Les Custom WEB Controls en ASP.NET

Comment créer son propre Control WEB

Lors de développement d'applications WEB, on a découvert précédemment comment utiliser les Contrôles Utilisateur (ASCX) intégrable dans les pages ASPX. On souhaite maintenant pouvoir avoir la même chose mais sans avoir de fichier ASCX afin de pouvoir avoir une DLL complète intégrant tout nos Contrôles.

ASP.NET : Les Custom Web Controls

Introduction

On a vu dans les articles précédents qu'on pouvait développer des Contrôles Utilisateur et les utiliser simplement dans les pages ASPX ou même dans d'autres User Controls.

- **Exemple de User Control en ASP.NET (FR)**
- **Exemple d'Application ASP.NET utilisant les Users Controls (FR)**
- **Exemple d'Application ASP.NET au maximum les Users Controls (FR)**

Maintenant que cette utilisation est acquise, on souhaiterait améliorer le principe afin de créer une bibliothèque de contrôles. Le problème avec les Users Controls est que ceux-ci nécessitent l'usage de fichier ASCX qui ne peuvent pas être ajoutés directement dans une DLL (lors de la compilation).

En effet, la compilation des User Controls n'utilise que les fichiers 'Code Behind' (xxxx.ascx.vb ou xxxx.ascx.cs). De ce fait, les fichiers ASCX qui sont indispensables lors de l'utilisation des contrôles ne peuvent pas être stockés dans la DLL.

Maintenant comment peut-on obtenir ce résultat sans avoir le problème expliqué et donc pouvoir avoir le Contrôle entier dans la DLL. Cela se fait par l'utilisation des Custom Web Controls. Voyons comment faire.

Présentation

Les Custom Web Controls sont des composants ASP.NET que l'on peut placer directement dans des pages ASPX ou des User Controls (ASCX), ceci depuis une DLL compilée.

Ceci est possible parce que les Custom Web Controls ont génèrent dynamiquement un flux HTML par l'usage du Render (comme les composants ASP.NET standards) qui va renvoyer durectement le flux HTML en sortie.

Une présentation des Custom Web Controls est disponible sur le site MSDN :

- **Contrôles utilisateur Web et contrôles Web personnalisés (FR)**

On y trouve aussi une comparaison entre les Users Controls et les Custom Web Controls :

- **Comparaison entre les contrôles utilisateur Web et les contrôles Web personnalisés (FR)**

Voyons maintenant comment créer et utiliser ces Custom Web Controls.

Création d'un Custom Web Control

Notre exemple est très simple, il s'agit d'avoir un Custom Web Control fournissant une entête que l'on placera simplement en début de page en enrichissant les propriétés de ce contrôle.

Ce controle aura quatres propriété :

- Titre de la Page
- Soustitre de la Page
- Commentaire de la Page
- Image de la Page

Ainsi le résultat ressemblera à ceci :

Voici la source de la Classe qui compose le Custom Web Control. Il suffit d'ajouter celle-ci dans le projet voulu qui peut être le projet du Site WEB ASP.NET ou dans le projet de Librairie de Classes.

```
'*****************************************************
' $Archive: $
' $Author: $
' $Date: $ $Revision: $
'  Description : Exemple de développement d'un CustomWebControl basé sur un Exemple d'Entête
'              de Page que l'on placera simplement dans la page ASPX
'
'*****************************************************
Imports System.ComponentModel
Imports System.Web.UI

''' ----------------------------------------------------------------
''' Project    : DotNetSysInfo
''' Class      : FabWebControl
'''
''' ----------------------------------------------------------------
''' <summary>
''' Exemple de développement d'un CustomWebControl basé sur un Exemple d'Entête
''' de Page que l'on placera simplement dans la page ASPX
''' </summary>
''' <remarks>
''' </remarks>
''' <history>
'''   [moi]  04/08/2004  Created
''' </history>
''' ----------------------------------------------------------------
<DefaultProperty("TitreEntete"), ToolboxData("<{0}:FabWebControl runat=server></{0}:FabWebControl>")> &_
Public Class FabWebControl
   Inherits System.Web.UI.WebControls.WebControl

   Dim _TitreEntete As String = ""
   Dim _SousTitreEntete As String = ""
   Dim _CommentaireTitreEntete As String = ""
   Dim _URLImageEntete As String = ""

#Region "Propriétés du WebControl"

   ''' ----------------------------------------------------------------
   ''' <summary>
   ''' Titre de l'entete que l'on utilisera dans la page
   ''' </summary>
   ''' <returns></returns>
   ''' <remarks>
   ''' </remarks>
   ''' <history>
   '''   [moi]  04/08/2004  Created
   ''' </history>
```

```
    ''' -----------------------------------------------------------------
    <Bindable(True), Category("Appearance"), DefaultValue("")> Property [TitreEntete]() As String
        Get
            Return _TitreEntete
        End Get

        Set(ByVal Value As String)
            _TitreEntete = Value
        End Set
    End Property

    ''' -----------------------------------------------------------------
    ''' <summary>
    ''' SousTitre de l'entete que l'on utilisera dans la page
    ''' </summary>
    ''' <returns></returns>
    ''' <remarks>
    ''' </remarks>
    ''' <history>
    '''   [moi]  04/08/2004  Created
    ''' </history>
    ''' -----------------------------------------------------------------
    <Bindable(True), Category("Appearance"), DefaultValue("")> Property [SousTitreEntete]() As String
        Get
            Return _SousTitreEntete
        End Get

        Set(ByVal Value As String)
            _SousTitreEntete = Value
        End Set
    End Property

    ''' -----------------------------------------------------------------
    ''' <summary>
    ''' Commentaire de l'entete que l'on utilisera dans la page
    ''' </summary>
    ''' <returns></returns>
    ''' <remarks>
    ''' </remarks>
    ''' <history>
    '''   [moi]  04/08/2004  Created
    ''' </history>
    ''' -----------------------------------------------------------------
    <Bindable(True), Category("Appearance"), DefaultValue("")> Property [CommentaireTitreEntete]() As String
        Get
            Return _CommentaireTitreEntete
        End Get

        Set(ByVal Value As String)
            _CommentaireTitreEntete = Value
        End Set
    End Property

    ''' -----------------------------------------------------------------
    ''' <summary>
    ''' URL de l'image qui sera associée à l'entête de la page
    ''' </summary>
    ''' <returns></returns>
    ''' <remarks>
    ''' </remarks>
    ''' <history>
    '''   [moi]  04/08/2004  Created
    ''' </history>
    ''' -----------------------------------------------------------------
    <Bindable(True), Category("Appearance"), DefaultValue("")> Property [URLImageEntete]() As String
        Get
            Return _URLImageEntete
        End Get

        Set(ByVal Value As String)
            _URLImageEntete = Value
        End Set
    End Property

    ' -----------------------------------------------------------------
#End Region
```

```
    ''' ----------------------------------------------------------------
    ''' <summary>
    ''' Sortie envoyée par le WebControl lors de son placement dans la page
    ''' </summary>
    ''' <param name="output"></param>
    ''' <remarks>
    ''' </remarks>
    ''' <history>
    '''   [moi]  04/08/2004  Created
    ''' </history>
    ''' ----------------------------------------------------------------
    Protected Overrides Sub Render(ByVal output As System.Web.UI.HtmlTextWriter)

        ' On va déja paramétrer la date courante afin de l'afficher dans l'entête
        Dim c As New System.Globalization.CultureInfo("fr-FR")
        Dim tempdate As String = DateTime.Now.ToString("f",c.DateTimeFormat)

        ' Variable temporaire qui contiendra le flux HTML à renvoyer
        Dim Temp As New System.Text.StringBuilder

        ' Chargement du flux HTML que l'on aura en sortie
        Temp.Append("<TABLE cellSpacing='1' cellPadding='1' width='100%' border='0'>")
        Temp.Append("<TR width='100%'>")
        Temp.Append("<TD>")
        If Not [URLImageEntete] Is Nothing AndAlso [URLImageEntete] <> "" Then
            Temp.Append("<IMG src='" & [URLImageEntete] & "' align='left' Height='62'>")
        End If
        Temp.Append("<FONT SIZE=4><B>")
        Temp.Append([TitreEntete])
        Temp.Append("</B></FONT><BR><FONT SIZE=2><B>")
        Temp.Append([SousTitreEntete])
        Temp.Append("</B></FONT><BR>")
        Temp.Append([_CommentaireTitreEntete])
        Temp.Append("<TD vAlign='top' align='right'>")
        Temp.Append(tempdate)
        Temp.Append("</TD></TR></TABLE><BR><BR>")

        output.Write(Temp.ToString)
    End Sub

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

Utilisation du Custom Web Control développé

Une fois la classe disponible, il existe 2 méthodes d'utiliser ce contrôle :

- Par la page ASPX
- Par le Code Behind

Nous présenterons les deux méthodes dans cet article.

Dans la page ASPX (ou ASCX) :

```
....
<%@ Register
            TagPrefix="cc1"
            Namespace="LeNamSpace"
            Assembly="LeNomDuProjet" %>
....
<cc1:FabWebControl
            id="FabWebControl1"
            runat="server"
            SousTitreEntete="Sous Titre 2"
            TitreEntete="Titre 2"
            CommentaireTitreEntete="Commentaire 2"
            URLImageEntete="./images/information.gif">
</cc1:FabWebControl>

....
```

Le référencement du Custom Web Control nécessite de fournir le Nom du NameSpace de la classe (s'il n'est pas définit, il faut fournir le nom du projet) ainsi que le nom de l'Assembly (c'est à dire le nom de la DLL). Les propriétés sont alors accessibles directement dans Visual Studio.NET dans la partie des propriétés.

Dans le Code Behind :

```
Dim Testcontrol As New FabWebControl

Testcontrol.CommentaireTitreEntete = "Le commentaire"
Testcontrol.URLImageEntete = "./images/question.gif"
Testcontrol.TitreEntete = "Le titre"
Testcontrol.SousTitreEntete = "Le Sous Titre"

Me.Controls.Add(Testcontrol)
```

Dans le code Behind il suffit de créer un nouvel Objet avec le nom de la classe, ensuite de charger les valeurs des propriétés.

Maintenant vous avez vu comment créer un Custom Web Control et s'en servir. Vous pouvez donc créer vos propres contrôles selon vos besoins.

Conclusion

Le développement des Custom WEB Control est une étape importante dans l'apprentissage du développement ASP.NET qu'il est indispensable de le maîtriser. En effet, il vous permettra de créer des bibliothèques de composants que vous pourrez enregistrer dans le Global Assembly Cache et ainsi réutiliser à volonté ces contrôles dans tous vos projets.

Celui-ci permet aussi d'ajouter votre control dans le menu de Visual Studio.NET afin de ne plus avoir qu'à faire un glisser-déposer dans votre page comme n'importe quel composant ASP.NET (Datagrid, Datalist, Label, ...).

Pour ceci, la méthode est complètement expliquée ici :

- **Procédure pas à pas : création d'un contrôle Web personnalisé (FR)**

J'espère que cet article vous aura servi. Voila quelques articles ou exemples sur ce sujet :

- **Contrôles utilisateur Web et contrôles Web personnalisés (FR)**
- **Une librairie de UserControl ? C'est possible ! (FR)**
- **Creating a Web Custom Control (US)**
- **Creating a Custom Web Control to Consume an XML Web Service (US)**
- **Creating a Custom Web Control to Consume an XML Web Service - Part 2 (US)**
- **Date Validator Custom Web Control (US)**
- **Creating Custom Web Controls (US)**
- **Creating Custom ASP.NET Controls in VS.NET (US)**

En vous souhaitant de bons projets de développement.

Romelard Fabrice (alias F___)