Création d'un User Control
et utilisation de celui-ci en ASP.NET
(sous Visual Studio.NET et en VB.NET)

Pré-requis

Dans cet exemple, je vais utiliser la librairie de classes présentée juste avant (ASP.NET Chaînes Utilitaires) afin de traîter les strings qui seront ajoutées à ce User Control.

- **ASP.NET Chaînes Utilitaires** (http://www.asp-php.net/scripts/asp.net/lib_class.php)

Définition

Tout d'abord, définissons simplement ce qu'est un User Control dans ASP.NET.
Il s'agit simplement d'un composant (Souvent graphique) sous forme de classe et intégré dans la page en cours de développement.

Les Users Controls sont la réponse de Microsoft face à la technique des inclusions sous PHP, contrairement à l'include de ASP 3, les ajouts de Users Controls dans un pages ASPX peuvent être conditionnées et peuvent être ajoutés avec des paramêtres transmis à ceux-ci.

Comme expliqué précédemment, un User Control est une classe et à ce titre peut posséder des fonctions, des propriétés, des procédures, et des variables le tout publiques ou privées.

En revanche, à la différence des classes simple, les User Controls ont une parties présentation (xxx.ascx) et une partie Code Behind (xxx.ascx.vb (ou cs si on est en C#)).
Ainsi, on a donc l'évènement de base page_load qui permet de définir les actions se faisant au chargement de la page.

Exemple de Résultat

Source du User Control

Dans cet exemple de User Control, le but est d'être le plus proche possible d'une classe VB, ainsi la page ASCX ne contient que l'entete spécifiant le fichier de Code Behind :

Fichier URLRollOverInfo.ascx

```
<%@
Control Language="vb"
AutoEventWireup="false"  Codebehind="URLRollOverInfo.ascx.vb"
Inherits="URLRollOverInfo"
TargetSchema="http://schemas.microsoft.com/intellisense/ie5"
%>
```

Présentation des Composantes du Code Behind

Le Code Behind contient la procédure principale chargée à l'instanciation de ce User Control (page_load) dans une autre page, mais aussi la liste des propriétés publiques et des variables privées.

- **Association de bloc HTML avec une clé**

Ce User control gère aussi les requis de la librairie de script Overlib :
Ajout de la ligne lançant la demande du fichier de script :

```
<script language="javascript" src="Répertoire de Scripts JS/overlib.js">
</script>
```

Ajout de la ligne DIV qui sera cachée :

```
<div id='overDiv' style='position:absolute; visibility:hidden; z-index:1000;'>
</div>
```

Ces 2 lignes doivent être présentes dans le fichier Source HTML une seule fois et donc dans le Code Behind, un appel permet d'associer un ajout de code dans la page avec une clé et de tester à chaque instanciation de ce User Control si cette clé n'est pas déja présente. Si c'est le cas, il n'ajoute pas le source HTML, sinon il l'ajoute.

Cela se fait par le code suivant :

```
Dim MonSourceAAjouter As String = "<div id='overDiv' "_
        "style='position:absolute; visibility:hidden; z-index:1000;'></div>"

If Not Page.IsClientScriptBlockRegistered("MaClé") Then
  Page.RegisterClientScriptBlock("MaClé", MonSourceAAjouter)
End If
```

- **Ajout dynamique de string dans un autre String**

Dans le Code Behind figure une fonction qui permet d'ajouter automatiquement 3 variables dans une autre variable String.
Il faut donc définir une constante String préformattée et qui recevra les autres variables string.

```
Private Const IncludeScriptFormat As String = ControlChars.CrLf & _
   "<script language=""{0}"" src=""{1}{2}""></script>"
Private Const ScriptFileName As String = "overlib.js"

....

Dim location As String = Page.Request.ApplicationPath + "/script/"
Dim includeScript As String = [String].Format(IncludeScriptFormat, "javascript", location, ScriptFileName)

If Not Page.IsClientScriptBlockRegistered("DeclarationFichierJS") Then
  Page.RegisterClientScriptBlock("DeclarationFichierJS", includeScript)
End If
```

Fichier URLRollOverInfo.ascx.vb

```
Public MustInherit Class URLRollOverInfo
    Inherits System.Web.UI.UserControl

    ' Liste des Variables Privées
    Private _TexteURL As String = ""
    Private _AdresseURL As String = ""
    Private _CSSClass As String = ""
    Private _LeTarget As String = "_self"

    Private _TitreBoiteSurvol As String = ""
    Private _TexteBoiteSurvol As String = ""
    Private _TailleBordure As String = "1"

    Private _BackGroundPicture As String = ""
    Private _CouleurFondTitreBoite As String = ""
    Private _CouleurFondTexteBoite As String = "#999999"
    Private _TextColor As String = ""

    Private _Width As String = ""
    Private _Height As String = ""
    Private _TextSize As String = ""

    Private _IsHTMLPopup As Boolean = False
    Private _IsFullHTMLControl As Boolean = False

    Private Const IncludeScriptFormat As String = ControlChars.CrLf & "<script language=""{0}""
src=""{1}{2}""></script>"
    Private Const ScriptFileName As String = "overlib.js"

    ' Liste des Propriétées Publiques
    Public Property URLTexte() As String
        Get
            Return _TexteURL
        End Get
        Set(ByVal Value As String)
            _TexteURL = Value
```

```
    End Set
End Property
Public Property AdresseURL() As String
    Get
        Return _AdresseURL
    End Get
    Set(ByVal Value As String)
        _AdresseURL = Value
    End Set
End Property
Public Property CSSClass() As String
    Get
        Return _CSSClass
    End Get
    Set(ByVal Value As String)
        _CSSClass = Value
    End Set
End Property
Public Property LeTarget() As String
    Get
        Return _LeTarget
    End Get
    Set(ByVal Value As String)
        _LeTarget = Value
    End Set
End Property

Public Property TitreBoiteSurvol() As String
    Get
        Return _TitreBoiteSurvol
    End Get
    Set(ByVal Value As String)
        _TitreBoiteSurvol = Value
    End Set
End Property
Public Property TexteBoiteSurvol() As String
    Get
        Return _TexteBoiteSurvol
    End Get
    Set(ByVal Value As String)
        _TexteBoiteSurvol = Value
    End Set
End Property
Public Property TailleBordure() As String
    Get
        Return _TailleBordure
    End Get
    Set(ByVal Value As String)
        _TailleBordure = Value
    End Set
End Property

Public Property BackGroundPicture() As String
    Get
        Return _BackGroundPicture
    End Get
    Set(ByVal Value As String)
        _BackGroundPicture = Value
    End Set
End Property
```

```
Public Property CouleurFondTitreBoite() As String
    Get
        Return _CouleurFondTitreBoite
    End Get
    Set(ByVal Value As String)
        _CouleurFondTitreBoite = Value
    End Set
End Property
Public Property CouleurFondTexteBoite() As String
    Get
        Return _CouleurFondTexteBoite
    End Get
    Set(ByVal Value As String)
        _CouleurFondTexteBoite = Value
    End Set
End Property
Public Property TextColor() As String
    Get
        Return _TextColor
    End Get
    Set(ByVal Value As String)
        _TextColor = Value
    End Set
End Property

Public Property Width() As String
    Get
        Return _Width
    End Get
    Set(ByVal Value As String)
        _Width = Value
    End Set
End Property
Public Property Height() As String
    Get
        Return _Height
    End Get
    Set(ByVal Value As String)
        _Height = Value
    End Set
End Property
Public Property TextSize() As String
    Get
        Return _TextSize
    End Get
    Set(ByVal Value As String)
        _TextSize = Value
    End Set
End Property

Public Property IsHTMLPopup() As Boolean
    Get
        Return _IsHTMLPopup
    End Get
    Set(ByVal Value As Boolean)
        _IsHTMLPopup = Value
    End Set
End Property
Public Property IsFullHTMLControl() As Boolean
    Get
```

```
            Return _IsFullHTMLControl
        End Get
        Set(ByVal Value As Boolean)
            _IsFullHTMLControl = Value
        End Set
    End Property

#Region " Code généré par le Concepteur Web Form "

    'Cet appel est requis par le Concepteur Web Form.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
        'CODEGEN : cet appel de méthode est requis par le Concepteur Web Form
        'Ne le modifiez pas en utilisant l'éditeur de code.
        InitializeComponent()
    End Sub

#End Region

    ' -----------------------------------------------------
    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Dim DeclarationDIV As String = "<div id='overDiv' " _
            " style='position:absolute; visibility:hidden; z-index:1000;'></div>"
        Dim location As String = Page.Request.ApplicationPath + "/script/"
        Dim includeScript As String = [String].Format(IncludeScriptFormat, "javascript", location,
ScriptFileName)

        ' ------------------------
        ' Déclaration des deux parties obligatoires dans la page pour faire
        ' fonctionner le Script

        If Not Page.IsClientScriptBlockRegistered("DeclarationFichierJS") Then
            Page.RegisterClientScriptBlock("DeclarationFichierJS", includeScript)
        End If

        If Not Page.IsClientScriptBlockRegistered("DeclarationDIV") Then
            Page.RegisterClientScriptBlock("DeclarationDIV", DeclarationDIV)
        End If

        ' ------------------------
        ' Attribution de la couleur du fond du Texte du Survol
        Dim ActionOnMouseOver As String = "javascript:return overlib('" & TexteBoiteSurvol & _
            "' , FGCOLOR, '" & CouleurFondTexteBoite & "'"

        ' ------------------------
        'Si l'on souhaite un titre dans la cellule de survol
        If TitreBoiteSurvol <> "" Then
            ActionOnMouseOver &= ", CAPTION, '" & _
                Chaines_Utilitaires.Chaines_Utilitaires.ManipChaine.FormatteJavascript(TitreBoiteSurvol) & "'"
        End If

        ' ------------------------
        ' Couleur du Fond de la cellule de Survol
        If CouleurFondTitreBoite <> "" Then
            ActionOnMouseOver &= ", BGCOLOR, '" & CouleurFondTitreBoite & "'"
        End If
```

```
' ------------------------
' Fond de la cellule de Survol dans le cas d'une image
 If BackGroundPicture <> "" Then
     ActionOnMouseOver &= ", BACKGROUND, '" & BackGroundPicture & "'"
 End If

' ------------------------
' Cas ou on a de l'HTML avec un lien par exemple
' et que l'on souhaite que celui-ci soit cliquable
If IsHTMLPopup = True Then
    ActionOnMouseOver &= ", STICKY"
 End If

' ------------------------
' Si on est sur d'envoyer de l'HTML Brut
If IsFullHTMLControl = True Then
    ActionOnMouseOver &= ", FULLHTML"
 End If

' ------------------------
' Taille et couleur du Texte
If TextColor <> "" Then
    ActionOnMouseOver &= ", TEXTCOLOR, '" & TextColor & "'"
 End If
If TextSize <> "" Then
    ActionOnMouseOver &= ", TEXTSIZE, " & TextSize
 End If

' ------------------------
' Taille de la cellule RollOver
If Width <> "" Then
    ActionOnMouseOver &= ", WIDTH, " & Width
 End If
If Height <> "" Then
    ActionOnMouseOver &= ", HEIGHT, " & Height
 End If

 ActionOnMouseOver &= ", BORDER, " & TailleBordure & ");"

 If AdresseURL <> "" Then
    ' ------------------------
    ' Cas ou la cellule de survol apparait sur un Lien
    Dim MonLienSurvol As New System.Web.UI.WebControls.HyperLink()
    With MonLienSurvol
        .CssClass = CSSClass
        .NavigateUrl = AdresseURL
        .Target = LeTarget
        .Text = URLTexte
        .Attributes.Add("onmouseover", ActionOnMouseOver)
        .Attributes.Add("onmouseout", "javascript:return nd();")
    End With
    Me.Controls.Add(MonLienSurvol)
 Else
    ' ------------------------
    ' Cas ou la cellule de survol apparait sur un simple Div généré
    Dim MonLienSurvol As New System.Web.UI.WebControls.Label()
    With MonLienSurvol
        .CssClass = CSSClass
        .Text = URLTexte
```

```
            .Attributes.Add("onmouseover", ActionOnMouseOver)
            .Attributes.Add("onmouseout", "javascript:return nd();")
        End With
        Me.Controls.Add(MonLienSurvol)
    End If
  End Sub

  ' ----------------------------------------------------
End Class
```

Utilisation de ce User Control

Ce User Control est a ajouter dans votre projet directement dans le répertoire contenant tous vos UC (un peu comme une Bibliothèque).

Afin d'utiliser celui-ci sous Visual Studio.NET, il suffit de faire faire glisser le User Control sur la page ASPX (ou ASCX) voulu (en mode Design). Sinon mettre dans le fichier source de la page ASPX (ou ASCX) le texte suivant (ce qui revient à la même chose que le Cliqué-Glissé sous VS.NET) :

```
<%@ Register
        TagPrefix="UserControl"
        TagName="MonUserControl"
        src="URLRollOverInfo.ascx"
%>
```

Une fois le Register fait, il y a plusieurs possibilités :

- Soit ce User Control est dans un Datagrid (1)
- Soit ce User Control est ajouté dans la page ASPX directement (2)
- Soit ce User Control est géré directement par le Code Behind (3)

Cas N°1 : User Control dans un Datagrid

Dans la page ASPX, lors de la création du Datagrid, on peut paramétrer des colonnes avec un format particulier (Appelé les Templates Columns), et ainsi dire que dans la colonne spécifiée on aura un texte (issu du Bind du Datagrid) avec le survol (via ce User Control) dont les données nécessaires seront issues de la base.

Ceci se fait par le code suivant dans la page ASPX :

```
<asp:TemplateColumn HeaderText="Nom du Film" ItemStyle-Width="50%">
 <ItemTemplate>
    <UserControl:MonUserControl
   id="MonUserControlBoiteSurvol"
   URLTexte='<%# Container.DataItem("nom_du_film") %>'
   AdresseURL='<%# "../?page=fiche&film="& Container.DataItem("ref_film") %>'
   CouleurFondTexteBoite='#999999'
   CouleurFondTitreBoite='#333333'
   TailleBordure='2'
   TitreBoiteSurvol='Détails'
   TexteBoiteSurvol='<%# Container.DataItem("realisateurs") %>'
   runat="server">
    </UserControl:MonUserControl>
  </ItemTemplate>
</asp:TemplateColumn>
```

Pour bien comprendre ce code, il faut travailler sur les Datagrid et la gestion de ceux-ci.

Cas N°2 : User Control dans la page ASPX directement

Il se gère de la même facon que pour le datagrid avec le code suivant :

```
<%@ Register
       TagPrefix="uc1"
       TagName="URLRollOverInfo"
       Src="URLRollOverInfo.ascx" %>

...

<uc1:URLRollOverInfo
       id="URLRollOverInfo1"
       URLTexte='Le Texte du Lien'
       AdresseURL='MaPageDestination.asp'
       CouleurFondTexteBoite='#999999'
       CouleurFondTitreBoite='#333333'
       TailleBordure='2'
       TitreBoiteSurvol='Détails'
       TexteBoiteSurvol='Mon Texte Qui sera Affiché'
       runat="server">
</uc1:URLRollOverInfo>
```

Cas N°3 : User Control géré directement par le Code Behind

Dans ce cas, la ligne Register (dans la page ASPX) n'est pas nécessaire, tout ce gère alors par le Code Behind de la page ou on veut insérer le User Control.
Cela donnera donc le code suivant :

```
Dim ZoomCollab As URLRollOverInfo
ZoomCollab = LoadControl("./URLRollOverInfo.ascx")
With ZoomCollab
  .TitreBoiteSurvol = "Le Titre de la Boite"
  .CouleurFond = "#999999"
  .TexteURL = "Le Texte du Lien"
  .CSSClass = "ClassCSSDulien"
  .AdresseURL = "AdresseURL.asp"
  .TexteBoiteSurvol = "Le Texte dans la boite du Survol"
  ...

End With

...

Me.Controls.Add(ZoomCollab)
```

Ainsi, il ne vous reste plus qu'à adapter les différents exemples fournis à votre projet ASP.NET.

Conclusion

Une version de cet article est téléchargeable ici :

- ASP.NET - Javascript URL Survol (http://fromelard.free.fr/Scripts/ASPNET_URLRollOverInfo)

Cet article nous montre donc comment créer un User Control et utiliser celui-ci dans votre projet ASP.NET.

F___