```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  .....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'**************************************************************************************
'**************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  *************************
'**************************************************************************************
'**************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'**************************************************************************************
'**************************************************************************************
'******************************  PRIVATE MEMBERS  ******************************
'**************************************************************************************
'**************************************************************************************

'***************************************
'MODULE/CLASS-WIDE CONSTANTS
'***************************************
'Strings...
Private Const m_sMODULE_NAME As String = "ThisWorkbook"




'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub Workbook_Open()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "Workbook_Open()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

'Allow user to halt macro execution by hitting Esc or Ctrl+Break.  IF a user _
does this VBA will treat the action as a run-time error, with _
Err.Number = 18 (which is the value we have set for g_lUSER_CANCEL).  A user _
cancel will now be routed through our central error handler (which, in turn, _
will ignore the error)...
Application.EnableCancelKey = xlErrorHandler

On Error GoTo ErrorHandler
```

```
'Instantiate some globals variables (done as follows for those instances _
in which we are re-running Workbook_Open() whilst debugging)...
g_bDebugMode = False
g_bStepThruMode = False
'''g_bStepThruWsChangeEventMode = False
If g_objMsgBoxHandler Is Nothing Then _
    Set g_objMsgBoxHandler = New MsgBoxHandler

If Not ThisWorkbook.CustomDocumentProperties("Production Mode").value Then
    'Prompt for running in debug mode...
    Dim ansRunInDebugMode As VbMsgBoxResult
    g_objMsgBoxHandler.vSetRunDebugModeMsg
    ansRunInDebugMode = g_objMsgBoxHandler.ShowMsgBoxReturnAnswer(vbYesNo)
    If ansRunInDebugMode = vbYes Then g_bDebugMode = True
End If

If g_bDebugMode Then
    'Prompt for stepping through code...
    Dim ansStepThruMode As VbMsgBoxResult
    g_objMsgBoxHandler.vSetStepThruModeMsg
    ansStepThruMode = g_objMsgBoxHandler.ShowMsgBoxReturnAnswer(vbYesNo)
    If ansStepThruMode = vbYes Then g_bStepThruMode = True
'''    'Prompt for stepping through worksheet_change event(s)...
'''    Dim ansStepThruWsChng As VbMsgBoxResult
'''    g_objMsgBoxHandler.vSetStepThruWsChngMsg
'''    ansStepThruWsChng = g_objMsgBoxHandler.ShowMsgBoxReturnAnswer(vbYesNo)
'''    If ansStepThruWsChng = vbYes Then g_bStepThruWsChangeEventMode = True
End If

If g_bStepThruMode Then Stop
'Again, reset some variables in case we are debugging...
g_bPrepareComplete = False
g_bReviewMode = False

If StrComp(UCase(ThisWorkbook.ContentTypeProperties("Review Status").value), _
    UCase(g_sSP_PPTY_VAL_COMPLTD)) = 0 Then g_bPrepareComplete = True
g_bReviewMode = ThisWorkbook.CustomDocumentProperties("Review Mode").value

g_bBlankTemplateMode = _
    ThisWorkbook.CustomDocumentProperties("In Blank Template").value

If Not g_bDebugMode Then Application.ScreenUpdating = False

'Instantiate two global objects...
Set g_objWbInUseMgr = New WkbkInUseMgr
Set g_objWbOpenCloseMgr = New WkbkOpenCloseMgr

'Fire up wbk prep...
If g_bDebugMode Then Stop
g_objWbOpenCloseMgr.PrepWbkOnOpen RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Application.ScreenUpdating = True
Exit Sub

ErrorHandler:
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
```

```
                Resume ExitPoint
        End If
End Sub
'=============================================================================
Private Sub Workbook_BeforeClose(Cancel As Boolean)
'We must declare this variable, but we will do nothing with it...
Dim bCalledProcedureRanOk As Boolean

If g_bDebugMode Then Stop
If Not g_bDebugMode Then Application.ScreenUpdating = False

'Generally, we do NOT want to do any error-handling here, nor do we _
want remaining code execution here to be interrupted, so...
On Error Resume Next

'For debugging...
If g_objWbOpenCloseMgr Is Nothing Then _
    Set g_objWbOpenCloseMgr = New WkbkOpenCloseMgr

g_objWbOpenCloseMgr.PrepWorkbookForClose RanOkRESULT:=bCalledProcedureRanOk

'Clean-up code goes here...
Application.ScreenUpdating = True

End Sub
'=============================================================================
'=============================================================================
'=============================================================================
'=============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshCover"




'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub AIGLogo_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "AIGLogo_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler

If g_bDebugMode Then Stop
g_objWbInUseMgr.ToggleReviewMode _
    ClosingWbk:=False, _
    RanOkRESULT:=bCalledProcedureRanOk
```

```
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'============================================================================
'============================================================================
'============================================================================
'============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshCtxImport"






'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnUpdateEntNm_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnUpdateEntNm_Click()"
'''Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler

'The entity name formula is a little buggy.  By default when calc-ing a wb or _
ws the formula will display the name of the entity in the user's default _
Data Profile.  Correcting that is simply a matter of getting the cell _
itself to recalc.  I initally would have VBA grab the existing formula _
```

```
and then reenter it.  However, this only toggled the cell between giving the _
correct entity name and giving the name in the Data Profile.  HFC found that _
if we re-enter some other cell's formula the entity name formula always gives _
the correct result.  Since there happened to be a hidden formual in the _
worksheet I named and recalc that "random" cell - WHW...
WshCtxImport.Range("ptrSumFCIfAmts").Formula = _
    WshCtxImport.Range("ptrSumFCIfAmts").Formula

WshCtxImport.Range("ptrImportEntNm").Formula = _
    WshCtxImport.Range("ptrImportEntNm").Formula
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'===============================================================================
'===============================================================================
'===============================================================================
'===============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/10/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshReqrdData"




'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "Worksheet_SelectionChange()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
'''If g_bStepThruWsChangeEventMode Then Stop

'Call a lower-level sub-routine.  Make sure the SUB has a ByRef boolean _
parameter for error-handling...
'''NonEntryPointProcedure RanOkRESULT:= bCalledProcedureRanOk
```

```
'''If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR


ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub
'================================================================================
Private Sub Worksheet_Activate()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "Worksheet_Activate()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...


On Error GoTo ErrorHandler

'Call a lower-level sub-routine.  Make sure the SUB has a ByRef boolean _
parameter for error-handling...
'''NonEntryPointProcedure RanOkRESULT:= bCalledProcedureRanOk
'''If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR


ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub
'================================================================================
Private Sub Worksheet_Change(ByVal Target As Range)
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "Worksheet_Change()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...


On Error GoTo ErrorHandler
```

```
'Call a lower-level sub-routine.  Make sure the SUB has a ByRef boolean _
parameter for error-handling...
'''NonEntryPointProcedure RanOkRESULT:= bCalledProcedureRanOk
'''If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub
'============================================================================
Private Sub btnCreateFile_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnCreateFile_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...


On Error GoTo ErrorHandler

'Call a lower-level sub-routine.  Make sure the SUB has a ByRef boolean _
parameter for error-handling...
'''NonEntryPointProcedure RanOkRESULT:= bCalledProcedureRanOk
'''If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub
'============================================================================
'============================================================================
'============================================================================
'============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/11/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshReqrdData"




'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnCreateFile_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnCreateFile_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler

If g_bStepThruMode Then Stop
If Not g_bDebugMode Then Application.ScreenUpdating = False
g_objWbInUseMgr.CreateCFCSpecificFileMaestro RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Application.ScreenUpdating = True
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'================================================================================
'================================================================================
'================================================================================
'================================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshSched01"






'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnPrint_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnPrint_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
bCalledProcedureRanOk = True '...until proven otherwise

g_objWbInUseMgr.PrintAllVisibleSheets RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'===============================================================================
'===============================================================================
'===============================================================================
'===============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshSched02"






'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnPrint_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnPrint_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
bCalledProcedureRanOk = True '...until proven otherwise

g_objWbInUseMgr.PrintAllVisibleSheets RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'===============================================================================
'===============================================================================
'===============================================================================
'===============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshSched03"




'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnPrint_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnPrint_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
bCalledProcedureRanOk = True '...until proven otherwise

g_objWbInUseMgr.PrintAllVisibleSheets RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'===============================================================================
'===============================================================================
'===============================================================================
'===============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshSched04A"




'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnPrint_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnPrint_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
bCalledProcedureRanOk = True '...until proven otherwise

g_objWbInUseMgr.PrintAllVisibleSheets RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'===============================================================================
'===============================================================================
'===============================================================================
'===============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshSched04B"




'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnPrint_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnPrint_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
bCalledProcedureRanOk = True '...until proven otherwise

g_objWbInUseMgr.PrintAllVisibleSheets RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'===============================================================================
'===============================================================================
'===============================================================================
'===============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshSched05"






'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnPrint_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnPrint_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
bCalledProcedureRanOk = True '...until proven otherwise

g_objWbInUseMgr.PrintAllVisibleSheets RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'===============================================================================
'===============================================================================
'===============================================================================
'===============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshSched06"




'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnPrint_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnPrint_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
bCalledProcedureRanOk = True '...until proven otherwise

g_objWbInUseMgr.PrintAllVisibleSheets RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'===============================================================================
'===============================================================================
'===============================================================================
'===============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshSched07"




'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnPrint_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnPrint_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
bCalledProcedureRanOk = True '...until proven otherwise

g_objWbInUseMgr.PrintAllVisibleSheets RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'===============================================================================
'===============================================================================
'===============================================================================
'===============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshSched08"






'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnPrint_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnPrint_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
bCalledProcedureRanOk = True '...until proven otherwise

g_objWbInUseMgr.PrintAllVisibleSheets RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'===============================================================================
'===============================================================================
'===============================================================================
'===============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshSched09"






'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnPrint_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnPrint_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
bCalledProcedureRanOk = True '...until proven otherwise

g_objWbInUseMgr.PrintAllVisibleSheets RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'===============================================================================
'===============================================================================
'===============================================================================
'===============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshSched10"






'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub btnPrint_Click()
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = True
Const sSOURCE As String = "btnPrint_Click()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
bCalledProcedureRanOk = True '...until proven otherwise

g_objWbInUseMgr.PrintAllVisibleSheets RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'===============================================================================
'===============================================================================
'===============================================================================
'===============================================================================
```

```
ErrorHandling - 1

'.....................................................................
'.....................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................
'.....................................................................

'.....................................................................
'.........................  Class-level COMMENTS  ....................
'With some adaptations, the code in this module is taken from:
    'Professional Excel Development (2nd Ed.), by:
    'Rob Bovey, Dennis Wallentin, Stephen Bullen, & John Green
    'Addison-Wesley, 2009
    'ISBN-13:  978-0-321-50879-9
    'Pgs 482-3
    'http://www.informit.com/store/product.aspx?isbn=0321508793
'.....................................................................
'.....................................................................


'*********************************************************************
'*********************************************************************
'*************************  MODULE-LEVEL DECLARATIONS  *************************
'*********************************************************************
'*********************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons



'*********************************************************************
'*********************************************************************
'*******************************  PRIVATE MEMBERS  *******************************
'*********************************************************************
'*********************************************************************

'**************************************
'MODULE-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sUSER_CANCELED As String = "User canceled macro execution"
Private Const m_sSUB_FUNC_ERROR As String = _
        "Error in called Sub or invoked Function"
Private Const m_sTESTING_ERROR As String = "Testing our error-handling"
Private Const m_sMISMATCHED_PARENT_WS_ERROR As String = _
        "Header range (cell) and data column must exist on same worksheet"
Private Const m_sNO_DATA_IN_DEFINED_RNG_ERROR As String = _
        "The range we are attempting to define contains no data"
Private Const m_sSTRUCTURE_CHANGED As String = _
    "The workbook has been altered in such a way as to invalidate the VBA code"
Private Const m_sROWCOL_REF_INVALID As String = _
    "The row or column reference exceeds the size of the named range"
Private Const m_sFILE_ERROR_LOG As String = "Error.log"



'*********************************************************************
'*********************************************************************
'*******************************  PUBLIC MEMBERS  *******************************
```

```
'*****************************************************************************
'*****************************************************************************


'============================================================
'==================  (Public) PROCEDURE  ====================
'============================================================
'============================================================================
Public Sub CentralErrorHandler(ByVal ModuleName As String, _
    ByVal ProcedureName As String, _
    ByRef StepThroughErrorModeRESULT As Boolean, _
    Optional ByVal FileNm As String, _
    Optional ByVal IsEntryPoint As Boolean)
'NOTE 1:  If FileNm is not specified assume error was thrown by code _
from within this workbook.
'NOTE 2:  The boolean returned by this function sets whether or not you _
wind up stepping through the results of the error.

Static sErrMsgOrig As String
Static sFileNmOrigErr As String
Static sProcNmOrgErr As String
Dim iFile As Integer
Dim lErrNmbr As Long
Dim sErrMsgCurr As String
Dim sFullSource As String
Dim sPath As String
Dim sLogText As String
Dim sErrLogFile As String

Dim i As Integer

'Grab error info before it's cleared by On Error Resume Next below...
lErrNmbr = Err.Number

Select Case lErrNmbr
    Case g_lUSER_CANCEL
        sErrMsgCurr = m_sUSER_CANCELED
    Case g_lMONKEY_WRENCH
        'We've thrown a deliberate error...
        sErrMsgCurr = m_sTESTING_ERROR
    Case g_lHANDLED_ERROR
        'Error in a call sub or invoked function...
        sErrMsgCurr = m_sSUB_FUNC_ERROR
    Case g_lMISMATCHED_WSHS
        'Ranges must reside on same worksheet...
        sErrMsgCurr = m_sMISMATCHED_PARENT_WS_ERROR
    Case g_lNO_DATA_IN_RNG
        sErrMsgCurr = m_sNO_DATA_IN_DEFINED_RNG_ERROR
    Case g_lSTRUCTURE_ALTERED
        'Some wksheet, table, range, etc. has wrong number of cols...
        sErrMsgCurr = m_sSTRUCTURE_CHANGED
    Case g_lREF_OUTSIDE_RNG
        'Trying, for example, to size a 5-col range on col #6...
        sErrMsgCurr = m_sROWCOL_REF_INVALID
    Case Else
        sErrMsgCurr = Err.Description
End Select

'If this is the originating error the static error variables will be empty. _
In that case store the originating error information in these variables...
If sErrMsgOrig = vbNullString Then sErrMsgOrig = sErrMsgCurr
If sProcNmOrgErr = vbNullString Then sProcNmOrgErr = ProcedureName
If sFileNmOrigErr = vbNullString Then
    If FileNm = vbNullString Then
        'When no file name is supplied assume the error rose in this file...
        sFileNmOrigErr = ThisWorkbook.Name
    Else '...file name WAS supplied
        sFileNmOrigErr = FileNm
    End If
End If
```

```
ErrorHandling - 3

'Since we can't allow errors in the central error handler itself...
On Error Resume Next '...NOTE:  This clears all properties of the Err obj

'If no file name is supplied load the default file name...
If FileNm = vbNullString Then FileNm = ThisWorkbook.Name

'NOTE:
    'Chr(92) = backslash - i.e., "\"
    'Chr(91) and Chr(93) = left and right square brackets - i.e., "[", "]"
    'Chr(46) = period - i.e., "."
    'Chr(32) = space - i.e., " "

''''Get the application directory [when using on desktop apps]...
'''sPath = ThisWorkbook.Path
'''If StrComp(Right$(sPath, 1), Chr(92)) <> 0 Then sPath = sPath & Chr(92)

'Get the MyDocs directory [use this for SharePoint hosted apps]...
Dim WshShell As Object
Set WshShell = CreateObject("Wscript.Shell")
sPath = WshShell.SpecialFolders("MyDocuments") & Chr(92)

'Construct the fully-qualified error source name...
sFullSource = Chr(91) & FileNm & Chr(93) & ModuleName & Chr(46) & ProcedureName

'Create the error text to be logged...
sLogText = _
    Chr(32) & sFullSource & ", Error " & CStr(lErrNmbr) & ":  " & sErrMsgCurr

'Open the log file, write out the error information, and then close the file...
iFile = FreeFile()
Open sPath & ThisWorkbook.Name & m_sFILE_ERROR_LOG For Append As #iFile
Print #iFile, Format$(Now(), "mm/dd/yyyy hh:mm:ss"); sLogText
If IsEntryPoint Then Print #i,
Close #iFile

'Generate messages, if and where appropriate, and do other clean-up...
Application.ScreenUpdating = True
sErrLogFile = sPath & ThisWorkbook.Name & m_sFILE_ERROR_LOG
If StrComp(sErrMsgCurr, m_sUSER_CANCELED) <> 0 Then
    'Show the error message immediately if we are in debug mode.  Otherwise, _
    show the error message when we reach the entry point...
    If IsEntryPoint Or g_bDebugMode Then
        g_objMsgBoxHandler.SetFatalErrorMsg _
            ErrLogFileName:=sErrLogFile, _
            OriginatingErrFileName:=sFileNmOrigErr, _
            OriginatingProcName:=sProcNmOrgErr, _
            OriginatingErrMsg:=sErrMsgOrig
        g_objMsgBoxHandler.ShowMsgBoxNoAnswer vbCritical
        'So that we are ready to handle the next error, clear the static _
        variables once we've reached the entry point...
        sErrMsgOrig = vbNullString
        sFileNmOrigErr = vbNullString
        sProcNmOrgErr = vbNullString
    End If
    'The return value is the debug mode status...
    StepThroughErrorModeRESULT = g_bDebugMode
Else 'This is a UserCanceled error.
    'Show the error message when we reach the entry point, but only _
    if we are in production (i.e., not in debug) mode...
    If IsEntryPoint And Not g_bDebugMode Then
        g_objMsgBoxHandler.SetUserStoppedMacroMsg ErrLogFileName:=sErrLogFile
        g_objMsgBoxHandler.ShowMsgBoxNoAnswer vbInformation
    End If
    If IsEntryPoint Then sErrMsgOrig = vbNullString
    StepThroughErrorModeRESULT = False
End If
End Sub
'============================================================================
'============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'..............  REFERENCE LIBRARIES EMPLOYED/REQUIRED BY PROJECT  .............
'''Microsoft Shell Controls and Automation
'''Microsoft Scripting Runtime
'''IF USING THE MSSSDataConnectionClass include reference to highest numbered _
'''Microsoft ActiveX Data Objects 2.X Library
'
'.....................................................................................
'.....................................................................................

'.....................................................................................
'............................  Class-level COMMENTS  ..........................
'This module simply contains all global variable declarations
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'************************  MODULE-LEVEL DECLARATIONS  ************************
'*************************************************************************************
'*************************************************************************************
Option Explicit


'=====================================================================================
'=====================================================================================
'************************************
'CONSTANTS
'************************************
'Longs...
'[NOTE:  numbers 1024 through 65535 are available for custom errors]
Public Const g_lHANDLED_ERROR As Long = 9999
Public Const g_lMONKEY_WRENCH As Long = 9876
Public Const g_lMISMATCHED_WSHS As Long = 9998
Public Const g_lNO_DATA_IN_RNG As Long = 9997
Public Const g_lREF_OUTSIDE_RNG As Long = 9995
Public Const g_lSTRUCTURE_ALTERED As Long = 9994
Public Const g_lUSER_CANCEL As Long = 18
'Strings...
Public Const g_sAPP_TITLE As String = "CFC Template"
Public Const g_sCTX_07_ADDIN As String = _
    "CorpTax.Office.Excel.AddIn.Version2007"
'Orig worksheet pwd: "He@ther4260"
'Public Const g_sWSH_PASSWORD As String = "KF0sk0" '...thru 3/17/2013
Public Const g_sWSH_PASSWORD As String = "200Liberty" '...as of 3/18/2013
Public Const g_sFILE_NM_SUFFIX As String = "_Intl Tax Package"
Public Const g_sXL_MACRO_FILE_EXT As String = ".xlsm"
Public Const g_sSP_PPTY_NM_EOP As String = "Period End Date"
Public Const g_sSP_PPTY_NM_SI_CODE_NM As String = "SI Code and Name"
Public Const g_sSP_PPTY_NM_YR As String = "Tax Year"
Public Const g_sSP_PPTY_NM_ENT_GEMS As String = "Entity Code and GEMS ID"
```

```
Public Const g_sSP_PPTY_NM_DOC_TYPE As String = "Document Type"
Public Const g_sSP_PPTY_NM_JURIS As String = "Jurisdiction"
Public Const g_sSP_PPTY_NM_REVW_STATUS As String = "Review Status"

Public Const g_sSP_PPTY_VAL_DOC_TYPE As String = "P - CFC Tax Package"
Public Const g_sSP_PPTY_VAL_JURIS As String = "Federal"
Public Const g_sSP_PPTY_VAL_IN_PREP As String = "In Preparation"
Public Const g_sSP_PPTY_VAL_COMPLTD As String = "Completed"

Public Const g_sSP_URL_QA As String = _
    "https://taxqa.aig.net:444/Compliance/International%20Group/"
Public Const g_sSP_URL_PROD As String = _
    "https://tax.aig.net:444/Compliance/International%20Group/"
Public Const g_sSP2010_URL_PROD As String = _
    "https://sp.contact.aig.net/global/tax/corptax/Compliance/International%20Group/"
Public Const g_sSP2010_URL_QA As String = _
    "http://sp.contactq.aig.net/global/tax/home/Compliance/International%20Group/"

Public Const g_sLIST_NM_ENTITY As String = "lstIntlEntity"
Public Const g_sLIST_NM_CURR_LIST As String = "lstCurrency"
Public Const g_sLISTS_COL_SORT As String = "SortOrder"

'Bytes...
Public Const g_yMAX_CONNECT_ATTEMPTS As Byte = 3

'**************************************
'VARIABLES
'**************************************
'Objects...
Public g_objShPtPpts As ShPtProperties
Public g_objWbOpenCloseMgr As WkbkOpenCloseMgr
Public g_objWbInUseMgr As WkbkInUseMgr
Public g_objMsgBoxHandler As MsgBoxHandler
'''zzzPublic g_objCtxOffMgr As CtxOfficeMgr
Public g_objWshzToAlwaysHide As Wkshtz
Public g_objWshzToAlwaysShow As Wkshtz
Public g_objWshzToShowInFileMode As Wkshtz
Public g_objWshzWithTaxPrepCols As Wkshtz
Public g_objWshzToPwdProtect As Wkshtz

Public g_comaddinCTX As COMAddIn


'Booleans (default = FALSE)...
Public g_bStepThruMode As Boolean
'''Public g_bStepThruWsChangeEventMode As Boolean
Public g_bDebugMode As Boolean
Public g_bBlankTemplateMode As Boolean
Public g_bPrepareComplete As Boolean
Public g_bReviewMode As Boolean


'**************************************
'ENUMERATIONS
'**************************************
Public Enum ProjWbOpenCloseStatus
    projWbOpenTemplMode
    projWbOpenFileMode
    projWbCloseMode
End Enum
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Module-level COMMENTS  ...................................
'The procedures in this module are designed to be general-purpose - i.e., _
they are not specific to this project.
'These procedures DO, however, require integration with centralized _
error-trapping.
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'************************  MODULE-LEVEL DECLARATIONS  ********************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons



'*************************************************************************************
'*************************************************************************************
'******************************  PRIVATE MEMBERS  ************************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "Utilities"




'*************************************************************************************
'*************************************************************************************
'******************************  PUBLIC MEMBERS  *************************************
'*************************************************************************************
'*************************************************************************************

'============================================================
'==================  (Public) PROCEDURES  ===================
'============================================================
'=========================================================================
Private Sub ProtectWsh(ByVal WsToProtect As Worksheet, _
    ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = False
Const sSOURCE As String = "ProtectWsh()"
```

```
On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
WsToProtect.Protect _
    Password:=g_sWSH_PASSWORD, _
    contents:=True, _
    userinterfaceonly:=False

ExitPoint:
On Error Resume Next
'''Must-run procedure/function clean-up code...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
        ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
        IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'================================================================================
'================================================================================
'================================================================================
'================================================================================
'================================================================================
'================================================================================
'================================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................


'.....................................................................................
'.........................  Module-level COMMENTS  ...........................
'DEVELOPMENT-ONLY PROCEDURES!!!!
'Note:  Because of the nature of this module there is very little, if any, _
error-handling within this module's procedures and functions.
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'************************  MODULE-LEVEL DECLARATIONS  ************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons



'*************************************************************************************
'*************************************************************************************
'******************************  PRIVATE MEMBERS  ******************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "vDevUtilities"



'*************************************************************************************
'*************************************************************************************
'******************************  PUBLIC MEMBERS  ******************************
'*************************************************************************************
'*************************************************************************************

'============================================================
'==================  (Public) PROCEDURES  ==================
'============================================================
'=============================================================================
Private Sub PrintXlDefinedNamesToImmediateWindow()
Dim wbNmz As Names
Dim wbDefNm As Name
Dim sNameNm As String
Set wbNmz = ThisWorkbook.Names
If wbNmz.Count = 0 Then
        MsgBox "There are no names in this workbook", vbok, g_sAPP_TITLE
        Exit Sub
```

```
vDevUtilities - 2

End If

Debug.Print "*****  NAMES IN THE " & ThisWorkbook.Name & " WORKBOOK  *****" & _
    vbCr
Dim i As Integer
For i = 1 To wbNmz.Count
    Set wbDefNm = wbNmz(i)
    sNameNm = wbDefNm.Name
    Debug.Print sNameNm & Space(30 - Len(sNameNm)) & wbDefNm.RefersTo
Next i
End Sub
'================================================================================
Private Sub PrintXlDefinedNamesToTextFile()
Dim fso As FileSystemObject
Dim ts As TextStream
Dim namesInWb As File
Dim wbNmz As Names
Dim wbDefNm As Name
Dim sNameNm As String
Dim sFileNm As String
Dim iNmLen As Integer
Dim iMaxNamLen As Integer
Dim i As Integer
Dim bNameIsPrintRelated As Boolean
Dim ansOmitPrintStuff As VbMsgBoxResult

If g_objMsgBoxHandler Is Nothing Then
    Set g_objMsgBoxHandler = New MsgBoxHandler
End If
g_objMsgBoxHandler.vSetOmitPrintAreaTitlesMsg
ansOmitPrintStuff = g_objMsgBoxHandler.ShowMsgBoxReturnAnswer(vbYesNo)

Set wbNmz = ThisWorkbook.Names
Set fso = New FileSystemObject
sFileNm = ThisWorkbook.FullName & "_Names.txt"
If Not fso.FileExists(sFileNm) Then
    Set ts = fso.CreateTextFile(sFileNm)
    ts.Close
End If

'Set indent...
For i = 1 To wbNmz.Count
    Set wbDefNm = wbNmz(i)
    sNameNm = wbDefNm.Name
    'Set boolean...
    If ansOmitPrintStuff = vbYes Then
        bNameIsPrintRelated = _
            InStr(1, sNameNm, "!Print_Area") > 0 Or _
            InStr(1, sNameNm, "!Print_Titles") > 0
    End If
    'If we WANTED  print-related stuff boolean never reset from default(F)...
    If Not bNameIsPrintRelated Then
            iNmLen = Len(sNameNm)
        If iNmLen > iMaxNamLen Then iMaxNamLen = iNmLen
    End If
    bNameIsPrintRelated = False '...restore to default
Next i

Set namesInWb = fso.GetFile(sFileNm)
Set ts = namesInWb.OpenAsTextStream(ForWriting, TristateUseDefault)
ts.WriteLine "*****  NAMES IN THE " & ThisWorkbook.Name & " WORKBOOK  *****"
If ansOmitPrintStuff = vbYes Then
    ts.Write "[OMITS PRINT-RELATED NAMES!!!]"
End If
ts.WriteBlankLines (2)
For i = 1 To wbNmz.Count
    Set wbDefNm = wbNmz(i)
    sNameNm = wbDefNm.Name
    'Set boolean...
    If ansOmitPrintStuff = vbYes Then
        bNameIsPrintRelated = _
```

```
            InStr(1, sNameNm, "!Print_Area") > 0 Or _
            InStr(1, sNameNm, "!Print_Titles") > 0
    End If
    'If we WANTED  print-related stuff boolean never reset from default(F)...
    If Not bNameIsPrintRelated Then
        ts.WriteLine sNameNm & Space(iMaxNamLen + 5 - Len(sNameNm)) & wbDefNm.RefersTo
    End If
    bNameIsPrintRelated = False '...restore to default
Next i
ts.Close
Shell "Notepad.exe " & sFileNm, vbNormalFocus
End Sub
'================================================================================
Private Sub RemoveScrollAreas()
Dim ws As Worksheet

For Each ws In ThisWorkbook.Worksheets
    If ws.Visible = xlSheetVisible Then _
        ws.ScrollArea = ""
Next ws
MsgBox "Done"
End Sub
'================================================================================
Private Sub UnhideAndUnprotectAllWorksheets()
UnhideAndUnprotectMacroMsgWsh
Unprotect3IntroWorksheets
UnhideSchedWorksheets
UnprotectSchedWorksheets
UnhideAndUnprotectAlwaysHiddenWshs
MsgBox "Done", vbOKOnly, "Viola!"
End Sub
'================================================================================
Private Sub UnhideSchedWorksheets()
Dim i As Byte
For i = 1 To g_objWshzToShowInFileMode.Count
    g_objWshzToShowInFileMode.Item(i).Visible = xlSheetVisible
Next i
End Sub
'================================================================================
Private Sub UnhideAndUnprotectAlwaysHiddenWshs()
Dim i As Byte
Dim ws As Worksheet
For i = 1 To g_objWshzToAlwaysHide.Count
    Set ws = g_objWshzToAlwaysHide.Item(i)
    ws.Visible = xlSheetVisible
    ws.Unprotect g_sWSH_PASSWORD
Next i
End Sub
'================================================================================
Private Sub UnhideAndUnprotectMacroMsgWsh()
WshEnabMacros.Visible = xlSheetVisible
WshEnabMacros.Unprotect g_sWSH_PASSWORD
End Sub
'================================================================================
Private Sub UnprotectSchedWorksheets()
Dim i As Byte
For i = 1 To g_objWshzToShowInFileMode.Count
    g_objWshzToShowInFileMode.Item(i).Unprotect g_sWSH_PASSWORD
Next i
End Sub
'================================================================================
Private Sub Unprotect3IntroWorksheets()
Dim i As Byte
For i = 1 To g_objWshzToAlwaysShow.Count
    g_objWshzToAlwaysShow.Item(i).Unprotect g_sWSH_PASSWORD
Next i
End Sub
'================================================================================
Private Sub HideSchedWorksheets()
Dim i As Byte
WshReqrdData.Activate
```

```
For i = 1 To g_objWshzToShowInFileMode.Count
    g_objWshzToShowInFileMode.Item(i).Visible = xlSheetVeryHidden
Next i
MsgBox "Done!", vbOKOnly, "Perfectus"
End Sub
'================================================================================
Private Sub vCallToggleReviewMode()
'Error-handling declarations...
Const sSOURCE As String = "vCallToggleReviewMode()"
Const bSUB_IS_ENTRY_PT As Boolean = True
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler

g_bPrepareComplete = True


g_objWbInUseMgr.ToggleReviewMode _
    ClosingWbk:=False, _
    RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  .....................................
'1)  This class is designed to hold all of the clumsy string concatenations _
involved in message box prompts.
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "MsgBoxHandler"
Private Const m_sDEV_MSGBOX_TTL As String = "*** DEVELOPMENT-ONLY MESSAGE!!! ***"


'**************************************
'MODULE/CLASS-WIDE VARIABLES
'**************************************
'Strings...
Private m_sTitle As String
Private m_sPrompt As String
Private m_s2Crs  As String
Private m_s3Crs  As String
Private m_s2Crs1Tab As String
Private m_sCrTab As String
Private m_sCr2Tabs As String
Private m_s2Tabs As String




'=========================================================
'==================  (Private) PROCEDURES  ================
'=========================================================
```

```
MsgBoxHandler - 2

'============================================================================
Private Sub Class_Initialize()
m_s2Crs = vbCr & vbCr
m_s3Crs = vbCr & vbCr & vbCr
m_s2Crs1Tab = vbCr & vbCr & vbTab
m_sCrTab = vbCr & vbTab
m_sCr2Tabs = vbCr & vbTab & vbTab
m_s2Tabs = vbTab & vbTab
End Sub



'****************************************************************************
'****************************************************************************
'*****************************  PUBLIC MEMBERS  *****************************
'****************************************************************************
'****************************************************************************


'============================================================
'==================  (Public) FUNCTIONS  ==================
'============================================================
'============================================================================
Public Function ShowMsgBoxReturnAnswer(MsgBoxStyle As VbMsgBoxStyle) _
    As VbMsgBoxResult
ShowMsgBoxReturnAnswer = MsgBox(m_sPrompt, MsgBoxStyle, m_sTitle)
'Reset matters...
m_sPrompt = vbNullString
m_sTitle = vbNullString
End Function



'============================================================
'==================  (Public) PROCEDURES  ==================
'============================================================

'**************************************
'USER MESSAGES
'**************************************
'============================================================================
Public Sub ShowMsgBoxNoAnswer(MsgBoxStyle As VbMsgBoxStyle)
MsgBox m_sPrompt, MsgBoxStyle, m_sTitle
'Reset matters...
m_sPrompt = vbNullString
m_sTitle = vbNullString
End Sub
'============================================================================
Public Sub SetWillLetYouKnowWhenDonePrintingMsg()
m_sTitle = "FYI..."
m_sPrompt = _
    "We will let you know when the printing has been completed."
End Sub
'============================================================================
Public Sub SetUserStoppedMacroMsg(ByVal ErrLogFileName As String)
m_sTitle = "Code Interrupted"
m_sPrompt = _
        "You have elected to halt macro execution." & m_s2Crs & _
        "To be on the safe side we suggest that you close and reopen this " & _
        "file in order to ensure that the macro works as designed." & m_s2Crs & _
        "NOTE:  If you have halted or are halting macro execution because " & _
        "of problems with the workbook please contact the Tax Technology " & _
        "Department.  Further, if that is the case we ask that you also " & _
        "email us the following file:" & m_s2Crs1Tab & _
        ErrLogFileName
End Sub
'============================================================================
Public Sub SetFatalErrorMsg(ByVal ErrLogFileName As String, _
    ByVal OriginatingErrFileName As String, _
    ByVal OriginatingProcName As String, _
```

```
MsgBoxHandler - 3

    ByVal OriginatingErrMsg As String)

If Len(OriginatingErrMsg) = 0 Then
    OriginatingErrMsg = "[No error message was generated.]"
End If

m_sTitle = "Serious Error Encountered"
m_sPrompt = _
        "This application has encountered a serious error:" & m_s2Crs1Tab & _
        "Details:" & m_sCr2Tabs & _
        OriginatingErrMsg & m_s2Crs1Tab & _
        "Procedure:" & m_sCr2Tabs & _
        OriginatingProcName & m_s2Crs1Tab & _
        "File:" & m_sCr2Tabs & _
        OriginatingErrFileName & m_s2Crs & _
        "Please inform us at the Tax Technology Group of the error, and " & _
        "please email us the following file:" & m_s2Crs1Tab & _
        ErrLogFileName & m_s2Crs & _
        "Lastly, we strongly suggest that you discontinue working with this " & _
        "file until we have had an opportunity to resolve this issue."
End Sub
'================================================================================
Public Sub SetNoEntityChosenMsg()
m_sTitle = "No Entity Chosen"
m_sPrompt = _
    "You must pick an entity in the drop-down box " & _
    "before we can create a CFC-specific file."
End Sub
'================================================================================
Public Sub SetNoFunctionalCurrencyChosenMsg()
m_sTitle = "No Functional Currency Chosen"
m_sPrompt = _
    "You must pick a functional currency in the drop-down box " & _
    "before we can create a CFC-specific file."
End Sub
'================================================================================
Public Sub SetChosenEntityNotInSPLibraryMsg()
m_sTitle = "Entity Not in SharePoint Library"
m_sPrompt = _
    "The entity you have chosen is not currently registered in the " & _
    "SharePoint library.  Unfortunately, you cannot continue your work on " & _
    "this entity until SharePoint has been updated." & _
    m_s2Crs & _
    "Please contact any of the 3 contacts (J. McAvinue, K. Fosko, or " & _
    "R. Harris) listed on the " & WshReqrdData.Name & " worksheet/tab " & _
    "to have this issue addressed."
End Sub
'================================================================================
Public Sub SetSINotInSPLibraryMsg()
m_sTitle = "SI Not in SharePoint Library"
m_sPrompt = _
    "The SI of the entity you have chosen is not currently registered " & _
    "in the SharePoint library.  Unfortunately, you cannot continue " & _
    "your work on this entity until SharePoint has been updated." & _
    m_s2Crs & _
    "Please contact any of the 3 contacts (J. McAvinue, K. Fosko, or " & _
    "R. Harris) listed on the " & WshReqrdData.Name & " worksheet/tab " & _
    "to have this issue addressed."
End Sub
'================================================================================
Public Sub SetMustBeCompletedDocForReviewModeMsg()
m_sTitle = "Cannot Activate Review Mode"
m_sPrompt = _
    "Review mode can only be activated on Tax Packages where the " & _
    Chr(34) & "Review Status" & Chr(34) & " has been set to " & _
    Chr(34) & "Completed" & Chr(34) & Chr(46)
End Sub
'================================================================================
Public Sub SetCTXOfficeAddInNotInstalledModeMsg()
m_sTitle = "CTX for Office Not Found"
m_sPrompt = _
```

```
MsgBoxHandler - 4

    "Activating Review Mode requires the CorpTax Office Excel Add-In." & _
    m_s2Crs & _
    "Please contat your local admin to have this Add-In installed " & _
    "on your machine."
End Sub
'================================================================================
Public Sub SetCannotConnectToCTXOfficeMsg()
m_sTitle = "Problems Connecting to CorpTax Server"
m_sPrompt = _
    "You are experiencing problems connection with the CorpTax server.  " & _
    "Either wait a little while before trying again or exit completely " & _
    "out of Excel before reopening this workbook." & m_s2Crs & _
    "If the problem persists contact the Tax Technology group."
End Sub
'================================================================================
Public Sub SetNotLoggedIntoCTXOfficeMsg()
m_sTitle = "Not Logged Into CorpTax for Office"
m_sPrompt = _
    "You are either not logged into CorpTax for Excel, or you have not " & _
    "set up a valid data profile.  (Either of these scenarios will " & _
    "account for any system/CorpTax error messages you might have just " & _
    "received.)" & m_s2Crs & _
    "Please handle these issue(s) before proceeding."
End Sub
'================================================================================
Public Sub SetToggleModeMsg()
Const sMODE_ACTV As String = "Active"
Const sMODE_INACTV As String = "Inactive"
Const sMODE_ENABL As String = "enabled"
Const sMODE_DISABL As String = "disabled"
Dim sActvInactv As String
Dim sAblDisbl As String

If g_bReviewMode Then
    sActvInactv = sMODE_ACTV
    sAblDisbl = sMODE_ENABL
Else
    sActvInactv = sMODE_INACTV
    sAblDisbl = sMODE_DISABL
End If

m_sTitle = "Review Mode " & sActvInactv
m_sPrompt = _
    "Review mode has been " & sAblDisbl & "."
End Sub
'================================================================================
Public Sub PrepFileReadyForReviewQuest()
m_sTitle = "Tax Package Ready for Review?"
m_sPrompt = _
    "Is this tax package completed and ready for review?"
End Sub
'================================================================================
Public Sub PrepActivatePrintPreviewQuest()
m_sTitle = "Use Print Preview First?"
m_sPrompt = _
    "Do you want to see Print Preview before you print to your printer?"
End Sub
'================================================================================
Public Sub PrepConfirmCompletedModeQuest()
m_sTitle = "Should File Remain Marked as Completed?"
m_sPrompt = _
    "This file has been marked as " & _
    Chr(34) & "Completed" & Chr(34) & Chr(46) & m_s2Crs & _
    "Is that still the correct?" & m_s3Crs & _
    "[NOTE:  Answering NO will reset the " & _
    Chr(34) & "Review Status" & Chr(34) & " of this file to " & _
    Chr(34) & "In Preparation" & Chr(34) & Chr(46) & Chr(93)
End Sub
```

```
'**************************************
'DEVELOPER MESSAGES
'**************************************
'=============================================================================
Public Sub vSetRunDebugModeMsg()
m_sTitle = m_sDEV_MSGBOX_TTL
m_sPrompt = "Run in Debug Mode?"
End Sub
'=============================================================================
Public Sub vSetStepThruModeMsg()
m_sTitle = m_sDEV_MSGBOX_TTL
m_sPrompt = "Execute in Step-Through mode?"
End Sub
'=============================================================================
Public Sub vSetStepThruWsChngMsg()
m_sTitle = m_sDEV_MSGBOX_TTL
m_sPrompt = "Step through worksheet_change event?"
End Sub
'=============================================================================
Public Sub vSetOmitPrintAreaTitlesMsg()
m_sTitle = m_sDEV_MSGBOX_TTL
m_sPrompt = "OMIT Print_Titles and Print_Area names?"
End Sub
'=============================================================================
'=============================================================================
'=============================================================================
'=============================================================================
'=============================================================================
'=============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/05/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................


'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'This is a collection class.  Each object will have a property equal to _
the display name of the Share Point property.  The items collected into this _
class will simply be the enumerated string values for the SP Property
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ********************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "ShPtPptyEnums"


'**************************************
'MODULE/CLASS-WIDE VARIABLES
'**************************************
'Collections...
Private m_coll As Collection
'Strings...
Private m_sPptyDisplayNm As String


'=========================================================
'======================  EVENTS  =========================
'=========================================================
'=====================================================================================
Private Sub Class_Initialize()
Set m_coll = New Collection
End Sub
'=====================================================================================
Private Sub Class_Terminate()
```

```
ShPtPptyEnums - 2

'Since there's really nothing useful an error-handler can do at this point...
On Error Resume Next
Set m_coll = Nothing
End Sub




'*****************************************************************************
'*****************************************************************************
'*****************************  PUBLIC MEMBERS  ******************************
'*****************************************************************************
'*****************************************************************************


'============================================================
'==================  (Public) PROPERTIES  ==================
'============================================================
'============================================================================
Property Let ShPtPptyDisplayName(value As String)
    m_sPptyDisplayNm = value
End Property
Property Get ShPtPptyDisplayName() As String
    ShPtPptyDisplayName = m_sPptyDisplayNm
End Property


'============================================================
'==================  (Public) FUNCTIONS  ===================
'============================================================
'============================================================================
Public Function Items() As Collection
Set Items = m_coll
End Function
'============================================================================
Public Function Item(index) As String
Item = m_coll(index)
End Function
'============================================================================
Public Function Count() As Integer
Count = m_coll.Count
End Function
'============================================================================
Public Function ContainsItem(ItemName As String) As Boolean
Dim bContains As Boolean

If m_coll.Count > 0 Then
        Dim i As Integer
        For i = 1 To m_coll.Count
                If StrComp(m_coll.Item(i), ItemName) = 0 Then
                        bContains = True
                        Exit For
                End If
        Next i
End If
ContainsItem = bContains
End Function




'============================================================
'==================  (Public) PROCEDURES  ==================
'============================================================
'============================================================================
Public Sub Add(value As String)
'No need for error handling; incompatible types will not even compile...
m_coll.Add Item:=value
End Sub
'============================================================================
```

```
ShPtPptyEnums - 3

Public Sub Remove(index, ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = False
Const sSOURCE As String = "Remove()"
'Operating code declarations...

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

m_coll.Remove index

ExitPoint:
Exit Sub

ErrorHandler:
        RanOkRESULT = False
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub

'============================================================================
'============================================================================
'============================================================================
'============================================================================
'============================================================================
'============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/05/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................



'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Collection of the SharePoint properties.
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons



'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "ShPtProperties"


'**************************************
'MODULE/CLASS-WIDE VARIABLES
'**************************************
'Collections...
Private m_coll As Collection



'============================================================
'======================  EVENTS  ============================
'============================================================
'=====================================================================================
Private Sub Class_Initialize()
Set m_coll = New Collection
End Sub
'=====================================================================================
Private Sub Class_Terminate()
'Since there's really nothing useful an error-handler can do at this point...
On Error Resume Next
```

```
ShPtProperties - 2

Set m_coll = Nothing
End Sub




'*******************************************************************************
'*******************************************************************************
'*****************************  PUBLIC MEMBERS  ********************************
'*******************************************************************************
'*******************************************************************************


'============================================================
'==================  (Public) FUNCTIONS  ====================
'============================================================
'==========================================================================
Public Function Items() As Collection
Set Items = m_coll
End Function
'==========================================================================
Public Function Item(index) As ShPtPptyEnums
Set Item = m_coll(index)
End Function
'==========================================================================
Public Function Count() As Integer
Count = m_coll.Count
End Function
'==========================================================================
Public Function ContainsItem(ItemName As String) As Boolean
Dim bContains As Boolean

If m_coll.Count > 0 Then
        Dim obj As ShPtPptyEnums
        Set obj = New ShPtProperties
        Dim i As Integer
        For i = 1 To m_coll.Count
                Set obj = m_coll.Item(i)
                If StrComp(obj.ShPtPptyDisplayName, ItemName) = 0 Then
                        bContains = True
                        Exit For
                End If
        Next i
End If
ContainsItem = bContains
End Function
'==========================================================================
Public Function GetEnumsObjByName(ShPtPptyNm As String) As ShPtPptyEnums
Dim objSPPptyEnums As ShPtPptyEnums
Dim i As Integer
For i = 1 To m_coll.Count
        Set objSPPptyEnums = m_coll.Item(i)
        If StrComp(objSPPptyEnums.ShPtPptyDisplayName, ShPtPptyNm) = 0 Then
                Set GetEnumsObjByName = objSPPptyEnums
                i = m_coll.Count
        End If
Next i
End Function




'============================================================
'==================  (Public) PROCEDURES  ===================
'============================================================
'==========================================================================
Public Sub Add(value As ShPtPptyEnums)
'No need for error handling; incompatible types will not even compile...
m_coll.Add Item:=value, key:=value.ShPtPptyDisplayName
End Sub
```

```
'=============================================================================
Public Sub Remove(index, ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = False
Const sSOURCE As String = "Remove()"
'Operating code declarations...

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

m_coll.Remove index

ExitPoint:
Exit Sub

ErrorHandler:
        RanOkRESULT = False
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub


'=============================================================================
'=============================================================================
'=============================================================================
'=============================================================================
'=============================================================================
'=============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/11/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'Commentary...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WkbkInUseMgr"



'=============================================================
'======================  EVENTS  ============================
'=============================================================
'=====================================================================================
Private Sub Class_Initialize()
'[VBA QUIRK:  Errors in Intialize, Activate, and Terminate event handlers _
cannot be handled by procedures which create or destroy instances of the _
class. If any such error-handling IS required, create a separate public _
procedure - e.g., the Initialize procedure, below.]
End Sub
'=====================================================================================
Private Sub Class_Terminate()
'[VBA QUIRK:  Errors in Intialize, Activate, and Terminate event handlers _
cannot be handled by procedures which create or destroy instances of the _
class. If any such error-handling IS required, create a separate public _
procedure - e.g., the Initialize procedure, below.]

'Since there's really nothing useful an error-handler can do at this point...
On Error Resume Next
'clean-up code...
```

```
WkbkInUseMgr - 2

End Sub



'============================================================
'==================  (Private) FUNCTIONS  ==================
'============================================================
'================================================================================
Private Function RetrieveSICodeAndNameFmShPt(ByVal CtxID As String, _
    ByRef RanOkRESULT As Boolean) As String
'Error-handling declarations...
Const sSOURCE As String = "RetrieveSICodeAndNameFmShPt()"
Const bFUNCTION_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim objShPtSICodes As ShPtPptyEnums
Dim sSICodeAndNm As String
Dim sSICodeFromShPtPpty As String
Dim sSICodeFromCtxID As String
Dim i As Integer

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
sSICodeFromCtxID = Mid(CtxID, 3, 4)
Set objShPtSICodes = g_objShPtPpts.GetEnumsObjByName(g_sSP_PPTY_NM_SI_CODE_NM)
For i = 1 To objShPtSICodes.Count
    sSICodeAndNm = objShPtSICodes.Item(i)
    sSICodeFromShPtPpty = Left(Trim(sSICodeAndNm), 4)
    If StrComp(sSICodeFromShPtPpty, sSICodeFromCtxID) = 0 Then _
        i = objShPtSICodes.Count '...break FOR loop
Next i

RetrieveSICodeAndNameFmShPt = sSICodeAndNm '...return result

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Function

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bFUNCTION_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Function
'================================================================================
Private Function IsCTXOfficeInstalled(ByRef RanOkRESULT As Boolean) As Boolean
'Error-handling declarations...
Const sSOURCE As String = "IsCTXOfficeInstalled()"
Const bFUNCTION_IS_ENTRY_PT As Boolean = False
'''Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

RanOkRESULT = True '...until we discover otherwise

On Error Resume Next
Set g_comaddinCTX = Application.COMAddIns(g_sCTX_07_ADDIN)
If Err.Number <> 0 Then
    Err.Clear
```

```
WkbkInUseMgr - 3

    IsCTXOfficeInstalled = False
    GoTo ExitPoint
End If

On Error GoTo ErrorHandler '...resume our normal approach

'Now we know g_comaddinCTX is not nothing...
IsCTXOfficeInstalled = g_comaddinCTX.Connect

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Function

ErrorHandler:
        RanOkRESULT = False
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bFUNCTION_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Function

'============================================================
'==================  (Private) PROCEDURES  ==================
'============================================================
'================================================================================
Private Sub CreateCFCSpecificFile(ByRef FileCreatedRESULT As Boolean, _
    ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "CreateCFCSpecificFile()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Const sDD_SELECT_PROMPT As String = "Select" '...had problems including  "..."

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'Trap for user not yet having made an entity choice...
If StrComp(UCase(Left(WshReqrdData.Range("ptrSelectedCFC"), 6)), _
        UCase(sDD_SELECT_PROMPT), vbTextCompare) = 0 Then
    g_objMsgBoxHandler.SetNoEntityChosenMsg
    g_objMsgBoxHandler.ShowMsgBoxNoAnswer vbExclamation
    GoTo ExitPoint
End If

'Trap for user not yet having made a currency choice...
If StrComp(UCase(Left(WshReqrdData.Range("ptrSelectedFC"), 6)), _
        UCase(sDD_SELECT_PROMPT), vbTextCompare) = 0 Then
    g_objMsgBoxHandler.SetNoFunctionalCurrencyChosenMsg
    g_objMsgBoxHandler.ShowMsgBoxNoAnswer vbExclamation
    GoTo ExitPoint
End If

'Instantiate object and populate all of our SP enumeration objects...
Set g_objShPtPpts = New ShPtProperties
PopulateTwoShPtObjsController RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

'Now trap for user picking an entity which is not in the SP drop-down list...
Dim objEntNmEnums As ShPtPptyEnums
Dim sSelectedCtxId As String
```

```
WkbkInUseMgr - 4

Dim sPptyEnum As String
Dim bSPHasEntity As Boolean
Dim i As Integer
sSelectedCtxId = Trim(WshReqrdData.Range("ptrSelectedCtxId"))
Set objEntNmEnums = g_objShPtPpts.Item(g_sSP_PPTY_NM_ENT_GEMS)
For i = 1 To objEntNmEnums.Count
    sPptyEnum = objEntNmEnums.Items(i)
    If InStr(1, sPptyEnum, sSelectedCtxId, vbTextCompare) > 0 Then
        bSPHasEntity = True
        i = objEntNmEnums.Count '...break FOR loop
    End If
Next i
If Not bSPHasEntity Then
    g_objMsgBoxHandler.SetChosenEntityNotInSPLibraryMsg
    g_objMsgBoxHandler.ShowMsgBoxNoAnswer vbCritical
    GoTo ExitPoint
End If

If g_bDebugMode Then Stop
'Get/trap for SI Code and name...
Dim sSICodeName As String
sSICodeName = RetrieveSICodeAndNameFmShPt( _
    CtxID:=sSelectedCtxId, _
    RanOkRESULT:=bCalledProcedureRanOk)
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
If sSICodeName = vbNullString Then
    g_objMsgBoxHandler.SetSINotInSPLibraryMsg
    g_objMsgBoxHandler.ShowMsgBoxNoAnswer vbCritical
    GoTo ExitPoint
End If

If g_bDebugMode Then Stop
'If here we can go ahead an create the file...
g_bBlankTemplateMode = False
ThisWorkbook.CustomDocumentProperties("In Blank Template").value = _
    g_bBlankTemplateMode
SetShPtProperties _
    EntityAndShortNm:=sPptyEnum, _
    SICodeAndNm:=sSICodeName, _
    ReviewStatus:=g_sSP_PPTY_VAL_IN_PREP, _
    RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
Dim sFileNm As String
Dim sTaxYr As String '...for debugging
'sFileNm = g_sSP2010_URL_PROD
sFileNm = g_sSP2010_URL_QA
sTaxYr = Format(Range("ptrTaxYear"), "0000")
If g_bDebugMode Then sTaxYr = "2014"
sFileNm = sFileNm & _
    sTaxYr & "_" & _
    WshReqrdData.Range("ptrSelectedGemsId") & "_" & _
    WshReqrdData.Range("ptrSelectedCtxId") & "_" & _
    g_sFILE_NM_SUFFIX & _
    g_sXL_MACRO_FILE_EXT
If g_bStepThruMode Then Stop
ThisWorkbook.SaveAs sFileNm, XlFileFormat.xlOpenXMLWorkbookMacroEnabled
FileCreatedRESULT = True

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
        RanOkRESULT = False
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
```

```
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub
'================================================================================
Private Sub ExposeScheduleWshs(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "ExposeScheduleWshs()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim i As Byte

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
For i = 1 To g_objWshzToShowInFileMode.Count
    g_objWshzToShowInFileMode.Item(i).Visible = xlSheetVisible
Next i

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'================================================================================
Private Sub PopulateTwoShPtObjsController(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "PopulateTwoShPtObjsController()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim metaThisWbkContentTypeProps As MetaProperties
Dim varShPtSchema As Variant
Dim domdocShPtSchema As DOMDocument60
Dim elemShPtSchemaRoot As IXMLDOMElement
Dim lstXsdSchema As IXMLDOMNodeList
Dim nodXsdSchema As IXMLDOMNode
Dim ElemXsdSchema As IXMLDOMElement
Dim bDOMCreated As Boolean
Dim i As Byte

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
Set metaThisWbkContentTypeProps = ThisWorkbook.ContentTypeProperties
'Sometimes the size of these SP Schemas are too big for a string variable, _
so we use a variant instead...
varShPtSchema = metaThisWbkContentTypeProps.SchemaXml
```

```
WkbkInUseMgr - 6


'Create DOM object, with a trap...
Set domdocShPtSchema = New DOMDocument60
bDOMCreated = domdocShPtSchema.LoadXML(varShPtSchema)
    If Not bDOMCreated Then Err.Raise g_lHANDLED_ERROR

'Any xml doc has to have at least a root element...
Set elemShPtSchemaRoot = domdocShPtSchema.DocumentElement

'Collect the xsd:schema elements...
Set lstXsdSchema = elemShPtSchemaRoot.getElementsByTagName("xsd:schema")
    If lstXsdSchema Is Nothing Then Err.Raise g_lHANDLED_ERROR

'Loop through each xsd:schema element, populate objs...
For i = 1 To lstXsdSchema.Length
    Set ElemXsdSchema = lstXsdSchema.Item(i - 1)
    PopulateShPtObjsFromXsdSchemaElement _
        RanOkRESULT:=bCalledProcedureRanOk, _
        ElemXsdSchema:=ElemXsdSchema
            If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
Next i


ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'================================================================================
Private Sub PopulateShPtObjsFromXsdSchemaElement( _
    ByVal ElemXsdSchema As IXMLDOMElement, _
    ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "PopulateShPtObjsFromXsdSchemaElement()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim lstXsdElements As IXMLDOMNodeList
Dim elemXsdElement As IXMLDOMElement
Dim lstXsdSimpleTypes As IXMLDOMNodeList
Dim elemXsdSimpleType As IXMLDOMElement
Dim lstXsdRestriction As IXMLDOMNodeList
Dim elemXsdRestriction As IXMLDOMElement
Dim lstXsdEnumerations As IXMLDOMNodeList
Dim elemXsdEnumeration As IXMLDOMElement
Dim nodXml As IXMLDOMNode
Dim objSPPptEnums As ShPtPptyEnums
Dim i As Integer
Dim j As Integer

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
Set lstXsdElements = ElemXsdSchema.getElementsByTagName("xsd:element")
```

```
For i = 1 To lstXsdElements.Length
    Set elemXsdElement = lstXsdElements(i - 1)
    If StrComp(elemXsdElement.getAttribute("ma:format"), "Dropdown") = 0 Then
        Set objSPPptEnums = New ShPtPptyEnums
        'Capture displayed name of the enumerated SP property...
        objSPPptEnums.ShPtPptyDisplayName = _
            elemXsdElement.getAttribute("ma:displayName")

         '************************
        'NOTE:  I made several attempts to navigate to the xsd:restriction _
        element using commands like _
            Set nodXsdRestriction = _
            elemXsdElement.SelectSingleNode("xsd:simpleType/xsd:restriction") _
        However, while the SelectSingleNode method is written to take _
        XPath queries, I could not get the method to work.  It kept saying _
        that the xsd namespace was not recognized.  I tried things like _
        omitting the namespace and using a wildcard for a namespace prefix, _
        but, grrrr...., nothing worked.  Ergo, our drill down here is more _
        cumbersome than I would like - WHW...
        '************************

        Set lstXsdSimpleTypes = _
            elemXsdElement.getElementsByTagName("xsd:simpleType")
        'Since there can be only one "simpleType" element in an "xsd:element"...
        Set elemXsdSimpleType = lstXsdSimpleTypes(0)
        'Get xsd:restriction element...
        Set lstXsdRestriction = elemXsdSimpleType.getElementsByTagName("xsd:restriction")

        For Each nodXml In lstXsdRestriction '...there SHOULD only be one
            Set elemXsdRestriction = nodXml
            If StrComp(elemXsdRestriction.getAttribute("base"), _
                    "dms:Choice") = 0 Then
                Set lstXsdEnumerations = _
                    elemXsdRestriction.getElementsByTagName("xsd:enumeration")
                'Loop through the enumeration elements and populate our collection _
                object...
                For j = 1 To lstXsdEnumerations.Length
                    Set elemXsdEnumeration = lstXsdEnumerations(j - 1)
                    objSPPptEnums.Add elemXsdEnumeration.getAttribute("value")
                Next j
                'Now add our enumerations collection to the SP Enumerated Pptz coll...
                g_objShPtPpts.Add objSPPptEnums
            End If
        Next nodXml
    End If
Next i

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'=============================================================================
Private Sub SetShPtProperties(ByVal EntityAndShortNm As String, _
    ByVal SICodeAndNm As String, _
```

```
    ByVal ReviewStatus As String, _
    ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "SetShPtProperties()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim dtEOY As Date
Dim sTaxYr As String
Dim sSICode As String
Dim sSICodeAndNm As String

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'Set some variables...
sTaxYr = Trim(Str(Range("ptrTaxYear")))
dtEOY = CDate("12/31/" & sTaxYr)

If g_bStepThruMode Then Stop
With ThisWorkbook
    .ContentTypeProperties(g_sSP_PPTY_NM_DOC_TYPE).value = _
        g_sSP_PPTY_VAL_DOC_TYPE
    .ContentTypeProperties(g_sSP_PPTY_NM_ENT_GEMS).value = _
        EntityAndShortNm
    .ContentTypeProperties(g_sSP_PPTY_NM_EOP).value = dtEOY
    .ContentTypeProperties(g_sSP_PPTY_NM_JURIS).value = _
        g_sSP_PPTY_VAL_JURIS
    .ContentTypeProperties(g_sSP_PPTY_NM_SI_CODE_NM).value = _
        SICodeAndNm
    .ContentTypeProperties(g_sSP_PPTY_NM_YR).value = sTaxYr
    .ContentTypeProperties(g_sSP_PPTY_NM_REVW_STATUS).value = _
        ReviewStatus
End With


ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
        RanOkRESULT = False
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub




'***************************************************************************
'***************************************************************************
'*****************************  PUBLIC MEMBERS  ****************************
'***************************************************************************
'***************************************************************************

'=========================================================
```

```
WkbkInUseMgr - 9

'==================  (Public) PROCEDURES  ==================
'============================================================
'================================================================================
Public Sub ShowMacroMsgWshHideAllOthers(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "ShowMacroMsgWshHideAllOthers()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim ws As Worksheet
Dim i As Byte

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
With WshEnabMacros
    .Visible = xlSheetVisible
    .Select
    .Range("A1").Activate
End With
For i = 1 To ThisWorkbook.Worksheets.Count
    Set ws = ThisWorkbook.Worksheets(i)
    If StrComp(WshEnabMacros.Name, ws.Name) <> 0 Then _
        ws.Visible = xlSheetHidden
Next i

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'================================================================================
Public Sub SetGenlInfoWshRowVisibilities(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "SetGenlInfoWshRowVisibilities()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim rngToHide

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'We can use the global boolean to set whether rows are hidden or exposed...
With WshReqrdData
    .Range("ptrddCFC").EntireRow.Hidden = Not g_bBlankTemplateMode
    .Range("ptrSelectedCFC").EntireRow.Hidden = g_bBlankTemplateMode
    .Range("ptrddCurrency").EntireRow.Hidden = Not g_bBlankTemplateMode
    .Range("ptrSelectedFC").EntireRow.Hidden = g_bBlankTemplateMode
    .Range("rngQuestionnaire").EntireRow.Hidden = g_bBlankTemplateMode
    .Range("rngEntityNotInDropDownMsg").EntireRow.Hidden = _
        Not g_bBlankTemplateMode
    .btnCreateFile.Enabled = g_bBlankTemplateMode
End With
```

```
ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'================================================================================
Public Sub SetWshScrollAreas(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "SetWshScrollAreas()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Const sSCROLL_AREA_RNG_NM As String = "rngScrollArea"
Dim ws As Worksheet
Dim nm As Name
Dim i As Byte

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
For Each ws In ThisWorkbook.Worksheets
    If ws.Visible = xlSheetVisible Then
        For i = 1 To ws.Names.Count
            Set nm = ws.Names(i)
            If InStr(1, nm.Name, sSCROLL_AREA_RNG_NM) > 0 Then
                ws.ScrollArea = ws.Range(sSCROLL_AREA_RNG_NM).Address
                i = ws.Names.Count
            End If
        Next i
    End If
Next ws

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
```

```
WkbkInUseMgr - 11

'================================================================================
Public Sub CreateCFCSpecificFileMaestro(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "CreateCFCSpecificFileMaestro()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim bFileSuccessfullyCreated As Boolean
Dim i As Byte

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
CreateCFCSpecificFile _
    FileCreatedRESULT:=bFileSuccessfullyCreated, _
    RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
    If Not bFileSuccessfullyCreated Then GoTo ExitPoint
ExposeScheduleWshs RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

SetWshScrollAreas RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

WshReqrdData.Select

SetGenlInfoWshRowVisibilities RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
WshReqrdData.btnCreateFile.Enabled = False


ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'================================================================================
Public Sub SetTaxPreparerColumnsVisibility(ByVal HideColumns As Boolean, _
    ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "SetTaxPreparerColumnsVisibility()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim ws As Worksheet
Dim rngColPreparer As Range
Dim i As Byte

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
If g_bDebugMode Then Stop
For i = 1 To g_objWshzWithTaxPrepCols.Count
```

```
    Set ws = g_objWshzWithTaxPrepCols.Item(i)
    ws.Activate
    'If ws.ProtectionMode Then ws.Unprotect g_sWSH_PASSWORD
    Set rngColPreparer = ws.Range("colPrepOnly_RelatedPartyDD").EntireColumn
    rngColPreparer.EntireColumn.Hidden = HideColumns
    ws.Range("A1").Activate
Next i

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'================================================================================
Public Sub ToggleReviewMode(ByVal ClosingWbk As Boolean, _
    ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = False
Const sSOURCE As String = "ToggleReviewMode()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Const sNOT_CONNECTED As String = "#NotConnected"

RanOkRESULT = True '...until we discover otherwise
On Error GoTo ErrorHandler
If g_bDebugMode Then Stop

'If activating Review Mode trap for the workbook not being flagged _
as completed...
If Not g_bPrepareComplete And Not g_bReviewMode Then
    If StrComp(ThisWorkbook.ContentTypeProperties.Item( _
            g_sSP_PPTY_NM_REVW_STATUS).value, g_sSP_PPTY_VAL_COMPLTD) <> 0 Then
        g_objMsgBoxHandler.SetMustBeCompletedDocForReviewModeMsg
        g_objMsgBoxHandler.ShowMsgBoxNoAnswer vbExclamation
        GoTo ExitPoint
    End If
End If

'Trap for no CTXOffice add-in...
If Not g_bReviewMode Then '...about to activate Review Mode
    'Is the add-in installed?...
    If Not IsCTXOfficeInstalled(RanOkRESULT:=bCalledProcedureRanOk) Then
        If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
        g_objMsgBoxHandler.SetCTXOfficeAddInNotInstalledModeMsg
        g_objMsgBoxHandler.ShowMsgBoxNoAnswer vbCritical
        GoTo ExitPoint
    End If
End If

If g_bDebugMode Then Stop
If Not g_bDebugMode Then Application.ScreenUpdating = False

g_objWbInUseMgr.SetTaxPreparerColumnsVisibility _
    HideColumns:=g_bReviewMode, _
    RanOkRESULT:=bCalledProcedureRanOk
```

```
If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

'Reset visibility of CTX Import sheet...
If g_bReviewMode Then '...ergo, we are in process of turning off Rvw Mode
    WshCtxImport.Visible = xlSheetVeryHidden
    'Added by HFC 1/26/2014
    WshSummary.Visible = xlSheetVeryHidden
Else
    WshCtxImport.Visible = xlSheetVisible
    'Added by HFC 1/26/2014
    WshSummary.Visible = xlSheetVisible
End If

WshCover.Select

'Toggle flag and set Doc property...
g_bReviewMode = Not g_bReviewMode
ThisWorkbook.CustomDocumentProperties("Review Mode").value = g_bReviewMode

'Prompt user...
If ClosingWbk Then GoTo ExitPoint
g_objMsgBoxHandler.SetToggleModeMsg
g_objMsgBoxHandler.ShowMsgBoxNoAnswer vbInformation

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Application.ScreenUpdating = True
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'================================================================================
Public Sub PrintAllVisibleSheets(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = False
Const sSOURCE As String = "PrintAllVisibleSheets()"
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim wsOrig As Worksheet
Dim ansUsePrintPreview As VbMsgBoxResult
Dim bUsePrintPreview As Boolean

On Error GoTo ErrorHandler

'CODE HERE...
Set wsOrig = ActiveSheet
g_objMsgBoxHandler.PrepActivatePrintPreviewQuest
ansUsePrintPreview = g_objMsgBoxHandler.ShowMsgBoxReturnAnswer(vbYesNoCancel)
If ansUsePrintPreview = vbCancel Then GoTo ExitPoint
If ansUsePrintPreview = vbYes Then bUsePrintPreview = True

Application.StatusBar = "Currently printing:  " & ThisWorkbook.Name
ThisWorkbook.PrintOut Copies:=1, preview:=bUsePrintPreview, Collate:=True

'Without this XL will always show the 1st active sheet when _
done prining - in our case the Cover wsh...
```

```
wsOrig.Activate '...restore things

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Application.StatusBar = vbNullString
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub

'============================================================================
'============================================================================
'============================================================================
'============================================================================
'============================================================================
'============================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................


'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'This class primarily contains methods to hide and unhide worksheets, rows, _
and, for the most part, to control the visibility of various parts of the _
workbook.  How these methods operate depends on whether the workbook is _
being opened or closed, and whether it is being opened as an initial, blank _
template, or whether as an existing file.  The two public methods _
are called from the Workbook_Open and Workbook_Close events.  I prefer to have _
as little code as possible in XL objects themselves, so that is why I didn't _
simply make these methods private methods within the workbook object..... WHW
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WkbkOpenCloseMgr"




'============================================================
'======================  EVENTS  ============================
'============================================================
'=============================================================================
Private Sub Class_Initialize()
'[VBA QUIRK:  Errors in Intialize, Activate, and Terminate event handlers _
cannot be handled by procedures which create or destroy instances of the _
class. If any such error-handling IS required, create a separate public _
procedure - e.g., the Initialize procedure, below.]
End Sub
'=============================================================================
```

```
WkbkOpenCloseMgr - 2

Private Sub Class_Terminate()
'[VBA QUIRK:  Errors in Intialize, Activate, and Terminate event handlers _
cannot be handled by procedures which create or destroy instances of the _
class. If any such error-handling IS required, create a separate public _
procedure - e.g., the Initialize procedure, below.]

'Since there's really nothing useful an error-handler can do at this point...
On Error Resume Next
'clean-up code...
End Sub




'============================================================
'=================  (Private) PROCEDURES  ================
'============================================================
'============================================================================
Private Sub PopulateWshCollectionsMaestro(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "PopulateWshCollectionsMaestro()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
PopulateAlwaysVisbWshzCollection RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
PopulateAlwaysHiddenWshzCollection RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
PopulateVisbWshzFileModeCollection RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
PopulateTaxPreparerColumnWshzCollection RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
PopulateWshzToPwdProtectCollection RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR


ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'============================================================================
Private Sub PopulateAlwaysVisbWshzCollection(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "PopulateAlwaysVisbWshzCollection()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise
```

```
WkbkOpenCloseMgr - 3


'CODE HERE...
Set g_objWshzToAlwaysShow = New Wkshtz
'Add wshs needed for raw template...
g_objWshzToAlwaysShow.Add WshCover
g_objWshzToAlwaysShow.Add WshInstruc
g_objWshzToAlwaysShow.Add WshReqrdData

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
        RanOkRESULT = False
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub
'============================================================================
Private Sub PopulateAlwaysHiddenWshzCollection(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "PopulateAlwaysHiddenWshzCollection()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
Set g_objWshzToAlwaysHide = New Wkshtz '...collection object
'Add always-hidden wshs...
g_objWshzToAlwaysHide.Add WshCtxDataCache
g_objWshzToAlwaysHide.Add WshData
g_objWshzToAlwaysHide.Add WshLkupLsts
g_objWshzToAlwaysHide.Add WshCtxImport
g_objWshzToAlwaysHide.Add WshCurrencyLst
g_objWshzToAlwaysHide.Add WshIntlEntityList
'Added by HFC 1/26/2014
g_objWshzToAlwaysHide.Add WshSummary


ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
        RanOkRESULT = False
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
```

```
WkbkOpenCloseMgr - 4

        End If
End Sub
'=============================================================================
Private Sub PopulateVisbWshzFileModeCollection(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "PopulateVisbWshzFileModeCollection()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
Set g_objWshzToShowInFileMode = New Wkshtz
g_objWshzToShowInFileMode.Add WshSched01
g_objWshzToShowInFileMode.Add WshSched02
g_objWshzToShowInFileMode.Add WshSched03
g_objWshzToShowInFileMode.Add WshSched04A
g_objWshzToShowInFileMode.Add WshSched04B
g_objWshzToShowInFileMode.Add WshSched05
g_objWshzToShowInFileMode.Add WshSched06
g_objWshzToShowInFileMode.Add WshSched07
g_objWshzToShowInFileMode.Add WshSched08
g_objWshzToShowInFileMode.Add WshSched09
g_objWshzToShowInFileMode.Add WshSched10


ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
        RanOkRESULT = False
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub
'=============================================================================
Private Sub PopulateTaxPreparerColumnWshzCollection(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "PopulateTaxPreparerColumnWshzCollection()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
Set g_objWshzWithTaxPrepCols = New Wkshtz '...collection object
'Add wshs that have Tax Preparer columns (i.e., columns to hide)...
g_objWshzWithTaxPrepCols.Add WshSched03
g_objWshzWithTaxPrepCols.Add WshSched04A
g_objWshzWithTaxPrepCols.Add WshSched04B
g_objWshzWithTaxPrepCols.Add WshSched06
g_objWshzWithTaxPrepCols.Add WshSched07
g_objWshzWithTaxPrepCols.Add WshSched08


ExitPoint:
On Error Resume Next
```

```
WkbkOpenCloseMgr - 5

'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
        RanOkRESULT = False
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub
'============================================================================
Private Sub PopulateWshzToPwdProtectCollection(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "PopulateWshzToPwdProtectCollection()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'CODE HERE...
Set g_objWshzToPwdProtect = New Wkshtz '...collection object
'Add wshs that we need to password-protect...
g_objWshzToPwdProtect.Add WshCover
g_objWshzToPwdProtect.Add WshCtxImport
g_objWshzToPwdProtect.Add WshEnabMacros
g_objWshzToPwdProtect.Add WshInstruc
g_objWshzToPwdProtect.Add WshReqrdData
g_objWshzToPwdProtect.Add WshSched01
g_objWshzToPwdProtect.Add WshSched02
g_objWshzToPwdProtect.Add WshSched03
g_objWshzToPwdProtect.Add WshSched04A
g_objWshzToPwdProtect.Add WshSched04B
g_objWshzToPwdProtect.Add WshSched05
g_objWshzToPwdProtect.Add WshSched06
g_objWshzToPwdProtect.Add WshSched07
g_objWshzToPwdProtect.Add WshSched08
g_objWshzToPwdProtect.Add WshSched09
g_objWshzToPwdProtect.Add WshSched10
'Added by HFC 1/26/2014
g_objWshzToPwdProtect.Add WshSummary

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
        RanOkRESULT = False
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
```

```
WkbkOpenCloseMgr - 6

End Sub
'================================================================================
Private Sub SetWorksheetVisibility(ByRef RanOkRESULT As Boolean, _
    WbOpenCloseStatus As ProjWbOpenCloseStatus)
'Error-handling declarations...
Const sSOURCE As String = "SetWorksheetVisibility()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim ws As Worksheet
Dim yWshs As Byte
Dim i As Byte
Dim visbOpenInFileMode As XlSheetVisibility
Dim visbCloseTempMode As XlSheetVisibility
Dim visbCloseFileMode As XlSheetVisibility
Dim visbHiddenStatus As XlSheetVisibility
Dim visbSchedWshs As XlSheetVisibility

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

visbHiddenStatus = xlSheetVeryHidden '...default/production mode
    If g_bDebugMode Then visbHiddenStatus = xlSheetHidden

'Things to always do when opening a file...
If WbOpenCloseStatus = projWbOpenFileMode Or _
    WbOpenCloseStatus = projWbOpenTemplMode Then
    For i = 1 To g_objWshzToAlwaysShow.Count
        Set ws = g_objWshzToAlwaysShow.Item(i)
        With ws
            .Visible = xlSheetVisible
            .Select
            .Range("A1").Activate
        End With
    Next i
    For i = 1 To g_objWshzToAlwaysHide.Count
        g_objWshzToAlwaysHide.Item(i).Visible = visbHiddenStatus
    Next i
End If

'Now handle schedule worksheets when opening...
If WbOpenCloseStatus = projWbOpenFileMode Or _
        WbOpenCloseStatus = projWbOpenTemplMode Then
    If WbOpenCloseStatus = projWbOpenTemplMode Then
        visbSchedWshs = visbHiddenStatus
    Else '...open file mode
        visbSchedWshs = xlSheetVisible
    End If
    For i = 1 To g_objWshzToShowInFileMode.Count
        g_objWshzToShowInFileMode.Item(i).Visible = visbSchedWshs
    Next i
End If

'Last touches on opening a file...
If WbOpenCloseStatus <> projWbCloseMode Then
With WshCover
    .Select
    .Range("A2").Activate
End With
WshEnabMacros.Visible = visbHiddenStatus
End If

'Lastly, handle closing files...
If WbOpenCloseStatus = projWbCloseMode Then
    'First, check on status of our worksheet collection objects...
    Dim bMissingAWshCollection As Boolean
    If g_objWshzToAlwaysShow Is Nothing Then bMissingAWshCollection = True
    If g_objWshzToAlwaysHide Is Nothing Then bMissingAWshCollection = True
    If g_objWshzToShowInFileMode Is Nothing Then bMissingAWshCollection = True
    If bMissingAWshCollection Then
        PopulateWshCollectionsMaestro RanOkRESULT:=bCalledProcedureRanOk
```

```
    End If

    'Fire up macro message sheet...
    With WshEnabMacros
        .Visible = xlSheetVisible
        .Select
        .Range("A1").Activate
    End With
    For i = 1 To g_objWshzToAlwaysShow.Count
        g_objWshzToAlwaysShow.Item(i).Visible = xlSheetVeryHidden
    Next i
    For i = 1 To g_objWshzToShowInFileMode.Count
        g_objWshzToShowInFileMode.Item(i).Visible = xlSheetVeryHidden
    Next i
    'And just in case any perma-hide sheets were made visible in debugging...
    For i = 1 To g_objWshzToAlwaysHide.Count
        g_objWshzToAlwaysHide.Item(i).Visible = xlSheetVeryHidden
    Next i
End If

ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
        RanOkRESULT = False
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub
'================================================================================
Private Sub SortTable( _
    ByVal WshWithTable As Worksheet, _
    ByVal TableName As String, _
    ByVal SortColHeading As String, _
    ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "SortTable()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim lstToSort As ListObject
Dim rngSortCol As Range
Dim sColRngRef As String


On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'NOTE:  chr(91) & chr(93) are, respectively, "[", "]"
sColRngRef = TableName & Chr(91) & SortColHeading & Chr(93)
Set lstToSort = WshWithTable.ListObjects(TableName)
Set rngSortCol = lstToSort.ListColumns(SortColHeading).Range
With lstToSort.Sort
    .SortFields.Clear
    .SortFields.Add key:=rngSortCol.Cells(2, 1), _
        SortOn:=xlSortOnValues, _
        Order:=xlAscending
    .Apply
End With
```

```
WkbkOpenCloseMgr - 8


ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub




'***************************************************************************
'***************************************************************************
'*****************************  PUBLIC MEMBERS  ****************************
'***************************************************************************
'***************************************************************************


'=============================================================
'==================  (Public) PROCEDURES  ==================
'=============================================================
'=============================================================================
Public Sub PrepWbkOnOpen(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "PrepWbkOnOpen()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...
Dim enumWbOpenCloseStatus As ProjWbOpenCloseStatus
Dim ws As Worksheet
Dim i As Byte

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

If g_bDebugMode Then
    Application.DisplayDocumentInformationPanel = True
Else
    Application.DisplayDocumentInformationPanel = False
End If

'When debugging set so we only have EnableMacros ws open...
If g_bDebugMode Then
    g_objWbInUseMgr.ShowMacroMsgWshHideAllOthers _
        RanOkRESULT:=bCalledProcedureRanOk
        If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
End If

'Populate our worksheet collections re:  visibility...
PopulateWshCollectionsMaestro RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

'Set variable...
If g_bBlankTemplateMode Then
```

```
    enumWbOpenCloseStatus = projWbOpenTemplMode
Else
    enumWbOpenCloseStatus = projWbOpenFileMode
End If

'Protect appropriate worksheets...
For i = 1 To g_objWshzToPwdProtect.Count
    Set ws = g_objWshzToPwdProtect.Item(i)
    ws.Protect Password:=g_sWSH_PASSWORD, userinterfaceonly:=True
Next i

'Show/hide appropriate wshs...
SetWorksheetVisibility _
    RanOkRESULT:=bCalledProcedureRanOk, _
    WbOpenCloseStatus:=enumWbOpenCloseStatus
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

'Show/hide rows in Required Data worksheet...
g_objWbInUseMgr.SetGenlInfoWshRowVisibilities _
    RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

'Set scroll areas...
g_objWbInUseMgr.SetWshScrollAreas RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

'Sort hidden tables, just in case they are not sorted in SP...
WshCurrencyLst.ListObjects(g_sLIST_NM_CURR_LIST).Refresh
SortTable WshWithTable:=WshCurrencyLst, _
    TableName:=g_sLIST_NM_CURR_LIST, _
    SortColHeading:=g_sLISTS_COL_SORT, _
    RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR

WshIntlEntityList.ListObjects(g_sLIST_NM_ENTITY).Refresh
SortTable WshWithTable:=WshIntlEntityList, _
    TableName:=g_sLIST_NM_ENTITY, _
    SortColHeading:=g_sLISTS_COL_SORT, _
    RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR


'Confirm status with user if file is marked as "Completed"...
If g_bPrepareComplete Then
    Dim ansRetainCompleteStatus As VbMsgBoxResult
    g_objMsgBoxHandler.PrepConfirmCompletedModeQuest
    ansRetainCompleteStatus = _
        g_objMsgBoxHandler.ShowMsgBoxReturnAnswer(vbExclamation + vbYesNo)
    If ansRetainCompleteStatus = vbNo Then
        ThisWorkbook.ContentTypeProperties(g_sSP_PPTY_NM_REVW_STATUS).value = _
            g_sSP_PPTY_VAL_IN_PREP
        g_bPrepareComplete = False
    End If
End If


ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
```

```
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'================================================================================
Public Sub PrepWorkbookForClose(ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const sSOURCE As String = "PrepWorkbookForClose()"
Const bSUB_IS_ENTRY_PT As Boolean = False
Dim bCalledProcedureRanOk As Boolean
'Operating code declarations...

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'Handle closing a wbk In Preparation...
If StrComp(ThisWorkbook.ContentTypeProperties(g_sSP_PPTY_NM_REVW_STATUS).value, _
        g_sSP_PPTY_VAL_IN_PREP) = 0 Then
    Dim ansFileReadyForReview As VbMsgBoxResult
    g_objMsgBoxHandler.PrepFileReadyForReviewQuest
    ansFileReadyForReview = g_objMsgBoxHandler.ShowMsgBoxReturnAnswer(vbYesNo)
    If ansFileReadyForReview = vbYes Then _
        ThisWorkbook.ContentTypeProperties(g_sSP_PPTY_NM_REVW_STATUS).value = _
            g_sSP_PPTY_VAL_COMPLTD
End If

'To handle cases of files sent back to field for rework we will be on the _
safe side and always close up review columns when closing a file...
If g_bReviewMode Then
    g_objWbInUseMgr.ToggleReviewMode _
        ClosingWbk:=True, _
        RanOkRESULT:=bCalledProcedureRanOk
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR
End If

SetWorksheetVisibility _
    RanOkRESULT:=bCalledProcedureRanOk, _
    WbOpenCloseStatus:=projWbCloseMode
    If Not bCalledProcedureRanOk Then Err.Raise g_lHANDLED_ERROR


ExitPoint:
On Error Resume Next
'Any must-do and/or clean-up code goes here...
Exit Sub

ErrorHandler:
    RanOkRESULT = False
    Dim bRetraceErrorMode As Boolean
    ErrorHandling.CentralErrorHandler _
        ModuleName:=m_sMODULE_NAME, _
            ProcedureName:=sSOURCE, _
        StepThroughErrorModeRESULT:=bRetraceErrorMode, _
            IsEntryPoint:=bSUB_IS_ENTRY_PT
    If bRetraceErrorMode Then
        If g_bDebugMode Then Stop
        'So we can step through the code which caused the error...
        Resume
    Else
        Resume ExitPoint
    End If
End Sub
'================================================================================
'================================================================================
'================================================================================
'================================================================================
'================================================================================
'================================================================================
```

```
'.....................................................................................
'.....................................................................................
'.......        Written by:             William H. White (consultant)
'.......        With:                   TEKsystems, Inc.
'.......                                www.teksystems.com
'.......        For:                    AIG
'.......                                Tax Technology Group
'.......                                1 WFC, 14th floor
'.......        Current contact:        william.white@aig.com
'.......                                (212) 551-5846
'.......        Permanent contact:      www.rcpconsulting.biz
'.......                                billwhite@rcpconsulting.biz
'.......                                New Providence, NJ
'.......        Module created:         02/03/2013
'.......        Proj finished:          03/05/2013
'.......        Proj updated:           01/29/2014
'.....................................................................................
'.....................................................................................

'.....................................................................................
'.........................  Class-level COMMENTS  ....................................
'This is a collection object.  It is used simply to hold the group of _
worksheets which are to remain hidden from the user throughout this app...
'.....................................................................................
'.....................................................................................


'*************************************************************************************
'*************************************************************************************
'*************************  CLASS-LEVEL DECLARATIONS  ************************************
'*************************************************************************************
'*************************************************************************************
Option Explicit
Option Compare Binary '...use cAsE senSITive string comparisons




'*************************************************************************************
'*************************************************************************************
'*******************************  PRIVATE MEMBERS  ***********************************
'*************************************************************************************
'*************************************************************************************

'**************************************
'MODULE/CLASS-WIDE CONSTANTS
'**************************************
'Strings...
Private Const m_sMODULE_NAME As String = "WshzToHide"


'**************************************
'MODULE/CLASS-WIDE VARIABLES
'**************************************
'Collections...
Private m_coll As Collection



'============================================================
'======================  EVENTS  ============================
'============================================================
'=====================================================================================
Private Sub Class_Initialize()
Set m_coll = New Collection
End Sub
'=====================================================================================
Private Sub Class_Terminate()
'Since there's really nothing useful an error-handler can do at this point...
On Error Resume Next
Set m_coll = Nothing
```

```
Wkshtz - 2

End Sub




'*********************************************************************************
'*********************************************************************************
'*******************************  PUBLIC MEMBERS  ********************************
'*********************************************************************************
'*********************************************************************************


'============================================================
'==================  (Public) FUNCTIONS  ====================
'============================================================
'=============================================================================
Public Function Items() As Collection
Set Items = m_coll
End Function
'=============================================================================
Public Function Item(index) As Worksheet
Set Item = m_coll(index)
End Function
'=============================================================================
Public Function Count() As Integer
Count = m_coll.Count
End Function
'=============================================================================
Public Function ContainsItem(ItemName As String) As Boolean
Dim bContains As Boolean

If m_coll.Count > 0 Then
        Dim ws As Worksheet
        Dim i As Integer
        For i = 1 To m_coll.Count
                Set ws = m_coll.Item(i)
                If StrComp(ws.Name, ItemName) = 0 Then
                        bContains = True
                        Exit For
                End If
        Next i
End If
ContainsItem = bContains
End Function




'============================================================
'==================  (Public) PROCEDURES  ===================
'============================================================
'=============================================================================
Public Sub Add(value As Worksheet)
'No need for error handling; incompatible types will not even compile...
m_coll.Add Item:=value, key:=value.Name '...use appropriate string value
End Sub
'=============================================================================
Public Sub Remove(index, ByRef RanOkRESULT As Boolean)
'Error-handling declarations...
Const bSUB_IS_ENTRY_PT As Boolean = False
Const sSOURCE As String = "Remove()"

On Error GoTo ErrorHandler
RanOkRESULT = True '...until we discover otherwise

'Operating code declarations...
m_coll.Remove index

ExitPoint:
Exit Sub
```

```
ErrorHandler:
        RanOkRESULT = False
        Dim bRetraceErrorMode As Boolean
        ErrorHandling.CentralErrorHandler _
                ModuleName:=m_sMODULE_NAME, _
                ProcedureName:=sSOURCE, _
                StepThroughErrorModeRESULT:=bRetraceErrorMode, _
                IsEntryPoint:=bSUB_IS_ENTRY_PT
        If bRetraceErrorMode Then
                If g_bDebugMode Then Stop
                'So we can step through the code which caused the error...
                Resume
        Else
                Resume ExitPoint
        End If
End Sub
'================================================================================
'================================================================================
'================================================================================
'================================================================================
'================================================================================
'================================================================================
```