath = GetXlsDataPath(pDate)
    Dim prefix
    Dim i, cnt
    Dim arrUsers: arrUsers = Split(allusers, vbCrlf)
    Dim absentusers
    Dim user_attachment
    For i = LBound(arrUsers) To UBound(arrUsers)
        If Len(Trim(arrUsers(i))) > 0 Then
            prefix = Trim(Left(arrUsers(i), InStr(arrUsers(i), "=")-1))
            'wscript.echo xlspath & " ### :" & prefix
            user_attachment = FindOneFile(xlspath, prefix)
            If Len(user_attachment) = 0 Then
                'wscript.echo prefix & ", attachment: " & user_attachment
                If IsEmpty(absentusers) Then
                    absentusers = prefix
                Else
                    absentusers = absentusers & ";" & prefix
                End If
            End If
        End If
    Next
    pUserList = Split(absentusers, ";")
    GetAbsentUserList = (UBound(pUserList) + 1)
End Function

Function SavePrjectInfoToFile(ByVal pArrPrjInfoList, ByVal pFilePath)
    Dim objXlsApp: Set objXlsApp = CreateExcelObject
    Dim objWorkbook: Set objWorkbook = objXlsApp.Workbooks.Add()
    Dim objWeeklySheet: Set ObjWeeklySheet = objWorkBook.Sheets(1)
    Dim objAbsenceSheet: Set objAbsenceSheet = objWorkBook.Sheets(2)
    Dim i, strBuf
    Dim absentlist
    Dim absentcnt: absentcnt = GetAbsentUserList(ThisDate, absentlist)
    Dim cellval
    With objWeeklySheet
        .Name = GetYYYYIWRev(ThisDate) & _
            Split(pArrPrjInfoList(0, IDX_CurWeekPrjTime), vbCrlf)(1) & _
            "统计结果"
        .Cells(1,1) = "统计时间:"
        .Cells(1,2) = .Name
        .Cells(1,2).HorizontalAlignment = xlCenter
        .Cells(2,1) = "生成日期:" & CStr(Now)
        .Cells(2,2) = "共有(" & absentcnt & ")人未参与统计"
        .Cells(2,2).HorizontalAlignment = xlCenter
        .Hyperlinks.Add .Range("B2"), "", "'" & .Cells(2,2) & "'!A1", "点击查看详情"

        'Resize column width to fit on one-page-printing, unit is mm
        .Columns(1).ColumnWidth = 18 ' 工程编号
        .Columns(2).ColumnWidth = 45.5 ' 工程名称
        .Columns(3).ColumnWidth = 6.5 '工数(累计)
        .Columns(4).ColumnWidth = 6.5 '工数(xx周)

'Range("C5").AddComment
'    Range("C5").Comment.Visible = False
'    Range("C5").Comment.Text Text:="jia.tao:" & Chr(10) & "abc"
        For i = LBound(pArrPrjInfoList) To UBound(pArrPrjInfoList)
        ' Start at Row3, Row1=main title, Row2=sub title
            .Cells(i+3, IDX_PrjNum+1) = pArrPrjInfoList(i, IDX_PrjNum)
            .Cells(i+3, IDX_PrjName+1) = pArrPrjInfoList(i, IDX_PrjName)
            .Cells(i+3, IDX_CurWeekPrjTime+1) = pArrPrjInfoList(i, IDX_CurWeekPrjTime)
            .Cells(i+3, IDX_TotalPrjTime+1) = pArrPrjInfoList(i, IDX_TotalPrjTime)

            cellval = .Cells(i+3, IDX_TotalPrjTime+1)
            If IsNumeric(cellval) Then
                If CDbl(cellval) > 0 Then
                    Dim hm
                    'a Mod b <=> a - (b * (a \ b))
                    hm = CDbl(cellval) / (21.5 * 8)
                    .Cells(i+3, IDX_TotalPrjTime+1).AddComment "约： " & FormatNumber(hm,3,-1)
                     & "人月"
                End If
            End If
        Next
    End With

    With objAbsenceSheet
        .Columns(1).ColumnWidth = 15 ' 序号
        .Columns(2).ColumnWidth = 45 ' 邮件地址
        .Name = objWeeklySheet.Cells(2,2)
        .Cells(1,1) = "返回"
        .Hyperlinks.Add .Range("A1"), "", "'" & objWeeklySheet.Name & "'!B2",
        "点击返回统计结果"
        .Cells(1,2) = .Name
        .Cells(2,1) = "序号"
        .Cells(2,1).HorizontalAlignment = xlRight

        .Cells(2,2) = "邮件地址"
        For i = LBound(absentlist) To UBound(absentlist)
            .Cells(i+3, 1) = (i+1)
            .Cells(i+3, 2) = absentlist(i)
        Next
    End With

    objXlsApp.DisplayAlerts = False
    objWorkbook.SaveAs(pFilePath)

    Call FreeExcelObject(objXlsApp)
End Function

Function GetStatisticDataFilePath(ByVal pDate)
    Dim strTargetPath: strTargetPath = DataHome
    Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystemObject")
    Dim objFolder: Set objFolder = objFSO.GetFolder(strTargetPath)
    GetStatisticDataFilePath = objFolder.Path & "\sent\" & GetYYYYIWRev(pDate) & ".xls"
End Function

Function GetPrevStatisticDataFilePath(ByVal pDate)
```

```
    Dim strTargetPath: strTargetPath = DataHome
    Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystemObject")
    Dim objFolder: Set objFolder = objFSO.GetFolder(strTargetPath)
    GetPrevStatisticDataFilePath = objFolder.Path & "\sent\" & GetYYYYIWRev(DateAdd("ww", -1,
     pDate)) & ".xls"
End Function

Function CreateExcelObject()
    Set CreateExcelObject = CreateObject("Excel.Application")
End Function

Sub FreeExcelObject(ByRef pExcelApp)
    pExcelApp.Quit
    Set pExcelApp = Nothing
End Sub

Function GetRecipientList()
    Dim recipientlist
    Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystemObject")
    Dim objFolder: Set objFolder = objFSO.GetFolder(RecipientListHome)
    Dim objOneList, objListFile, strLine
    For Each objOneList In objFolder.Files
        If Not Left(LCase(objOneList.Name), 3) = "bak" Then
            Set objListFile = objFSO.OpenTextFile(objOneList.Path, 1, False)
            Do While objListFile.AtEndOfStream <> True
                strLine = objListFile.ReadLine
                If Len(Trim(strLine)) > 0 Then
                    If IsEmpty(recipientlist) Then
                        recipientlist = strLine
                    Else
                        recipientlist = recipientlist & ";" & strLine
                    End If
                End If
            Loop
            objListFile.Close
        End If
    Next
    GetRecipientList = recipientlist
End Function

Sub SendReportFile(ByVal pFile, ByVal pToList)
    Dim objShell: Set objShell = Wscript.CreateObject("WScript.Shell")
    Dim strSubj, strBody
    strSubj = "周报统计_" & CStr(Date)
    strBody = "详见附件" ' """""" '<-Empty Str("")
    'objShell.Run
    objShell.Run ModuleHome & _
        "\jcssendmail.vbs " & _
            MyEmail & " " & _
            pToList & " " & _
            """"" " & _
            "jia.tao@jedat-soft.com.cn " & _
            strSubj & " " & _
            strBody & " " & _
            """" & pFile & """"
    'wscript.echo "To " & vbCrlf & pToList
    Set objShell = Nothing
```

```
End Sub

Function GetConfigValueOf(ByVal pName)
    Dim strVal, strLine
    Dim aryParam, aryValue
    Dim strConfigfile: strConfigfile = "..\config\parameter.txt"
    Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystemObject")

    Dim objParamFile: Set objParamFile = objFSO.OpenTextFile(strConfigfile, 1, False)
    Do While objParamFile.AtEndOfStream <> True
        strLine = objParamFile.ReadLine
        aryParam = Split(strLine, "=")
        strVal = Trim(aryParam(0))
        If LCase(strVal) = LCase(pName) Then
            strVal = Trim(aryParam(1))
            aryValue = Split(strVal, """")
            strVal = aryValue(UBound(aryValue)-1)
            Exit Do
        End If
    Loop
    objParamFile.Close
    GetConfigValueOf = strVal
End Function

Function GetWeekColOffsetByWeekName(ByVal pStartWeekname, ByVal pPrevWeekname)
    Dim lOffset: lOffset = 0
    If Len(pStartWeekname) > 1 And Len(pPrevWeekname) > 1 Then
        Dim startwk: startwk = Trim(pStartWeekname)
        Dim prevwk: prevwk = Trim(pPrevWeekname)
        Dim x: x  = CLng(Left(prevwk, Len(prevwk)-1))
        Dim x0: x0  = CLng(Left(startwk, Len(startwk)-1))
        lOffset = x - x0 + 1
    End If
    GetWeekColOffsetByWeekName = lOffset
End Function

Function GetWeekColOffset()
    Dim strStartDate : strStartDate = GetConfigValueOf("StartDate")
    Dim iMon: iMon = CInt(Left(strStartDate, InStr(strStartDate, "-")-1))
    Dim iofs, lastyyyyiw
    If (iMon <= Month(Now)) Then
        iofs = CLng(GetYYYYIW(Now)) - CLng(GetYYYYIW(strStartDate))
    Else
        lastyyyyiw = GetYYYYIW(CDate(Year(Now)-1 & "-12-31"))
        iofs = CLng(lastyyyyiw) - CLng(GetYYYYIW(CDate(Year(Now)-1 & "-" & strStartDate)))
        iofs = iofs + CLng(GetYYYYIW(Now)) - Year(Now)*100
    End If
    If iofs > 0 Then iofs = iofs -1
    GetWeekColOffset = iofs
End Function

Function GetWeekColOffset2()
    Dim strStartDate : strStartDate = GetConfigValueOf("StartDate")
    Dim iMon: iMon = CInt(Left(strStartDate, InStr(strStartDate, "-")-1))
    Dim iofs, lastyyyyiw
    If (iMon <= Month(Now)) Then
        iofs = DateDiff("ww", Year(Now) & "-" & strStartDate, Now)
    Else
        lastyyyyiw = GetYYYYIW(CDate(Year(Now)-1 & "-12-31"))
        iofs = DateDiff("ww", Year(Now)-1 & "-" & strStartDate, Now)
    End If
    If iofs > 0 Then iofs = iofs -1
    GetWeekColOffset2 = iofs
End Function

Function GetPrevProjectTimeInfo(ByVal pPrevFilename, ByRef pPrevWeekname, ByRef
pArrPrjInfoList)
On Error Resume Next
    pPrevWeekname = ""
    Dim ret: ret = False
    GetPrevProjectTimeInfo = ret
    Dim objXlsApp: Set objXlsApp = CreateExcelObject
    Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystemObject")

    If Not objFSO.FileExists(pPrevFilename) Then
        Exit Function
    End If

    Dim objWorkBook : Set objWorkBook = objXlsApp.Workbooks.Open(pPrevFilename)
    Dim ObjStatisResultSheet : Set ObjStatisResultSheet = objWorkBook.Sheets(1)

    Dim strPrevWeekname : strPrevWeekname = ObjStatisResultSheet.Cells(3, 3)

    Dim iStartPos, iEndPos
    iStartPos = InStr(strPrevWeekname, "(")+1
    iEndPos = InStr(strPrevWeekname, ")")
    pPrevWeekname = Mid(strPrevWeekname, iStartPos,  iEndPos - iStartPos)

    Dim strPrevPrjNum, strCurPrjNum
    Dim lPrevPrjNum, lCurPrjNum
    Dim dblPrevTotalPrjTime
    Dim i, j
    For i = LBound(pArrPrjInfoList)+1 To UBound(pArrPrjInfoList)
        strCurPrjNum = CLng(pArrPrjInfoList(i, IDX_PrjNum))
        lCurPrjNum = CLng(strCurPrjNum)
        If (lCurPrjNum > 0) Then
            For j =  LBound(pArrPrjInfoList)+3 To UBound(pArrPrjInfoList)+3
                'msgbox pArrPrjInfoList(i, IDX_PrjNum) & pArrPrjInfoList(i, IDX_PrjName) &
                pArrPrjInfoList(i, IDX_TotalPrjTime)
                strPrevPrjNum = CLng(ObjStatisResultSheet.Cells(j, (IDX_PrjNum+1)))
                lPrevPrjNum = CLng(strPrevPrjNum)
                If lPrevPrjNum > 0 And lPrevPrjNum = lCurPrjNum Then
                    dblPrevTotalPrjTime = CDbl(ObjStatisResultSheet.Cells(j, (
                    IDX_TotalPrjTime+1)))
                    'msgbox lPrevPrjNum & "," & strCurPrjNum & "=>" & dblPrevTotalPrjTime
                    'Exit For
                    pArrPrjInfoList(i, IDX_TotalPrjTime) = dblPrevTotalPrjTime
                End If
            Next
        End If
    Next

    Call FreeExcelObject(objXlsApp)
    ret = True
    GetPrevProjectTimeInfo = ret
End Function
```