

# Microsoft Silverlight 4 - Tutoriel 14 : Éditeur de texte riche

Éditeur de texte riche avec la mise en forme, l'impression, ouvrir/enregistrer, etc.

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

Date de publication : 30/03/2012

Dernière mise à jour : 30/03/2012

Ce tutoriel fait partie d'une **série** de niveau débutant-intermédiaire pour apprendre Silverlight 4 par le biais d'exemples pas à pas.

**Langage** : [VB.NET](#)

**Public visé** : niveau **Intermédiaire**

Commentez ce tutoriel :

Traduction.....	3
Introduction.....	3
Éditeur de texte riche avec la mise en forme, l'impression, ouvrir/enregistrer, etc.....	3
Étape 1.....	3
Étape 2.....	4
Étape 3.....	5
Étape 4.....	5
Étape 5.....	6
Étape 6.....	7
Étape 7.....	8
Étape 8.....	9
Étape 9.....	9
Étape 10.....	11
Étape 11.....	11
Étape 12.....	12
Étape 13.....	13
Étape 14.....	15
Étape 15.....	16
Étape 16.....	16
Étape 17.....	17
Étape 18.....	18
Étape 19.....	18
Étape 20.....	19
Étape 21.....	20
Étape 22.....	20
Étape 23.....	20
Étape 24.....	21
Étape 25.....	21
Étape 26.....	22
Étape 27.....	22
Conclusion.....	22
Liens.....	23
Remerciements.....	23

## Traduction

Ce tutoriel est la traduction la plus fidèle possible du tutoriel original de **Peter Bull, Microsoft Silverlight 4 - Rich Text Editor with Formatting, Printing, Open/Save and more.**

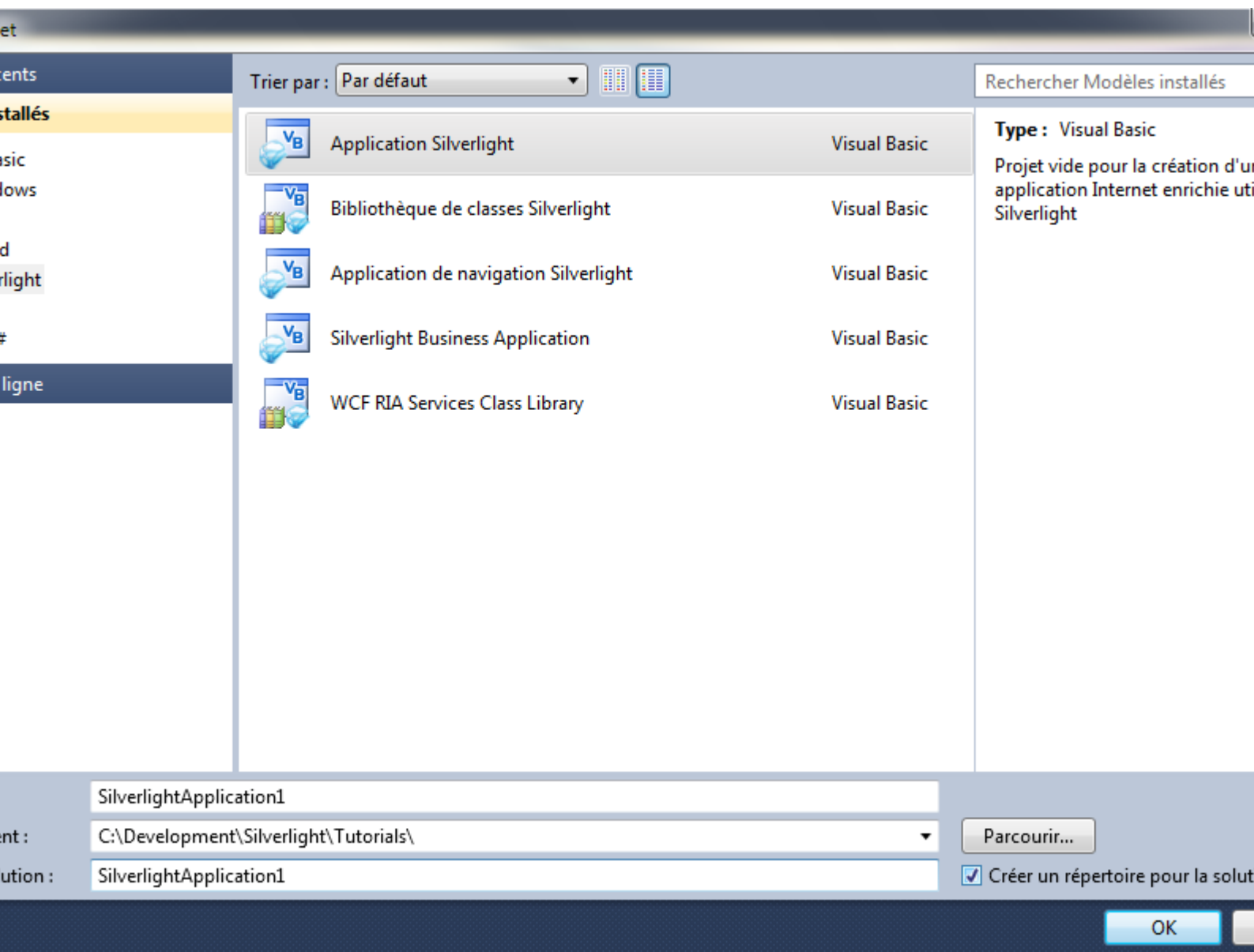
## Introduction

Créez un éditeur de texte riche en utilisant *RichTextArea* avec *Ouvrir (Open)* et *Enregistrer (Save)* en plus de la mise en forme telle que *Bold (Gras)*, *Italic (Italique)* et *Underline (Souligner)* avec Silverlight.

## Éditeur de texte riche avec la mise en forme, l'impression, ouvrir/enregistrer, etc.

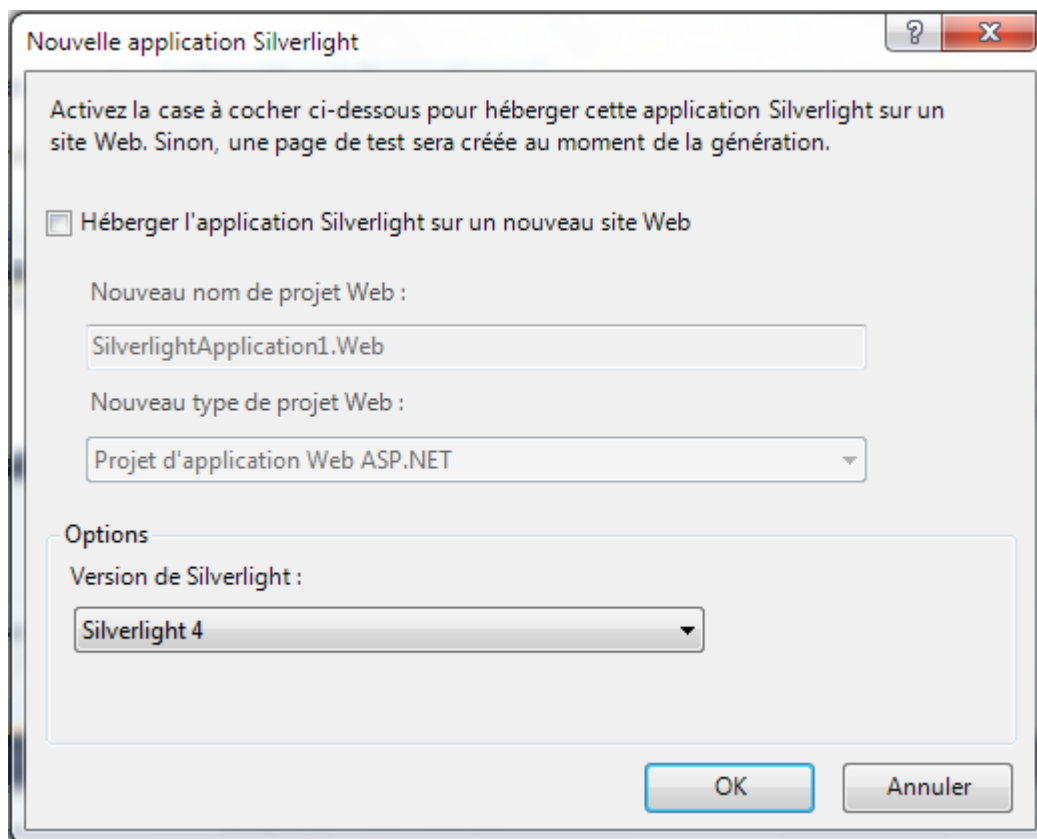
## Étape 1

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



## Étape 2

Une nouvelle fenêtre *Nouvelle application Silverlight* devrait apparaître, décochez la case *Héberger l'application Silverlight sur un nouveau site Web*, puis sélectionnez la *Version de Silverlight* ciblée :



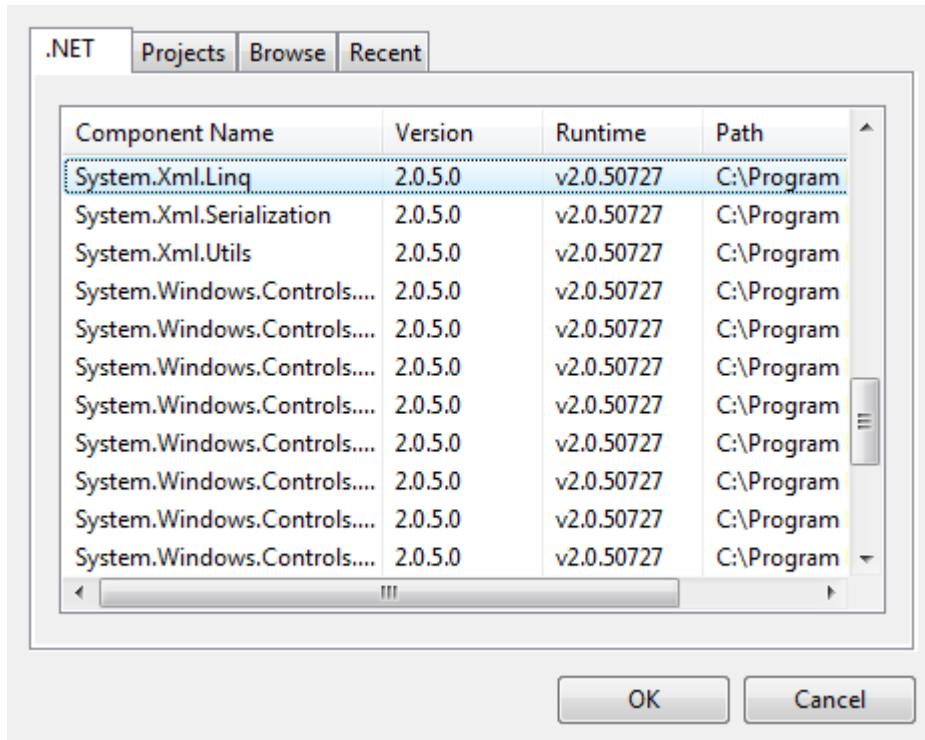
### Étape 3

Une page vierge nommée *MainPage.xaml* devrait alors apparaître :



### Étape 4

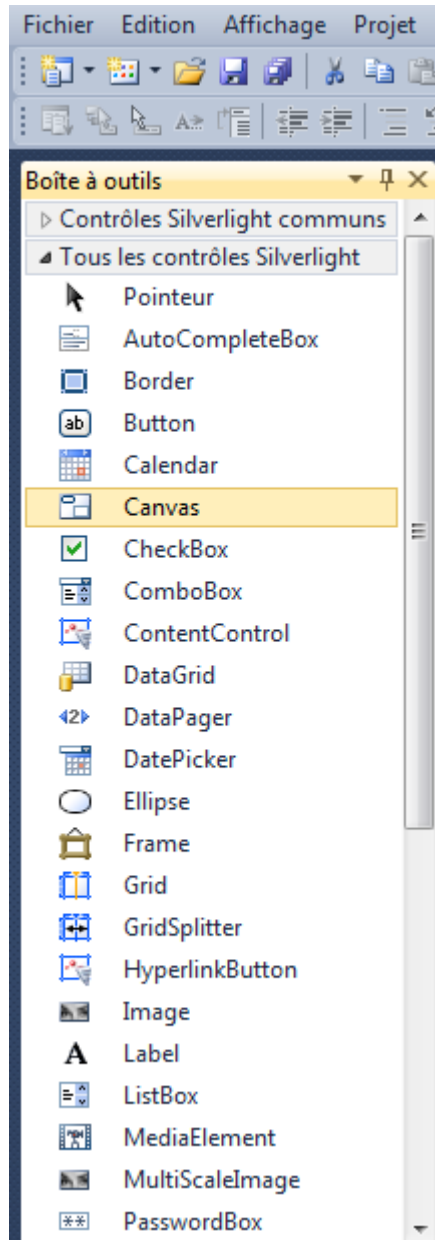
Sélectionnez *Projet* ensuite *Ajouter une référence...* La fenêtre *Ajouter une référence* devrait apparaître, sélectionnez *System.XML.Linq* dans la liste *.NET* :



## Étape 5

Ajouter la référence à *System.XML.Linq* en cliquant sur OK.

Puis dans la section *Tous les contrôles Silverlight* de la *Boîte à outils*, sélectionnez le contrôle *Canvas* :

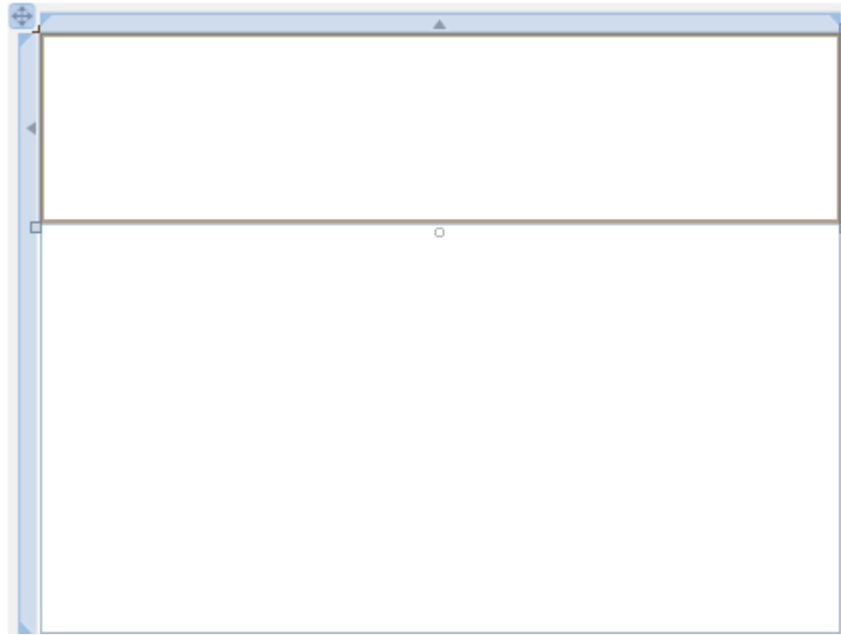


## Étape 6

Dessinez un *Canvas* sur la page, ensuite dans le volet XAML au-dessus du `</Grid>` modifiez la ligne *Canvas1* comme ceci :

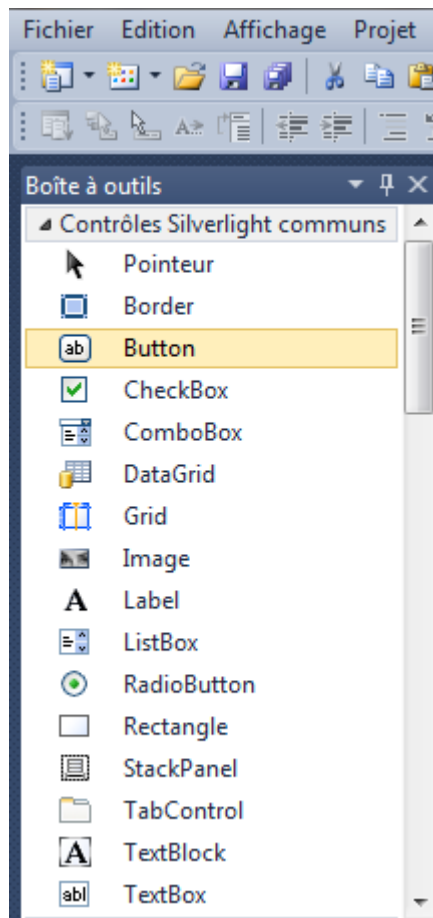
```
<Canvas Height="95" Width="400" VerticalAlignment="Top" HorizontalAlignment="Left" Name="Toolbar"></Canvas>
```

Voir ci-dessous :



## Étape 7

Puis dans la section *Contrôles Silverlight communs* de la *Boîte à outils*, sélectionnez le contrôle *Button* :



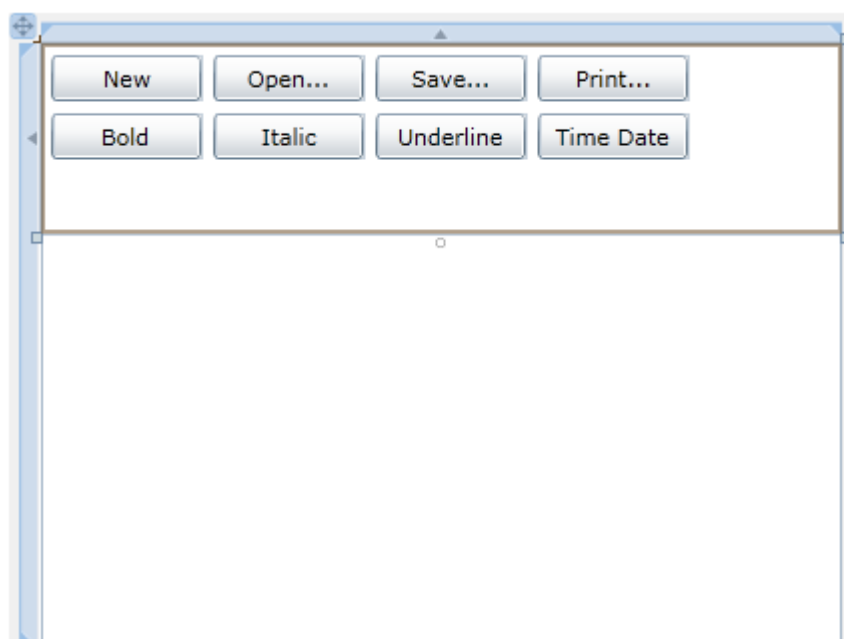


## Étape 8

Dessinez huit boutons sur le *Canvas* en glissant les boutons de la *Boite à outils* sur le *Canvas*, ensuite dans le volet XAML entre les balises `<Canvas>` et `</Canvas>` modifiez les lignes `<Button>` comme ceci :

```
<Button Canvas.Left="6" Canvas.Top="6" Height="23" Width="75" Name="New" Content="New"/>
<Button Canvas.Left="87" Canvas.Top="6" Height="23" Width="75" Name="Open" Content="Open..."/>
<Button Canvas.Left="168" Canvas.Top="6" Height="23" Width="75" Name="Save" Content="Save..."/>
<Button Canvas.Left="249" Canvas.Top="6" Height="23" Width="75" Name="Print" Content="Print..."/>
<Button Canvas.Left="6" Canvas.Top="35" Height="23" Width="75" Name="Bold" Content="Bold"/>
<Button Canvas.Left="87" Canvas.Top="35" Height="23" Width="75" Name="Italic" Content="Italic"/>
<Button Canvas.Left="168" Canvas.Top="35" Height="23" Width="75" Name="Underline" Content="Underline"/>
<Button Canvas.Left="249" Canvas.Top="35" Height="23" Width="75" Name="Time Date" Content="Time Date"/>
```

Voir ci-dessous :



## Étape 9

Etant toujours dans le volet XAML en dessous de la balise `</Canvas>` et au-dessus de la balise `</Grid>`, tapez le code *ComboBox XAML* suivant :

```
<ComboBox Canvas.Left="6" Canvas.Top="64" Height="23" Width="156" Name="Font"
  FontFamily="{Binding SelectedItem.FontFamily, RelativeSource={RelativeSource Self}}"
  FontSize="{Binding SelectedItem.FontSize, RelativeSource={RelativeSource Self}}">
  <ComboBoxItem Content="Arial" Tag="Arial" FontFamily="Arial" FontSize="12" IsSelected="True"/>
  <ComboBoxItem Content="Arial Black" Tag="Arial Black" FontFamily="Arial Black" FontSize="12"/>
  <ComboBoxItem Content="Comic Sans MS" Tag="Comic Sans MS" FontFamily="Comic Sans MS" FontSize="12"/>
  <ComboBoxItem Content="Courier New" Tag="Courier New" FontFamily="Courier New" FontSize="12"/>
  <ComboBoxItem Content="Georgia" Tag="Georgia" FontFamily="Georgia" FontSize="12"/>
  <ComboBoxItem Content="Lucida Sans Unicode" Tag="Lucida Sans Unicode" FontFamily="Lucida Sans Unicode" FontSize="12"/>
  <ComboBoxItem Content="Times New Roman" Tag="Times New Roman" FontFamily="Times New Roman" FontSize="12"/>
  <ComboBoxItem Content="Trebuchet MS" Tag="Trebuchet MS" FontFamily="Trebuchet MS" FontSize="12"/>
  <ComboBoxItem Content="Verdana" Tag="Verdana" FontFamily="Verdana" FontSize="12"/>
  <ComboBoxItem Content="Webdings" Tag="Webdings" FontSize="12"/>
</ComboBox>

<ComboBox Canvas.Left="168" Canvas.Top="64" Height="23" Width="75" Name="Size">
  <ComboBoxItem Content="8" Tag="8"/>
  <ComboBoxItem Content="10" Tag="10"/>
```

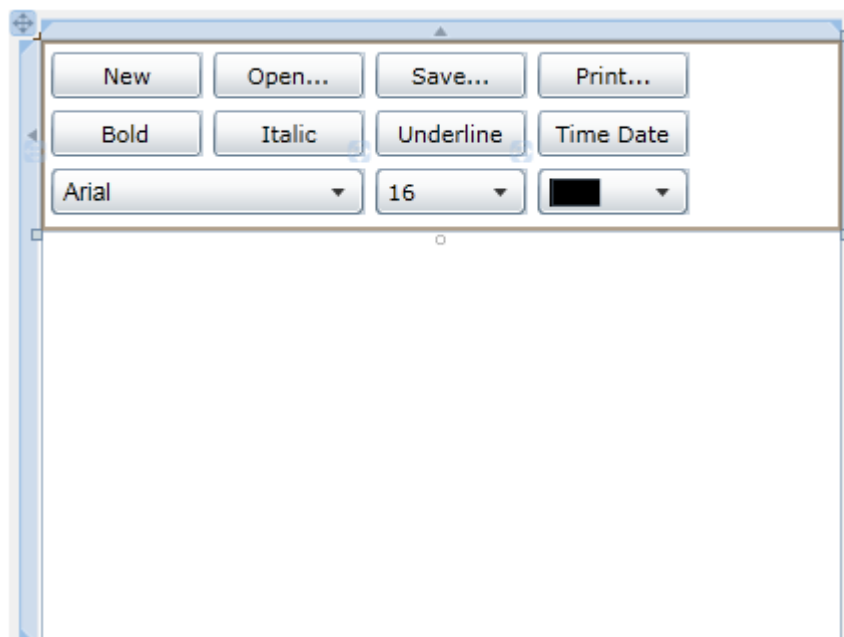
```

<ComboBoxItem Content="12" Tag="12"/>
<ComboBoxItem Content="16" Tag="16" IsSelected="True"/>
<ComboBoxItem Content="20" Tag="20"/>
<ComboBoxItem Content="24" Tag="24"/>
<ComboBoxItem Content="28" Tag="28"/>
<ComboBoxItem Content="36" Tag="36"/>
<ComboBoxItem Content="48" Tag="48"/>
<ComboBoxItem Content="72" Tag="72"/>
</ComboBox>

<ComboBox Canvas.Left="249" Canvas.Top="64" Height="23" Width="75" Name="Colour">
  <ComboBoxItem Tag="FF000000" IsSelected="True">
    <Rectangle Width="25" Height="14" Fill="Black"/>
  </ComboBoxItem>
  <ComboBoxItem Tag="FF808080">
    <Rectangle Width="25" Height="14" Fill="Gray"/>
  </ComboBoxItem>
  <ComboBoxItem Tag="FFFF0000">
    <Rectangle Width="25" Height="14" Fill="Red"/>
  </ComboBoxItem>
  <ComboBoxItem Tag="FFFA5000">
    <Rectangle Width="25" Height="14" Fill="Orange"/>
  </ComboBoxItem>
  <ComboBoxItem Tag="FFFFFF00">
    <Rectangle Width="25" Height="14" Fill="Yellow"/>
  </ComboBoxItem>
  <ComboBoxItem Tag="FF080000">
    <Rectangle Width="25" Height="14" Fill="Green"/>
  </ComboBoxItem>
  <ComboBoxItem Tag="FF00FFFF">
    <Rectangle Width="25" Height="14" Fill="Cyan"/>
  </ComboBoxItem>
  <ComboBoxItem Tag="FF0000FF">
    <Rectangle Width="25" Height="14" Fill="Blue"/>
  </ComboBoxItem>
  <ComboBoxItem Tag="FFFF00FF">
    <Rectangle Width="25" Height="14" Fill="Magenta"/>
  </ComboBoxItem>
  <ComboBoxItem Tag="FF800080">
    <Rectangle Width="25" Height="14" Fill="Purple"/>
  </ComboBoxItem>
</ComboBox>

```

Voir ci-dessous :

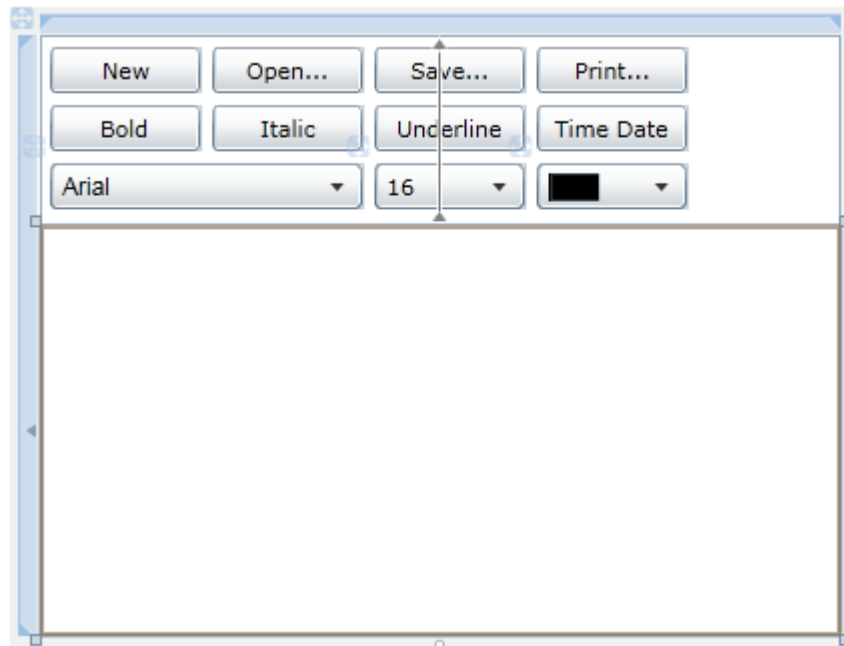


## Étape 10

Une fois de plus, étant toujours dans le volet XAML en dessous de la balise `</Canvas>` et au-dessus de la balise `</Grid>`, tapez le code *RichTextArea XAML* suivant :

```
<RichTextArea Height="205" Width="400" Margin="0,95,0,0" HorizontalAlignment="Left" VerticalAlignment="Top"
TextWrapping="Wrap" BorderThickness="0" Name="Editor"/>
```

Voir ci-dessous :



## Étape 11

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 la ligne `Partial Public Class MainPage`, tapez le code suivant :

```
Imports System.Xml.Linq
Imports System.Windows.Printing
```

```
Imports System.Xml.Linq
Imports System.Windows.Printing
Partial Public Class MainPage
    Inherits UserControl
    Public Sub New()
        InitializeComponent()
    End Sub
End Class
```

## Étape 12

Puisque nous en sommes à la *vue de code* pour *MainPage.xaml*, en dessous du *End Sub* pour le *constructeur Public Sub New()*, tapez le *Sub* suivant :

```

Private Sub Read(ByRef Stream As IO.Stream, ByRef Blocks As BlockCollection)
    Dim _doc As XDocument = XDocument.Load(Stream)
    Dim _colour As String
    For Each _element As XElement In _doc.Descendants("Paragraph")
        Dim _paragraph As New Paragraph
        _paragraph.FontFamily = New FontFamily(_element.Attribute(XName.[Get]("FontFamily")).Value)
        _paragraph.FontSize = Double.Parse(_element.Attribute(XName.[Get]("FontSize")).Value)
        _paragraph.FontStretch = DirectCast(GetType(FontStretches).GetProperty(
            _element.Attribute(XName.[Get]("FontStretch")).Value).GetValue(Nothing, Nothing), FontStretch)
        _paragraph.FontStyle = DirectCast(GetType(FontStyles).GetProperty(
            _element.Attribute(XName.[Get]("FontStyle")).Value).GetValue(Nothing, Nothing), FontStyle)
        _paragraph.FontWeight = DirectCast(GetType(FontWeights).GetProperty(
            _element.Attribute(XName.[Get]("FontWeight")).Value).GetValue(Nothing, Nothing), FontWeight)
        _colour = _element.Attribute(XName.[Get]("Foreground")).Value
        _colour = _colour.Remove(0, 1)
        _paragraph.Foreground = New SolidColorBrush(Color.FromArgb(
            Byte.Parse(_colour.Substring(0, 2), System.Globalization.NumberStyles.HexNumber),
            Byte.Parse(_colour.Substring(2, 2), System.Globalization.NumberStyles.HexNumber),
            Byte.Parse(_colour.Substring(4, 2), System.Globalization.NumberStyles.HexNumber),
            Byte.Parse(_colour.Substring(6, 2), System.Globalization.NumberStyles.HexNumber)))
        For Each _inline As XElement In _element.Descendants("Inline")
            If _inline.Attribute(XName.[Get]("Type")).Value = "Run" Then
                Dim _run As New Run
                _run.FontFamily = New FontFamily(_element.Attribute(XName.[Get]("FontFamily")).Value)
                _run.FontSize = Double.Parse(_element.Attribute(XName.[Get]("FontSize")).Value)
                _run.FontStretch = DirectCast(GetType(FontStretches).GetProperty(
                    _element.Attribute(XName.[Get]("FontStretch")).Value).GetValue(Nothing, Nothing), FontStretch)
                _run.FontStyle = DirectCast(GetType(FontStyles).GetProperty(
                    _element.Attribute(XName.[Get]("FontStyle")).Value).GetValue(Nothing, Nothing), FontStyle)
                _run.FontWeight = DirectCast(GetType(FontWeights).GetProperty(
                    _element.Attribute(XName.[Get]("FontWeight")).Value).GetValue(Nothing, Nothing), FontWeight)
                _colour = _element.Attribute(XName.[Get]("Foreground")).Value
                _colour = _colour.Remove(0, 1)
                _run.Foreground = New SolidColorBrush(Color.FromArgb(
                    Byte.Parse(_colour.Substring(0, 2), System.Globalization.NumberStyles.HexNumber),
                    Byte.Parse(_colour.Substring(2, 2), System.Globalization.NumberStyles.HexNumber),
                    Byte.Parse(_colour.Substring(4, 2), System.Globalization.NumberStyles.HexNumber),
                    Byte.Parse(_colour.Substring(6, 2), System.Globalization.NumberStyles.HexNumber)))
                If _inline.Attribute(XName.[Get]("TextDecorations")).Value = "Underline" Then
                    _run.TextDecorations = TextDecorations.Underline
                End If
                _run.Text = _inline.Attribute(XName.[Get]("Text")).Value
                _paragraph.Inlines.Add(_run)
            End If
        Next
        Blocks.Add(_paragraph)
    Next
End Sub
    
```

```

Private Sub Read(ByRef Stream As IO.Stream, ByRef Blocks As BlockCollection)
    Dim _doc As XDocument = XDocument.Load(Stream)
    Dim _colour As String
    For Each _element As XElement In _doc.Descendants("Paragraph")
        Dim _paragraph As New Paragraph
        _paragraph.FontFamily = New FontFamily(_element.Attribute(XName.[Get]("FontFamily")).Value)
        _paragraph.FontSize = Double.Parse(_element.Attribute(XName.[Get]("FontSize")).Value)
        _paragraph.FontStretch = DirectCast(GetType(FontStretches).GetProperty(
            _element.Attribute(XName.[Get]("FontStretch")).Value).GetValue(Nothing, Nothing), FontStretch)
        _paragraph.FontStyle = DirectCast(GetType(FontStyles).GetProperty(
            _element.Attribute(XName.[Get]("FontStyle")).Value).GetValue(Nothing, Nothing), FontStyle)
        _paragraph.FontWeight = DirectCast(GetType(FontWeights).GetProperty(
            _element.Attribute(XName.[Get]("FontWeight")).Value).GetValue(Nothing, Nothing), FontWeight)
        _colour = _element.Attribute(XName.[Get]("Foreground")).Value
        _colour = _colour.Remove(0, 1)
        _paragraph.Foreground = New SolidColorBrush(Color.FromArgb(
            Byte.Parse(_colour.Substring(0, 2), System.Globalization.NumberStyles.HexNumber),
            Byte.Parse(_colour.Substring(2, 2), System.Globalization.NumberStyles.HexNumber),
            Byte.Parse(_colour.Substring(4, 2), System.Globalization.NumberStyles.HexNumber),
            Byte.Parse(_colour.Substring(6, 2), System.Globalization.NumberStyles.HexNumber)))

        For Each _inline As XElement In _element.Descendants("Inline")
            If _inline.Attribute(XName.[Get]("Type")).Value = "Run" Then
                Dim _run As New Run
                _run.FontFamily = New FontFamily(_element.Attribute(XName.[Get]("FontFamily")).Value)
                _run.FontSize = Double.Parse(_element.Attribute(XName.[Get]("FontSize")).Value)
                _run.FontStretch = DirectCast(GetType(FontStretches).GetProperty(
                    _element.Attribute(XName.[Get]("FontStretch")).Value).GetValue(Nothing, Nothing), FontStretch)
                _run.FontStyle = DirectCast(GetType(FontStyles).GetProperty(
                    _element.Attribute(XName.[Get]("FontStyle")).Value).GetValue(Nothing, Nothing), FontStyle)
                _run.FontWeight = DirectCast(GetType(FontWeights).GetProperty(
                    _element.Attribute(XName.[Get]("FontWeight")).Value).GetValue(Nothing, Nothing), FontWeight)
                _colour = _element.Attribute(XName.[Get]("Foreground")).Value
                _colour = _colour.Remove(0, 1)
                _run.Foreground = New SolidColorBrush(Color.FromArgb(
                    Byte.Parse(_colour.Substring(0, 2), System.Globalization.NumberStyles.HexNumber),
                    Byte.Parse(_colour.Substring(2, 2), System.Globalization.NumberStyles.HexNumber),
                    Byte.Parse(_colour.Substring(4, 2), System.Globalization.NumberStyles.HexNumber),
                    Byte.Parse(_colour.Substring(6, 2), System.Globalization.NumberStyles.HexNumber)))
                If _inline.Attribute(XName.[Get]("TextDecorations")).Value = "Underline" Then
                    _run.TextDecorations = TextDecorations.Underline
                End If
                _run.Text = _inline.Attribute(XName.[Get]("Text")).Value
                _paragraph.Inlines.Add(_run)
            End If
        Next
        Blocks.Add(_paragraph)
    Next
End Sub

```

## Étape 13

Toujours dans la *vue de code* pour *MainPage.xaml*, en dessous du *End Sub* pour le constructeur `Private Sub Read()`, tapez la *fonction* suivante :

```

Private Function Write(ByRef Blocks As BlockCollection) As String
    Dim _write As New Text.StringBuilder
    Dim _doc = From block In Blocks _
    From inline In TryCast(block, Paragraph).Inlines
        Where inline.GetType Is GetType(InlineUIContainer) _
    Select inline
    If _doc.Count = 0 Then
        _write.Append("<doc>")
        For Each _block As Block In Blocks
            Dim _paragraph As New Paragraph
            _write.Append("<Paragraph ")
            _write.Append("Name='" & _paragraph.Name & "' ")
            _write.Append("FontFamily='" & _paragraph.FontFamily.ToString & "' ")
            _write.Append("FontSize='" & _paragraph.FontSize & "' ")
            _write.Append("FontStretch='" & _paragraph.FontStretch.ToString & "' ")
            _write.Append("FontStyle='" & _paragraph.FontStyle.ToString & "' ")
            _write.Append("FontWeight='" & _paragraph.FontWeight.ToString & "' ")
            _write.Append("Foreground='" & TryCast(_paragraph.Foreground,
                SolidColorBrush).Color.ToString & "'>")
            For Each _inline In TryCast(_block, Paragraph).Inlines
                If TypeOf _inline Is Run Then
                    Dim _run As Run = TryCast(_inline, Run)
                    _write.Append("<Inline Type='Run' ")
                    _write.Append("Name='" & _run.Name & "' ")
                    _write.Append("FontFamily='" & _run.FontFamily.ToString & "' ")
                    _write.Append("FontSize='" & _run.FontSize & "' ")
                    _write.Append("FontStretch='" & _run.FontStretch.ToString & "' ")
                    _write.Append("FontStyle='" & _run.FontStyle.ToString & "' ")
                    _write.Append("FontWeight='" & _run.FontWeight.ToString & "' ")
                    _write.Append("Foreground='" & TryCast(_run.Foreground,
                        SolidColorBrush).Color.ToString & "' ")
                    _write.Append("Text='" & _run.Text & "' ")

                    _write.Append("TextDecorations='" & If(_run.TextDecorations Is Nothing, "", "Underline") & "'/>")
                End If
            Next
            _write.Append("</Paragraph>")
        Next
        _write.Append("</doc>")
        Return _write.ToString
    Else
        Return Nothing
    End If
End Function

```



```

Private Function Write(ByRef Blocks As BlockCollection) As String
    Dim _write As New Text.StringBuilder
    Dim _doc = From block In Blocks _
        From inline In TryCast(block, Paragraph).Inlines
        Where inline.GetType Is GetType(InlineUIContainer) _
        Select inline
    If _doc.Count = 0 Then
        _write.Append("<doc>")
        For Each _block As Block In Blocks
            Dim _paragraph As New Paragraph
            _write.Append("<Paragraph ")
            _write.Append("Name='" & _paragraph.Name & "' ")
            _write.Append("FontFamily='" & _paragraph.FontFamily.ToString & "' ")
            _write.Append("FontSize='" & _paragraph.FontSize & "' ")
            _write.Append("FontStretch='" & _paragraph.FontStretch.ToString & "' ")
            _write.Append("FontStyle='" & _paragraph.FontStyle.ToString & "' ")
            _write.Append("FontWeight='" & _paragraph.FontWeight.ToString & "' ")
            _write.Append("Foreground='" & TryCast(_paragraph.Foreground, SolidColorBrush).Color.ToString & "'>")
            For Each _inline In TryCast(_block, Paragraph).Inlines
                If TypeOf _inline Is Run Then
                    Dim _run As Run = TryCast(_inline, Run)
                    _write.Append("<Inline Type='Run' ")
                    _write.Append("Name='" & _run.Name & "' ")
                    _write.Append("FontFamily='" & _run.FontFamily.ToString & "' ")
                    _write.Append("FontSize='" & _run.FontSize & "' ")
                    _write.Append("FontStretch='" & _run.FontStretch.ToString & "' ")
                    _write.Append("FontStyle='" & _run.FontStyle.ToString & "' ")
                    _write.Append("FontWeight='" & _run.FontWeight.ToString & "' ")
                    _write.Append("Foreground='" & TryCast(_run.Foreground, SolidColorBrush).Color.ToString & "' ")
                    _write.Append("Text='" & _run.Text & "' ")
                    _write.Append("TextDecorations='" & If(_run.TextDecorations Is Nothing, "", "Underline") & "' ")
                End If
            Next
            _write.Append("</Paragraph>")
        Next
        _write.Append("</doc>")
    Return _write.ToString
Else
    Return Nothing
End If
End Function
    
```

## Étape 14

Retournez au *concepteur de vues* en sélectionnant l'onglet *MainPage.xaml*, ou faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Concepteur de vues*.

Double-cliquez sur le bouton « *New* » et tapez ceci dans le Sub *New\_Click* :

```

If MessageBox.Show("Start a New Document?", "Rich Text Editor", _
    MessageBoxButton.OKCancel) = MessageBoxResult.OK Then
    Editor.Blocks.Clear()
End If
    
```

```

Private Sub New_Click(ByVal sender As System.Object, _
                    ByVal e As System.Windows.RoutedEventArgs) _
    Handles [New].Click
    If MessageBox.Show("Start a New Document?", "Rich Text Editor", _
        MessageBoxButton.OKCancel) = MessageBoxResult.OK Then
        Editor.Blocks.Clear()
    End If
End Sub
    
```

## Étape 15

Retournez au *concepteur de vues* en sélectionnant l'onglet *MainPage.xaml*, ou faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Concepteur de vues*.

Double-cliquez sur le bouton « *Open...* » et tapez ceci dans le Sub *Open\_Click* :

```

Dim OpenFileDialog As New OpenFileDialog
OpenDialog.Filter = "Saved Files (*.sav)|*.sav"
If OpenFileDialog.ShowDialog Then
    Try
        If OpenFileDialog.File.Exists Then
            Editor.Blocks.Clear()
            Read(OpenDialog.File.OpenRead, Editor.Blocks)
        End If
    Catch ex As Exception
        'Ignore Errors
    End Try
End If
    
```

```

Private Sub Open_Click(ByVal sender As System.Object, _
                    ByVal e As System.Windows.RoutedEventArgs) _
    Handles Open.Click
    Dim OpenFileDialog As New OpenFileDialog
    OpenFileDialog.Filter = "Saved Files (*.sav)|*.sav"
    If OpenFileDialog.ShowDialog Then
        Try
            If OpenFileDialog.File.Exists Then
                Editor.Blocks.Clear()
                Read(OpenDialog.File.OpenRead, Editor.Blocks)
            End If
        Catch ex As Exception
            ' Ignore Errors
        End Try
    End If
End Sub
    
```

## Étape 16

Retournez au *concepteur de vues* en sélectionnant l'onglet *MainPage.xaml*, ou faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Concepteur de vues*.

Double-cliquez sur le bouton « *Save...* » et tapez ceci dans le Sub *Save\_Click* :

```

Dim SaveDialog As New SaveFileDialog
SaveDialog.Filter = "Saved Files (*.sav)|*.sav"
If SaveDialog.ShowDialog Then
    
```



```

Try
    Using FileStream As IO.StreamWriter = _
        New IO.StreamWriter(SaveDialog.OpenFile)
        FileStream.Write(Write(Editor.Blocks))
    End Using
Catch ex As Exception
    'Ignore Errors
End Try
End If
    
```

```

Private Sub Save_Click(ByVal sender As System.Object, _
    ByVal e As System.Windows.RoutedEventArgs) _
    Handles Save.Click
    Dim SaveDialog As New SaveFileDialog
    SaveDialog.Filter = "Saved Files (*.sav)|*.sav"
    If SaveDialog.ShowDialog Then
        Try
            Using FileStream As IO.StreamWriter = _
                New IO.StreamWriter(SaveDialog.OpenFile)
                FileStream.Write(Write(Editor.Blocks))
            End Using
        Catch ex As Exception
            ' Ignore Errors
        End Try
    End If
End Sub
    
```

## Étape 17

Retournez au *concepteur de vues* en sélectionnant l'onglet *MainPage.xaml*, ou faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Concepteur de vues*.

Double-cliquez sur le bouton « *Print...* » et tapez ceci dans le Sub *Print\_Click* :

```

Dim _print As New PrintDocument
_print.DocumentName = "Rich Text Editor"
AddHandler _print.PrintPage, Sub(s As Object, args As PrintPageEventArgs)
    args.PageVisual = Editor
    args.HasMorePages = False
End Sub
AddHandler _print.EndPrint, Sub(s As Object, args As EndPrintEventArgs)
    MessageBox.Show("Document Printed", _
        "Rich Text Editor", MessageBoxButton.OK)
End Sub
_print.Print()
Editor.Focus()
    
```

```

Private Sub Print_Click(ByVal sender As System.Object, _
                        ByVal e As System.Windows.RoutedEventArgs) _
    Handles Print.Click
    Dim _print As New PrintDocument
    _print.DocumentName = "Rich Text Editor"
    AddHandler _print.PrintPage, Sub(s As Object, args As PrintPageEventArgs)
        args.PageVisual = Editor
        args.HasMorePages = False
    End Sub
    AddHandler _print.EndPrint, Sub(s As Object, args As EndPrintEventArgs)
        MessageBox.Show("Document Printed", _
                        "Rich Text Editor", MessageBoxButton.OK)
    End Sub

    _print.Print()
    Editor.Focus()
End Sub
    
```

## Étape 18

Retournez au *concepteur de vues* en sélectionnant l'onglet *MainPage.xaml*, ou faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Concepteur de vues*.

Double-cliquez sur le bouton « *Bold* » et tapez ceci dans le Sub *Bold\_Click* :

```

If Editor.Selection.Text.Length > 0 Then ' Text Selected
    If TypeOf Editor.Selection.GetPropertyvalue(Run.FontWeightProperty) Is FontWeight _
        AndAlso DirectCast(Editor.Selection.GetPropertyvalue( _
            Run.FontWeightProperty), FontWeight) = FontWeights.Normal Then
        Editor.Selection.SetPropertyvalue(Run.FontWeightProperty, FontWeights.Bold)
    Else
        Editor.Selection.SetPropertyvalue(Run.FontWeightProperty, FontWeights.Normal)
    End If
End If
Editor.Focus()
    
```

```

Private Sub Bold_Click(ByVal sender As System.Object, _
                       ByVal e As System.Windows.RoutedEventArgs) _
    Handles Bold.Click
    If Editor.Selection.Text.Length > 0 Then ' Text Selected
        If TypeOf Editor.Selection.GetPropertyvalue(Run.FontWeightProperty) Is FontWeight _
            AndAlso DirectCast(Editor.Selection.GetPropertyvalue( _
                Run.FontWeightProperty), FontWeight) = FontWeights.Normal Then
            Editor.Selection.SetPropertyvalue(Run.FontWeightProperty, FontWeights.Bold)
        Else
            Editor.Selection.SetPropertyvalue(Run.FontWeightProperty, FontWeights.Normal)
        End If
    End If
    Editor.Focus()
End Sub
    
```

## Étape 19

Retournez au *concepteur de vues* en sélectionnant l'onglet *MainPage.xaml*, ou faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Concepteur de vues*.

Double-cliquez sur le bouton « *Italic* » et tapez ceci dans le Sub *Italic\_Click* :

```

If Editor.Selection.Text.Length > 0 Then ' Text Selected
    If TypeOf Editor.Selection.GetPropertyvalue(Run.FontStyleProperty) Is FontStyle _
        AndAlso DirectCast(Editor.Selection.GetPropertyvalue( _
            Run.FontStyleProperty), FontStyle) = FontStyles.Normal Then
        Editor.Selection.SetPropertyvalue(Run.FontStyleProperty, FontStyles.Italic)
    Else
        Editor.Selection.SetPropertyvalue(Run.FontStyleProperty, FontStyles.Normal)
    End If
End If
Editor.Focus()
    
```

```

Private Sub Italic_Click(ByVal sender As System.Object, _
    ByVal e As System.Windows.RoutedEventArgs) _
    Handles Italic.Click
    If Editor.Selection.Text.Length > 0 Then ' Text Selected
        If TypeOf Editor.Selection.GetPropertyvalue(Run.FontStyleProperty) Is FontStyle _
            AndAlso DirectCast(Editor.Selection.GetPropertyvalue( _
                Run.FontStyleProperty), FontStyle) = FontStyles.Normal Then
            Editor.Selection.SetPropertyvalue(Run.FontStyleProperty, FontStyles.Italic)
        Else
            Editor.Selection.SetPropertyvalue(Run.FontStyleProperty, FontStyles.Normal)
        End If
    End If
    Editor.Focus()
End Sub
    
```

## Étape 20

Retournez au *concepteur de vues* en sélectionnant l'onglet *MainPage.xaml*, ou faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Concepteur de vues*.

Double-cliquez sur le bouton « *Underline* » et tapez ceci dans le Sub *Underline\_Click* :

```

If Editor.Selection.Text.Length > 0 Then ' Text Selected
    If Editor.Selection.GetPropertyvalue(Run.TextDecorationsProperty) Is Nothing Then
        Editor.Selection.SetPropertyvalue(Run.TextDecorationsProperty, TextDecorations.Underline)
    Else
        Editor.Selection.SetPropertyvalue(Run.TextDecorationsProperty, Nothing)
    End If
End If
Editor.Focus()
    
```

```

Private Sub Underline_Click(ByVal sender As System.Object, _
    ByVal e As System.Windows.RoutedEventArgs) _
    Handles Underline.Click
    If Editor.Selection.Text.Length > 0 Then ' Text Selected
        If Editor.Selection.GetPropertyvalue(Run.TextDecorationsProperty) Is Nothing Then
            Editor.Selection.SetPropertyvalue(Run.TextDecorationsProperty, TextDecorations.Underline)
        Else
            Editor.Selection.SetPropertyvalue(Run.TextDecorationsProperty, Nothing)
        End If
    End If
    Editor.Focus()
End Sub
    
```

## Étape 21

Retournez au *concepteur de vues* en sélectionnant l'onglet *MainPage.xaml*, ou faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Concepteur de vues*.

Double-cliquez sur le bouton « *Time Date* » et tapez ceci dans le Sub *TimeDate\_Click* :

```
Editor.Selection.Text = Format(Now, "HH:mm dd/MM/yyyy")
Editor.Focus()
```

```
Private Sub TimeDate_Click(ByVal sender As System.Object, _
    ByVal e As System.Windows.RoutedEventArgs) _
    Handles TimeDate.Click
    Editor.Selection.Text = Format(Now, "HH:mm dd/MM/yyyy")
    Editor.Focus()
End Sub
```

## Étape 22

Retournez au *concepteur de vues* en sélectionnant l'onglet *MainPage.xaml*, ou faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Concepteur de vues*.

Double-cliquez sur le premier *ComboBox (Font)* ou sur celui à l'extrême gauche et tapez ceci dans le Sub *Font\_SelectionChanged* :

```
If Editor IsNot Nothing AndAlso Editor.Selection.Text.Length > 0 Then ' Text Selected
    Editor.Selection.SetPropertyValue(Run.FontFamilyProperty, _
        New FontFamily(CType(Font.SelectedItem, ComboBoxItem).Tag))
    Editor.Focus()
End If
```

```
Private Sub Font_SelectionChanged(ByVal sender As System.Object, _
    ByVal e As System.Windows.Controls.SelectionChangedEventArgs) _
    Handles Font.SelectionChanged
    If Editor IsNot Nothing AndAlso Editor.Selection.Text.Length > 0 Then ' Text Selected
        Editor.Selection.SetPropertyValue(Run.FontFamilyProperty, _
            New FontFamily(CType(Font.SelectedItem, ComboBoxItem).Tag))
        Editor.Focus()
    End If
End Sub
```

## Étape 23

Retournez au *concepteur de vues* en sélectionnant l'onglet *MainPage.xaml*, ou faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Concepteur de vues*.

Double-cliquez sur le second *ComboBox (Size)* et tapez ceci dans le Sub *Size\_SelectionChanged* :

```
If Editor IsNot Nothing AndAlso Editor.Selection.Text.Length > 0 Then ' Text Selected
    Editor.Selection.SetPropertyValue(Run.FontSizeProperty, _
        Double.Parse(TryCast(Size.SelectedItem, ComboBoxItem).Tag))
    Editor.Focus()
End If
```

```

Private Sub Size_SelectionChanged(ByVal sender As System.Object, _
    ByVal e As System.Windows.Controls.SelectionChangedEventArgs) _
    Handles Size.SelectionChanged
    If Editor IsNot Nothing AndAlso Editor.Selection.Text.Length > 0 Then ' Text Selected
        Editor.Selection.SetPropertyValue(Run.FontSizeProperty, _
            Double.Parse(TryCast(Size.SelectedItem, ComboBoxItem).Tag))
        Editor.Focus()
    End If
End Sub
    
```

## Étape 24

Retournez au *concepteur de vues* en sélectionnant l'onglet *MainPage.xaml*, ou faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Concepteur de vues*.

Double-cliquez sur le troisième ComboBox (Colour) et tapez ceci dans le Sub *Colour\_SelectionChanged* :

```

If Editor IsNot Nothing AndAlso Editor.Selection.Text.Length > 0 Then ' Text Selected
    Dim _colour As String = CType(Colour.SelectedItem, ComboBoxItem).Tag
    Dim _brush As New SolidColorBrush(Color.FromArgb(
        Byte.Parse(_colour.Substring(0, 2), System.Globalization.NumberStyles.HexNumber),
        Byte.Parse(_colour.Substring(2, 2), System.Globalization.NumberStyles.HexNumber),
        Byte.Parse(_colour.Substring(4, 2), System.Globalization.NumberStyles.HexNumber),
        Byte.Parse(_colour.Substring(6, 2), System.Globalization.NumberStyles.HexNumber)))
    Editor.Selection.SetPropertyValue(Run.ForegroundProperty, _brush)
    Editor.Focus()
End If
    
```

```

Private Sub Colour_SelectionChanged(ByVal sender As System.Object, _
    ByVal e As System.Windows.Controls.SelectionChangedEventArgs) _
    Handles Colour.SelectionChanged
    If Editor IsNot Nothing AndAlso Editor.Selection.Text.Length > 0 Then ' Text Selected
        Dim _colour As String = CType(Colour.SelectedItem, ComboBoxItem).Tag
        Dim _brush As New SolidColorBrush(Color.FromArgb(
            Byte.Parse(_colour.Substring(0, 2), System.Globalization.NumberStyles.HexNumber),
            Byte.Parse(_colour.Substring(2, 2), System.Globalization.NumberStyles.HexNumber),
            Byte.Parse(_colour.Substring(4, 2), System.Globalization.NumberStyles.HexNumber),
            Byte.Parse(_colour.Substring(6, 2), System.Globalization.NumberStyles.HexNumber)))
        Editor.Selection.SetPropertyValue(Run.ForegroundProperty, _brush)
        Editor.Focus()
    End If
End Sub
    
```

## Étape 25

Enregistrez le projet maintenant que vous avez terminé l'application Silverlight. Sélectionnez *Debug* ensuite *Démarrer le débogage* ou cliquez sur *Démarrer le débogage* :

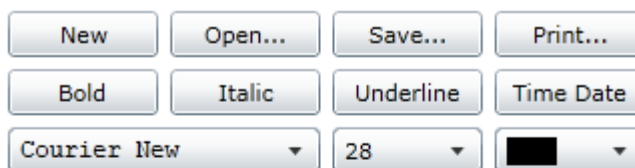


Une fois que vous l'aurez fait, ce qui suit apparaîtra dans une nouvelle fenêtre du navigateur :



## Étape 26

Vous pouvez saisir du texte dans le *RichTextArea*, et ensuite appliquez la mise en forme, les polices, les tailles et les couleurs - ensuite *Enregistrer (Save)* le fichier pour l'*Ouvrir (Open)* plus tard :



13:39 03/01/2010

**Bold**

*Italic*

Underline

Colour

**Arial Black**

Courier New 28pt

## Étape 27

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 un éditeur de texte riche très simple avec quelques-unes des fonctionnalités communes prises en charge par Silverlight. Voyez si vous pouvez ajouter davantage de fonctionnalités à cette application telles que *l'alignement de texte (Text Alignment)*.

## Liens

**Télécharger le code source du tutoriel.**

**Démonstration en ligne 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.