Enterprise product data management api programs:

//To load vault name in form //

Public vault As IEdmVault8 = New EdmVault5

Public views() As EdmViewInfo

Private Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

vault.GetVaultViews(views, False)

For i = LBound(views) To UBound(views)

ComboBox1.Items.Add(views(i).mbsVaultName)

Next

If ComboBox1.Items.Count > 0 Then

j = ComboBox1.Items.Count

ComboBox1.Text = ComboBox1.Items.Item(0)

End If

End Sub

//To login in vault either autologin or login method //

username = TextBox1.Text

password = TextBox2.Text

vaultname = ComboBox1.Text

Try

If vault.IsLoggedIn = False Then

If TextBox1.Text = "" And TextBox2.Text = "" Then

vault.LoginAuto(vaultname, 0)

Else

vault.Login(username, password, vaultname)

MsgBox(" login sucessfully ")

End If

End If

Me.Hide()

Form3.Show()

Catch ex As Exception

ex.ToString()

End Try

// Get vaultname and path //

Private Sub Form2\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Label2.Text = vaultname

Label3.Text = vault.RootFolderPath

ToolTip1.SetToolTip(Label3, vault.RootFolderPath)

Exit Sub

End Sub

// Get group names in vault //

Private Sub B2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles B2.Click

ListBox3.Items.Clear()

ListBox4.Items.Clear()

ListBox5.Items.Clear()

wkf = vault.CreateUtility(EdmUtility.EdmUtil\_WorkflowMgr)

pos = wkf.GetFirstWorkflowPosition

While Not pos.IsNull

wrkflw = wkf.GetNextWorkflow(pos)

ListBox3.Items.Add(wrkflw.Name)

End While

End Sub

// Get User names in vault //

Private Sub ListBox1\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ListBox1.SelectedIndexChanged

ListBox2.Items.Clear()

uname = ListBox1.SelectedItem

If uname <> "" Then

pdmgrp = usermgr.GetUserGroup(uname)

pos = pdmgrp.GetFirstUserPosition()

While Not pos.IsNull

pdmuser = pdmgrp.GetNextUser(pos)

ListBox2.Items.Add(pdmuser.Name)

End While

End If

End Sub

// Get Workflow names in vault //

Private Sub B2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles B2.Click

ListBox3.Items.Clear()

ListBox4.Items.Clear()

ListBox5.Items.Clear()

wkf = vault.CreateUtility(EdmUtility.EdmUtil\_WorkflowMgr)

pos = wkf.GetFirstWorkflowPosition

While Not pos.IsNull

wrkflw = wkf.GetNextWorkflow(pos)

ListBox3.Items.Add(wrkflw.Name)

End While

End Sub

// Get State and transition name in vault //

Private Sub ListBox3\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ListBox3.SelectedIndexChanged

ListBox4.Items.Clear()

ListBox5.Items.Clear()

Try

pos = wkf.GetFirstWorkflowPosition

While Not pos.IsNull

wrkflw = wkf.GetNextWorkflow(pos)

If wrkflw.Name = ListBox3.SelectedItem Then

stspos = wrkflw.GetFirstStatePosition()

tranpos = wrkflw.GetFirstTransitionPosition()

While Not stspos.IsNull

wfstat = wrkflw.GetNextState(stspos)

ListBox4.Items.Add(wfstat.Name)

End While

While Not tranpos.IsNull

wftran = wrkflw.GetNextTransition(tranpos)

ListBox5.Items.Add(wftran.Name)

End While

End If

End While

Catch ex As Exception

ex.ToString()

End Try

End Sub

// Get Folder and File from vault //

Rootpath = vault.RootFolderPath

mainfolder = vault.GetFolderFromPath(Rootpath)

pos = mainfolder.GetFirstSubFolderPosition

While Not pos.IsNull

folder = mainfolder.GetNextSubFolder(pos)

lbox1.Items.Add(folder.Name)

If lbox1.Items.Contains("<items>") Then

lbox1.Items.Remove("<items>")

End If

End While

pos = mainfolder.GetFirstFilePosition

While Not pos.IsNull

file = mainfolder.GetNextFile(pos)

lbox2.Items.Add(file.Name)

End While

// go through folder to get folder files //

a = lbox1.SelectedItem

If folderpath = "" Then

Rootpath = vault.RootFolderPath

mainfolder = vault.GetFolderFromPath(Rootpath)

pos = mainfolder.GetFirstSubFolderPosition

While Not pos.IsNull

folder = mainfolder.GetNextSubFolder(pos)

foldername = folder.Name

If foldername = a Then

Exit While

End If

End While

folderpath = folder.LocalPath

mainfolder = vault.GetFolderFromPath(folderpath)

pos = mainfolder.GetFirstSubFolderPosition

lbox1.Items.Clear()

If Not pos.IsNull Then

While Not pos.IsNull

folder = mainfolder.GetNextSubFolder(pos)

lbox1.Items.Add(folder.Name)

End While

Else

lbox1.Items.Add("Empty")

End If

pos = mainfolder.GetFirstFilePosition

lbox2.Items.Clear()

If Not pos.IsNull Then

While Not pos.IsNull

file = mainfolder.GetNextFile(pos)

lbox2.Items.Add(file.Name)

End While

Else

lbox2.Items.Add("Empty")

End If

Else

mainfolder = vault.GetFolderFromPath(folderpath)

pos = mainfolder.GetFirstSubFolderPosition

While Not pos.IsNull

folder = mainfolder.GetNextSubFolder(pos)

foldername = folder.Name

If foldername = a Then

Exit While

End If

End While

folderpath = folder.LocalPath

mainfolder = vault.GetFolderFromPath(folderpath)

pos = mainfolder.GetFirstSubFolderPosition

lbox1.Items.Clear()

If Not pos.IsNull Then

While Not pos.IsNull

folder = mainfolder.GetNextSubFolder(pos)

lbox1.Items.Add(folder.Name)

End While

Else

lbox1.Items.Add("Empty")

End If

pos = mainfolder.GetFirstFilePosition

lbox2.Items.Clear()

If Not pos.IsNull Then

While Not pos.IsNull

file = mainfolder.GetNextFile(pos)

lbox2.Items.Add(file.Name)

End While

Else

lbox2.Items.Add("Empty")

End If

End If

//Initial Loading of Version //

filename = lbox2.SelectedItem

If folderpath = "" Then

folder = vault.GetFolderFromPath(Rootpath)

file = folder.GetFile(filename)

Else

folder = vault.GetFolderFromPath(folderpath)

file = folder.GetFile(filename)

End If

TB1.Text = file.GetLocalVersionNo(folder.ID)

Tb2.Text = file.CurrentVersion

verenum = file

pos = verenum.GetFirstVersionPosition

Cb2.Items.Clear()

While Not pos.IsNull

ver = verenum.GetNextVersion(pos)

Cb2.Items.Add(ver.VersionNo)

End While

If Cb2.Items.Count > 0 Then

Cb2.Text = Cb2.Items.Item(0)

End If

TextBox1.Text = ver.VersionDate

d = ver.FileSize

TextBox2.Text = ((d / 1000) & "KB")

TextBox3.Text = ver.User.Name

TextBox4.Text = ver.Comment

// Get Version of the file //

TextBox1.Text = ""

TextBox2.Text = ""

TextBox3.Text = ""

TextBox4.Text = ""

b = Cb2.SelectedItem

If b = "" Then

MsgBox("Select Version", MsgBoxStyle.OkOnly)

Else

verenum = file

ver = verenum.GetVersion(CInt(b))

ver.GetFileCopy(0)

TB1.Text = ver.VersionNo

End If

TextBox1.Text = ver.VersionDate

d = ver.FileSize

TextBox2.Text = ((d / 1000) & "KB")

TextBox3.Text = ver.User.Name

TextBox4.Text = ver.Comment

// Get Latest Version of the file //

TextBox1.Text = ""

TextBox2.Text = ""

TextBox3.Text = ""

TextBox4.Text = ""

c = Cb2.Items.Count

ver = verenum.GetVersion(c)

ver.GetFileCopy(0)

TB1.Text = ver.VersionNo

TextBox1.Text = ver.VersionDate

d = ver.FileSize

TextBox2.Text = ((d / 1000) & "KB")

TextBox3.Text = ver.User.Name

TextBox4.Text = ver.Comment

// Get version History of the file //

filename = lbox2.SelectedItem

If folderpath = "" Then

folder = vault.GetFolderFromPath(Rootpath)

file = folder.GetFile(filename)

Else

folder = vault.GetFolderFromPath(folderpath)

file = folder.GetFile(filename)

End If

TB1.Text = file.GetLocalVersionNo(folder.ID)

Tb2.Text = file.CurrentVersion

verenum = file

pos = verenum.GetFirstVersionPosition

'Dim ver As IEdmVersion5

Dim message As String

message = "History of " + file.Name + ":" + vbLf

Dim str As String

While Not pos.IsNull

ver = verenum.GetNextVersion(pos)

str = ver.VersionNo

message = message + str

str = ver.FileDate

message = message + ", file date=" + str + ", user=" + ver.User.Name

str = ver.FileSize

message = message + ", file size=" + str + " bytes, comment=" + ver.Comment + vbLf

End While

MsgBox(message)

// Get Transition target msgbox //

Try

wrkmgr = vault

pos = wrkmgr.GetFirstWorkflowPosition

lbwork.Items.Clear()

If Not pos.IsNull Then

While Not pos.IsNull

workflow = wrkmgr.GetNextWorkflow(pos)

lbwork.Items.Add(workflow.Name)

If workflow.Name = f Then

Exit While

End If

End While

End If

Dim message As String

message = "The transitions of " + vault.Name + ":" + vbLf + vbLf

pos = workflow.GetFirstTransitionPosition

While Not pos.IsNull

transition = workflow.GetNextTransition(pos)

message = message + transition.Name + " ("

If transition.FromStateID > 1 Then

message = message + transition.FromState.Name

End If

message = message + " => " + transition.ToState.Name + ")" + vbLf

End While

MsgBox(message)

wrkmgr = vault

pos = wrkmgr.GetFirstWorkflowPosition

lbwork.Items.Clear()

If Not pos.IsNull Then

While Not pos.IsNull

workflow = wrkmgr.GetNextWorkflow(pos)

lbwork.Items.Add(workflow.Name)

End While

End If

Catch ex As Exception

Dim ename As String

Dim edesc As String

ename = Err.ToString()

edesc = Err.Description

vault.GetErrorString(Err.Number, ename, edesc)

MsgBox(ename + vbLf + edesc)

End Try

// Check in files into the vault //

Try

filename = ListBox2.SelectedItem

If filepath = "" Then

filepath = rootpath & "\" & filename

Else

file = vault.GetFileFromPath(filepath)

length = Len(filename)

lenofpath = Len(filepath)

lenofpath = lenofpath - length

folderpath = folderpath.Substring(0, lenofpath - 1)

folder = vault.GetFolderFromPath(folderpath)

End If

file.UnlockFile(folder.ID, "", 0)

MsgBox("File is Check In Sucessfully ", MsgBoxStyle.Information, "Check In Information")

TextBox1.Text = ""

TextBox2.Text = ""

folderpath = ""

// or checkin //

Try

If folderpath = "" Then

Folder = vault.GetFolderFromPath(Rootpath)

file = Folder.GetFile(filename)

Else

Folder = vault.GetFolderFromPath(folderpath)

file = Folder.GetFile(filename)

End If

If file.IsLocked Then

file.UnlockFile(Folder.ID, "", 0)

MsgBox("file is Check in sucessfully")

TextBox1.Text = ""

TextBox3.Text = ""

End If

Catch ex As Exception

ex.ToString()

End Try

// checkout file in the vault and check out details //

Try

filename = ListBox2.SelectedItem

If folderpath = "" Then

filepath = rootpath & "\" & filename

folderpath = rootpath

Else

filepath = folderpath & filename

End If

'\\\\ To Initialize File \\\\\

file = vault.GetFileFromPath(filepath)

chkoutpath = folderpath

chkoutfolder = vault.GetFolderFromPath(chkoutpath)

Dim id As Integer

id = chkoutfolder.ID

file.LockFile(id, 0)

MsgBox("File is Check out Sucessfully ", MsgBoxStyle.Information, "Check Out Information")

Catch ex As Exception

ex.ToString()

End Try

// or checkout //

Try

If folderpath = "" Then

Folder = vault.GetFolderFromPath(Rootpath)

file = Folder.GetFile(filename)

Else

Folder = vault.GetFolderFromPath(folderpath)

file = Folder.GetFile(filename)

End If

If file.IsLocked Then

Else

file.LockFile(Folder.ID, 0)

MsgBox("file is Check out sucessfully")

TextBox1.Text = file.LockedByUser.Name

TextBox3.Text = file.LockedOnComputer

End If

Catch ex As Exception

ex.ToString()

End Try

// Get Properties of the file //

pos = folder.GetFirstFilePosition

While Not pos.IsNull

file = folder.GetNextFile(pos)

i = file.Name

If i = ListBox2.SelectedItem Then

TextBox6.Text = file.GetLocalFileDate(folder.ID)

TextBox7.Text = file.GetLocalFileSize(folder.ID)

TextBox1.Text = file.Name

TextBox2.Text = file.GetLocalPath(folder.ID)

TextBox3.Text = file.GetLocalVersionNo(folder.ID)

TextBox4.Text = file.CurrentRevision

TextBox5.Text = file.CurrentVersion

Exit Sub

End If

End While

// Load different variable names in form //

vrbmgr = vault.CreateUtility(EdmUtility.EdmUtil\_VariableMgr)

pos = vrbmgr.GetFirstVariablePosition

Cvar.Items.Clear()

If Not pos.IsNull Then

While Not pos.IsNull

var = vrbmgr.GetNextVariable(pos)

Cvar.Items.Add(var.Name)

End While

If Cvar.Items.Count > 0 Then

Cvar.Text = Cvar.Items.Item(0)

End If

Else

Cvar.Text = "Empty"

MsgBox("No varibles are present")

End If

// To get Transition target as per initial state selected //

Try

Dim stname As String

wrkmgr = vault.CreateUtility(EdmUtility.EdmUtil\_WorkflowMgr)

pos = wrkmgr.GetFirstWorkflowPosition

While Not pos.IsNull

wrkflw = wrkmgr.GetNextWorkflow(pos)

If wrkflw.name = a Then

Exit While

End If

End While

Dim stinitial, stid As Integer

pos = wrkflw.GetFirstStatePosition

While Not pos.IsNull

state = wrkflw.GetNextState(pos)

stname = state.Name

If wrkflw.InitialState.Name = stname Then

stinitial = state.ID

Exit While

End If

End While

Dim message As String

Dim tosta As Integer

Dim n, m, o, p, q, r, s, t As Integer

pos = wrkflw.GetFirstTransitionPosition

While Not pos.IsNull

trans = wrkflw.GetNextTransition(pos)

If trans.FromStateID = stinitial Then

message = message + trans.Name + " ("

message = message + trans.FromState.Name

message = message + "=>" + trans.ToState.Name + ")" + vbLf

tosta = trans.ToStateID

n = tosta

i1 = trans.FromState.Name

t1 = trans.Name

s2 = trans.ToState.Name

Exit While

End If

End While

pos = wrkflw.GetFirstTransitionPosition

While Not pos.IsNull

trans = wrkflw.GetNextTransition(pos)

If trans.FromStateID = stinitial And trans.ToStateID <> tosta Then

message = message + trans.Name + " ("

message = message + trans.FromState.Name

message = message + "=>" + trans.ToState.Name + ")" + vbLf

tosta = trans.ToStateID

m = tosta

Exit While

End If

End While

pos = wrkflw.GetFirstTransitionPosition

While Not pos.IsNull

trans = wrkflw.GetNextTransition(pos)

If trans.FromStateID = n Then

message = message + trans.Name + " ("

message = message + trans.FromState.Name

message = message + "=>" + trans.ToState.Name + ")" + vbLf

tosta = trans.ToStateID

o = tosta

End If

End While

pos = wrkflw.GetFirstTransitionPosition

While Not pos.IsNull

trans = wrkflw.GetNextTransition(pos)

If trans.FromStateID = n And trans.ToStateID <> tosta Then

message = message + trans.Name + " ("

message = message + trans.FromState.Name

message = message + "=>" + trans.ToState.Name + ")" + vbLf

tosta = trans.ToStateID

p = tosta

Exit While

End If

End While

pos = wrkflw.GetFirstTransitionPosition

While Not pos.IsNull

trans = wrkflw.GetNextTransition(pos)

If trans.FromStateID = m Then

message = message + trans.Name + " ("

message = message + trans.FromState.Name

message = message + "=>" + trans.ToState.Name + ")" + vbLf

tosta = trans.ToStateID

q = tosta

Exit While

End If

End While

pos = wrkflw.GetFirstTransitionPosition

While Not pos.IsNull

trans = wrkflw.GetNextTransition(pos)

If trans.FromStateID = m And trans.ToStateID <> tosta Then

message = message + trans.Name + " ("

message = message + trans.FromState.Name

message = message + "=>" + trans.ToState.Name + ")" + vbLf

tosta = trans.ToStateID

r = tosta

End If

End While

pos = wrkflw.GetFirstTransitionPosition

While Not pos.IsNull

trans = wrkflw.GetNextTransition(pos)

If trans.FromStateID = p Then

message = message + trans.Name + " ("

message = message + trans.FromState.Name

message = message + "=>" + trans.ToState.Name + ")" + vbLf

tosta = trans.ToStateID

s = tosta

Exit While

End If

End While

MsgBox(message)

// To Create runtime button //

Dim b As New Button

b.Size = New Size(145, 29)

b.Location = New Point(187, 359)

b.Visible = True

Me.Controls.Add(b)

b.Text = "Flowchart"

b.BringToFront()

AddHandler b.Click, AddressOf test