Les nouveautés de Silverlight 4

par Benjamin Roux (Retour aux articles)

Date de publication : 07/12/2009

Dernière mise à jour :

Cet article vous présentera les nouveautés de Silverlight 4.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

1 - Introduction

La beta de Silverlight 4 vient tout juste d'être annoncée à la PDC à Los Angeles durant le mois de novembre, c'est le moment pour nous de vous présenter les différentes nouveautés.

Parmi celles-ci, on retrouve beaucoup de choses réclamées par la communauté depuis longtemps, une preuve que les équipes de Silverlight nous écoutent !

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

2 - Pré-requis

Afin de développer avec Silverlight 4, il vous faut

- **Visual Studio 2010 Beta 2**
- **Silverlight Tools for Visual Studio 2010**

On peut également s'amuser avec d'autres outils

- **Expression Blend for .NET 4 Preview**
- **Silverlight Toolkit**
- **Silverlight Media Framework**

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

3 - L'impression

Première des nouveautés, et il me semble la plus demandée, est le support de l'impression.

Trois petites lignes de code suffisent pour l'implémenter

```
PrintDocument document = new PrintDocument();

// on définit la partie à afficher via la propriété PageVisual
document.PrintPage += (o, e2) => e2.PageVisual = this;

document.Print();
```

Comme vous pouvez le voir, on peut choisir ce que l'on veut imprimer : toute notre application ou seulement une partie (une image, un canvas et ses enfants).

Il est également possible de définir des actions à effectuer avant et après l'impression via deux évènements : StartPrint et EndPrint.

Vraiment pratique.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

4 - Clic droit

Une autre demande qui revenait souvent, la surcharge du menu contextuel.

Les équipes de Silverlight ont répondu à nos appels via un nouvel évènement permettant de capter le clic droit.

Prenons ce Xaml simple

```
<Grid x:Name="LayoutRoot" Background="White">
        <Image Source="Images/
moi.jpg" VerticalAlignment="Center" HorizontalAlignment="Center" Width="100" MouseRightButtonDown="Image_MouseRig
</Grid>
```

Pour afficher un menu il suffit donc d'afficher par exemple une ListBox à l'endroit où on a cliqué.

```
public partial class MainPage : UserControl
{
    public MainPage()
    {
        InitializeComponent();
    }

    private void Image_MouseRightButtonDown(object sender, MouseButtonEventArgs e)
    {
        /* A modifier */
        ListBox menu = new ListBox();
        menu.Items.Add("Save as");
        menu.Items.Add("Print");

        PerformPlacement(menu, e.GetPosition(null));

        e.Handled = true;
    }

    #region Popup

    internal Popup Popup { get; set; }

    private void PerformPlacement(FrameworkElement content, Point point)
    {
        PerformPlacement(content, point.X, point.Y);
    }

    private void PerformPlacement(FrameworkElement content, double x, double y)
    {
        Canvas elementOutside = new Canvas();
        Canvas childCanvas = new Canvas();

        elementOutside.Background = new SolidColorBrush(Colors.Transparent);

        if (Popup != null)
        {
            Popup.IsOpen = false;
            if (Popup.Child is Canvas) ((Canvas)Popup.Child).Children.Clear();
        }
        Popup = new Popup();

        Popup.Child = childCanvas;

        elementOutside.MouseLeftButtonDown += new MouseButtonEventHandler((o, e) =>
 Popup.IsOpen = false);
        elementOutside.Width = Application.Current.Host.Content.ActualWidth;
        elementOutside.Height = Application.Current.Host.Content.ActualHeight;

        childCanvas.Children.Add(elementOutside);
        childCanvas.Children.Add(content);

        Canvas.SetLeft(content, x);
```


Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

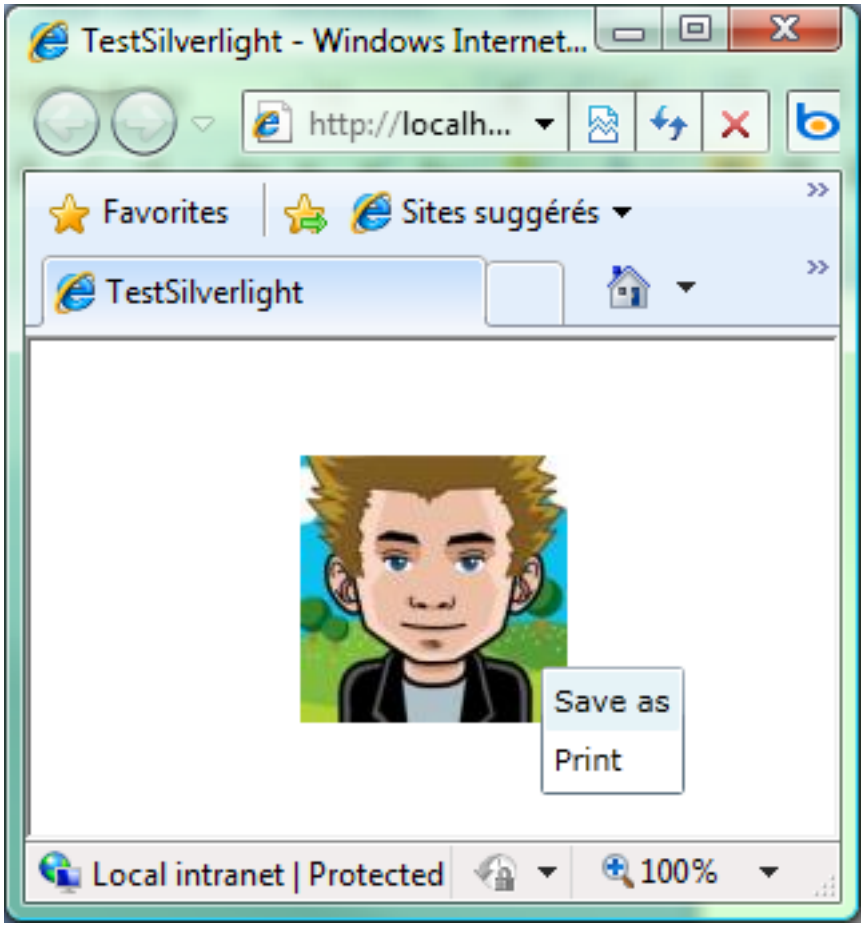
```
            Canvas.SetTop(content, y);

            Popup.IsOpen = true;
        }

        #endregion
    }
```

Le code pour placer la ListBox au bon endroit (tout ce qui se trouve dans la #region) vient de mon ancien code pour afficher un Menu Contextuel que vous pouvez trouver sur mon blog.

Le résultat :



On peut bien entendu ajouter des évènements pour les clics sur le menu mais ce n'est pas le sujet.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

5 - La molette de la souris

Une nouvelle nouveauté est le support de la molette de la souris pour tous les contrôles.

Exemple simple pour agrandir une image avec la molette

```
<Grid x:Name="LayoutRoot" Background="White">
    <Image x:Name="img" Source="Images/moi.jpg"
           Width="100" Height="100"
           MouseWheel="Image_MouseWheel">
        <Image.RenderTransform>
            <ScaleTransform ScaleX="1" ScaleY="1" />
        </Image.RenderTransform>
    </Image>
</Grid>
```

```
private void Image_MouseWheel(object sender, MouseWheelEventArgs e)
{
    ScaleTransform transform = ((ScaleTransform)img.RenderTransform);

    double scale = 0;

    if (e.Delta > 0) scale = 0.1;
    else scale = -0.1;

    if (transform.ScaleX > 0.5 || scale > 0)
    {
        transform.ScaleX += scale;
        transform.ScaleY += scale;
    }
}
```

e.Delta correspond au « montant » de la molette.

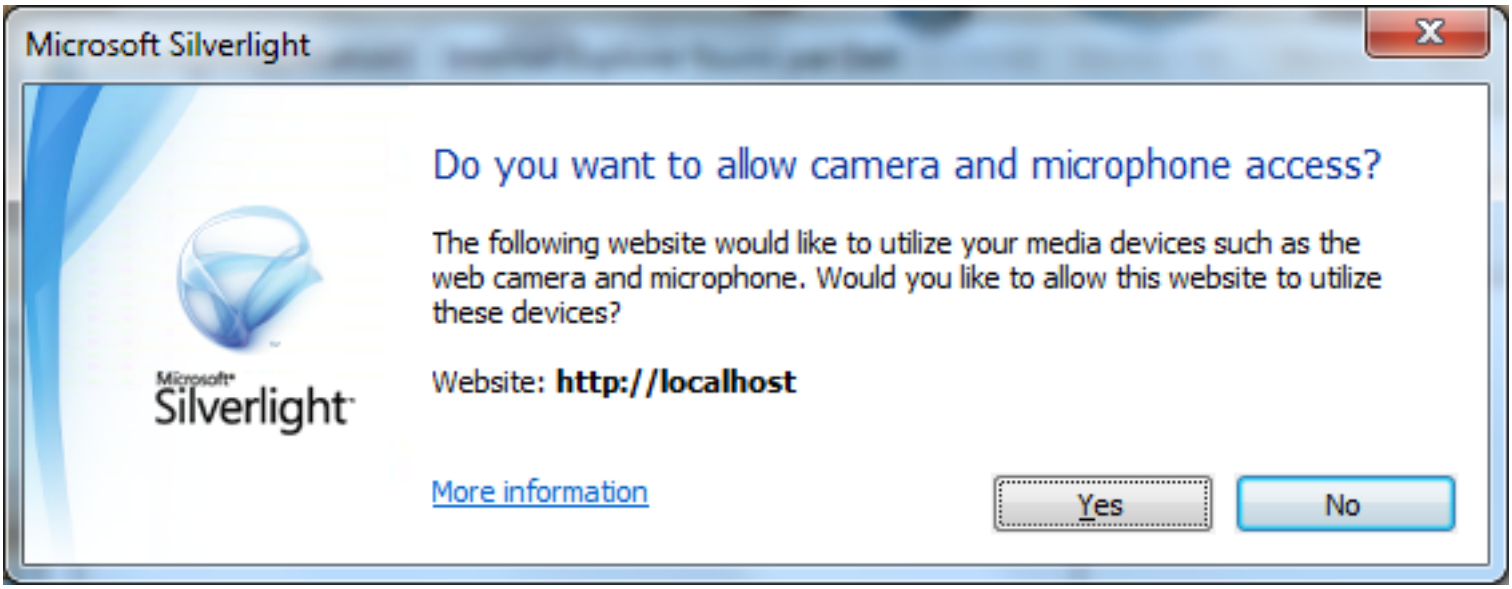
Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

6 - Webcam et microphone

Cette partie a été écrite par **Jérôme Lambert**.

Voici une nouveauté attendue depuis longtemps, le support de la Webcam et du microphone. L'API Silverlight permet à présent de détecter et d'utiliser les périphériques audio et vidéo connectés à l'ordinateur.

De nouvelles classes sont à présent disponibles au sein de l'espace de noms « System.Windows.Media » pour intéragir avec les périphériques audio et vidéo. La première classe dont vous aurez besoin est « CaptureDeviceConfiguration » qui permet de demander à l'utilisateur l'autorisation d'utiliser les périphériques de Webcam et microphone grâce à la méthode statique « RequestDeviceAccess ». L'appel à cette méthode lancera une boite de dialogue de confirmation auprès de l'utilisateur.



Pour éviter de devoir redemander l'accès aux périphériques, il est possible d'utiliser la propriété « AllowedDeviceAccess » qui renvoie vrai dans le cas où l'utilisateur a déjà donné son accord pour l'accès à ses périphériques au cours de la session.

```
if (CaptureDeviceConfiguration.AllowedDeviceAccess
|| CaptureDeviceConfiguration.RequestDeviceAccess())
{

}
```

Une fois que l'utilisateur a confirmé l'accès à ses périphériques, on va pouvoir retrouver les périphériques disponibles. Pour cela, il est possible de récupérer l'accès au périphérique vidéo par défaut grâce à la méthode statique « GetDefaultVideoCaptureDevice ». Il est aussi possible de récupérer le périphérique audio par défaut via la méthode statique « GetDefaultAudioCaptureDevice ».

```
VideoCaptureDevice videoDevice =CaptureDeviceConfiguration.GetDefaultVideoCaptureDevice();
AudioCaptureDevice audioDevice = CaptureDeviceConfiguration.GetDefaultAudioCaptureDevice();

if (videoDevice != null
&& audioDevice != null)
{
}
```

Pour initialiser la capture vidéo, on va devoir créer une instance de la classe « CaptureSource » en initialisant ses propriétés « VideoCaptureDevice » et « AudioCaptureDevice » pour spécifier les périphériques vidéo et audio à utiliser.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

```
var captureSource = new CaptureSource();
captureSource.VideoCaptureDevice = videoDevice;
captureSource.AudioCaptureDevice = audioDevice;
```

Enfin, il ne reste plus qu'à démarrer le périphérique grâce à la méthode « Start » de notre classe « CaptureSource ».

Pour l'arrêt de la capture, c'est aussi simple que le démarrage grâce à la méthode « Stop ». Vous pourrez aussi vous aider de la propriété « State » pour déterminer si la capture est bien démarrée.

```
if (_captureSource != null
&& _captureSource.State == CaptureState.Started)
{
_captureSource.Stop();
}
```

Pour finir, vous allez pouvoir afficher votre capture vidéo directement dans un élément de votre application Silverlight grâce à la classe « VideoBrush ».

```
VideoBrush videoBrush = new VideoBrush();
videoBrush.Stretch = Stretch.Fill;
videoBrush.SetSource(captureSource);
rectangle1.Fill = videoBrush;
```

Dans notre exemple, nous avons décidé d'utiliser les périphériques vidéo et audio par défaut, mais il est aussi possible de récupérer l'ensemble des périphériques vidéo et audio disponibles sur la machine de l'utilisateur. Pour ce faire, on utilisera les méthodes statiques « GetAvailableVideoCaptureDevices » et « GetAvailableAudioCaptureDevices » de la classe « CaptureDeviceConfiguration ».

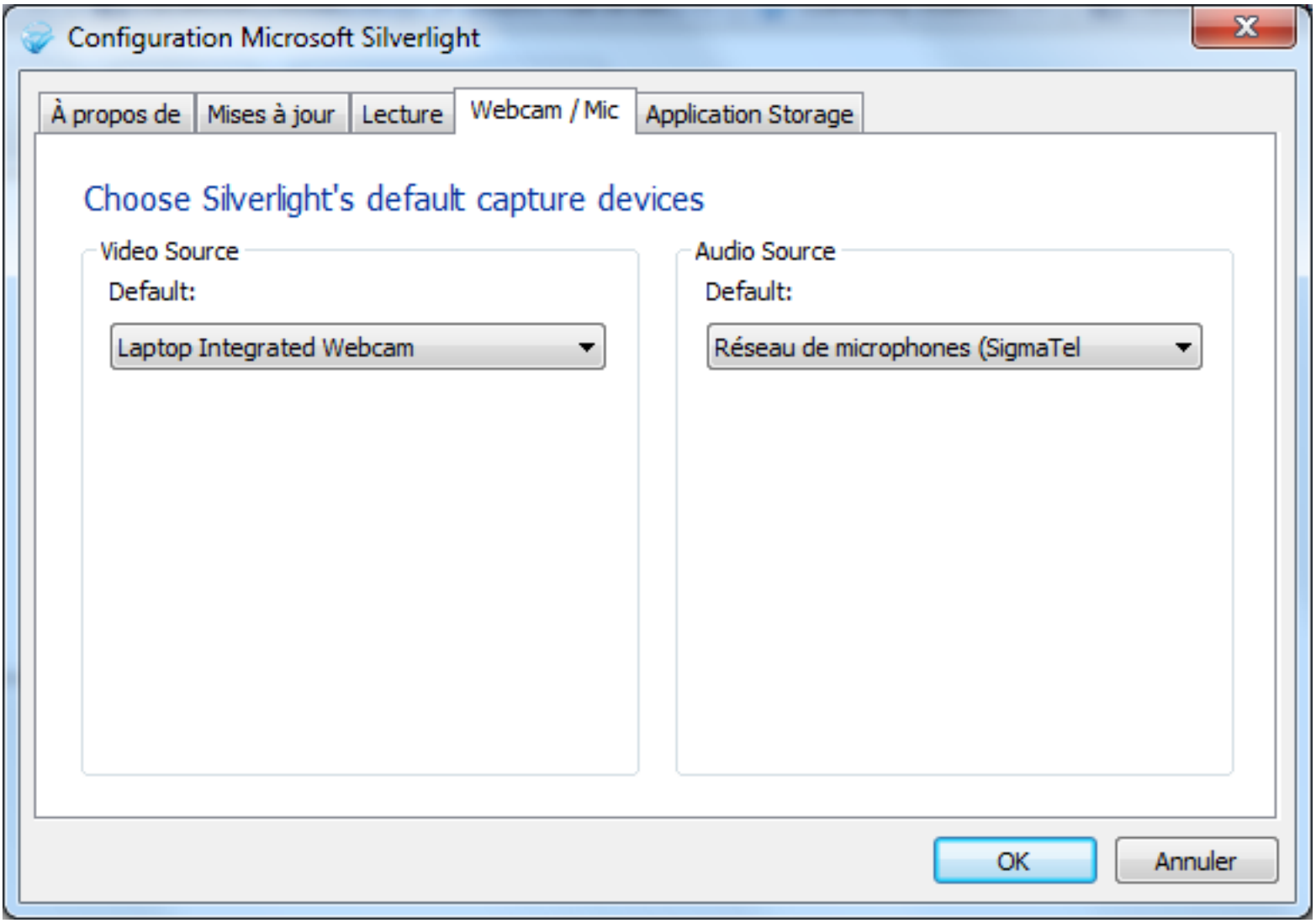
```
var videoDevices = CaptureDeviceConfiguration.GetAvailableVideoCaptureDevices();
var audioDevices = CaptureDeviceConfiguration.GetAvailableAudioCaptureDevices();

foreach (var videoDevice in videoDevices)
MessageBox.Show(videoDevice.FriendlyName);

foreach (var audioDevice in audioDevices)
MessageBox.Show(audioDevice.FriendlyName);
```

L'utilisateur a aussi la possibilité de consulter la liste de ses périphériques vidéo et audio disponibles et ainsi changer ceux par défaut. Pour cela, il suffit de se rendre dans les paramètres de l'application Silverlight en faisant un clique droit sur l'application Silverlight et en se rendant dans l'onglet « Webcam / Mic ».

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.



Le code complet :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;

namespace SilverlightApplication1
{
    public partial class MainPage : UserControl
    {
        CaptureSource _captureSource = null;

        public MainPage()
        {
            InitializeComponent();
        }

        private void buttonStart_Click(object sender, RoutedEventArgs e)
        {
            if (_captureSource != null
```


Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

```
            && _captureSource.State == CaptureState.Started)
            {
                _captureSource.Stop();
            }

            if (CaptureDeviceConfiguration.AllowedDeviceAccess
            || CaptureDeviceConfiguration.RequestDeviceAccess())
            {
                VideoCaptureDevice videoDevice =
CaptureDeviceConfiguration.GetDefaultVideoCaptureDevice();
                AudioCaptureDevice audioDevice =
CaptureDeviceConfiguration.GetDefaultAudioCaptureDevice();

                if (videoDevice != null
                && audioDevice != null)
                {
                    _captureSource = new CaptureSource();
                    _captureSource.VideoCaptureDevice = videoDevice;
                    _captureSource.AudioCaptureDevice = audioDevice;

                    _captureSource.Start();

                    VideoBrush videoBrush = new VideoBrush();
                    videoBrush.Stretch = Stretch.Fill;
                    videoBrush.SetSource(_captureSource);
                    rectangle1.Fill = videoBrush;
                }
            }
            else
            {

MessageBox.Show("AccÃ¨s refusÃ© par l'utilisateur ou aucun pÃ©riphÃ©rique vidÃ©o ou audio n'est disponible.");
            }
        }

        private void buttonStop_Click(object sender, RoutedEventArgs e)
        {
            if (_captureSource != null
            && _captureSource.State == CaptureState.Started)
            {
                _captureSource.Stop();
            }
        }

        private void buttonDevices_Click(object sender, RoutedEventArgs e)
        {
            var videoDevices = CaptureDeviceConfiguration.GetAvailableVideoCaptureDevices();
            var audioDevices = CaptureDeviceConfiguration.GetAvailableAudioCaptureDevices();

            foreach (var videoDevice in videoDevices)
                MessageBox.Show(videoDevice.FriendlyName);

            foreach (var audioDevice in audioDevices)
                MessageBox.Show(audioDevice.FriendlyName);
        }

    }
}
```

Vous pouvez également tester une démo de l'application ici : **http://dotnetdemos.developpez.com/jlambert/silverlight-4-webcam-and-mic/**

6-1 - Prendre une photo à partir du flux vidéo

Il est également possible de prendre une photo à partir du flux vidéo. Pour cela la classe *CaptureSource* met à notre disposition une méthode *AsyncCaptureImage*.

```
Source.CaptureImageCompleted += (o, e) => { var bmp = e.Result; ... };
```


Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

```
Source.CaptureImageAsync();
```


Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

7 - RichtextArea

Un nouveau contrôle fait également son apparition, il s'agit d'un éditeur de texte riche.

Exemple basique :

```
<Grid x:Name="LayoutRoot" Background="White">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto" />
        <RowDefinition Height="*" />
    </Grid.RowDefinitions>
    <StackPanel Grid.Row="0" Orientation="Horizontal">
        <Button x:Name="bold" Content="B" FontWeight="Bold" Click="bold_Click" />
        <Button x:Name="underline" Click="underline_Click">
            <Button.Content>
                <TextBlock Text="U" TextDecorations="Underline" />
            </Button.Content>
        </Button>
        <Button x:Name="italic" Click="italic_Click" >
            <Button.Content>
                <TextBlock Text="U" FontStyle="Italic" />
            </Button.Content>
        </Button>
        <Button x:Name="addImage" Content="Add image" Click="addImage_Click" />
    </StackPanel>
    <RichTextArea x:Name="richText" Grid.Row="1" />
</Grid>
```

On définit notre interface composée de 4 boutons (gras, souligné, italique et ajouter une image) et de notre éditeur.

Maintenant il nous reste à écrire un peu de code

```
private void bold_Click(object sender, RoutedEventArgs e)
{
    richText.Selection.SetPropertyValue(TextElement.FontWeightProperty, FontWeights.Bold);
}

private void underline_Click(object sender, RoutedEventArgs e)
{
    richText.Selection.SetPropertyValue(TextElement.TextDecorationsProperty,
 TextDecorations.Underline);
}

private void italic_Click(object sender, RoutedEventArgs e)
{
    richText.Selection.SetPropertyValue(TextElement.FontStyleProperty, FontStyles.Italic);
}

private void addImage_Click(object sender, RoutedEventArgs e)
{
    InlineUIContainer container = new InlineUIContainer();
    container.Child = new Image { Source = new BitmapImage(new Uri("Images/moi.jpg",
 UriKind.Relative)) };

    richText.Selection.Insert(container);
}
```

Comme vous pouvez le voir le RichText possède une propriété *Selection* représentant le texte sélectionné et auquel nous allons appliquer nos « styles ».

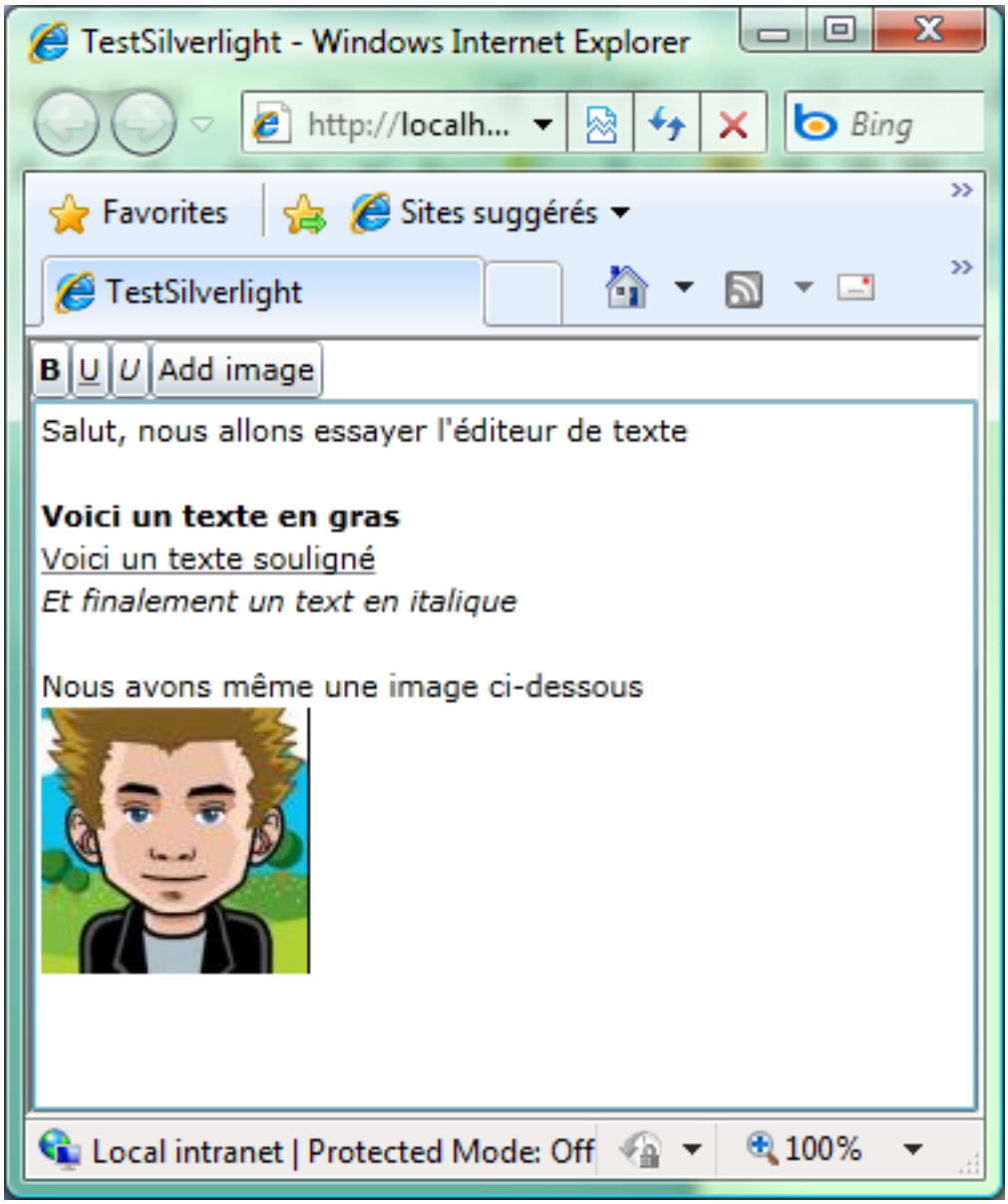
La chose merveilleuse est que nous pouvons également insérer n'importe quel type de contrôle via l'objet *InlineUiContainer*. Ici j'ai mis une image mais on pourrait mettre un DataGrid, un bouton ou même un TreeView !

Un RichTextBox est un enchainement de **Paragraph**. Chaque *Paragraph* peut contenir d'autres éléments tels que :

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

- Inline (peut être un InlineUiContainer, un LineBreak, un Run ou un Span)
- InlineUiContainer (Bouton, DataGrid.)
- Run (possède une propriété Text)
- Span (qui est une collection de Inline)
- Bold (pour mettre un texte en gras) <Bold>Texte</Bold>
- Hyperlink (un lien hypertexte)
- Italic (pour mettre un texte en italique)
- Underline (pour souligner un texte)

Le code précédent peut donner le résultat suivant



Ce contrôle peut devenir tout à fait puissant !

A mon avis il ne faudra pas attendre beaucoup avant que des personnes ne distribuent un joli éditeur de texte en Open-Source.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

8 - Support du presse-papier

Dans les versions précédentes de Silverlight il fallait passer par un appel en Javascript qui fonctionnait le plus souvent uniquement sous Internet Explorer.

Cette fonctionnalité fait donc son apparition pour le bonheur des développeurs.

Exemple simple en rajoutant 2 boutons Copy et Paste à notre code précédent (RichTextArea).

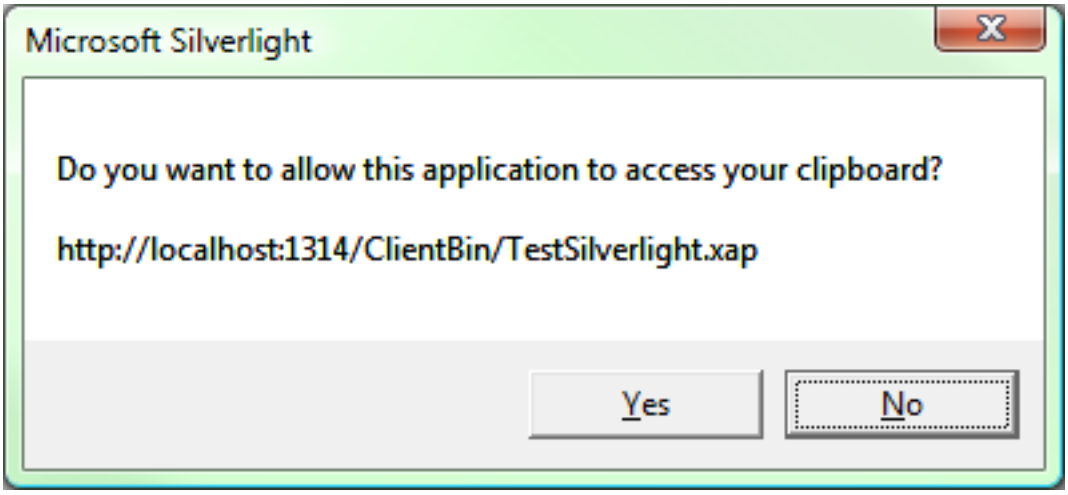
```
private void copy_Click(object sender, RoutedEventArgs e)
{
    Clipboard.SetText(richText.Selection.Text);
}

private void paste_Click(object sender, RoutedEventArgs e)
{
    Paragraph paragraph = new Paragraph();
    paragraph.Inlines.Add(new Run { Text = Clipboard.GetText() });

    richText.Blocks.Add(paragraph);
}
```

Une méthode permet également de savoir si du texte est présent dans le presse-papier.

Comme pour plusieurs fonctionnalités dans Silverlight, une copie dans le presse-papier ne peut se faire que par une action de l'utilisateur (clic sur un bouton, appui sur une touche du clavier.) et provoque l'affichage de ce message



Ce message n'apparait qu'une seule fois par session.

Le Clipboard Silverlight est commun à toutes les applications Silverlight mais également à la machine de l'utilisateur. On peut donc copier/coller un texte de/vers Word par exemple (simplement le texte pas les styles).

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

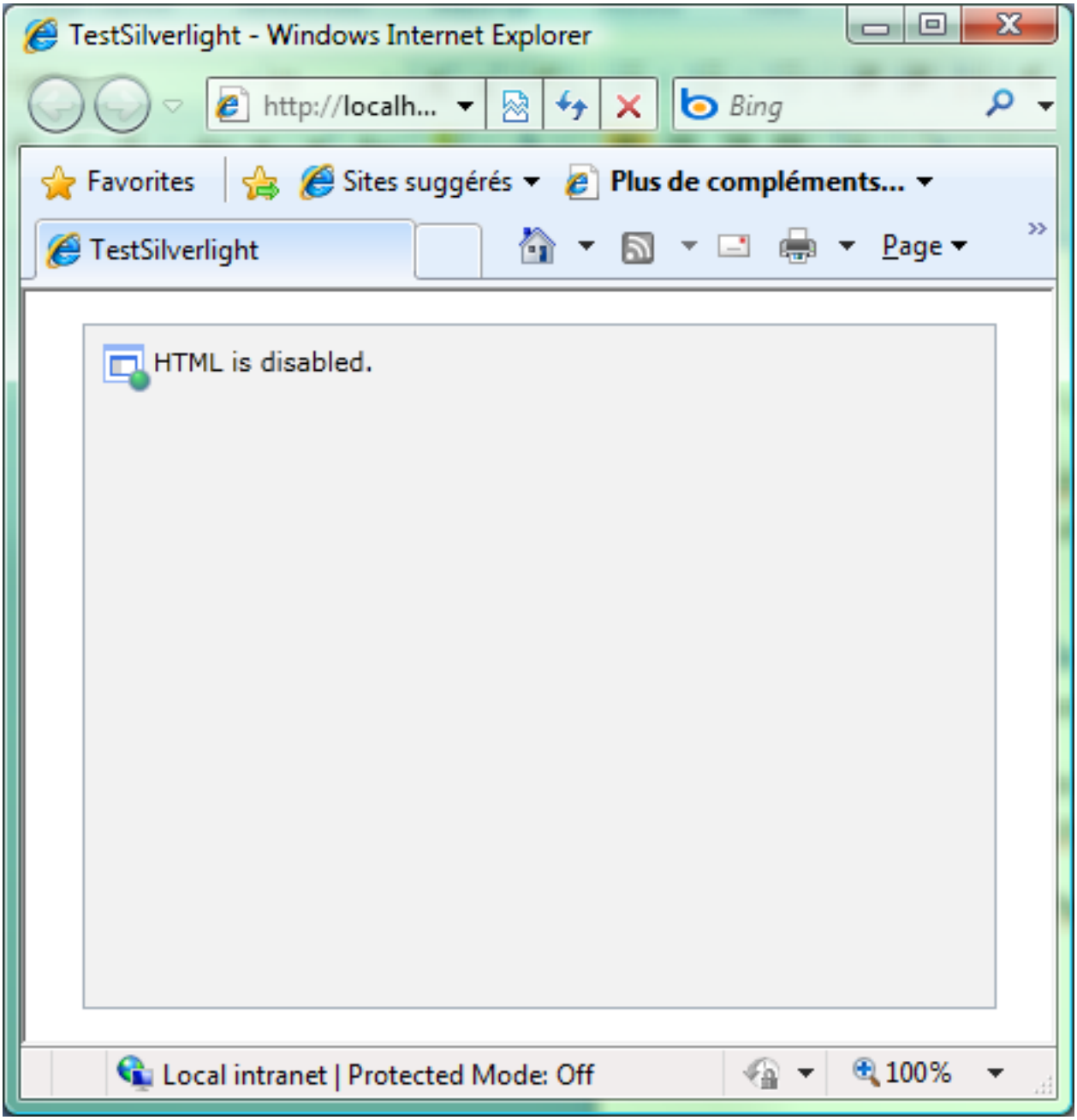
9 - Hébergement de HTML

Silverlight 4 permet également d'afficher du HTML au sein d'une application.

Dans les précédentes versions, c'était faisable en affichant une div HTML par-dessus l'application Silverlight là où on voulait qu'elle s'affiche.

*A noter que l'hébergement de HTML ne fonctionne que dans une application **Out Of Browser**.*

Dans le cas contraire vous obtiendrez ce résultat



Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

9-1 - WebBrowser

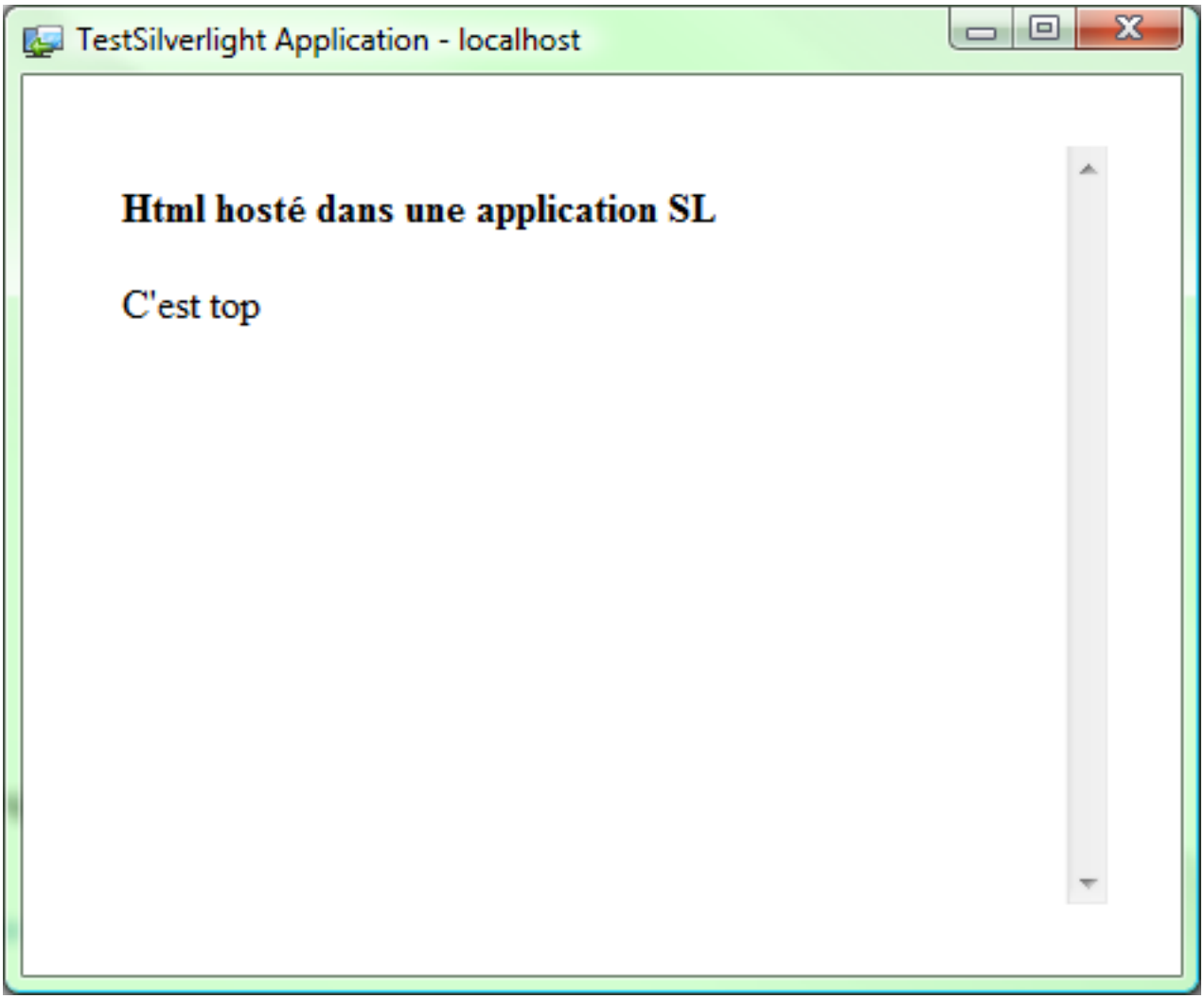
Un contrôle fait donc son apparition, le WebBrowser.

Deux méthodes utiles : *Navigate(url)* et *NavigateToString(html)*

```
<WebBrowser x:Name="browser" Width="400" Height="300" />
```

On peut donc écrire en code behind.

```
browser.NavigateToString("<b>Html hosté dans une application SL</b><br/><p>C'est top<p>");
```



Ou bien

```
browser.Navigate(new Uri("http://localhost/index.html", UriKind.Absolute));
```

Pour des raisons de sécurité le contrôle WebBrowser ne peut afficher que du contenu se trouvant dans le même domaine que l'application Silverlight, ce qui inclut le même domaine, le même protocole et le même port.

Pour afficher du contenu d'un autre domaine on peut passer par une *iframe* hostée dans notre page se trouvant sur notre domaine.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

Le WebBrowser est capable d'afficher n'importe quel type de contenu même des applications Flash (si le plugin est installé sur la machine bien entendu).

```
browser.NavigateToString("<iframe src=\"http://www.youtube.com/watch?v=FzRH3iTQPrk\" width=\"600px
\" height=\"600px\" />");
```


Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

9-2 - HtmlBrush

Il est également possible d'appliquer un brush de type Html via le *HtmlBrush*.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

Exemple :

```
<StackPanel x:Name="LayoutRoot" Background="White">

  <WebBrowser x:Name="browser" Source="youtube.html" Width="300" Height="300" Visibility="Collapsed" />
    <Rectangle x:Name="rect" Width="300" Height="300">
        <Rectangle.Fill>
            <HtmlBrush SourceName="browser" />
        </Rectangle.Fill>
    </Rectangle>
</StackPanel>
```

L'astuce est qu'il faut rafraichir notre *HtmlBrush* via la méthode *Redraw*.

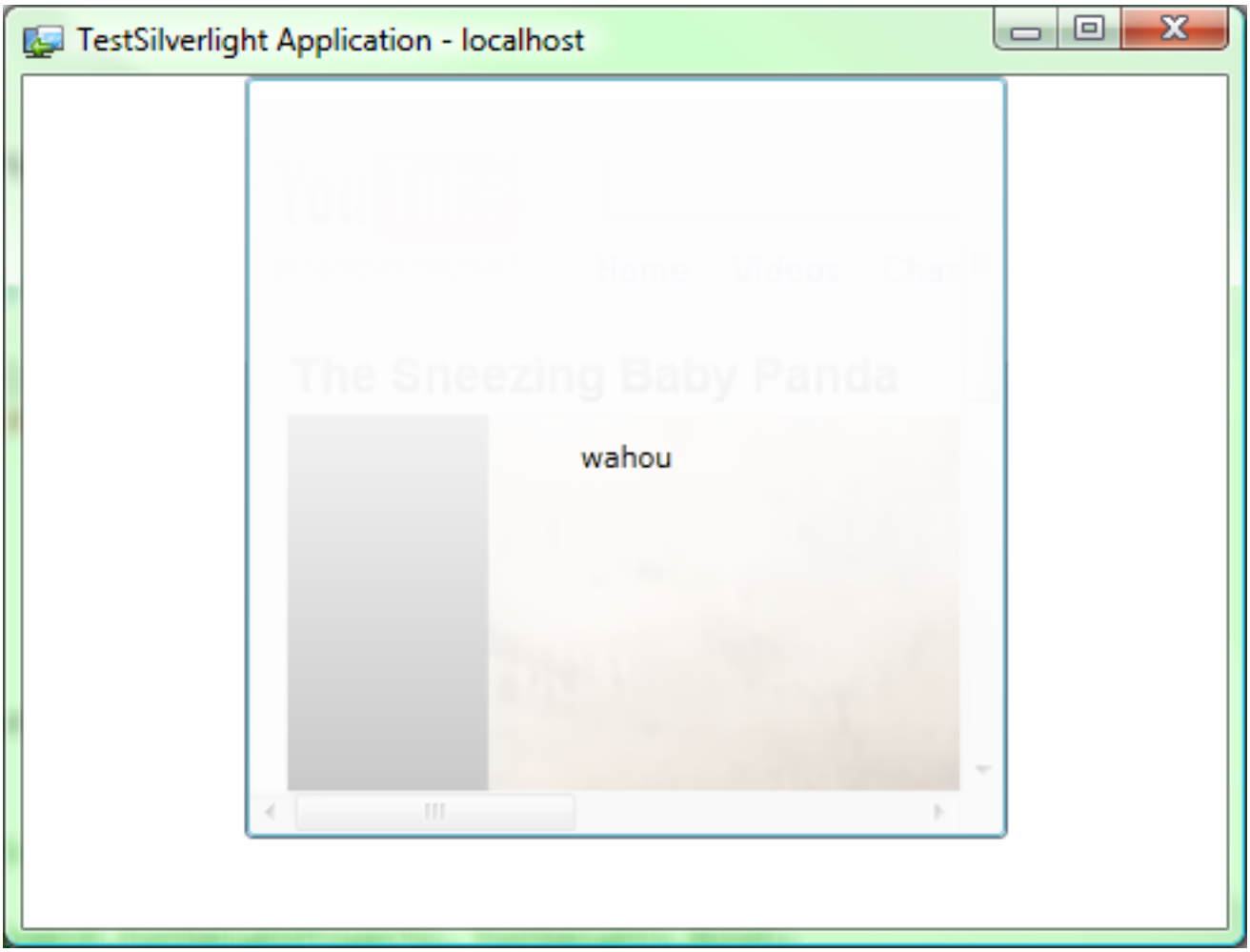
```
void MainPage_Loaded(object sender, RoutedEventArgs e)
{
    DispatcherTimer timer = new DispatcherTimer();
    timer.Interval = TimeSpan.FromMilliseconds(100);
    timer.Tick += new EventHandler(timer_Tick);
    timer.Start();
}

void timer_Tick(object sender, EventArgs e)
{
    if (rect.Fill != null) ((HtmlBrush)rect.Fill).Redraw();
}
```

On peut alors obtenir un rectangle dont le remplissage provient d'un *WebBrowser*.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

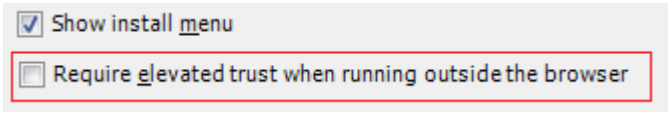
Exemple avec un bouton :



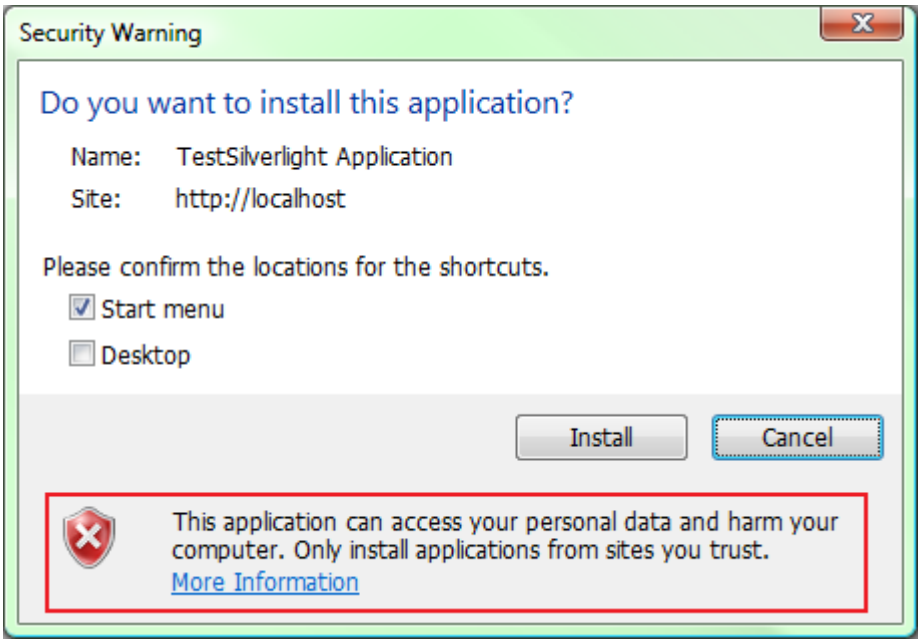
Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

10 - Trusted Application

Un nouveau mode d'application ***Out Of Browser*** fait également son apparition, il s'agit d'un mode dit ***« Trusted »*** (« de confiance » en français).



A l'installation la fenêtre change quelque peu



Dans ce mode plusieurs privilèges sont accordés.

Une propriété permet de savoir si l'application tourne en mode privilégiée.

```
App.Current.HasElevatedPermissions
```

10-1 - Accès au système de fichier local

Une application ***Out Of Browser*** en mode ***Trusted*** permet l'accès au système de fichiers local.

```
if (App.Current.HasElevatedPermissions)
{
    files.ItemsSource =
 Directory.EnumerateDirectories(Environment.GetFolderPath(Environment.SpecialFolder.MyMusic));
}
```

10-2 - Interopérabilité COM

Une application ***Trusted*** peut également interagir avec des composants COM comme par exemple Word ou Excel !

Il suffit pour cela de passer par l'API ***AutomationFactory***.

Exemple avec de l'automation Word.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.
http://broux.developpez.com/articles/csharp/silverlight-4

```
if (App.Current.HasElevatedPermissions)
{
    dynamic wordApp = AutomationFactory.CreateObject("Word.Application");
    wordApp.Visible = true;

    dynamic wordDoc = wordApp.Documents.Add();

    /* automation Word */
}
```

Ce code ouvre Word et crée un document. On pourrait par la suite bien entendu rajouter du texte dans le document.

L'automation peut également permettre d'accéder aux périphériques de la machine, à Windows Media Player et à tout plein d'autres fonctionnalités !

10-3 - Support total du clavier en plein-écran

Une application ***Trusted*** offre également le support complet du clavier en plein-écran.

Pour rappel, lorsqu'une application Silverlight « normale » se trouve en plein écran, seules les touches directionnelles et la barre d'espace sont disponibles.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

11 - API de notification

Une application ***Out Of Browser*** peut maintenant afficher des notifications style Live Messenger lorsqu'un contact se connecte.

```
if (App.Current.HasElevatedPermissions)
{
    /* Notification API */
    NotificationWindow window = new NotificationWindow();
    window.Height = 100;
    window.Width = 200;

    /* le contrôle à afficher */
    window.Content = new CustomNotification { Width = 200, Height = 100 };

    /* affichage de 2 secondes (2000 ms) */
    window.Show(2000);
}
```

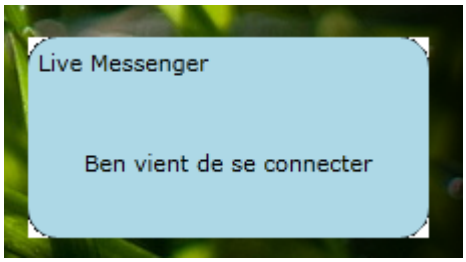
Avec CustomNotification, un simple UserControl.

```
<Grid x:Name="LayoutRoot" Background="Transparent">
    <Border CornerRadius="15" Background="LightBlue">
        <Grid>
            <Grid.RowDefinitions>
                <RowDefinition Height="25" />
                <RowDefinition />
            </Grid.RowDefinitions>

  <TextBlock Text="Live Messenger" Grid.Row="0" VerticalAlignment="Center" Margin="5,0,0,0" />

  <TextBlock Text="Ben vient de se connecter" Grid.Row="1" VerticalAlignment="Center" HorizontalAlignment="Center
        </Grid>
    </Border>
</Grid>
```

Le résultat :

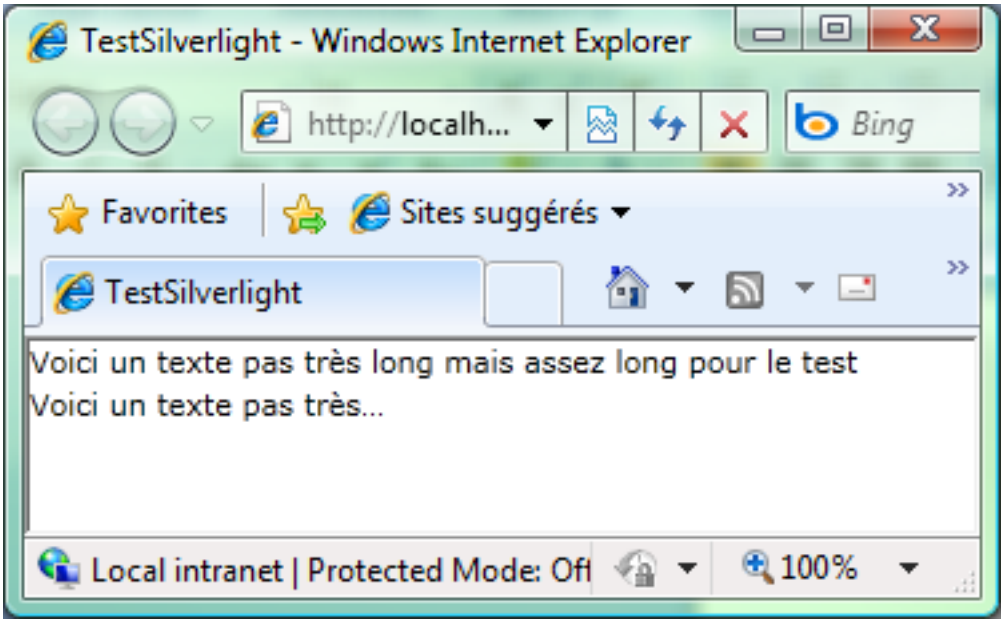


Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

12 - TextTrimming

Silverlight 4 apporte également quelques améliorations par rapport à l'affichage de texte.

```
<TextBlock Text="Voici un texte pas très long mais assez long pour le test" />
<TextBlock Text="Voici un texte pas très long mais assez long pour le test" TextTrimming="WordEllipsis" Width="15
```



La propriété *TextTrimming* permet de couper un texte trop long par « . ».

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

13 - Drag&Drop à partir du bureau

Silverlight 4 permet maintenant de glisser-déposer un élément du bureau vers notre application Silverlight.

```
void MainPage_Loaded(object sender, RoutedEventArgs e)
{
    textBox.AllowDrop = true;

    textBox.Drop += new DragEventHandler(textBox_Drop);
}

void textBox_Drop(object sender, DragEventArgs e)
{
    /* retourne un tableau de FileInfo (multi drag&drop), je recupère seulement le premier */
    FileInfo info = (e.Data.GetData(DataFormats.FileDrop) as FileInfo[])[0];

    textBox.Text = info.OpenText().ReadToEnd();
}
```

Ces quelques lignes de code permettent d'afficher le contenu d'un fichier du bureau dans notre TextBox Silverlight.

D'autres évènements peuvent également servir comme *DragEnter*, *DragLeave* et *DragOver*.

Très utile pour un Upload de fichier !

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

14 - Authentification Réseau

Il arrive parfois que l'on veuille accéder à une ressource sécurisée. Maintenant avec SL4 il est possible de passer des credentials (identifiants) lors d'une requête avec la pile réseau *ClientHttp*.

```
/* on spécifie que les requêtes sur http passe par la pile réseau ClientHttp */
WebRequest.RegisterPrefix("http://", System.Net.Browser.WebRequestCreator.ClientHttp);

WebClient client = new WebClient();
client.Credentials = new NetworkCredential("login", "password", "domain");
client.UseDefaultCredentials = false; // nous voulons utiliser nos propres credentials
client.DownloadStringCompleted += (o, e2) => ...
client.DownloadStringAsync(new Uri("http://mondomaine.com/maressourceprotegee"));
```


Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

15 - Amélioration du Data Binding

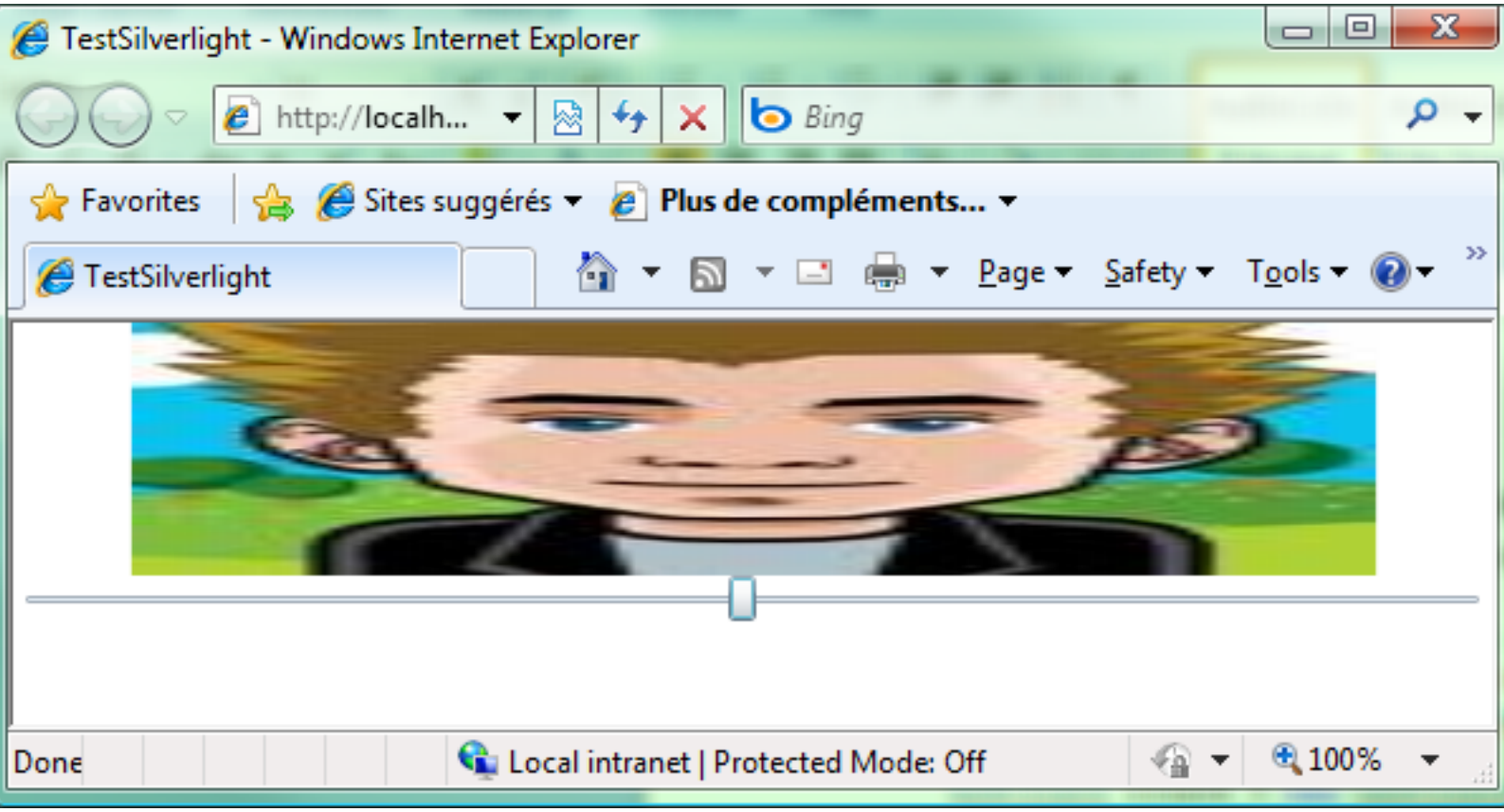
15-1 - Binding sur DependencyObject

Il est désormais possible de réaliser un *Binding* sur un *DependencyObject*. Auparavant seul le binding sur des *FrameworkElement* était supporté.

On peut donc maintenant binder des propriétés comme l'*Angle* d'un *RotateTransform* ou le *ScaleX* d'un *ScaleTransform*.

```
<Image Source="Images/moi.jpg" Width="100" Height="100">
    <Image.RenderTransform>

  <ScaleTransform ScaleX="{Binding Path=Value, ElementName=slider}" ScaleY="1" CenterX="50" CenterY="50" />
    </Image.RenderTransform>
</Image>
<Slider x:Name="slider" Maximum="10" />
```



15-2 - StringFormat, TargetNullValue, FallbackValue

D'autres propriétés ont également été ajoutées pour améliorer le Binding.

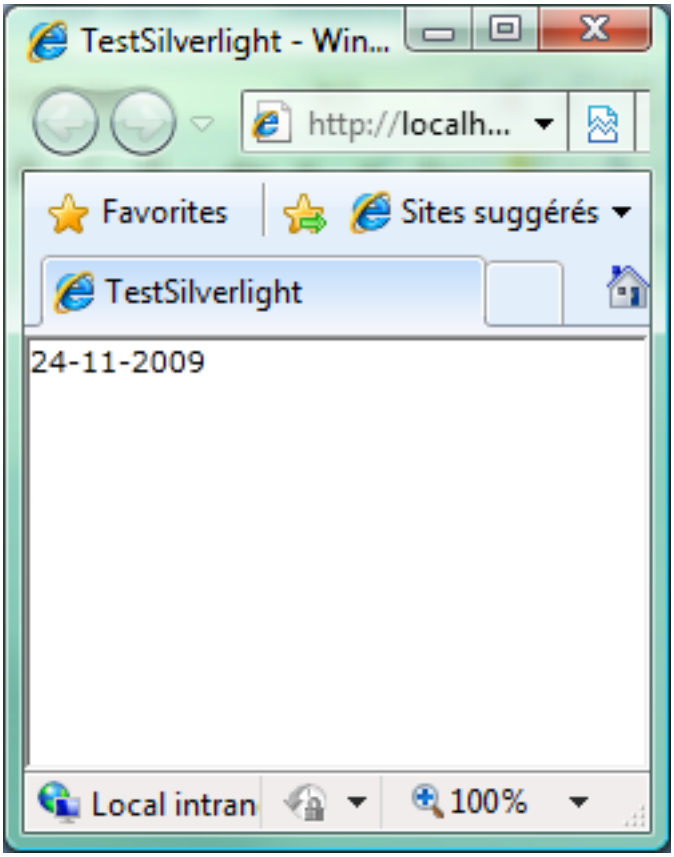
15-2-1 - StringFormat

Permet de formater un objet. Auparavant il fallait passer par un Converter.

```
<TextBlock Text="{Binding StringFormat='dd-MM-yyyy'}" />
```


Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.
http://broux.developpez.com/articles/csharp/silverlight-4

```
this.DataContext = DateTime.Now;
```



15-2-2 - TargetNullValue

Affiche une valeur lorsque la valeur utilisée pour le *Binding* vaut *null*.

```
<TextBlock Text="{Binding TargetNullValue=Oups}" />
```

15-2-3 - FallbackValue

Affiche une valeur lorsque l'opération de *Binding* est incapable de retourner une valeur (mauvais Path par exemple).

```
<TextBlock Text="{Binding FallbackValue=Oups}" />
```

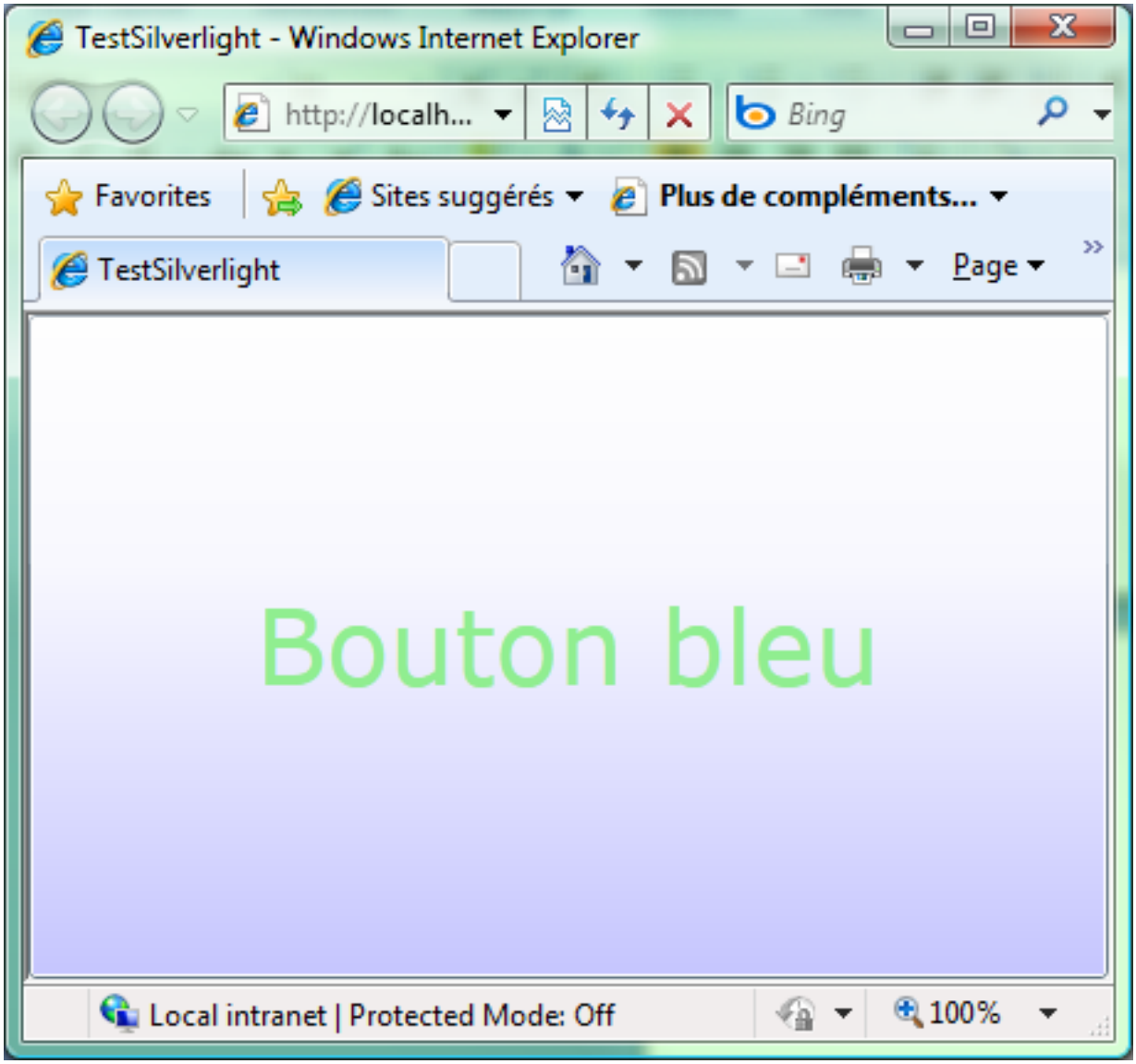

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

16 - Thèmes implicites

Silverlight 4 intègre le système des thèmes implicites, auparavant présent dans le Silverlight Toolkit.

```
<Grid.Resources>
    <Style TargetType="Button">
        <Setter Property="Background" Value="Blue" />
        <Setter Property="Foreground" Value="LightGreen" />
        <Setter Property="FontSize" Value="42" />
    </Style>
</Grid.Resources>
```

Maintenant tous les boutons de notre application auront le style définit plus haut. Plus besoin de spécifier de clé.



Il se peut bien entendu que l'on ne veuille pas appliquer le thème à un bouton spécifique, pour cela rien de plus simple :

```
<Button Content="Bouton normal" Style="{x:Null}" />
```


Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

17 - Managed Extensibility Framework (MEF)

Silverlight 4 supporte désormais MEF qui apporte un système de plug-ins.

Pour plus d'informations : **http://www.codeplex.com/MEF**

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

18 - ICommand et Command

Silverlight 4 apporte également le support de l'interface *ICommand* et de la propriété *Command*, utiles pour des patterns style MVVM (Model-View-ViewModel).

Les propriétés *Command* et *CommandParameter* sont disponibles sur le contrôle *ButtonBase* et *Hyperlink*.

Un travail en vue pour les développeurs de framework MVVM !

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

19 - Divers

Des évènements ont également rajoutés sur le contrôle *ItemsControl* : *BeforeLoaded*, *Loaded*, *Unloaded*, afin, par exemple, de pouvoir rajouter des animations.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

20 - Conclusion

Silverlight 4 apporte avec lui son lot de surprises, et ce n'est pas tout : ce n'est qu'une beta !

En espérant d'autres améliorations dans la version finale.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.

Remerciements

Merci à toute l'équipe .NET pour leurs relecture et corrections, ainsi que **Jérôme Lambert** pour sa partie sur la webcam et le microphone.

Merci également à **Furr** pour sa relecture finale.

Copyright 2009 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Droits de diffusion permanents accordés à developpez LLC.