Imports Microsoft.VisualBasic

Public Class FBBuilder

Protected sType, sMedia, sURL, sThumb, sTitle, sCaption, sCaption1, sAlt, sLinkStyle, sLinkText, sTip, sInfo, sPrint As String

Protected sCaptionText As String

Protected sHeight, sWidth, sPct As String

Protected iImageSize As Integer

Protected fbReturn As String

Protected m As Match

Protected ReadOnly sPlayerPath As String = "https://r21concierge.s3.amazonaws.com/js/"

Protected ReadOnly sPlayerFile As String = "flowplayer-3.2.7.swf"

WriteOnly Property mediaType() As String

Set(ByVal value As String)

sType = value

End Set

End Property

WriteOnly Property mediaURL() As String

Set(ByVal value As String)

sMedia = value

End Set

End Property

WriteOnly Property thumb() As String

Set(ByVal value As String)

sThumb = value

End Set

End Property

WriteOnly Property height() As String

Set(ByVal value As String)

sHeight = value

End Set

End Property

WriteOnly Property width() As String

Set(ByVal value As String)

sWidth = value

End Set

End Property

WriteOnly Property size() As Integer

Set(ByVal value As Integer)

iImageSize = value

End Set

End Property

WriteOnly Property title() As String

Set(ByVal value As String)

sTitle = value

If sTitle <> "" Then

sCaptionText += "caption:`" + sTitle + "` "

End If

End Set

End Property

WriteOnly Property caption() As String

Set(ByVal value As String)

sCaption = value

If sCaption <> "" Then

sCaptionText += "caption2:`" + sCaption + "` "

End If

End Set

End Property

WriteOnly Property alt() As String

Set(ByVal value As String)

sAlt = value

End Set

End Property

WriteOnly Property linkType() As String

Set(ByVal value As String)

sLinkStyle = value

End Set

End Property

WriteOnly Property linkText() As String

Set(ByVal value As String)

sLinkText = value

End Set

End Property

WriteOnly Property tip() As String

Set(ByVal value As String)

sTip = value

End Set

End Property

WriteOnly Property infoLink() As String

Set(ByVal value As String)

sInfo = value

End Set

End Property

WriteOnly Property print() As String

Set(ByVal value As String)

sPrint=value

End Set

End Property

Public Function getFB() As String

Call getURL(sMedia)

Return AssembleFB()

End Function

Protected Sub getURL(ByVal embedString As String) 'checks to see if flv (via media type for now…)

'Dim rFlv As Regex = New Regex("https?://\S+?\.(flv|mp4)", RegexOptions.IgnoreCase)

'm = rFlv.Match(embedString)

'If m.Success Then

If sType = "R21Video" Then

Dim sPlayerParams As String = "'plugins':{" \_

+ "'controls':{'height':24,'sliderColor':'#ff1207','sliderGradient':'none'," \_

+ "'progressColor':'#00a89d','volumeColor':'#00a89d','volumeSliderColor':'#ff1207'," \_

+ "'autoHide':'always'}}"

sMedia = sPlayerPath + sPlayerFile

sMedia += "?config={'clip':{'" + m.Value + "'}," + sPlayerParams + "}"

End If

End Sub

Protected Function AssembleFB() As String

'// FLOATBOX STRING ASSEMBLER

'PUTS TOGETHER THE FLOATBOX STRINGS TO BE ACTIVATED ON THE FEEDS PAGE, AS FOLLOWS:

'routine for sText = (1)sPreamble + (2)sFbOps + (3)sAnchorText

'(1) preamble

Dim sFbClass As String = "floatbox"

Dim sDispClass As String = " rc" + Replace(StrConv(sType, VbStrConv.ProperCase), " ", "") 'sets second class element according to media type

Dim sPreamble As String = "<div>" + "<a class=""" + sFbClass + " " + sDispClass + """ href=""" + sMedia + """ "

'(2) sFbOps = sDispFormat + sCaptionText + sDialogText + sGroupOps + " "

Dim sDispFormat As String = ""

'-------------------------------------------------------------------------------

'[TODO] Display size routines BOB

'using this for testing

If iImageSize = 2 Then 'Full Window

If sMedia = "Video" Then

'sDispFormat = "sizeRatio:" + sRatio = " " 'sWidth + "/" + sHeight + " "

Else

End If

Else

If sMedia Like "\*R21\*Video\*" Then

sDispFormat = "scrolling:no "

End If

If sWidth <> "" Then

sDispFormat = "width:" + sWidth + " "

End If

If sHeight <> "" Then

sDispFormat += "height:" + sHeight + " "

End If

End If

'// End Display size Testing

'----------------------------------------------------------------------------

'[TODO]Dim sButton As String = "Full Window" 'placeholder Button declaration for sDispFormat Option group; [DEV] make Display As an option group

'sCaptionText built above in properties

Dim sDialogText As String = "" 'collects string data from Info/Tooltip dialog below

'[TODO] sDialogText - test in properties set FB options vals

Dim sGroupOps As String = ""

'[TODO] part of Media\_groups code now? sGroupOps = (sThisGroup + " " + sDoSlide + " " + sDoShow + " " + sEndShow + " ")

Dim sFbOps As String = "data-fb-options=""" + sDispFormat + sCaptionText + sDialogText + sGroupOps + """"

'(3) sAnchorText

Dim sAnchorText As String = "" 'final constructor, includes ...><img...></a>

'Dim isActive As Boolean = True 'whether a particular image link is active; normally one image link per gallery...

'If isActive Then ' if marked as "active" (meaning visible) link

Dim sSrcClass As String = "rc" + Replace(StrConv(sLinkStyle, VbStrConv.ProperCase), " ", "") 'linkType css class

Select Case sLinkStyle ' tell us which link type – button, text link or thumbnail; sLinkStyle Dim'd to = ddnLinkStyle selected value

Case "Text"

sAnchorText = "><span class=""" + sSrcClass + """>" + sLinkText + "</span></a>"

Case Else

Dim sImageText As String = "<img class=""" + sSrcClass + """ src=""" + sLinkText + """ "

If sAlt <> "" Then

sImageText += "alt=""" + sAlt + """ "

sAnchorText = " title=""" + sAlt + """ "

End If

sAnchorText += ">" + sImageText + "/></a>"

End Select

'Else

' sAnchorText = "</a>"

'End If ' isActive

Dim sText As String = sPreamble + sFbOps + sAnchorText + "</div>"

Return sText

End Function

End Class

Imports System.IO

Imports System.Drawing.Imaging

Imports System.Drawing

Imports System.Net

Imports System.Text.RegularExpressions

Imports System.Web

'uses flowplayer to play flv files; flowplayer is placed in the url string if .input is

'regex'd as a flv...

Public Class ParseEmbed\_BD 'TODO lose the \_BD

Dim dict As Dictionary(Of String, String)

Private sInput As String

Private iMedia As Integer

Private fileDate As String

ReadOnly sThumbVideo As String = "http://r21concierge.s3.amazonaws.com/images/video\_thumb.png"

ReadOnly sThumbImage As String = ""

Private sThumbURL As String = ""

Private sImgHeight As String

Private simgWidth As String

Private sHeight As String

Private bmHeight As String = ""

Private bmWidth As String = ""

Private sWidth As String

Private iPct As Integer

Private fPercent As Single

Private sTitle As String

'regex strings

Dim rZero As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rVid As Regex = New Regex("(?<=src[="":]+)http[s]?://[^"" ]+", RegexOptions.IgnoreCase)

Dim r21Vid As Regex = New Regex("http[s]?://[^"" ]+\.(flv|mp4)", RegexOptions.IgnoreCase)

Dim rImg As Regex = New Regex("https?[:%0-9a-z\_./-]+\.[a-z]{2,4}([%0-9a-z\_./~-])\*[%0-9a-z \_.~-](.bmp|.gif|.jpe?g|.png)(?:\?\w+\=+\w+($|(?=[ "";>])))?", RegexOptions.IgnoreCase) '(?<=(?:img|src)+[='"" ]+)?https?[^'""> ]+\.(?:jpg|jpeg|gif|png)(?:\?\w+\=\w+($|(?=[ "";>])))?

Dim rGroup As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rDoc As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rSite As Regex = New Regex("http://[\w./]+[.][\w]{2,3}", RegexOptions.IgnoreCase)

Dim rPage As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rHTML As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rZeroThm As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rVidThm As Regex = New Regex(sThumbVideo, RegexOptions.IgnoreCase)

Dim r21VidThm As Regex = New Regex("(?<=https?://).+(?=(?:.bmp|.gif|.jpe?g|.png))", RegexOptions.IgnoreCase)

Dim rImgThm As Regex = New Regex("(?<=https?://).+(?=(?:.bmp|.gif|.jpe?g|.png))", RegexOptions.IgnoreCase)

Dim rGroupThm As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rDocThm As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rSiteThm As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rPageThm As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rHTMLThm As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rEmbedIframe As Regex = New Regex("<iframe.\*?(vimeo|youtube).\*?>", RegexOptions.IgnoreCase)

Dim rEmbedObject As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rEmbedScript As Regex = New Regex("", RegexOptions.IgnoreCase)

Dim rEmbedFlv As Regex = New Regex("r21\_production.s3.amazonaws.com/videos/[1-9][0-9]{0,}/[A-z0-9\_-]+?\.(flv|mp4)", RegexOptions.IgnoreCase)

Dim rWidth As Regex = New Regex("(?<=width[:="" ]+)\d{2,4}%?", RegexOptions.IgnoreCase)

Dim rHeight As Regex = New Regex("(?<=height[:="" ]+)\d{2,4}%?", RegexOptions.IgnoreCase)

Dim rWebsite As Regex = New Regex("https?://[^/]\*(?=\/)")

Dim regMObjs() As Regex = {rZero, rVid, r21Vid, rImg, rGroup, rDoc, rSite, rPage, rHTML}

Dim regTObjs() As Regex = {rZeroThm, rVidThm, r21VidThm, rImgThm, rGroupThm, rDocThm, rSiteThm, rPageThm, rHTMLThm}

'[TODO] searches for embed patterns to set up parse routines for <iframe, <object, <script, etc.

'may be better done by website name...

Dim regEObjs() As Regex = {rEmbedIframe} ', rEmbedObject, rEmbedScript, rEmbedFlv}

Dim regURI As Regex

Dim m As Match

Dim mThumb As Match

'Sub New(ByVal input As String, ByVal mtype As Integer, ByVal pct As Integer)

' sInput = input 'from txtEmbed.Text

' iMedia = mtype 'from MediaType.index'

' iPct = pct 'from CInt(txtPct.Text)

'End Sub

Public WriteOnly Property input() As String

Set(ByVal value As String)

'do a function that looks at the type of input string...

m = regMObjs(iMedia).Match(value)

Select Case iMedia

Case 1

sInput = NormalizeEmbed(value) 'find out what embed style & normalize params

Case 2

'Do While Not m.Success

'MsgBox("Embed entry is not valid for type R21Video.", MsgBoxStyle.OkOnly)

'Loop

If m.Success Then

sInput = value

End If

Case 3

If m.Success Then

Call GetImageDims(value)

sInput = m.Value + " width:" + bmWidth + " height:" + bmHeight

End If

'Case 1, 2

' Dim oVidPage As New VideoPage

' sInput = oVidPage.makeMovie(value, sTitle, fileDate, iPct)

Case Else

If m.Success Then

sInput = value

End If

End Select

End Set

End Property

Public WriteOnly Property mtype() As Integer

Set(ByVal value As Integer)

iMedia = value

End Set

End Property

Public WriteOnly Property pct() As Integer

Set(ByVal value As Integer)

iPct = value

End Set

End Property

Public WriteOnly Property sdate() As String

Set(ByVal value As String)

fileDate = value

End Set

End Property

Public WriteOnly Property title() As String

Set(ByVal value As String)

sTitle = value

End Set

End Property

Public WriteOnly Property thumb() As String '[BOB][TODO]

Set(ByVal value As String)

m = rImg.Match(value)

If m.Success Then

sThumbURL = value

End If

End Set

End Property

'this populates a Dictionary collection with calls to the appropriate functions.

'In Catalog\_New, Get ClassInstanceName.results pulls in dictionary & assigns it to locals there.

Public ReadOnly Property results() As Dictionary(Of String, String)

Get

'Return GetMedia(iMedia)

dict = New Dictionary(Of String, String)

dict.Add("url", GetMedia(iMedia)) 'media url

dict.Add("thumb", GetThumb(iMedia)) 'thumb url

dict.Add("web", GetWeb()) 'website url

dict.Add("height", GetHeight()) 'height of media; for images, calculated in MakeThumb() and assigned to class var sImgHeight

dict.Add("width", GetWidth()) 'width of media; for images, calculated in MakeThumb() and assigned to class var sImgWidth

'dict.Add("ratio", GetRatio(sWidth, sHeight)) 'media ratio or w/h

Return dict

dict = Nothing

End Get

End Property

Protected Function NormalizeEmbed(ByVal embedRef As String) As String

For i = 0 To Len(regEObjs)

m = regEObjs(i).Match(embedRef)

If m.Success Then

Return m.Value

End If

Next

Dim oVidPage As New VideoPage

Return oVidPage.makeMovie(embedRef, sTitle, fileDate, iPct)

End Function

'Array of Media regex expressions regMObjs(n) - n is passed Media Type index in order of array members

Private Function GetMedia(ByVal n As Integer) As String

m = regMObjs(n).Match(sInput)

If Not m.Success Then

Return "Unable to Parse Valid Media URL (Function GetMedia)"

Else

Return m.Value

End If

End Function

'Array of Thumb regex expressions regMObjs(n) - n is passed Media Type index in order of array members

'returned thumb is either:

'GetThumb(iMedia) (normal route parsing sInput for an image url - typically images...) or

'ParseThumb(sThumbURL) if there is a valid URL passed to thumb() property (entered by user in txtLinkInfo

Private Function GetThumb(ByVal n As Integer) As String

mThumb = rImg.Match(sThumbURL)

If mThumb.Success Then

Dim iMediaTemp As Integer = iMedia

Dim sTempReturn As String

iMedia = 3 'THIS IS A TEMPORARY FIX for the case problem in MakeThumb

sTempReturn = MakeThumb(mThumb.Value)

iMedia = iMediaTemp

Return sTempReturn

Else

m = regMObjs(n).Match(sInput)

Return MakeThumb(m.Value)

End If

End Function

'

Private Function GetWeb() As String 'Parses website url

m = rWebsite.Match(sInput)

If Not m.Success Then

Return "Unable to Parse Web Address (Fn GetWeb())"

Else

Return m.Value

End If

End Function

'

Protected Function GetHeight() As String 'Parses Height attribute (if exixts...)

m = rHeight.Match(sInput)

If Not m.Success Then

sHeight = ""

Else

Select Case iMedia

Case 1, 2

sHeight = m.Value

Case Else

sHeight = (Integer.Parse(m.Value) \* iPct / 100).ToString

End Select

End If

Return sHeight

End Function

'

Protected Function GetWidth() As String 'Parses Height attribute (if exixts...)

m = rWidth.Match(sInput)

If Not m.Success Then

sWidth = ""

Else

Select Case iMedia

Case 1, 2

sWidth = m.Value

Case Else

sWidth = (Integer.Parse(m.Value) \* iPct / 100).ToString

End Select

End If

Return sWidth

End Function

'

Protected Function GetRatio(ByVal width As String, ByVal height As String) As String

Dim w = Single.Parse(width)

Dim h = Single.Parse(height)

If (h <= 0 Or w <= 0) Then

Return "0"

Else

Dim r = (w / h)

Return String.Format("{0:n3}", r)

End If

End Function

Protected Sub GetImageDims(ByVal sfilename As String)

'Create a new Bitmap Image loading from location of origional file

'create the web request

Dim wr As WebRequest = WebRequest.Create(sfilename)

wr.Timeout = 3000

'get the response into a stream

Dim st As Stream = wr.GetResponse.GetResponseStream

'create the image

Dim bm As Image = Image.FromStream(st)

st.Close()

bmWidth = bm.Width.ToString

bmHeight = bm.Height.ToString

End Sub

Protected Function MakeThumb(ByVal sfilename As String) As String

'draws & stores image; returns image url & exposes original height/weight

'[TODO] This needs to be same url for any number of passes. For now, using

Select Case iMedia

Case 3

'Create a new Bitmap Image loading from location of origional file

'create the web request

Dim wr As WebRequest = WebRequest.Create(sfilename)

wr.Timeout = 3000

'get the response into a stream

Dim st As Stream = wr.GetResponse.GetResponseStream

'create the image

Dim bm As Image = Image.FromStream(st)

st.Close()

Dim bmWidth As String = bm.Width.ToString

Dim bmHeight As String = bm.Height.ToString

'Dim bm As Bitmap = Image.FromStream(st)

Dim maxHeight% = 80

Dim maxWidth% = 80

Dim newHeight%, newWidth%

Dim ratio As Single = (bm.Width / bm.Height)

'Set Thumbnails Height and Width

If ratio < 1 Then

newHeight = maxHeight

Else

newHeight = (maxWidth / ratio)

End If

newWidth = (newHeight \* ratio)

'Create the new image as a blank bitmap & new graphic obj

Dim thumbnail As Bitmap = New Bitmap(newWidth, newHeight)

Dim g As Graphics = Graphics.FromImage(thumbnail)

'resize graphics object to fit onto the resized image

g.DrawImage(bm, New Rectangle(0, 0, thumbnail.Width, thumbnail.Height), 0, 0, bm.Width, bm.Height, GraphicsUnit.Pixel)

'dispose of Graphic obj.

g.Dispose()

'Create new path and filename for the resized image

Dim sThumb As String

Dim sThumbInner As String

'//For Server (Concierge site)

'Dim sRootPath As String = HttpContext.Current.Server.MapPath("~/") + "thumbs\"

'//For Server (dev-Concierge site)

Dim sRootPath As String = "G:\ftproot\concierge\thumbs\"

'//For Bob Local

'Dim sThumbPath As String = "D:\Users\Bob Day\Documents\Visual Studio 2008\Websites\ConciergeDev\thumbs\"

Dim sthumbRelPath As String = "thumbs/"

sThumb = regTObjs(iMedia).Match(sfilename).Value

If sThumb <> "" Then

sThumbInner = Replace(sThumb, ".", "\_")

sThumbInner = Replace(sThumbInner, "/", "\_") + DateTime.Now.ToString("yyMMddhhmmssff") + ".png"

'sThumb = sthumbRelPath + sThumbInner

'[TODO] fix the below reference (remove dev-)

sThumb = "http://concierge.rocket21.com/thumbs/" + sThumbInner

'thumbnail.Save(sThumbPath + sThumbInner, ImageFormat.Png)

thumbnail.Save(sRootPath + sThumbInner, ImageFormat.Png)

'[DEPLOY] On server, this should be "thumbnail.Save(sRootPath + sThumbInner, ImageFormat.Png)"

Else

sThumb = "Unable to Create Thumbnail (MakeThumb)"

End If

'//Make image dimensions available to class (CANT DO, BECAUSE an image's thumbnail will resize image...)

'sWidth = bmWidth

'sHeight = bmHeight

Return sThumb

'[TODO] Case statement will be replaced by individual functions.

' These use the regTObjs array for parsing.

Case 1 'Video

Return sThumbVideo

Case 2 'R21 Video

Return sThumbVideo

Case 4 'Group

Return ""

Case 5 'Document

Return ""

Case 6 'Site

Return ""

Case 7 'Page

Return ""

Case 8 'HTML

Return ""

Case Else

Return ""

End Select

End Function

End Class