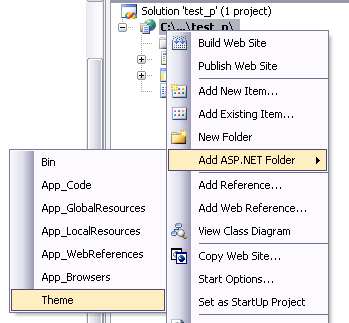
# Thèmes et Skins

## *Introduction aux thèmes*

Nous avons vu précédemment l’utilisation des MasterPages facilitant la mise en place et la duplication du design sur toutes les pages d’une application WEB, il manque cependant le formatage du site, plus exactement les couleurs, les polices, la taille des tableaux, la taille des différents composants etc….De plus les pages ne suivent pas forcément les même règles de formatages. C’est là qu’intervient les thèmes, plus exactement la mise en place d’un style sur un site.   
La mise en place du style sur un site se faisait souvent par l’intermédiaire des sources HTML, par l’utilisation des CSS (Cascading Style Sheet). La différence entre les CSS et les thèmes est que les CSS ne permettent pas le formatage des contrôles ASP.NET. C’est pourquoi les fonctionnalités des CSS et thèmes sont tout à fait complémentaire dans l'implémentation un site en ASP.NET 2.0.

## *Création d'un thème*

Les thèmes d’une application WEB en ASP.NET 2.0 sont stockés obligatoirement dans le dossier App\_Themes, qui est un dossier spécifique ASP.NET. De plus, un thème représente un dossier composé au minimum d’un fichier Skin (\*.skin).   
  
Pour créer le dossier App\_Themes, faites un clique-droit sur le nom de votre application WEB (et non la solution) puis «Add an ASP.NET Folder» puis « Theme ».



Vous avez ensuite la possiblité d’ajouter des nouveaux thèmes en allant dans App\_Themes > Add ASP.NET Folder > Theme   
Ensuite, pour ajouter un nouveau fichier Skin ou CSS à votre thème, il suffit tout simplement de faire un clique-droit sur le thème et « Add New Item ».

## *Les fichiers Skins*

Les fichiers Skins sont composés essentiellement de côntrole ASP.NET avec leurs propriétés de formattage (police, couleur, taille…). Vous avez la possibilité d’ajouter des contrôles communs (sans SkinID) ou des contrôles spécifiques (avec SkinID). Si le thème est appliqué, les contrôles communs s’appliqueront à tous les contrôles du même type alors que les contrôles spécifiques seront appliqués à tous les contrôles du même type et ayant le même SkinID.

Exemple d’un contrôle commun de type TextBox :

<asp:TextBox runat="server"

Font-Size="10px" Font-Names="Verdana" ForeColor="red" />

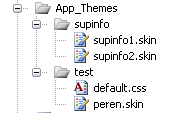
Si le thème est appliqué sur tout le site, toutes les TextBox auront le même style

Exemple d’un contrôle spécifique de type TextBox :

<asp:TextBox runat="server" SkinID="specifique"

Font-Size="10px" Font-Names="Verdana" ForeColor="red" />

Si le thème est appliqué sur tout le site, toutes les TextBox ayant comme SkinID « specifique » auront le même style



Voici un exemple de fichier Skin non exhaustif :

<asp:TextBox runat="server"

ForeColor="#000000"

font-name="Verdana"

Font-Size="9px"

height="13px"

borderWidth="1px"

borderStyle="Solid"

BorderColor="#82ACFF"

BackColor="#FFFFFF"

font-bold="false" />

<asp:DropDownList runat="server"

Font-Name="Verdana"

Font-Size="10px"

ForeColor="#000000"

Font-Bold="false"

borderWidth="1px"

borderStyle="Solid"

BorderColor="#82ACFF"

BackColor="#FFFFFF" />

<asp:ListBox Runat="Server" Font-Name="Verdana"

Font-Size="10pt"

BackColor="#FFFBFF"

ForeColor="gray"/>

<asp:GridView Runat="Server" SkinId="test"

CellPadding="1"

BorderStyle="solid"

BorderWidth="1px"

BackColor="whitesmoke"

ForeColor="#93B41A"

HeaderStyle-Font-Names="Verdana"

HeaderStyle-Font-Size="10px"

HeaderStyle-ForeColor="gray"

HeaderStyle-BackColor="white"

HeaderStyle-Font-Bold="true"

RowStyle-Font-Names="Verdana"

RowStyle-Font-Size="10pt"

RowStyle-ForeColor="#93B41A"

RowStyle-BackColor="whitesmoke"

AlternatingRowStyle-BackColor="white"

AlternatingRowStyle-ForeColor="#93B41A" />

<asp:Label runat="server" SkinID="LabelWhite"

ForeColor="white"

font-name="Verdana"

Font-size="10pt" />

<asp:Label runat="server" SkinID="LabelUnderline"

ForeColor="Gray"

font-name="Verdana"

Font-size="10pt"

Font-Underline="true" />

<asp:FileUpload runat="server"

ForeColor="Gray"

font-name="Verdana"

Font-Size="10pt"

borderWidth="1px"

BorderColor="#CCCCCC"

BackColor="#FFFBFF" />

<asp:Button runat="server"

BackColor="#FFFBFF"

BorderColor="#CCCCCC"

BorderStyle="solid"

BorderWidth="1px"

Font-name="Verdana"

Font-size="8pt"

ForeColor="Darkgray" />

<asp:RadioButtonList runat="server"

Font-Name="Verdana"

Font-Size="11px"

ForeColor="gray" />

<asp:CheckBox runat="server"

BackColor="#FFFBFF"

Font-Name="Verdana"

Font-Size="11px"

ForeColor="gray" />

<asp:RequiredFieldValidator runat="server"

Font-Name="Verdana"

Font-Size="11px"

ForeColor="red" />

<asp:CompareValidator runat="server"

Font-Name="Verdana"

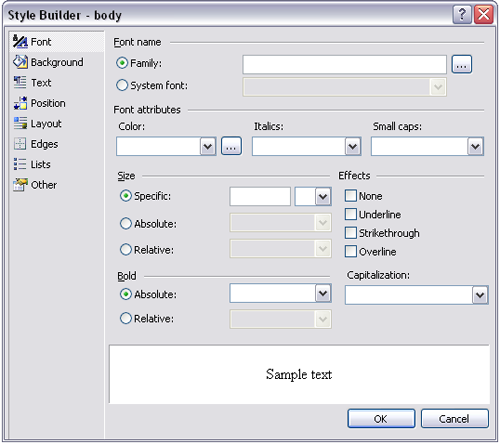
Font-Size="11px"

ForeColor="red" />

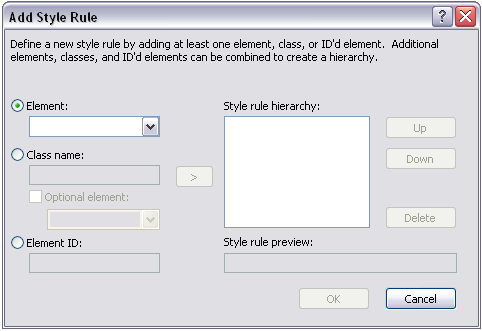
Il est tout à fait possible de créer plusieurs fichiers Skins dans un même thème, cependant les contrôles ayant un SkinID doivent être unique dans un thème, sinon vous ne pourrez pas compiler votre application WEB.

## *Les fichiers CSS*

Visual Studio 2005 intègre un outil de génération de style CSS par l’interface « Style Builder » permettant de définir vos style par le biais d’une interface simple. Pour afficher l’interface, il suffit de faire un clique-droit dans votre fichier css (dans une zone de style) puis « Build Style ».



Une interface pour ajouter des style CSS est disponible par un clique-droit dans votre feuille de style CSS puis « Add Style Rule ».



L’utilisation des CSS dans le cas d’une application WEB permet uniquement d’appliquer un style aux contrôles HTML, mais pas aux contrôles ASP.NET

## *Application d'un thème*

Un style peut être directement appliqué au contrôle ASP.NET en modifiant directement les propriétés d’un contrôle. Bien entendu, l’intérêt des Thèmes est de pouvoir « stocker » différents thèmes et pouvoir les appliquer plusieurs fois sans avoir à redéfinir le thème.

Voici les principales propriétés de style des contrôles :

|  |  |  |
| --- | --- | --- |
| Propriétés | Type | Description |
| BackColor | Texte | Couleur de l’arrière plan |
| BorderColor | Texte | Couleur de la bordure |
| BorderStyle | Texte | Style de la bordure |
| BorderWidth | Pixels | Largeur de la bordure |
| Font-Bold | Booleen | Caractère en gras |
| Font-Italic | Booleen | Caractère en italique |
| Font-Name | Texte | Police |
| Font-Overline | Booleen | Caractère surligné |
| Font-Size | Pixels | Taille des caractères |
| Font-Strikeout | Booleen | Caractère barré |
| Font-Underline | Booleen | Caractère sougliné |
| ForeColor | Texte | Couleur des caractères |
| Height | Pixels | Hauteur du contrôle |
| Visible | Booleen | Visibilité du contrôle |
| Width | Pixels | Largeur du contrôle |

Pour appliqer un thème automatiquement, il suffit de spécifier le nom du thème dans l’en-tête de la page.

<%@ Page Language="C#" AutoEventWireup="true" Theme="supinfo" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

Enfin, il faudra spécifier dans chaque contrôle le SkinID si vous souhaitez appliquer des styles spécifiques ayant un SkinID.

|  |  |
| --- | --- |
| Sans thème | Avec thème |
|  |  |

## *Appliquer un thème global*

Pour appliquer un thème à tout le site, pour éviter d’appliquer le thème page par page, il faut définir le thème à appliquer dans le fichier de configuration (Web.config) entre les balises system.web , en voici un exemple :

<?xml version="1.0"?>

<configuration>

<appSettings/>

<connectionStrings/>

<system.web>

<compilation debug="true"/>

<authentication mode="Windows"/>

<pages theme="test"/>

</system.web>

</configuration>

Dans ce cas, vous n’aurez pas à spécifier la propriété « Theme » sur chaque en-tête de chaque page. Par contre si votre thème contient des contrôles avec SkinID, il faudra les spécifier dans toutes vos pages.

## *Désactiver un thème*

Les thèmes appliqués sont prioritaires, ce qui n’est pas pratique si l’on souhaite appliquer un style spécifier à un et un seul contrôle. Dans ce cas vous avez la possibilité d’utiliser la propriété EnableTheming (booleen) disponible pour un contrôle ou une page.

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="test.aspx.cs" EnableTheming="false" Inherits="test " %>

Les thèmes sont désactivés sur la page test.aspx.

<asp:TextBox ID="TextBox1" ForeColor="red" EnableTheming="false" runat="server"></asp:TextBox>

La TextBox n’aura pas de thème appliqué, seul les propriétés de style définit dans le contrôle sera valable.