

# Windows Phone 7 - Tutoriel 5 : Liste des taches

Application Liste des taches basée sur XML avec Nouveau, Ouvrir, Enregistrer, Ajouter et Retirer items en utilisant Linq-to-XML sur Windows Phone 7

par [Peter Bull \(Blog\)](#) [Deepin Prayag \(Traduction\)](#) ([Home](#))

Date de publication : 22/03/2012

Dernière mise à jour : 23/03/2012

*Ce tutoriel fait partie d'une série de niveau débutant-intermédiaire pour apprendre Windows Phone 7 par le biais d'exemples pas à pas.*

**Langage :** [C#](#)

**Public visé :** niveau *Intermédiaire*

Commentez ce tutoriel :

Traduction.....	3
Introduction.....	3
Application Liste des taches basée sur XML avec Nouveau, Ouvrir, Enregistrer, Ajouter et Retirer items en utilisant Linq-to-XML sur Windows Phone 7.....	3
Étape 1.....	3
Étape 2.....	4
Étape 3.....	5
Étape 4.....	6
Étape 5.....	7
Étape 6.....	7
Étape 7.....	9
Étape 8.....	10
Étape 9.....	12
Étape 10.....	13
Étape 11.....	14
Étape 12.....	16
Étape 13.....	17
Étape 14.....	19
Étape 15.....	20
Étape 16.....	21
Étape 17.....	22
Étape 18.....	23
Étape 19.....	24
Étape 20.....	26
Étape 21.....	27
Étape 22.....	28
Conclusion.....	28
Liens.....	28
Remerciements.....	29

## Traduction

Ce tutoriel est la traduction la plus fidèle possible du tutoriel original de **Peter Bull**, **Windows Phone 7 - XML-based Task List application with New, Open, Save, Add and Remove items using Linq-to-XML on Windows Phone 7**.

## Introduction

Créer avec ce tutoriel une application Liste des taches pour Windows Phone 7 pour des todo list basées sur des *Listbox* avec les supports *Ouvrir / Enregistrer* et *Ajouter / Retirer Items* en utilisant *LINQ to XML*.

Application Liste des taches basée sur XML avec Nouveau, Ouvrir, Enregistrer, Ajouter et Retirer items en utilisant Linq-to-XML sur Windows Phone 7

## Étape 1

Démarrez Microsoft Visual Web Developer 2010 Express pour Windows Phone, puis sélectionnez *Fichier* puis *Nouveau Projet...* Sélectionnez *Visual C#* ensuite *Silverlight for Windows Phone* puis *Application Windows Phone* dans les modèles installés, sélectionnez un emplacement si vous le souhaitez, puis entrez un nom pour le projet et appuyez sur OK :

The screenshot shows the 'Add New Project' dialog in Visual Studio. The 'Trier par' dropdown is set to 'Par défaut'. The list of templates includes:

- Application Windows Phone (Visual C#)
- Application liée aux données Windows Phone (Visual C#)
- Bibliothèque de classes Windows Phone (Visual C#)
- Application Panorama Windows Phone (Visual C#)
- Application Pivot Windows Phone (Visual C#)

The 'Application Windows Phone' template is selected. The 'Type' is 'Visual C#'. The project name is 'WindowsPhoneApplication' and the location is 'C:\Dev\WP7\Tutorials\'.

Buttons: Parcourir..., Créer un répertoire pour la solution, OK

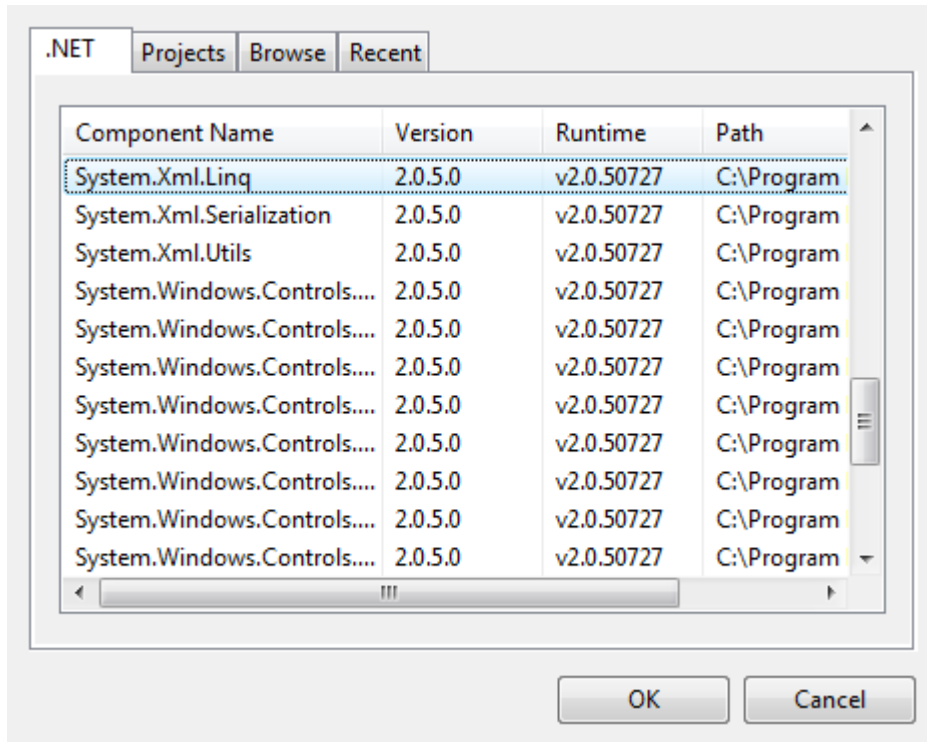
## Étape 2

Une page *Application Windows Phone* nommée *MainPage.xaml* devrait alors apparaître :



### Étape 3

Sélectionnez *Projet* ensuite *Ajouter une référence...* La fenêtre « *Ajouter une référence* » devrait apparaître, sélectionnez *System.XML.Linq* dans la liste *.NET* :



## Étape 4

Faites un clic droit sur *App.xaml* dans l'*Explorateur de solutions* et choisissez *Afficher le code*, ensuite dans la *vue de code* pour *App.xaml.cs* au-dessus de la ligne `public App ()` tapez le code suivant :

```

private string _filename = "";
private object _content = null;

public string Filename
{
    get { return _filename; }
    set { _filename = value; }
}

public object Content
{
    get { return _content; }
    set { _content = value; }
}
    
```

```

namespace TaskList
{
    public partial class App : Application
    {
        private string _filename = "";
        private object _content = null;

        public string Filename
        {
            get { return _filename; }
            set { _filename = value; }
        }

        public object Content
        {
            get { return _content; }
            set { _content = value; }
        }

        public App()
    }
}

```

## Étape 5

Retournez au *Concepteur de vues MainPage* en sélectionnant l'onglet *MainPage.xaml*. Ensuite dans le volet XAML au-dessus de la ligne `<Grid x:Name="LayoutRoot" Background="{StaticResource PhoneBackgroundBrush}">` tapez le code XAML suivant :

```

<phone:PhoneApplicationPage.ApplicationBar>
  <shell:ApplicationBar IsVisible="True" IsMenuEnabled="True">
    <shell:ApplicationBar.MenuItems>
      <shell:ApplicationBarItem Text="new" Click="New_Click"/>
      <shell:ApplicationBarItem Text="open" Click="Open_Click"/>
      <shell:ApplicationBarItem Text="save" Click="Save_Click"/>
    </shell:ApplicationBar.MenuItems>
  </shell:ApplicationBar>
</phone:PhoneApplicationPage.ApplicationBar>

```

```

<phone:PhoneApplicationPage.ApplicationBar>
  <shell:ApplicationBar IsVisible="True" IsMenuEnabled="True">
    <shell:ApplicationBar.MenuItems>
      <shell:ApplicationBarItem Text="new" Click="New_Click"/>
      <shell:ApplicationBarItem Text="open" Click="Open_Click"/>
      <shell:ApplicationBarItem Text="save" Click="Save_Click"/>
    </shell:ApplicationBar.MenuItems>
  </shell:ApplicationBar>
</phone:PhoneApplicationPage.ApplicationBar>

```

## Étape 6

Étant toujours dans le volet XAML, entre les lignes `<Grid x:Name="ContentPanel" Grid.Row="1">` et `</Grid>` , tapez le code XAML suivant :

```

<Grid x:Name="ContentMain">
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="*/>
  </Grid.RowDefinitions>

```

```

</Grid.RowDefinitions>
<!-- Content -->
<TextBox Grid.Row="0" FontSize="24" Name="Subject"/>
<Grid Grid.Row="1" x:Name="Buttons" >
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="160"/>
    <ColumnDefinition Width="160"/>
    <ColumnDefinition Width="160"/>
  </Grid.ColumnDefinitions>
  <!-- Buttons -->
  <Button Grid.Column="0" Content="add" Click="Add_Click"/>
  <Button Grid.Column="2" Content="remove" Click="Remove_Click"/>
</Grid>
<ListBox Grid.Row="2" Name="Tasks"/>
</Grid>

```

XAML :

```

<Grid x:Name="ContentGrid" Grid.Row="1">
  <Grid x:Name="ContentMain">
    <Grid.RowDefinitions>
      <RowDefinition Height="Auto"/>
      <RowDefinition Height="Auto"/>
      <RowDefinition Height="*/>
    </Grid.RowDefinitions>
    <!-- Content -->
    <TextBox Grid.Row="0" FontSize="24" Name="Subject"/>
    <Grid Grid.Row="1" x:Name="Buttons" >
      <Grid.ColumnDefinitions>
        <ColumnDefinition Width="160"/>
        <ColumnDefinition Width="160"/>
        <ColumnDefinition Width="160"/>
      </Grid.ColumnDefinitions>
      <!-- Buttons -->
      <Button Grid.Column="0" Content="add" Click="Add_Click"/>
      <Button Grid.Column="2" Content="remove" Click="Remove_Click"/>
    </Grid>
    <ListBox Grid.Row="2" Name="Tasks"/>
  </Grid>
</Grid>

```

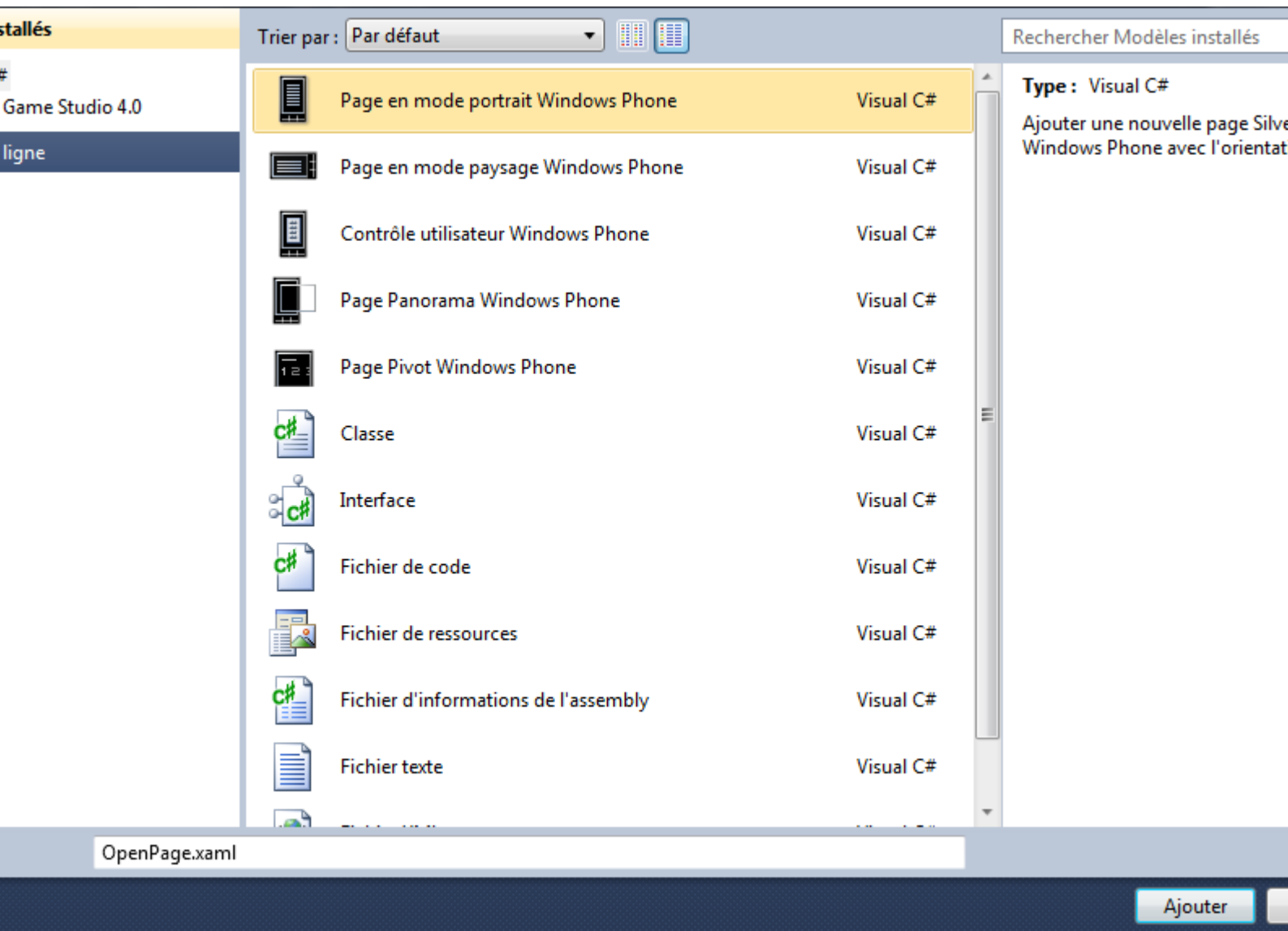
Design :





## Étape 7

Sélectionnez *Projet* ensuite *Ajouter un nouvel élément...*, sélectionnez le modèle *Page en mode portrait Windows Phone*, ensuite modifiez le *Nom* à *OpenPage.xaml* :



## Étape 8

Dans le *Concepteur de vues* pour *OpenPage.xaml*, dans le volet XAML entrez les lignes `<Grid Grid.Row="1" x:Name="ContentPanel">` et `</Grid>` tapez le code XAML suivant :

```
<Grid x:Name="ContentMain">
  <Grid.RowDefinitions>
    <RowDefinition Height="*" />
    <RowDefinition Height="Auto" />
  </Grid.RowDefinitions>
  <!-- Content -->
  <ListBox Grid.Row="0" Margin="10" FontSize="48" Name="Files" />
  <Grid Grid.Row="1" x:Name="Buttons" >
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="160" />
      <ColumnDefinition Width="160" />
      <ColumnDefinition Width="160" />
    </Grid.ColumnDefinitions>
    <!-- Buttons -->
    <Button Grid.Column="0" Content="open" Click="Open_Click" />
    <Button Grid.Column="1" Content="delete" Click="Delete_Click" />
    <Button Grid.Column="2" Content="cancel" Click="Cancel_Click" />
  </Grid>
</Grid>
```

XAML :

```

<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
  <Grid x:Name="ContentMain">
    <Grid.RowDefinitions>
      <RowDefinition Height="*" />
      <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>
    <!-- Content -->
    <ListBox Grid.Row="0" Margin="10" FontSize="48" Name="Files" />
    <Grid Grid.Row="1" x:Name="Buttons" >
      <Grid.ColumnDefinitions>
        <ColumnDefinition Width="160" />
        <ColumnDefinition Width="160" />
        <ColumnDefinition Width="160" />
      </Grid.ColumnDefinitions>
      <!-- Buttons -->
      <Button Grid.Column="0" Content="open" Click="Open_Click" />
      <Button Grid.Column="1" Content="delete" Click="Delete_Click" />
      <Button Grid.Column="2" Content="cancel" Click="Cancel_Click" />
    </Grid>
  </Grid>
</Grid>

```

Design :



## Étape 9

Faites un clic droit sur la page ou sur l'entrée pour *OpenPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Afficher le code*. Dans la *vue de code* au-dessus de `namespace TaskList` tapez ceci :

```
using System.IO.IsolatedStorage;
using System.Xml.Linq;
```

Toujours dans la *vue de code*, au-dessus de `public OpenPage ()` tapez la déclaration *Application* suivante :

```
public App app = (App) Application.Current;
```

```

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;
using Microsoft.Phone.Controls;
using System.IO.IsolatedStorage;
using System.Xml.Linq;

namespace TaskList
{
    public partial class OpenPage : PhoneApplicationPage
    {
        public App app = (App)Application.Current;

        public OpenPage()
        {
            InitializeComponent();
        }
    }
}

```

## Étape 10

Puisque nous en sommes à la *vue de code* pour *OpenPage.xaml.cs*, dans le *constructeur* `public OpenPage()` en dessous de `InitializeComponent()` ; tapez ceci :

```

ApplicationTitle.Text = "TASK LIST";
PageTitle.Text = "open";
Loaded += (object sender, RoutedEventArgs e) =>
{
    Files.Items.Clear();
    using (IsolatedStorageFile storage = IsolatedStorageFile.GetUserStoreForApplication())
    {
        foreach (string filename in storage.GetFileNames("*.tsk"))
        {
            Files.Items.Add(filename.ToLower());
        }
    }
};

```

```

public OpenPage()
{
    InitializeComponent();
    ApplicationTitle.Text = "TASK LIST";
    PageTitle.Text = "open";
    Loaded += (object sender, RoutedEventArgs e) =>
    {
        Files.Items.Clear();
        using (IsolatedStorageFile storage = IsolatedStorageFile.GetUserStoreForApplication())
        {
            foreach (string filename in storage.GetFileNames("*.tsk"))
            {
                Files.Items.Add(filename.ToLower());
            }
        }
    };
}
    
```

## Étape 11

Puisque nous en sommes toujours à la *vue de code* pour *OpenPage.xaml.cs*, en dessous du « } » de la méthode `public OpenPage()` tapez les *gestionnaires d'évènements* suivants :

```

private void Open_Click(object sender, RoutedEventArgs e)
{
    if (Files.SelectedItem != null)
    {
        app.Filename = (string)Files.SelectedItem;
        using (IsolatedStorageFile storage = IsolatedStorageFile.GetUserStoreForApplication())
        {
            XElement _xml;
            ListBox _tasks = new ListBox();
            IsolatedStorageFileStream location = new IsolatedStorageFileStream(app.Filename,
                System.IO.FileMode.Open, storage);
            System.IO.StreamReader file = new System.IO.StreamReader(location);
            _xml = XElement.Parse(file.ReadToEnd());
            if (_xml.Name.LocalName == "tasklist") // Root
            {
                _tasks.Items.Clear();
                foreach (XElement _task in _xml.Descendants("task"))
                {
                    CheckBox _item = new CheckBox();
                    _item.IsChecked = _task.FirstAttribute.Value.ToLower() == "checked";
                    _item.Content = _task.Value;
                    _tasks.Items.Add(_item);
                }
            }
            app.Content = _tasks;
            file.Dispose();
            location.Dispose();
        }
        NavigationService.GoBack();
    }
}

private void Delete_Click(object sender, RoutedEventArgs e)
{
    if (Files.SelectedItem != null)
    {
    
```

```

string _selected = (string)Files.SelectedItem;
if (MessageBox.Show("Delete selected item " + _selected + "?", "Task List",
    MessageBoxButton.OKCancel) == MessageBoxResult.OK)
{
    using (IsolatedStorageFile storage = IsolatedStorageFile.GetUserStoreForApplication())
    {
        if (storage.FileExists(_selected))
        {
            storage.DeleteFile(_selected);
        }
    }
    NavigationService.GoBack();
}
}

private void Cancel_Click(object sender, RoutedEventArgs e)
{
    NavigationService.GoBack();
}

```

```

private void Open_Click(object sender, RoutedEventArgs e)
{
    if (Files.SelectedItem != null)
    {
        app.Filename = (string)Files.SelectedItem;
        using (IsolatedStorageFile storage = IsolatedStorageFile.GetUserStoreForApplication())
        {
            XElement _xml;
            ListBox _tasks = new ListBox();
            IsolatedStorageFileStream location = new IsolatedStorageFileStream(app.Filename,
                System.IO.FileMode.Open, storage);
            System.IO.StreamReader file = new System.IO.StreamReader(location);
            _xml = XElement.Parse(file.ReadToEnd());
            if (_xml.Name.LocalName == "tasklist") // Root
            {
                _tasks.Items.Clear();
                foreach (XElement _task in _xml.Descendants("task"))
                {
                    CheckBox _item = new CheckBox();
                    _item.IsChecked = _task.FirstAttribute.Value.ToLower() == "checked";
                    _item.Content = _task.Value;
                    _tasks.Items.Add(_item);
                }
            }
            app.Content = _tasks;
            file.Dispose();
            location.Dispose();
        }
        NavigationService.GoBack();
    }
}

```

```

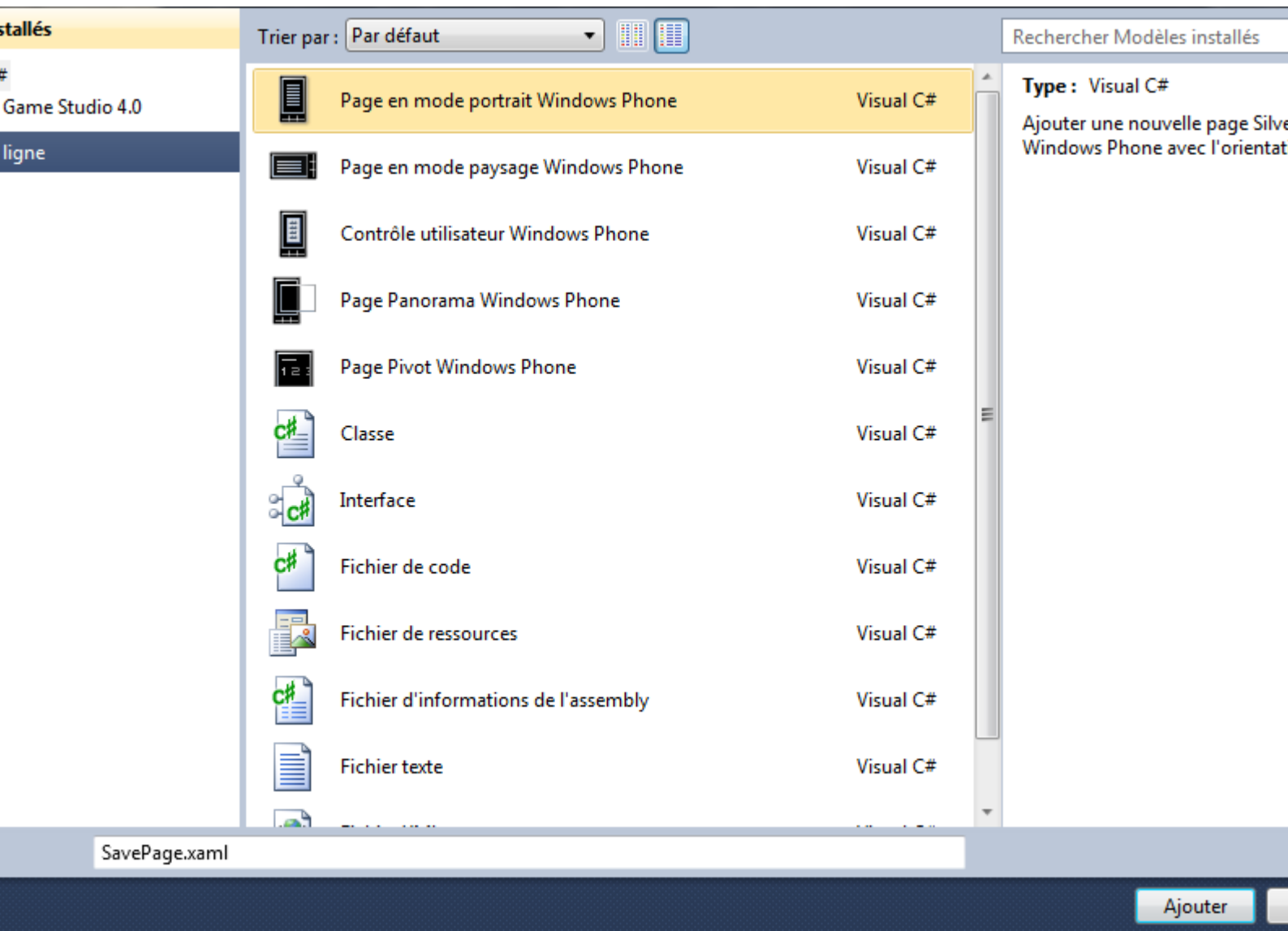
private void Delete_Click(object sender, RoutedEventArgs e)
{
    if (Files.SelectedItem != null)
    {
        string _selected = (string)Files.SelectedItem;
        if (MessageBox.Show("Delete selected Item " + _selected + "?", "Task List",
            MessageBoxButton.OKCancel) == MessageBoxResult.OK)
        {
            using (IsolatedStorageFile storage = IsolatedStorageFile.GetUserStoreForApplication())
            {
                if (storage.FileExists(_selected))
                {
                    storage.DeleteFile(_selected);
                }
            }
            NavigationService.GoBack();
        }
    }
}

private void Cancel_Click(object sender, RoutedEventArgs e)
{
    NavigationService.GoBack();
}
    
```

## Étape 12

Sélectionnez *Projet* ensuite *Ajouter un nouvel élément...*, sélectionnez le modèle *Page en mode portrait Windows Phone*, ensuite modifiez le *Nom* à *SavePage.xaml* :





## Étape 13

Dans le *Concepteur de vues* pour *SavePage.xaml*, et dans le volet XAML entre les lignes `<Grid Grid.Row="1" x:Name="ContentGrid">` et `</Grid>` tapez le code XAML suivant :

```
<Grid x:Name="ContentMain">
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="*/>
    <RowDefinition Height="Auto"/>
  </Grid.RowDefinitions>
  <!-- Content -->
  <TextBox Grid.Row="0" FontSize="24" Name="Filename">
    <TextBox.InputScope>
      <InputScope>
        <InputScopeName NameValue="FileName"/>
      </InputScope>
    </TextBox.InputScope>
  </TextBox>
  <Grid Grid.Row="2" x:Name="Buttons" >
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="160"/>
      <ColumnDefinition Width="160"/>
      <ColumnDefinition Width="160"/>
    </Grid.ColumnDefinitions>
  </Grid>
</Grid>
```

```
</Grid.ColumnDefinitions>
<!-- Buttons -->
<Button Grid.Column="0" Content="save" Click="Save_Click"/>
<Button Grid.Column="2" Content="cancel" Click="Cancel_Click"/>
</Grid>
</Grid>
```

XAML :

```
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
  <Grid x:Name="ContentMain">
    <Grid.RowDefinitions>
      <RowDefinition Height="Auto"/>
      <RowDefinition Height="*/>
      <RowDefinition Height="Auto"/>
    </Grid.RowDefinitions>
    <!-- Content -->
    <TextBox Grid.Row="0" FontSize="24" Name="Filename">
      <TextBox.InputScope>
        <InputScope>
          <InputScopeName NameValue="FileName"/>
        </InputScope>
      </TextBox.InputScope>
    </TextBox>
    <Grid Grid.Row="2" x:Name="Buttons" >
      <Grid.ColumnDefinitions>
        <ColumnDefinition Width="160"/>
        <ColumnDefinition Width="160"/>
        <ColumnDefinition Width="160"/>
      </Grid.ColumnDefinitions>
      <!-- Buttons -->
      <Button Grid.Column="0" Content="save" Click="Save_Click"/>
      <Button Grid.Column="2" Content="cancel" Click="Cancel_Click"/>
    </Grid>
  </Grid>
</Grid>
```

Design :



## Étape 14

Faites un clic droit sur la page ou sur l'entrée pour *SavePage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Afficher le code*. Dans la *vue de code* au-dessus de `namespace TaskList` tapez ceci :

```
using System.IO.IsolatedStorage;
using System.Xml.Linq;
```

Toujours dans la *vue de code* au-dessus de `public SavePage ()` tapez la déclaration *Application* suivante :

```
public App app = (App) Application.Current;
```

```

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;
using Microsoft.Phone.Controls;
using System.IO.IsolatedStorage;
using System.Xml.Linq;

namespace TaskList
{
    public partial class SavePage : PhoneApplicationPage
    {
        public App app = (App)Application.Current;

        public SavePage()
        {
            InitializeComponent();
        }
    }
}

```

## Étape 15

Puisque nous en sommes à la *vue de code* pour *SavePage.xaml.cs*, dans le *constructeur* `public SavePage()` en dessous de `InitializeComponent();` ; tapez ceci :

```

ApplicationTitle.Text = "TASK LIST";
PageTitle.Text = "save";
Loaded += (object sender, RoutedEventArgs e) =>
{
    if (app.Filename == "")
    {
        Filename.Text = "untitled.tsk";
    }
    else
    {
        Filename.Text = app.Filename;
    }
};

```

```

public SavePage()
{
    InitializeComponent();
    ApplicationTitle.Text = "TASK LIST";
    PageTitle.Text = "save";
    Loaded += (object sender, RoutedEventArgs e) =>
    {
        if (app.Filename == "")
        {
            Filename.Text = "untitled.tsk";
        }
        else
        {
            Filename.Text = app.Filename;
        }
    };
}
    
```

## Étape 16

Puisque nous en sommes toujours à la *vue de code* pour *SavePage.xaml.cs*, en dessous du « } » de la méthode `public SavePage()` tapez les *gestionnaires d'évènements* suivants :

```

private void Save_Click(object sender, RoutedEventArgs e)
{
    if (Filename.Text != "")
    {
        try
        {
            app.Filename = Filename.Text.Trim().ToLower();
            using (IsolatedStorageFile storage = IsolatedStorageFile.GetUserStoreForApplication())
            {
                XDocument _doc = new XDocument();
                ListBox _tasks = new ListBox();
                _tasks = (ListBox)app.Content;
                XElement _items = new XElement("tasklist");
                foreach (CheckBox _task in _tasks.Items)
                {
                    _items.Add(new XElement("task", _task.Content, new XAttribute("value",
                        ((bool)_task.IsChecked ? "checked" : "unchecked"))));
                }
                _doc = new XDocument(new XDeclaration("1.0", "utf-8", "yes"), _items);
                IsolatedStorageFileStream location = new IsolatedStorageFileStream(app.Filename,
                    System.IO.FileMode.Create, storage);
                System.IO.StreamWriter file = new System.IO.StreamWriter(location);
                _doc.Save(file);
                app.Content = null;
                file.Dispose();
                location.Dispose();
            }
            NavigationService.GoBack();
        }
        catch
        {
            // Ignore Errors
        }
    }
}

private void Cancel_Click(object sender, RoutedEventArgs e)
{
    NavigationService.GoBack();
}
    
```

```

private void Save_Click(object sender, RoutedEventArgs e)
{
    if (Filename.Text != "")
    {
        try
        {
            app.Filename = Filename.Text.Trim().ToLower();
            using (IsolatedStorageFile storage = IsolatedStorageFile.GetUserStoreForApplication())
            {
                XDocument _doc = new XDocument();
                ListBox _tasks = new ListBox();
                _tasks = (ListBox)app.Content;
                XElement _items = new XElement("tasklist");
                foreach (CheckBox _task in _tasks.Items)
                {
                    _items.Add(new XElement("task", _task.Content, new XAttribute("value",
                        ((bool)_task.IsChecked ? "checked" : "unchecked"))));
                }
                _doc = new XDocument(new XDeclaration("1.0", "utf-8", "yes"), _items);
                IsolatedStorageFileStream location = new IsolatedStorageFileStream(app.Filename,
                    System.IO.FileMode.Create, storage);
                System.IO.StreamWriter file = new System.IO.StreamWriter(location);
                _doc.Save(file);
                app.Content = null;
                file.Dispose();
                location.Dispose();
            }
            NavigationService.GoBack();
        }
        catch
        {
            // Ignore Errors
        }
    }
}

private void Cancel_Click(object sender, RoutedEventArgs e)
{
    NavigationService.GoBack();
}

```

## Étape 17

Retournez au *MainPage.xaml* en sélectionnant l'onglet *MainPage.xaml*, ensuite faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Afficher le code*. Dans la *vue de code* au-dessus de `public MainPage()` tapez la déclaration *Application* suivante :

```
public App app = (App)Application.Current;
```

```

namespace TaskList
{
    public partial class MainPage : PhoneApplicationPage
    {
        public App app = (App)Application.Current;

        // Constructeur
        public MainPage()
        {
            InitializeComponent();
        }
    }
}

```

## Étape 18

En étant toujours dans la *vue de code* pour *MainPage.xaml.cs*, dans le *constructeur* `public MainPage()` en dessous de *InitializeComponent();* tapez ou remplacez par ceci :

```

ApplicationTitle.Text = "TASK LIST";
PageTitle.Text = "untitled.tsk";
Loaded += (object sender, RoutedEventArgs e) =>
{
    if (app.Content != null)
    {
        ListBox _tasks = new ListBox();
        Tasks.Items.Clear();
        _tasks = (ListBox)app.Content;
        foreach (CheckBox Item in _tasks.Items)
        {
            CheckBox _task = new CheckBox();
            _task.Content = Item.Content;
            _task.IsChecked = Item.IsChecked;
            Tasks.Items.Add(_task);
        }
    }
    if (app.Filename != "")
    {
        PageTitle.Text = app.Filename;
    }
    else
    {
        PageTitle.Text = "untitled.tsk";
    }
};

```

```

public MainPage()
{
    InitializeComponent();
    ApplicationTitle.Text = "TASK LIST";
    PageTitle.Text = "untitled.tsk";
    Loaded += (object sender, RoutedEventArgs e) =>
    {
        if (app.Content != null)
        {
            ListBox _tasks = new ListBox();
            Tasks.Items.Clear();
            _tasks = (ListBox)app.Content;
            foreach (CheckBox Item in _tasks.Items)
            {
                CheckBox _task = new CheckBox();
                _task.Content = Item.Content;
                _task.IsChecked = Item.IsChecked;
                Tasks.Items.Add(_task);
            }
        }
        if (app.Filename != "")
        {
            PageTitle.Text = app.Filename;
        }
        else
        {
            PageTitle.Text = "untitled.tsk";
        }
    }
};
}
    
```

## Étape 19

Puisque nous en sommes à la *vue de code* pour *MainPage.xaml.cs*, en dessous du « } » de la méthode `public MainPage()` tapez les *gestionnaires d'événements* suivants :

```

private void Add_Click(object sender, EventArgs e)
{
    if (Subject.Text != "") // Has Subject
    {
        CheckBox _item = new CheckBox();
        _item.Content = Subject.Text;
        if (Tasks.SelectedIndex > -1) // Item Selected
        {
            // Insert before Selected
            Tasks.Items.Insert(Tasks.SelectedIndex, _item);
        }
        else
        {
            // Add to List End
            Tasks.Items.Add(_item);
        }
    }
}

private void Remove_Click(object sender, EventArgs e)
{
    if (Tasks.SelectedIndex > -1) // Item Selected
    {
        // Remove Selected
        Tasks.Items.RemoveAt(Tasks.SelectedIndex);
    }
}
    
```



```
}  
  
private void New_Click(object sender, EventArgs e)  
{  
    if (MessageBox.Show("Start a new Task List?", "Task List",  
        MessageBoxButton.OKCancel) == MessageBoxResult.OK)  
    {  
        PageTitle.Text = "untitled.tsk";  
        Tasks.Items.Clear();  
        app.Filename = PageTitle.Text;  
        app.Content = Tasks.Items;  
    }  
}  
  
private void Open_Click(object sender, EventArgs e)  
{  
    app.Content = Tasks;  
    NavigationService.Navigate(new Uri("/OpenPage.xaml", UriKind.Relative));  
}  
  
private void Save_Click(object sender, EventArgs e)  
{  
    app.Content = Tasks;  
    NavigationService.Navigate(new Uri("/SavePage.xaml", UriKind.Relative));  
}
```

```

private void Add_Click(object sender, EventArgs e)
{
    if (Subject.Text != "") // Has Subject
    {
        CheckBox _item = new CheckBox();
        _item.Content = Subject.Text;
        if (Tasks.SelectedIndex > -1) // Item Selected
        {
            // Insert before Selected
            Tasks.Items.Insert(Tasks.SelectedIndex, _item);
        }
        else
        {
            // Add to List End
            Tasks.Items.Add(_item);
        }
    }
}

private void Remove_Click(object sender, EventArgs e)
{
    if (Tasks.SelectedIndex > -1) // Item Selected
    {
        // Remove Selected
        Tasks.Items.RemoveAt(Tasks.SelectedIndex);
    }
}

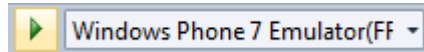
private void New_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Start a new Task List?", "Task List",
        MessageBoxButton.OKCancel) == MessageBoxResult.OK)
    {
        PageTitle.Text = "untitled.tsk";
        Tasks.Items.Clear();
        app.Filename = PageTitle.Text;
        app.Content = Tasks.Items;
    }
}

private void Open_Click(object sender, EventArgs e)
{
    app.Content = Tasks;
    NavigationService.Navigate(new Uri("/OpenPage.xaml", UriKind.Relative));
}

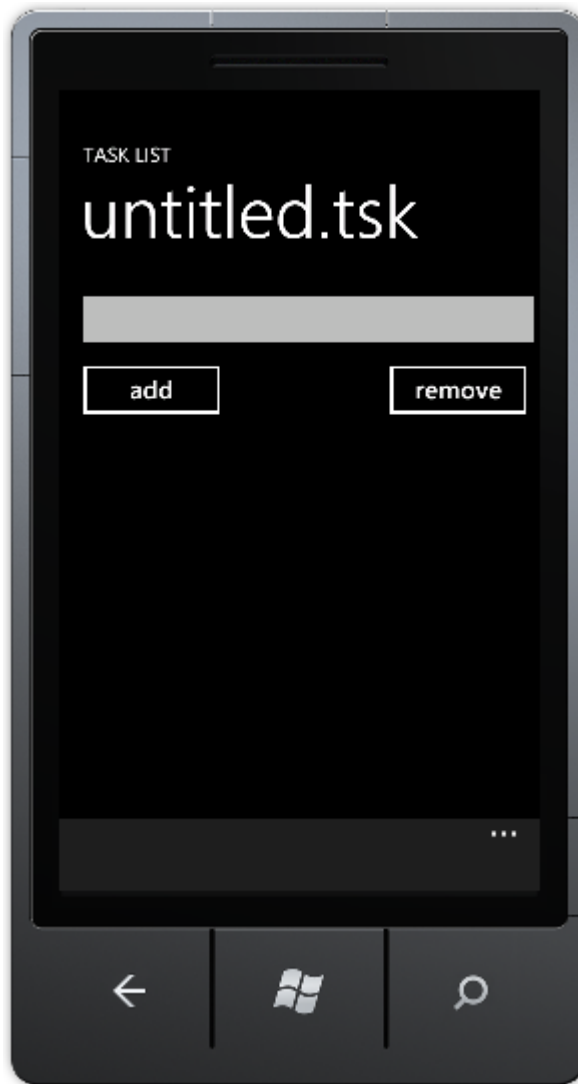
private void Save_Click(object sender, EventArgs e)
{
    app.Content = Tasks;
    NavigationService.Navigate(new Uri("/SavePage.xaml", UriKind.Relative));
}
    
```

## Étape 20

Enregistrez le *Projet* maintenant que vous avez terminé l'application *Windows Phone Silverlight*. Sélectionnez l'option *Windows Phone Emulator*, ensuite sélectionnez *Debug* puis *Démarrer le débogage* ou cliquez sur *Démarrer le débogage* :

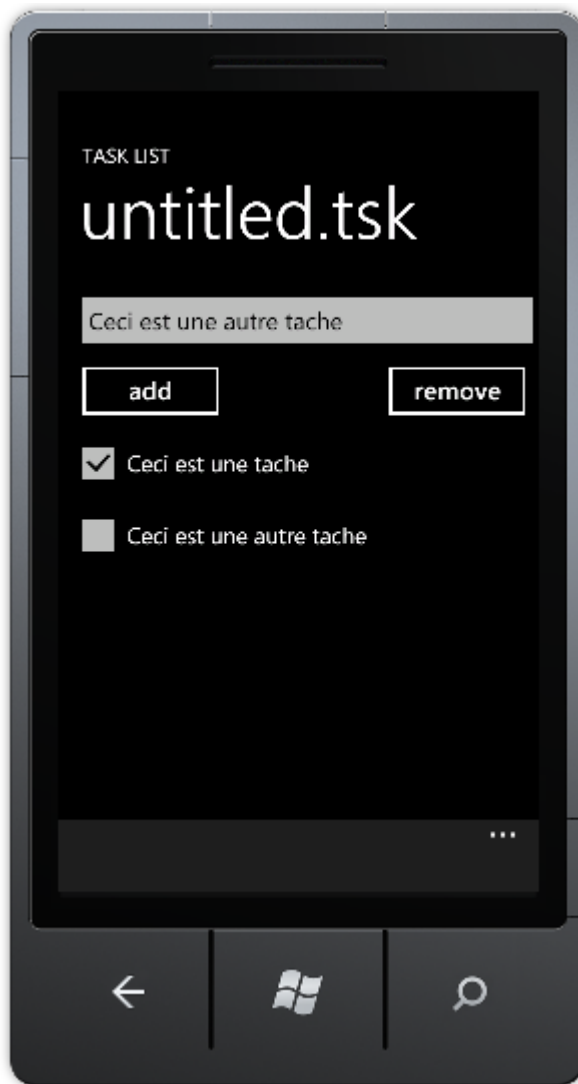


Une fois que vous l'aurez fait, ce qui suit apparaîtra dans *l'émulateur Windows Phone* après qu'il aura été chargé :



## Étape 21

Si vous sélectionnez et saisissez dans le *TextBox* le sujet d'une tâche, qu'ensuite vous cliquez sur le bouton *Add*, vous pouvez ajouter quelques tâches et cochez la case une fois que ce sera fait. Vous pouvez alors *enregistrer* (*Save*) cette liste pour *l'ouvrir* (*Open*) plus tard via le menu en cliquant sur « ... » :



## Étape 22

Fermez la fenêtre du navigateur en cliquant sur le bouton Fermer



en haut à droite du navigateur Web pour *Arrêter l'application*.

## Conclusion

Ceci est une application Liste de taches simple avec la possibilité d'ouvrir et d'enregistrer des fichiers de listes de taches sous XML y compris les éléments cochés et leur ordre d'apparition. Essayez d'ajouter des nouvelles fonctionnalités comme la possibilité de déplacer les items vers le haut et le bas de la liste et la possibilité d'éditer des tâches déjà saisies, ou utilisez l'application comme base pour des applications Windows Phone 7 plus complexes basées sur XML - personnalisez-la !

## Liens

**Télécharger le code source du tutoriel.**

## Remerciements

Je tiens ici à remercier **Peter Bull** de m'avoir autorisé à traduire son tutoriel.

Je remercie **xxx** pour sa relecture technique et ses propositions.

Je remercie également **yyy** pour sa relecture orthographique et ses propositions.