

Create WebTextEditor Field Type in Microsoft SharePoint 2007

This white paper describes the techniques and walkthrough about using WebTextEditor as a custom field type using WebTextEditor.

Prerequisites

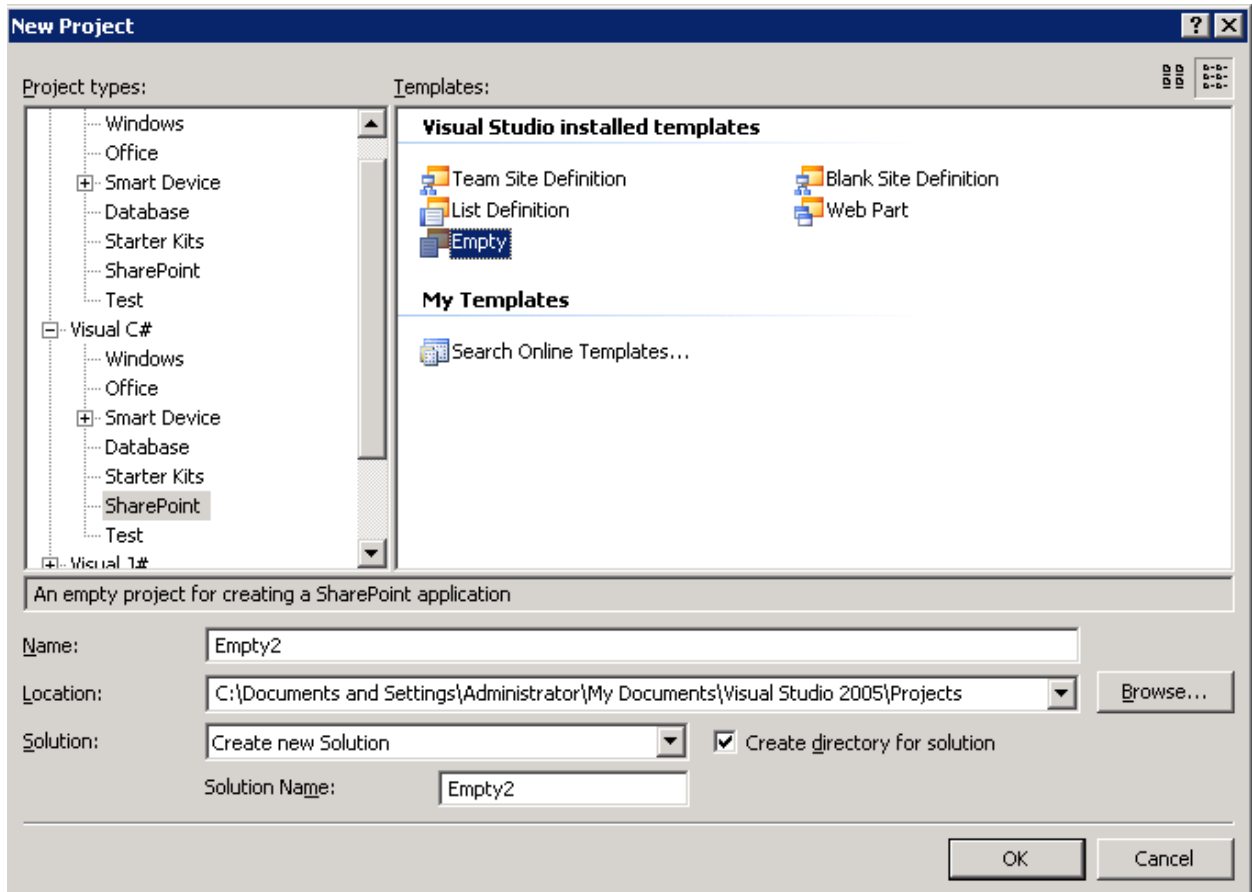
The following are the prerequisites development environments before proceeding further.

- Windows 2003 Server with IIS installed.
- Microsoft Office SharePoint Server 2007.
- Intersoft WebUI Studio.NET 2009 R2.

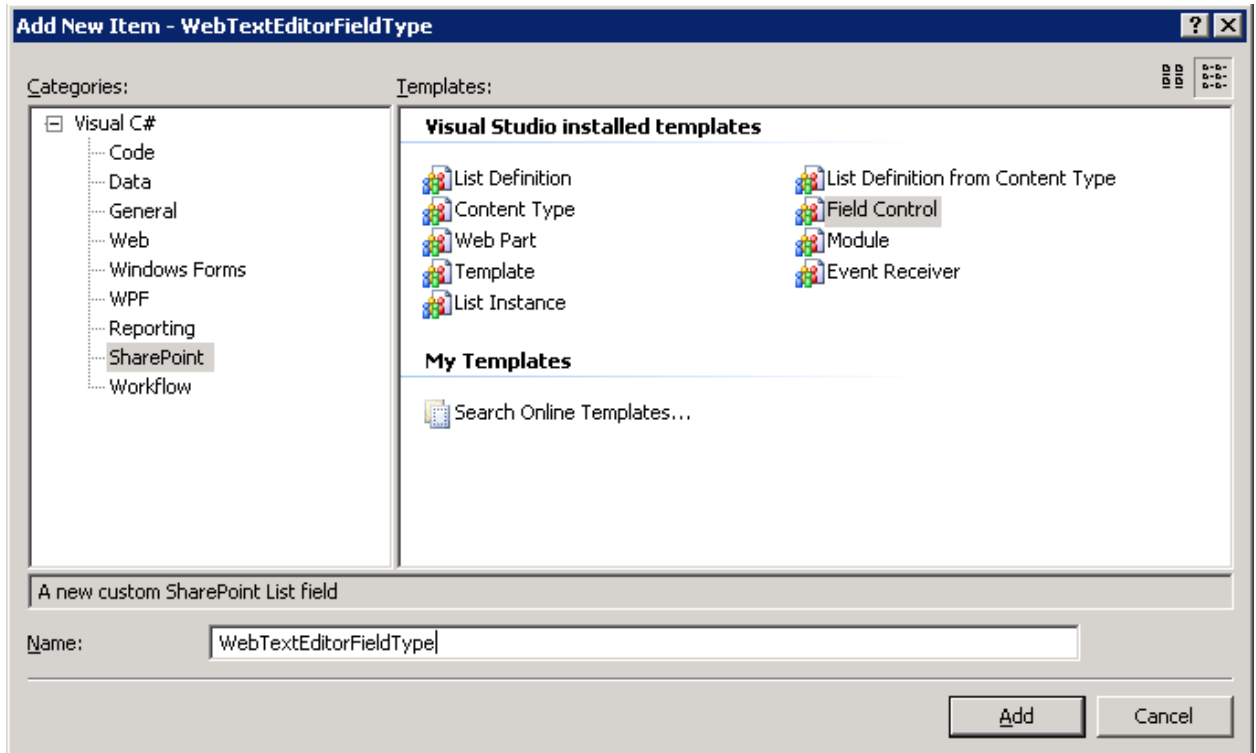
The virtual PC image of Microsoft Office SharePoint Server 2007 can be obtained [here](#). Click [here](#) to download the 30-day fully functional WebUI Studio 2009 R2.

Preparing your new SharePoint Project

1. Run your Visual Studio 2008. Create an empty SharePoint template. Please go to *New Project > Visual C# > SharePoint > Empty template.*



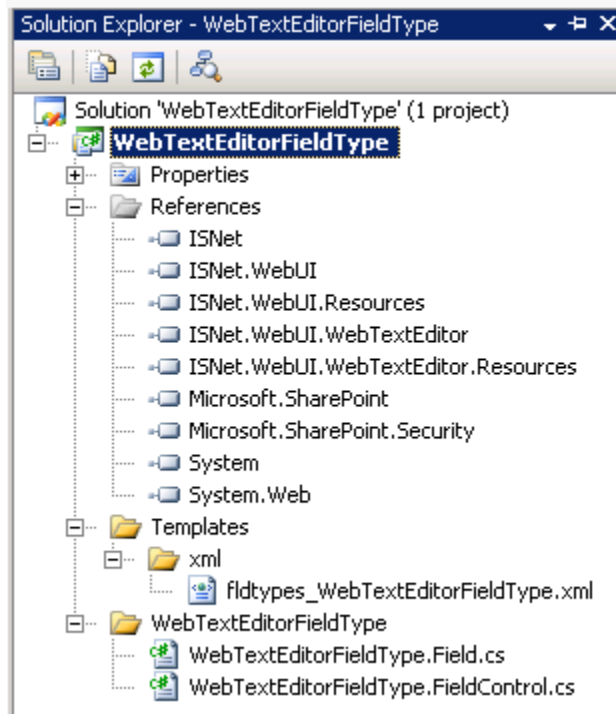
2. Give it a name. In this sample, it is named **WebTextEditorFieldType**.
3. Press Ok button and you'll see the empty project. On the Solution Explorer, right-click on the project name, select *Add*, and choose *New Item*.
4. Choose *Field Control Templates*.
5. In the *Name* field, enter the exact same name as your project name. For example: **WebTextEditorFieldType**.



6. Click *Add*. Two files will be created automatically: ***WebTextEditorFieldType.Field.cs*** and ***WebTextEditorFieldType.FieldControl.cs*** inside the ***WebTextEditorFieldType*** folder.

7. Next, add reference to all required assemblies. Right click on the References in Solution Explorer and navigate to:

- %Installation Path%\Intersoft Solutions\WebUI Studio for ASP.NET\ WebUI.NET Framework 3.0\Bin. Add **ISNet.dll**, **ISNet.WebUI.dll**, and **ISNet.WebUI.Resources.dll**.
- %Installation Path%\Intersoft Solutions\WebUI Studio for ASP.NET\ WebTextEditor.NET 1.1\Bin. Add **ISNet.WebUI.WebTextEditor.dll** and **ISNet.WebUI.WebTextEditor.Resources.dll**.



Configuring the Control

1. Open the *WebTextEditorFieldType.Field.cs* file. By default, the class is inherited from *SPFieldText*.
2. Open *WebTextEditorFieldType.FieldControl.cs*. By default, the class is inherited from *TextField*. Since we won't use any standard .NET controls, the class must be changed to *BaseFieldControl*.
3. Override the *CreateChildControls* method the *Value* property.

```
using System;
using System.Runtime.InteropServices;

using Microsoft.SharePoint;
using Microsoft.SharePoint.WebControls;
using ISNet.WebUI.WebTextEditor;
using System.Web.UI.WebControls;

namespace WebTextEditorFieldType
{
    // TODO: Replace, as needed, "TextField" with some other class
```

```

derived from Microsoft.SharePoint.WebControls.BaseFieldControl.
[CLSCompliant(false)]
[Guid("db82fa4d-4470-4386-a4bd-1661d32300aa")]
public class WebTextEditorFieldTypeFieldControl : BaseFieldControl
{
    protected WebTextEditor WebEditor;

    protected override void CreateChildControls()
    {
        if (this.ControlMode == SPControlMode.Edit ||
this.ControlMode == SPControlMode.New)
        {
            // Make sure inherited child controls are completely
rendered.
            base.CreateChildControls();

            this.WebEditor = new WebTextEditor();

            if (!this.Page.IsPostBack)
            {
                if (this.ControlMode == SPControlMode.New)
                {
                    this.WebEditor.Content = "";
                } // end assign default value in New mode

                this.WebEditor.Width = Unit.Pixel(500);
                this.WebEditor.Height = Unit.Pixel(600);
                this.WebEditor.ImagesDirectory = "http://budianto-
pc/CommonLibrary/Images/WebTextEditor/";
                this.WebEditor.ScriptDirectory = "http://budianto-
pc/CommonLibraryTemp08/WebTextEditor/V1_0_1000/";
                this.WebEditor.SharedScriptDirectory =
"http://budianto-pc/CommonLibrary/Shared/";
                this.WebEditor.SharedScriptDirectory =
"http://budianto-pc/CommonLibrary/Shared/";
                this.WebEditor.WebDesktopScriptDirectory =
"http://budianto-pc/CommonLibraryTemp08/WebDesktop/V3_0_7200/";
                this.WebEditor.ToolBarSettings.ToolBarMode =
WebTextEditorToolBarMode.Minimal;
            } // end if this is not a postback

            Controls.Add(this.WebEditor);
        }
    }

    public override object Value
    {
        get
        {
            //EnsureChildControls();
            return this.WebEditor.Content;
        }
        set
        {
            //EnsureChildControls();
            this.WebEditor.Content = (String)value;
        }
    }
}

```

```
}  
  }  
}  
}
```

The main concept is to render WebTextEditor when user enters the Edit or the New mode.

Creating the Field Type Definition

By default, Field type definition file (*fldtypes_CustomWeb.xml*) is created inside the Templates/xml folder. Build the project and Visual Studio will automatically generate the content, but not the *RenderPattern*.

Add *RenderPattern* definition under the `</FieldType>`.

```
<RenderPattern Name="DisplayPattern" DisplayName="DisplayPattern">  
</RenderPattern>
```

If you want to store more than 255 characters in your control, you should change the *ParentType* of your control to Note.

Your final XML file should look like this.

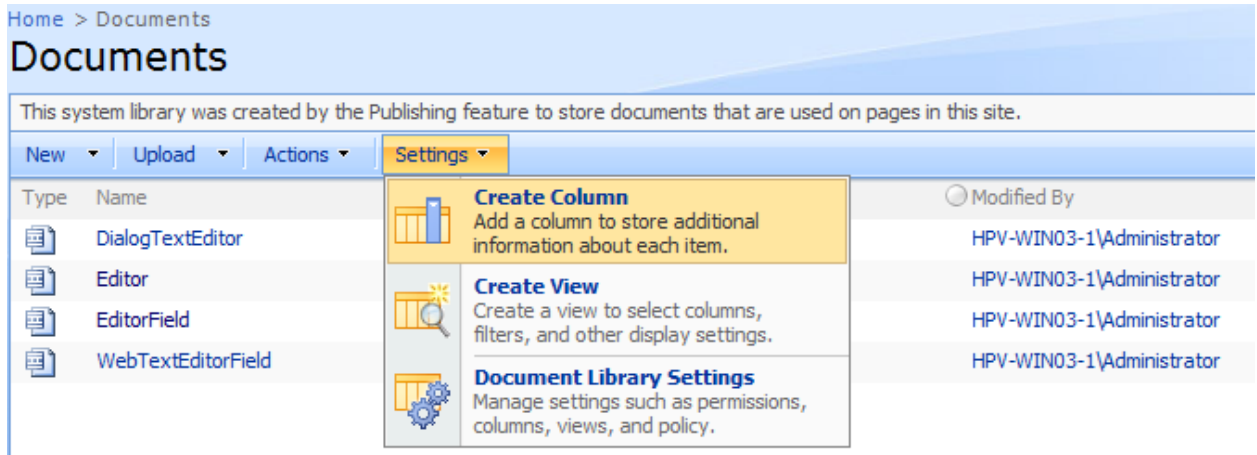
```
<?xml version="1.0" encoding="utf-8"?>  
<FieldTypes>  
  <RenderPattern Name="DisplayPattern" DisplayName="DisplayPattern">  
  </RenderPattern>  
  <FieldType>  
    <Field Name="TypeName">WebTextEditorFieldTypeField</Field>  
    <Field Name="TypeDisplayName">WebTextEditorFieldTypeField</Field>  
    <Field Name="TypeShortDescription">WebTextEditorFieldTypeField</Field>  
    <Field Name="ParentType">Text</Field>  
    <Field Name="UserCreatable">TRUE</Field>  
    <Field Name="FieldTypeClass">01041dd8-3c5e-496c-8d50-5ca20669a72b</Field>  
  </FieldType>  
</FieldTypes>
```

Deploy and Create Column

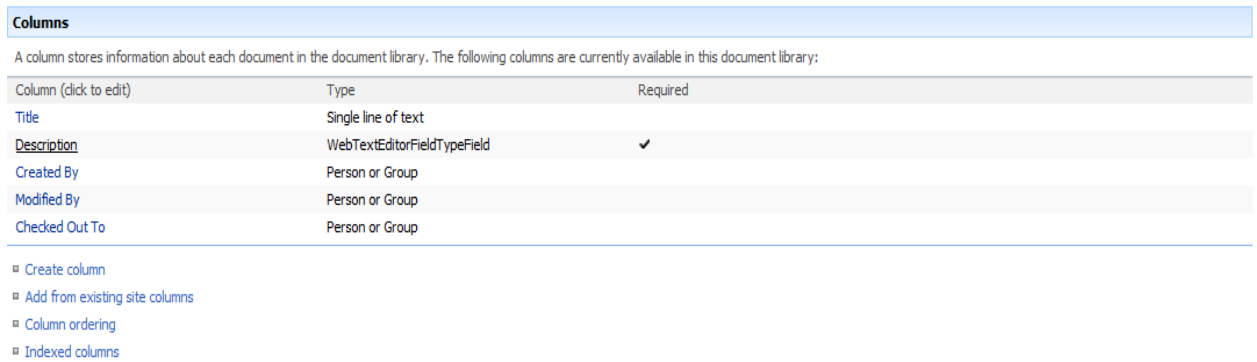
Now, you are ready to deploy your solution. Simply click on *Deploy* from your solution and the field will be added to SharePoint directly.

1. Go to your local Microsoft SharePoint page. Eg, <http://hpv-win03-1/Pages/Default.aspx>

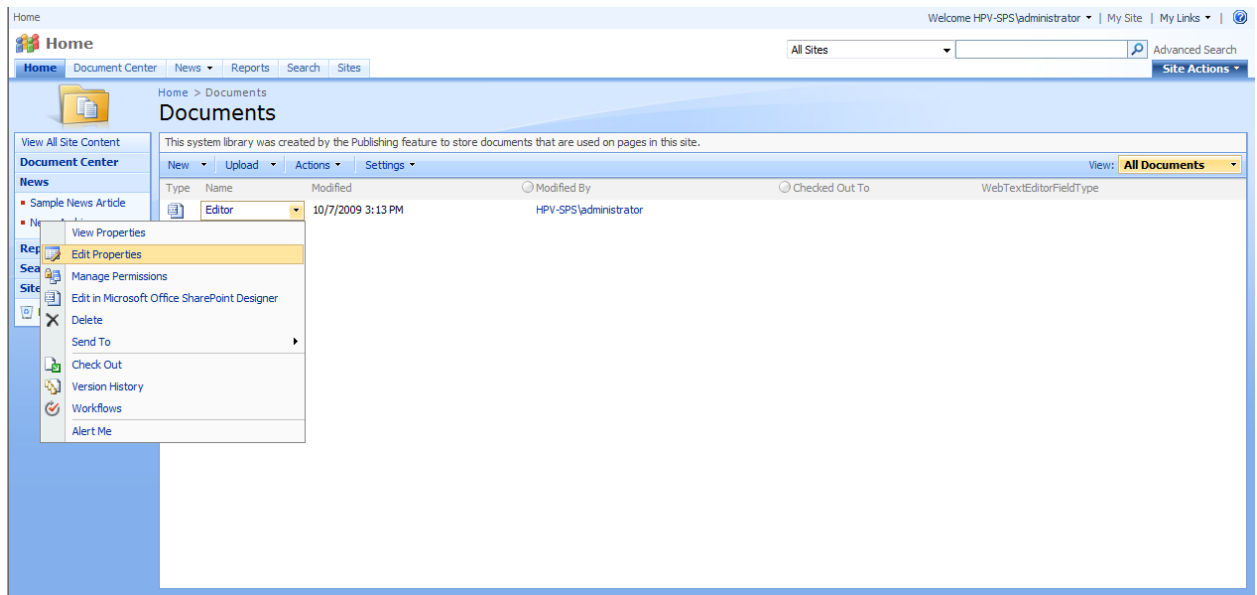
- Go to **View All Site Content > Document > Settings > CreateColumn**.



- Enter the column name and set the type to *“WebTextEditorFieldType”* (your custom template name) and hit OK. If you go to *“Document Library Settings”*, you will see something like this.



- When you open your list item in a new form, it will show you:



Documents: Editor

OK Cancel

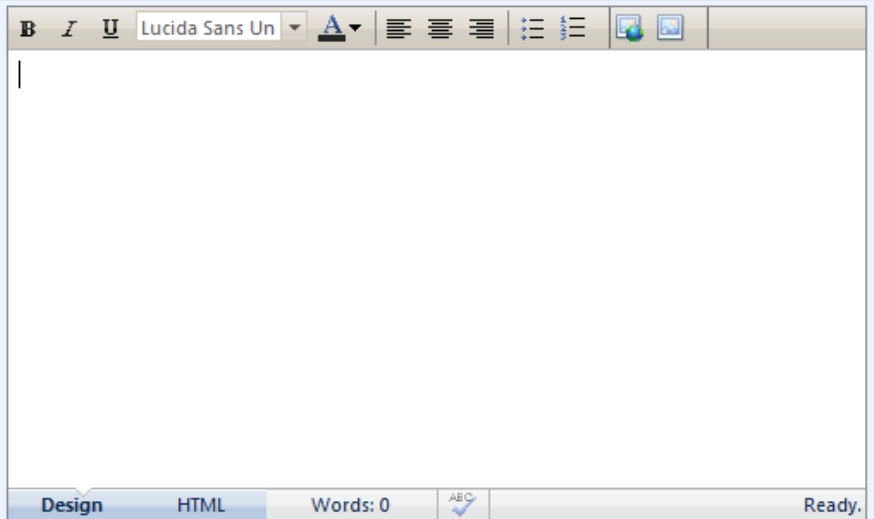
X Delete Item | Spelling...

* indicates a required field

Name *

Editor .aspx

Description *



The image shows a rich text editor interface. The toolbar includes icons for bold (B), italic (I), underline (U), font color (A), background color (■), bulleted list, numbered list, link, and unlink. The font face is set to 'Lucida Sans Un'. The main content area is empty with a vertical cursor. At the bottom, there are tabs for 'Design' and 'HTML', a 'Words: 0' counter, a spelling icon, and a 'Ready.' status indicator.

Open this Web Part Page in maintenance view to delete problem Web Parts and remove personal settings.
Open Web Part Page in maintenance view