EPDM ADD-IN FOR MS OFFICE

References:

Imports Microsoft.Office.Tools.Ribbon

Imports Microsoft.Office.Interop.Excel

Imports Microsoft.Office.Interop.Word

Imports Microsoft.Office.Interop.PowerPoint

Imports System.Runtime.InteropServices

Imports System.Windows.Forms

Imports EdmLib

Imports Microsoft.Win32

Declarations:

Dim fname As String

Dim xl As Excel.Application / word.Application / Powerpoint.application

Dim wb As Excel.Workbook / Word.Document / Powerpoint.Presentation

Dim vault As EdmVault5

Dim vault1 As IEdmVault8

Dim vinfo() As EdmViewInfo

Dim fol As IEdmFolder5

Dim dfile As IEdmFile5

Dim verenum As IEdmEnumeratorVersion5

Dim vers As IEdmVersion5

Dim view As IEdmCardView5

Dim pos As IEdmPos5

Dim statpos As IEdmPos5

Dim wfmgr As IEdmWorkflowMgr6

Dim wf As IEdmWorkflow6

Dim wfstat As IEdmState6

Dim wftrans As IEdmTransition5

Dim bul As IEdmBatchUnlock

Dim bget As IEdmBatchGet

Dim sel() As EdmSelItem

'Dim sel(1) As Integer

Dim cont As IEdmRefItemContainer

Dim poitem As IEdmRefItem

Dim btn As RibbonButton

Dim id As Integer

Dim vitem() As Object

Dim folpath As String

Dim err As String

Dim vpath As Object

Dim i As Integer

Dim findex As Integer

Dim vname As String

Dim bret As Boolean

//Get Object or current Document//

// For Excel //

xl = GetObject(, "Excel.Application")

wb = xl.ActiveWorkbook

// For Word //

xl = GetObject(, "Word.Application")

wb = xl.ActiveDocument

// For Powerpoint //

xl = GetObject(, "Powerpoint.Application")

wb = xl.ActivePresentation

// To Determine Active Document’s Vault //

If wb Is Nothing Then

MsgBox("There is no active document!!", MsgBoxStyle.Information, "EPDM-MS Office Connector")

Else

fname = wb.FullName

vault = New EdmVault5

vault1 = vault

vault1.GetVaultViews(vinfo, False)

findex = 0

For Me.i = 0 To UBound(vinfo)

If fname.Contains(vinfo(i).mbsPath) = True Then

findex = 1

vname = vinfo(i).mbsVaultName

Exit For

End If

Next

// To Get Active Document’s File And Folder From Vault //

If findex = 1 Then

vault.LoginAuto(vname, 0)

fol = vault.GetFolderFromPath(wb.Path)

dfile = vault.GetFileFromPath(fname, fol)

// To Check In More Number of File into Vault //

If dfile.IsLocked = True Then

Try

wb.Close(Excel.XlSaveAction.xlDoNotSaveChanges) // For Excel //

wb.Close(Word.Wdsaveoptions.wddonotsavechanges) // For Word //

wb.close() // For Powerpoint //

bul = vault.CreateUtility(EdmUtility.EdmUtil\_BatchUnlock)

ReDim sel(0)

sel(0).mlDocID = dfile.ID

sel(0).mlProjID = fol.ID

bul.AddSelection(vault, sel)

If bul.CreateTree(0, EdmUnlockBuildTreeFlags.Eubtf\_MayUnlock) = False Then

Exit Sub

End If

cont = bul

cont.GetItems(EdmRefItemType.Edmrit\_All, vitem)

id = LBound(vitem)

While id <= UBound(vitem)

poitem = vitem(id)

poitem.SetProperty(EdmRefItemProperty.Edmrip\_CheckKeepLocked, False)

id = id + 1

End While

If bul.ShowDlg(0) = True Then

bul.UnlockFiles(0)

End If

xl.Workbooks.Open(fname)

Catch ex As Exception

err = ex.Message

MsgBox(err, MsgBoxStyle.Information)

If err.Length > 0 Then

Exit Sub

End If

End Try

Else

MsgBox("The file was already Checked In!!", MsgBoxStyle.Information, "EPDM-MS Office Connector")

End If

Else

MsgBox("This is not a vault document!!", MsgBoxStyle.Information, "EPDM-MS Office Connector")

End If

End If

// To Check out More Number of File from Vault //

If dfile.IsLocked = False Then

Try

wb.Close(Excel.XlSaveAction.xlDoNotSaveChanges)

bget = vault.CreateUtility(EdmUtility.EdmUtil\_BatchGet)

ReDim sel(0)

sel(0).mlDocID = dfile.ID

sel(0).mlProjID = fol.ID

bget.AddSelection(vault, sel)

bget.CreateTree(0, EdmGetCmdFlags.Egcf\_Lock)

If bget.ShowDlg(0) = True Then

bget.GetFiles(0)

End If

xl.Workbooks.Open(fname)

Catch ex As Exception

err = ex.Message

MsgBox(err, MsgBoxStyle.Information)

If err.Length > 0 Then

Exit Sub

End If

End Try

Else

MsgBox("The file was already Checked Out!!", MsgBoxStyle.Information, "EPDM-MS Office Connector")

End If

Else

MsgBox("This is not a vault document!!", MsgBoxStyle.Information, "EPDM-MS Office Connector")

End If

End If

// To UndoCheck out More Number of File from Vault //

If dfile.IsLocked = True Then

Try

wb.Close(Excel.XlSaveAction.xlDoNotSaveChanges)

bul = vault.CreateUtility(EdmUtility.EdmUtil\_BatchUnlock)

ReDim sel(0)

sel(0).mlDocID = dfile.ID

sel(0).mlProjID = fol.ID

bul.AddSelection(vault, sel)

If bul.CreateTree(0, EdmUnlockBuildTreeFlags.Eubtf\_MayUndoLock) = False Then

Exit Sub

End If

cont = bul

cont.GetItems(EdmRefItemType.Edmrit\_All, vitem)

id = LBound(vitem)

While id <= UBound(vitem)

poitem = vitem(id)

poitem.SetProperty(EdmRefItemProperty.Edmrip\_CheckKeepLocked, False)

id = id + 1

End While

If bul.ShowDlg(0) = True Then

bul.UnlockFiles(0)

End If

xl.Workbooks.Open(fname)

// To Show Form as Dialogue //

Dim verform As Form

verform = New VersionInfo (VersionInfo is form name)

verform.ShowDialog()

// To Get Latest Version //

If dfile.IsLocked = False Then

Try

wb.Close(Excel.XlSaveAction.xlDoNotSaveChanges)

bget = vault.CreateUtility(EdmUtility.EdmUtil\_BatchGet)

ReDim sel(0)

sel(0).mlDocID = dfile.ID

sel(0).mlProjID = fol.ID

bget.AddSelection(vault, sel)

bget.CreateTree(0, EdmGetCmdFlags.Egcf\_Nothing)

If bget.ShowDlg(0) = True Then

bget.GetFiles(0)

End If

xl.Workbooks.Open(fname)

// To Get Inbox Window //

Dim reg As RegistryKey

Dim kval As String

Dim mailpath As String

Dim vstr() As String

kval = "SOFTWARE\\SolidWorks\\Applications\\PDMWorks Enterprise"

reg = Registry.LocalMachine.OpenSubKey(kval)

If Not reg Is Nothing Then

kval = reg.GetValue("LanFileDir", 0, RegistryValueOptions.None)

reg.Close()

vstr = Split(kval, "\")

mailpath = vstr(0)

For Me.i = 1 To (UBound(vstr) - 2)

mailpath = mailpath & "\" & vstr(i)

Next

mailpath = mailpath & "\Inbox.exe"

'MsgBox(mailpath)

System.Diagnostics.Process.Start(mailpath)

End If

// To Get Search Window //

Dim reg As RegistryKey

Dim kval As String

Dim searchpath As String

Dim vstr() As String

kval = "SOFTWARE\\SolidWorks\\Applications\\PDMWorks Enterprise"

reg = Registry.LocalMachine.OpenSubKey(kval)

If Not reg Is Nothing Then

kval = reg.GetValue("LanFileDir", 0, RegistryValueOptions.None)

reg.Close()

vstr = Split(kval, "\")

searchpath = vstr(0)

For Me.i = 1 To (UBound(vstr) - 2)

searchpath = searchpath & "\" & vstr(i)

Next

searchpath = searchpath & "\Search.exe"

'MsgBox(mailpath)

System.Diagnostics.Process.Start(searchpath)

End If

// To Get version Names on Ribbon combo box //

Dim ddi As RibbonDropDownItem

If findex = 1 Then

vault.LoginAuto(vname, 0)

fol = vault.GetFolderFromPath(wb.Path)

dfile = vault.GetFileFromPath(fname, fol)

verenum = dfile

cmbver.Items.Clear()

For Me.i = 0 To (dfile.CurrentVersion - 1)

vers = verenum.GetVersion(i + 1)

ddi = Globals.Factory.GetRibbonFactory.CreateRibbonDropDownItem

If i = 0 Then

ddi.Label = (i + 1) & ". " & "<Created>"

Else

If vers.Comment.ToString = "" Then

ddi.Label = (i + 1) & ". " & "<No Comment>"

Else

ddi.Label = (i + 1) & ". " & vers.Comment.ToString

End If

End If

cmbver.Items.Add(ddi)

Next i

// To Create Ribbon Button on runtime //

btn = Me.Factory.CreateRibbonButton

btn.ControlSize = Microsoft.Office.Core.RibbonControlSize.RibbonControlSizeRegular

btn.Label = wftrans.Name

btn.Name = "mnubtn\_" & btn.Id

AddHandler btn.Click, AddressOf btn\_OnClick

// To Get Transition for Current State //

wfstat = dfile.CurrentState

pos = wfstat.GetFirstTransitionPosition

Dim idx As Integer = -1

mnucstat.Items.Clear()

While Not pos.IsNull

wftrans = wfstat.GetNextTransition(pos)

'MsgBox(wftrans.Name)

idx = idx + 1

btn = Me.Factory.CreateRibbonButton

btn.ControlSize = Microsoft.Office.Core.RibbonControlSize.RibbonControlSizeRegular

btn.Label = wftrans.Name

btn.Name = "mnubtn\_" & btn.Id

AddHandler btn.Click, AddressOf btn\_OnClick

mnucstat.Items.Add(btn)

End While

// To Change State for the file //

Dim bcstate As IEdmBatchChangeState

Dim butn As RibbonButton

Dim csfile As IEdmFile5

Dim csfol As IEdmFolder5

Dim boolstat As Boolean

'MsgBox(butn.Label)

bcstate = vault.CreateUtility(EdmUtility.EdmUtil\_BatchChangeState)

xl = GetObject(, "Excel.Application")

wb = xl.ActiveWorkbook

csfile = vault.GetFileFromPath(wb.FullName)

If csfile.IsLocked = False Then

csfol = vault.GetFolderFromPath(wb.Path)

bcstate.AddFile(csfile.ID, csfol.ID)

butn = DirectCast(sender, RibbonButton)

bcstate.CreateTree(butn.Label)

boolstat = bcstate.ShowDlg(0)

If boolstat = True Then

bcstate.ChangeState(0)

End If

// To Get Version //

If findex = 1 Then

vault.LoginAuto(vname, 0)

fol = vault.GetFolderFromPath(wb.Path)

dfile = vault.GetFileFromPath(fname, fol)

Try

wb.Close(Excel.XlSaveAction.xlDoNotSaveChanges)

verenum = dfile

vno = Split(cmbver.Text, ".")(0)

vers = verenum.GetVersion(vno)

vers.GetFileCopy(0)

xl.Workbooks.Open(fname)