

Thèmes et apparences pour vos sites web avec ASP.NET 2.0

par Jérôme Lambert ([Mes articles](#))

Date de publication : 01/11/2007

Dernière mise à jour :

Au cours de cet article, vous découvrirez ce que sont les Thèmes de ASP.NET 2.0 et réaliserez votre premier thème en quelques minutes et sans effort.

I - Introduction.....	3
II - Les éléments d'un thème.....	3
II-A - Les apparences.....	3
II-B - Les feuilles de style (CSS).....	4
II-C - Les images et autres ressources.....	4
III - La portée des thèmes.....	4
III-A - Thèmes de page.....	4
III-B - Thèmes globaux.....	5
IV - Création de son premier thème.....	5
IV-A - Création du répertoire de gestion des thèmes.....	5
IV-B - Création de notre fichier Skin.....	6
IV-C - Création de notre feuille de style.....	7
IV-D - Application du thème à une page web.....	8
IV-E - Changement dynamique du thème.....	10
IV-F - Thème pour tout le site Web.....	11
V - Conclusion.....	12
VI - Téléchargement sources.....	12
VI - Liens.....	12
VIII - Remerciements.....	13

I - Introduction

Lorsque vous développez un site web, il arrive de vouloir proposer aux visiteurs plusieurs apparences différentes afin que chacun y trouve son bonheur. Cela va de la mise en page à la personnalisation de contrôles en passant par des changements au niveau des Fonts utilisées pour vos textes.



Avec ASP.NET 2.0, vous allez pouvoir développer vos sites et sur le côté décrire une série d'apparences, appelés thèmes. Ces thèmes vont vous permettre de définir l'apparence de vos pages et de vos contrôles afin de les appliquer facilement à une ou plusieurs pages voir à toute votre application web.

Au long de cet article, vous allez découvrir comment mettre en place des thèmes pour des sites ASP.NET 2.0 sous Visual Studio 2005/2008.

II - Les éléments d'un thème

Un thème se compose d'un ensemble d'éléments qui sont :

- Les apparences
- Les feuilles de style (CSS)
- Les images et autres ressources comme des fichiers textes

Les éléments cités ci-dessus ne sont pas tous obligatoires pour définir un thème. Un thème aura au minimum des apparences.

II-A - Les apparences

Les apparences permettent de paramétrer un ensemble de propriétés pour un ou plusieurs contrôles. Quand je parle de contrôles, je fais bien allusion aux contrôles ASP.NET tel que **TextBox**, **Label**, **Calendar**, **GridView**, etc. et quand je parle de propriétés, je veux bien sûr parler des propriétés des contrôles tel que **BackColor**, **ForeColor**, **BorderColor**, etc.

Il est important de savoir que vous ne pouvez pas utiliser n'importe quelle propriété d'un contrôle. Seules les propriétés ayant l'attribut **ThemeableAttribute** à **true** peuvent être redéfinies. Par exemple, vous ne pourrez pas utiliser la propriété **CommandName** du contrôle **Button** ou encore la propriété **DataSource** du contrôle **GridView**.

Pour définir un ensemble d'apparences pour un thème, on va utiliser un fichier de type **Skin** (extension **.skin**).

Avant de vous donner un exemple d'apparence dans un fichier skin, il vous faut d'abord savoir qu'il existe deux types d'apparences de contrôle :

- **L'apparence par défaut** : dans ce cas ci, on va définir un ensemble de propriétés pour un contrôle (exemple **TextBox**) et si on applique le thème à une page, l'apparence va s'appliquer à tous les contrôles du même type de la page en question (dans notre cas à tous les **TextBox**).

- **L'apparence nommée** : l'apparence nommée va être identifiée par un nom qui sera placé dans la propriété **SkinID**. Tous les contrôles du même type d'une page web qui auront comme **SkinID** le même nom auront donc l'apparence définie.

Exemple d'une apparence par défaut

```
<asp:TextBox runat="server" BorderColor="Red" BorderStyle="Dashed" BorderWidth="1px"></asp:TextBox>
```

Exemple d'une apparence nommée

```
<asp:TextBox ID="TextBox1" SkinID="TextBoxApparenceExemple" runat="server" BorderColor="Blue" BorderStyle="Groove" BorderWidth="1px"></asp:TextBox>
```



i A noter que vous pouvez avoir autant d'apparences de type de contrôles que vous le désirez dans un fichier Skin mais aussi répartir vos apparences dans plusieurs fichiers Skin.

II-B - Les feuilles de style (CSS)

En plus de définir des apparences dans un thème, vous pouvez y ajouter vos propres feuilles de style (CSS). Ci-dessous, un petit exemple qui définit la couleur de fond de la page ainsi que la couleur utilisée pour la police.

Exemple de fichier CSS

```
body
{
background-color: Silver;
color: Maroon;
}
```

II-C - Les images et autres ressources

Il est aussi possible d'inclure des fichiers ressources liés à un thème, comme des images, des fichiers textes, audio, vidéo, etc.

En général, vos fichiers ressources se trouveront dans un sous dossier de votre dossier des thèmes mais ce n'est pas une obligation. Ces fichiers peuvent très bien se trouver dans un dossier en dehors du dossier des thèmes.

Exemple d'une image se trouvant dans le dossier Thème

```
<asp:Image ID="t2" runat="server" ImageUrl="App_Themes/ThemeSubFolder_Ressources/dotnet.gif" />
```

Exemple d'une image se trouvant en dehors du dossier Thème

```
<asp:Image ID="t1" runat="server" ImageUrl="~/ApplicationSubFolder_Ressources/dotnet.gif" />
```

III - La portée des thèmes

Avant d'entrée dans le vif du sujet comme certains l'attendent avec impatience, il nous reste à parler d'un dernier point qui est la portée des thèmes.

III-A - Thèmes de page

Un thème de page est représenté par un répertoire de thème constitué de fichiers d'apparences de contrôles, de feuilles de style et de fichiers ressources.

III-B - Thèmes globaux

Le thème global est un thème qui peut être appliqué à tous les sites web d'un serveur. Tout comme les thèmes de pages, vous avez des fichiers d'apparences de contrôles, des feuilles de style et des fichiers de ressources.

A la différence des thèmes de pages, les thèmes globaux sont stockés dans un répertoire global du serveur Web intitulé **Themes**.

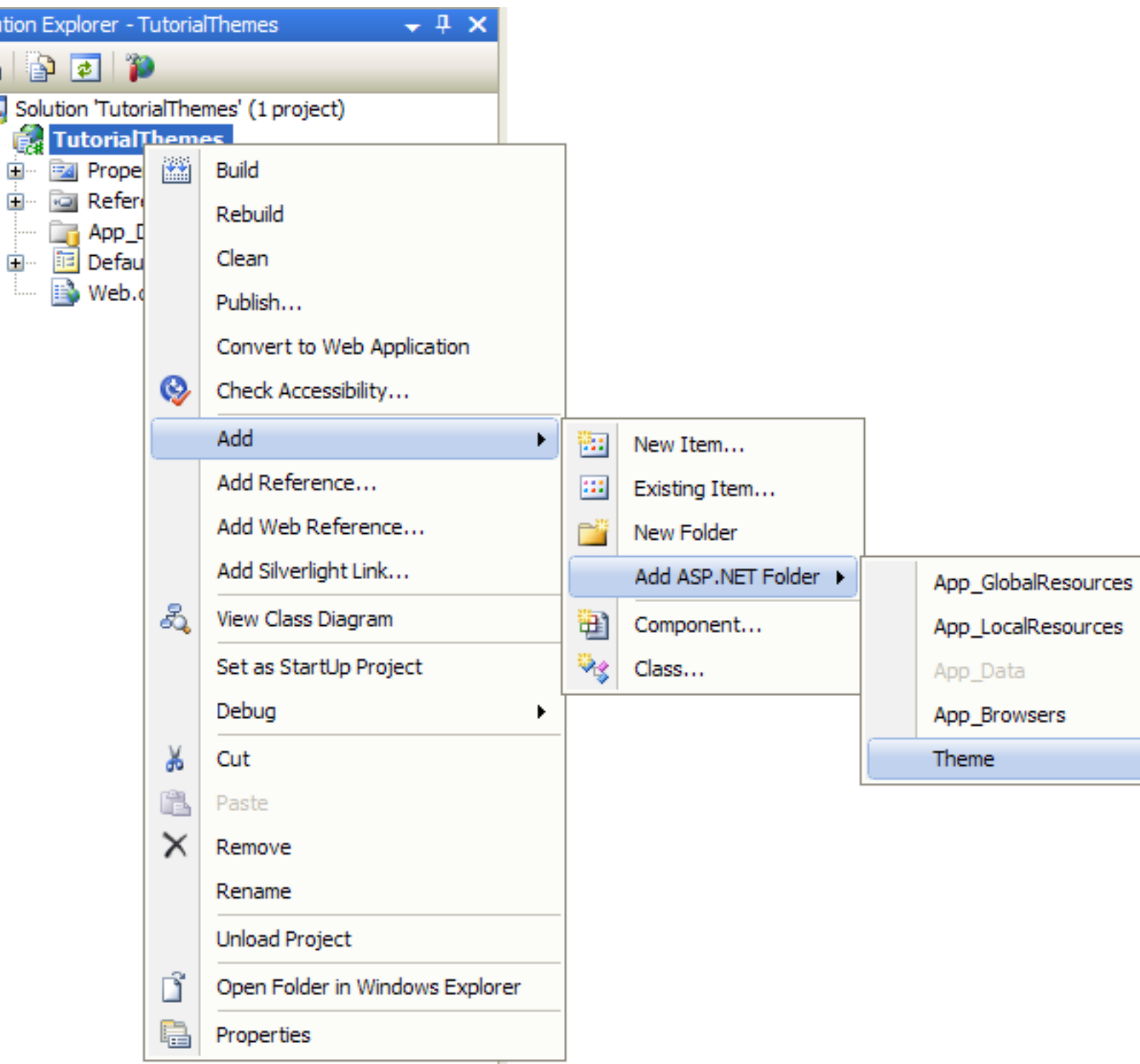
Étant donné qu'entre la création d'un thème de page et celle d'un thème global, il n'y a aucune différence mise à part l'endroit où se trouvent les thèmes en question, l'exemple qui suivra se basera sur le principe de thème de page.

IV - Création de son premier thème

À présent que vous savez presque tout en ce qui concerne les thèmes, nous allons voir comment les utiliser dans vos pages Web à l'aide de Visual Studio (2005/2008).

IV-A - Création du répertoire de gestion des thèmes

Une fois le projet créé, nous allons ajouter le répertoire qui gère les thèmes ASP.NET. Pour cela, faites cliquer droit sur votre projet dans la fenêtre "*Solution Explorer*" et cliquez sur "*Add/Add ASP.NET Folder/Theme*".



Le répertoire **App_Themes** est le répertoire de base permettant de gérer vos thèmes. D'ailleurs, lors de la création de ce répertoire, Visual Studio a créé par la même occasion le répertoire **Theme1** représentant votre premier thème de départ. Nous utiliserons ce répertoire pour établir notre premier thème en le renommant **MonPremierTheme**.

IV-B - Création de notre fichier Skin

Nous allons à présent créer notre fichier d'apparences à l'aide d'un fichier .skin ; faites clique droit sur le répertoire Theme1 puis sélectionnez "Add/New Item...". Dans la boîte de dialogue, sélectionnez le fichier de type .skin et nommez le fichier **monApparence.skin**.

Maintenant que le fichier d'apparences est créé, nous allons y définir les choses suivantes :

- Label nommé : texte en gras et souligné.
- Image nommée : représentée par une flèche qui sera placée devant notre Label nommé.
- Label par défaut : texte en gras.
- TextBox par défaut : bordure en pointillés et de couleur verte.
- Button par défaut : couleur de fond mauve et couleur de texte en blanc.

Ce qui nous donne le résultat suivant :

Fichier monSkin.skin

```
<asp:Label runat="server" SkinID="Titre" Font-Bold="true" Font-Underline="true"></asp:Label>
<asp:Image runat="server" SkinID="Fleche" ImageUrl="~/fleche.gif" />
<asp:Label runat="server" Font-Bold="true"></asp:Label>
<asp:TextBox runat="server" BorderStyle="Dotted" BorderColor="#008040" Width="98%"></asp:TextBox>
<asp:Button runat="server" BackColor="#8080C0" ForeColor="white" />
```

IV-C - Création de notre feuille de style

Nous allons aussi créer notre feuille de style (Clique droit sur le répertoire **monPremierTheme** et sélectionnez "Add/New Item...". Puis dans la boîte de dialogue, sélectionnez le fichier de type **Style Sheet** et nommez le **monStyle.css**) avec aura les propriétés suivantes :

- Fond d'écran gris.
- Police Verdana de taille 12.
- Les tableaux auront une bordure continue grise de 2 pixels.

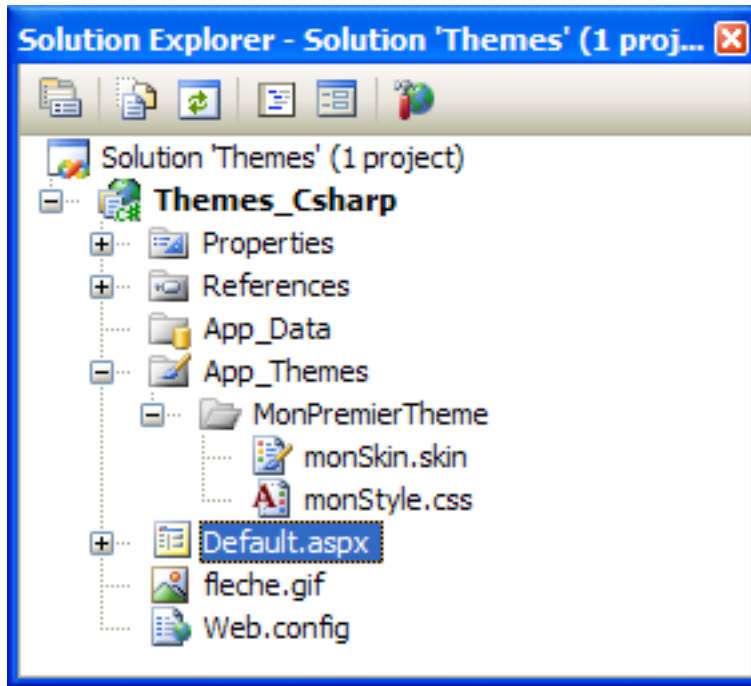
Ce qui nous donne le résultat suivant :

Fichier Style.css

```
body
{
    background-color: #E8E8E8;
    font-family: Verdana;
    font-size: 12px;
}

table
{
    border: 2px solid #CCC;
}
```

Pour ceux qui s'y sont perdus, vous devez obtenir l'organisation suivante dans votre projet Web :



IV-D - Application du thème à une page web

Passons à la création de notre page aspx qui va représenter un formulaire (sans logique de traitement derrière) afin que l'utilisateur puisse y insérer son nom, prénom et adresse email. Au passage, vous verrez trois boutons qui permettront de changer de thème dynamiquement (mais ce n'est pas pour tout de suite).

On obtient donc facilement le code suivant :

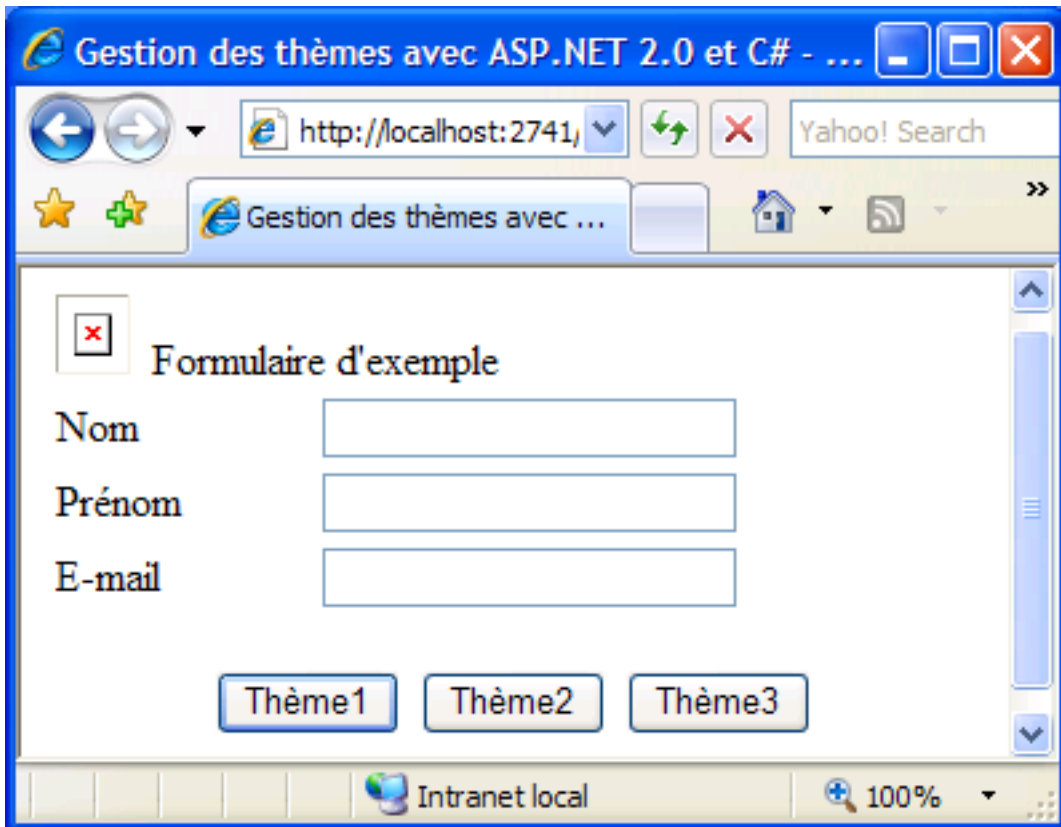
Fichier Default.aspx

```

<form id="form1" runat="server">
  <div>
    <table width="350px">
      <tr>
        <td colspan="2">
          <asp:Image ID="Image_Fleche" runat="server" />&nbsp;
          <asp:Label ID="Label_Titre" runat="server" Text="Formulaire d'exemple"></asp:Label>
        </td>
      </tr>
      <tr>
        <td style="width: 100px;"><asp:Label ID="Label_Nom" runat="server" Text="Nom "></asp:Label></td>
        <td style="width: 250px;"><asp:TextBox ID="TextBox_Nom" runat="server"></asp:TextBox></td>
      </tr>
      <tr>
        <td><asp:Label ID="Label_Prenom" runat="server" Text="PrÃ©nom "></asp:Label></td>
        <td><asp:TextBox ID="TextBox_Prenom" runat="server"></asp:TextBox></td>
      </tr>
      <tr>
        <td><asp:Label ID="Label_Email" runat="server" Text="E-mail "></asp:Label></td>
        <td><asp:TextBox ID="TextBox_Email" runat="server"></asp:TextBox></td>
      </tr>
      <tr>
        <td colspan="2" style="text-align: center;">
          <br />
          <asp:Button ID="Button_Theme1" runat="server" Text="ThÃ©me1" />&nbsp;
          <asp:Button ID="Button_Theme2" runat="server" Text="ThÃ©me2" />&nbsp;
          <asp:Button ID="Button_Theme3" runat="server" Text="ThÃ©me3" />
        </td>
      </tr>
    </table>
  </div>
</form>

```


Avec comme résultat dans votre navigateur :



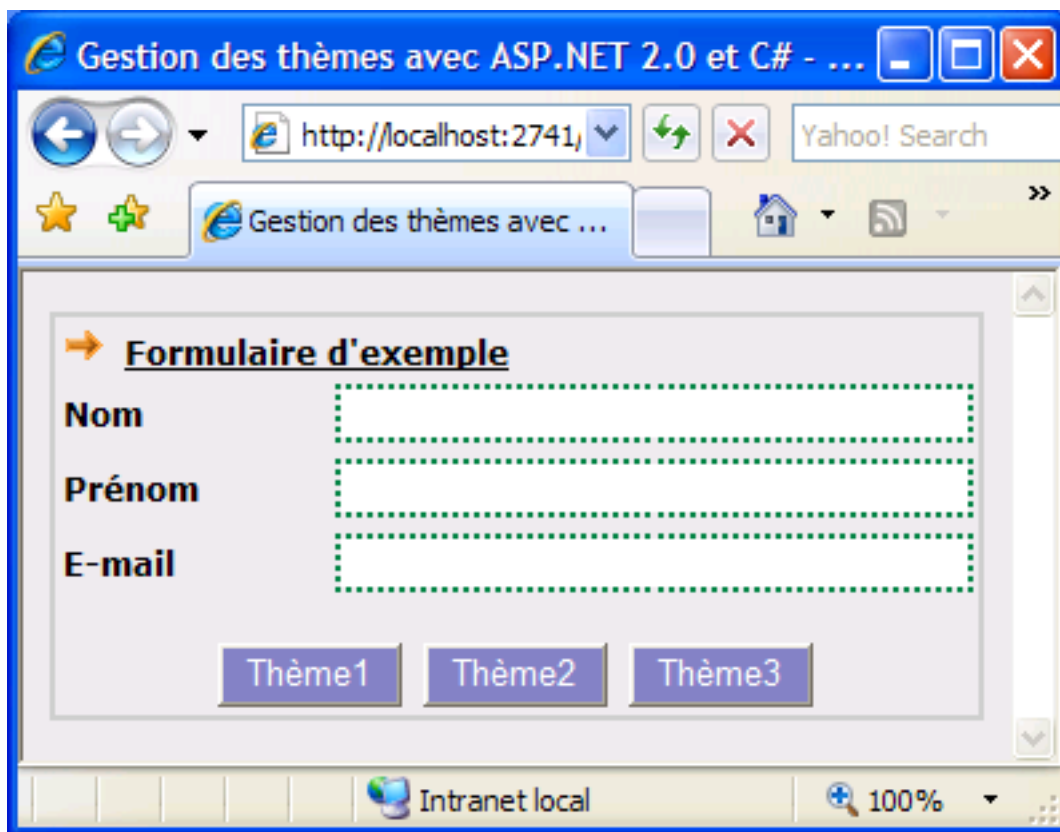
Nous allons à présent appliquer le thème **monPremierTheme**. Pour ce faire, vous devez assigner la propriété **Theme** de la balise Page de votre page Web avec le nom de votre thème :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
    Inherits="TutorialThemes._Default" Theme="MonPremierTheme" %>
```

Au passage, nous utiliserons les **SkinID** Titre et Fleche que nous appliquerons respectivement aux contrôles **Label_Titre** et **Image_Fleche** :

```
<asp:Image ID="Image_Fleche" SkinID="Fleche" runat="server" />&nbsp;  
<asp:Label ID="Label_Titre" SkinID="Titre" runat="server" Text="Formulaire d'exemple"></asp:Label>
```

Et sans rien faire d'autres, nous obtiendrons le résultat suivant :



i A la place d'utiliser la propriété **Theme**, vous pouvez utiliser la propriété **StyleSheetTheme**. Dans le cas de **Theme**, les informations du thème seront fusionnées aux informations de la page (utilisation d'une CSS, des propriétés déjà initialisées à des contrôles). Alors que dans le cas de **StyleSheetTheme**, ce sont les informations de la page qui seront prioritaires par rapport à celles du thème.

IV-E - Changement dynamique du thème

Précédemment, nous avons initialisé le thème via le designer mais il est tout à fait possible de le modifier durant l'exécution de l'application Web. Pour ce faire, il suffit tout simplement de setter la propriété **Theme** de votre objet **Page** avec le nom du thème à utiliser. A noter que ceci ne peut être fait qu'au moment de l'évènement **PreInit** ou alors avant cet évènement.

Prenons l'exemple de trois thèmes appelés **MonPremierTheme**, **MonDeuxiemeTheme** et **MonTroisiemeTheme**. Lorsque l'utilisateur cliquera sur un des trois boutons, on sélectionnera le thème correspondant :

```
Version C#
protected override void OnPreInit(EventArgs e)
{
    string objBtn1 = Page.Request["Button_Theme1"];
    string objBtn2 = Page.Request["Button_Theme2"];
    string objBtn3 = Page.Request["Button_Theme3"];

    if (objBtn1 != null)
    { Page.Theme = "MonPremierTheme"; }
    else if (objBtn2 != null)
    { Page.Theme = "MonDeuxiemeTheme"; }
    else if (objBtn3 != null)
    { Page.Theme = "MonTroisiemeTheme"; }
    else
    { Page.Theme = "MonPremierTheme"; }

    base.OnPreInit(e);
}
```

Version C#

```
}
}
```

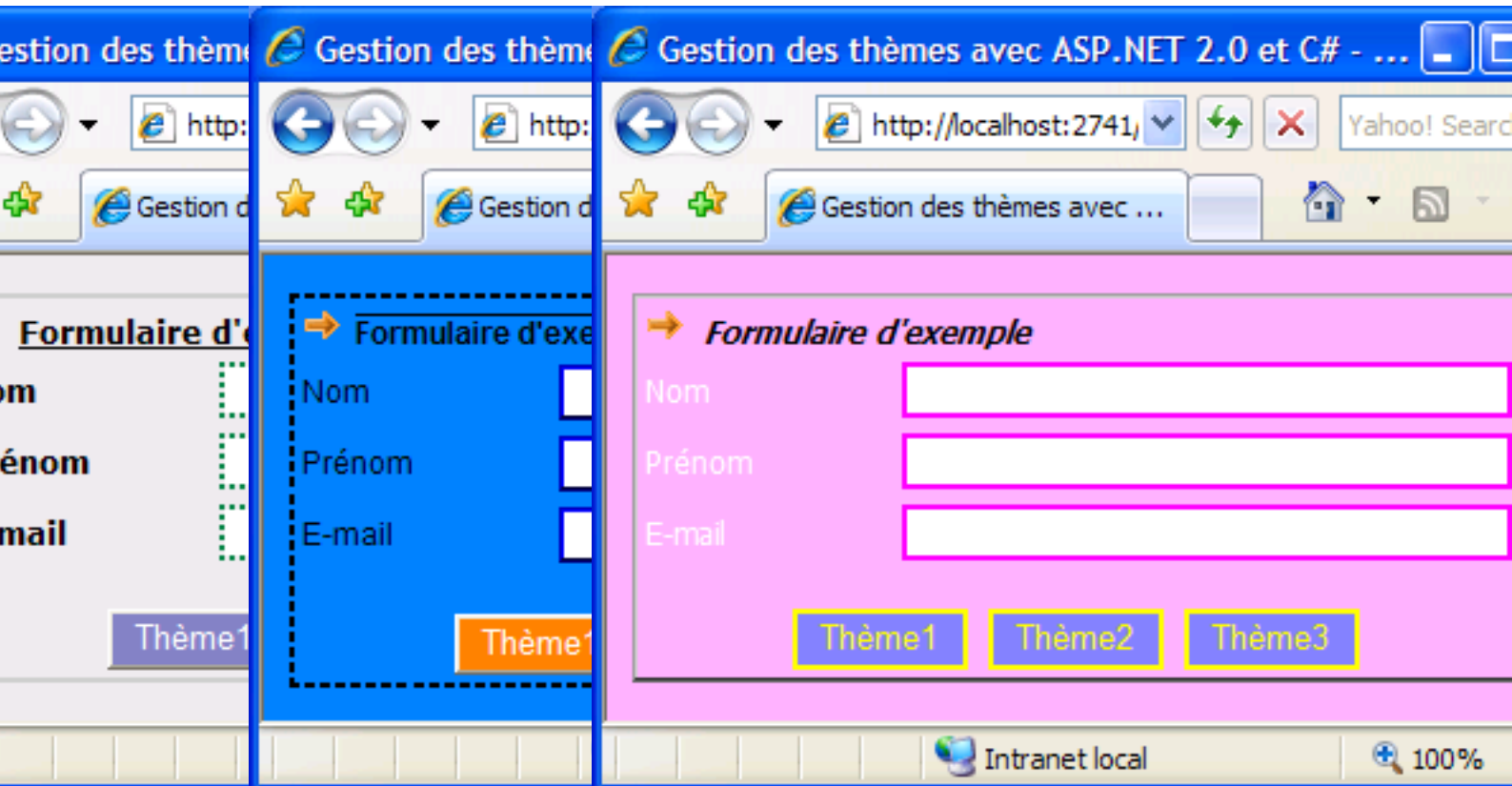
Version VB.NET

```
Protected Overrides Sub OnPreInit(ByVal e As System.EventArgs)
    Dim objBtn1 As String = Page.Request("Button_Theme1")
    Dim objBtn2 As String = Page.Request("Button_Theme2")
    Dim objBtn3 As String = Page.Request("Button_Theme3")

    If (Not objBtn1 Is Nothing) Then
        Page.Theme = "MonPremierTheme"
    ElseIf (Not objBtn2 Is Nothing) Then
        Page.Theme = "MonDeuxiemeTheme"
    ElseIf (Not objBtn3 Is Nothing) Then
        Page.Theme = "MonTroisiemeTheme"
    Else
        Page.Theme = "MonPremierTheme"
    End If

    MyBase.OnPreInit(e)
End Sub
```

Avec comme résultat en passant d'un thème à l'autre :



IV-F - Thème pour tout le site Web

Jusqu'à maintenant, nous n'avons appliqué un thème qu'à une page mais il est possible de définir un thème pour l'entièreté des pages d'un site Web. Pour cela, on passera par le fichier de config où on précisera le nom du thème à utiliser :

Fichier Web.config

```
<?xml version="1.0"?>
<configuration>
```

Fichier Web.config

```
<system.web>
    <pages theme="MonPremierTheme" />
</system.web>
</configuration>
```

Vous n'aurez donc plus besoin de préciser le thème pour chaque thème via la propriété **Theme** de la balise **page** de vos pages aspx.

V - Conclusion

Après avoir personnalisé des contrôles avec des fichiers d'apparences, habillé vos pages à l'aide de feuilles de style et enfin inséré des fichiers ressources dans vos pages, vous avez pu voir toute l'étendue qu'offre ASP.NET 2.0 avec sa gestion des thèmes. Mais surtout, vous avez pu constater avec quelle simplicité on peut créer plusieurs Template graphique en très peu de temps et bien séparé de votre code métier.

VI - Téléchargement sources

Vous pouvez télécharger la solution complète (en version C# et VB.NET) avec l'exemple qui a été construit durant ce tutoriel.

Cette solution a été réalisée avec Visual Studio 2008 beta 2. Toutefois, si certains n'arrivent pas l'utiliser sous Visual Studio 2005, vous pouvez toujours créer un nouvelle solution et copier les fichiers, l'entièreté de l'exemple est compatible Framework 2.0.

- [Source](#) **Version ZIP**
- [Source](#) **Version RAR**

VI - Liens

	<ul style="list-style-type: none">  Vue d'ensemble des thèmes et des apparences ASP.NET  Comment : définir des thèmes ASP.NET  Comment : appliquer des thèmes ASP.NET
	<ul style="list-style-type: none">  Les nouveautés de ASP.NET 2.0 avec Visual Studio 2005  ASP.Net 2.0 : les Master Pages  Introduction aux WebParts d'ASP.Net 2.0  Internationalisation d'un site web avec Visual Studio 2005

VIII - Remerciements

Merci à OD pour cette opportunité de proposer cette nouvelle version et à l'équipe de Developpez.com pour sa réactivité.