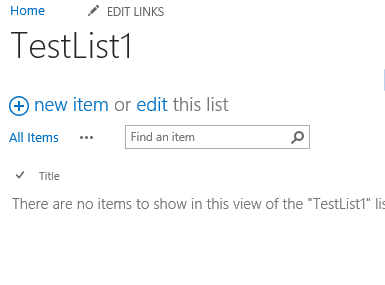
# How to use SPPageStatusSetter to set SharePoint page status

## Introduction

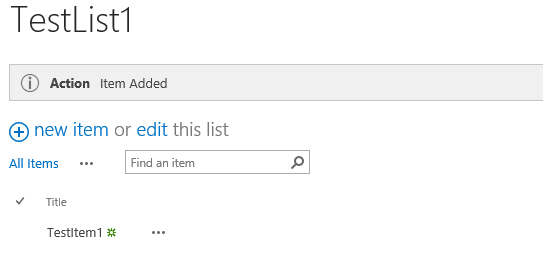
This sample demonstrates how to use SPPageStatusSetter to insert page status for SharePoint. The sample includes two scenarios: 1) add page status for an application page. 2) Add page status from a list event receiver.

## Running the Sample

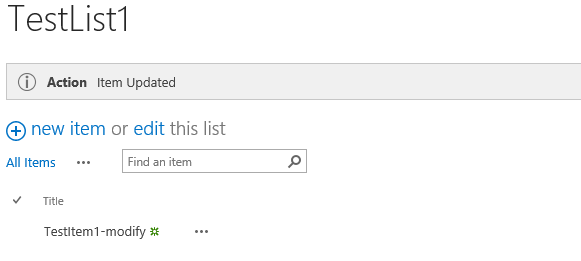
1. Deploy the demo project.
2. Suppose we have the following list:



Click the new item link to add a new item and then you will see the page as shown below:



Select one of the list item and modify it. Then you will see the page as shown below:



1. Validation is finished.

## Using the Code

1. Start **Visual Studio**.
2. On the **File** menu, click **New**, and then click **Project**
3. In the Installed Templates section of the **New Project** dialog box, expand either Visual Basic or Visual C#, expand **SharePoint**, and then click **Empty SharePoint Project**
4. Right-click the project to add the desired folder

"**Add**" >>"**SharePoint Layouts Mapped Folder**".

1. Right-click the "**Layouts**" folder then add a new folder named " SharePointSetPageStatus", then add a new Page named "Test.aspx".
2. The Test page is used to demo how to add page status to an application page.
3. Next, we will begin to demo how to add page status from a list event receiver. We realize this purpose through the following actions: create a web part with **SPPageStatusSetter** control in it and then add the web part to list view page from an event receiver along with status message dynamically.
4. Add a web part to the project. In the web part, we will add a **SPPageStatusSetter** control to it and define a custom Message property. The Message property is used to store the status bar message received from Event Receiver. The code resembles the following:

|  |
| --- |
| -Code block start-  --C# code snippet start--  [ToolboxItemAttribute(false)]  public class StatusBarWebPart : WebPart  {  SPPageStatusSetter statusBar;  string strMessage;  public StatusBarWebPart()  {  this.Message = string.Empty;  }  [Category("Custom Properties")]  [Browsable(false)]  public string Message  {  get  {  return strMessage;  }  set  {  strMessage = value;  }  }  protected override void CreateChildControls()  {  statusBar = new SPPageStatusSetter();  statusBar.AddStatus("Action", Message, SPPageStatusColor.Blue);    if (!string.IsNullOrEmpty(Message))  {  Controls.Add(statusBar);  }  }  protected override void RenderContents(HtmlTextWriter writer)  {  writer.Write("Status Bar demo");  RenderChildren(writer);  }  }  --C# code snippet end--  --VB code snippet start--  <ToolboxItemAttribute(false)> \_  Public Class StatusBarWebPart  Inherits WebPart  Private statusBar As SPPageStatusSetter  Private strMessage As String  Public Sub New()  Me.Message = String.Empty  End Sub  <Category("Custom Properties")> \_  <Browsable(False)> \_  Public Property Message() As String  Get  Return strMessage  End Get  Set(value As String)  strMessage = value  End Set  End Property  Protected Overrides Sub CreateChildControls()  statusBar = New SPPageStatusSetter()  statusBar.AddStatus("Action", Message, SPPageStatusColor.Blue)  If Not String.IsNullOrEmpty(Message) Then  Controls.Add(statusBar)  End If  End Sub  Protected Overrides Sub RenderContents(writer As HtmlTextWriter)  writer.Write("Status Bar demo")  RenderChildren(writer)  End Sub  End Class  --VB code snippet end--  -Code block end- |

1. Add an event receiver to the project. For demo purpose, we will handle the list item’s ItemAdded event and ItemUpdated event. Create an instance of the web part, then we will set custom status bar message to the Message property and then add (or update) the web part to the list page.

|  |
| --- |
| -Code block start-  --C# code snippet start--  /// <summary>  /// List Item Events.  /// </summary>  public class PageStatusEventReceiver : SPItemEventReceiver  {  StatusBarWebPart.StatusBarWebPart statusBarWebPart;  /// <summary>  /// An item was added.  /// </summary>  public override void ItemAdded(SPItemEventProperties properties)  {  base.ItemAdded(properties);  SPWeb web = properties.Web;  string oUrl = properties.List.DefaultViewUrl;  Microsoft.SharePoint.WebPartPages.SPLimitedWebPartManager coll = web.GetLimitedWebPartManager(oUrl, System.Web.UI.WebControls.WebParts.PersonalizationScope.Shared);  if (coll.WebParts.Count > 1)  {  statusBarWebPart = (StatusBarWebPart.StatusBarWebPart)coll.WebParts[1];  if (statusBarWebPart != null)  {  statusBarWebPart.Message = "Item Added";  coll.SaveChanges(statusBarWebPart);  }  }  else  {  statusBarWebPart = new StatusBarWebPart.StatusBarWebPart();  statusBarWebPart.Message = "Item Added";  statusBarWebPart.Hidden = true;  coll.AddWebPart(statusBarWebPart, "Left", 1);  coll.SaveChanges(statusBarWebPart);  }  }    /// <summary>  /// An item was updated.  /// </summary>  public override void ItemUpdated(SPItemEventProperties properties)  {  base.ItemUpdated(properties);  SPWeb web = properties.Web;  string oUrl = properties.List.DefaultViewUrl;  Microsoft.SharePoint.WebPartPages.SPLimitedWebPartManager coll = web.GetLimitedWebPartManager(oUrl, System.Web.UI.WebControls.WebParts.PersonalizationScope.Shared);  if (coll.WebParts.Count > 1)  {  statusBarWebPart = (StatusBarWebPart.StatusBarWebPart)coll.WebParts[1];  if (statusBarWebPart != null)  {  statusBarWebPart.Message = "Item Updated";  coll.SaveChanges(statusBarWebPart);  }  }  else  {  statusBarWebPart = new StatusBarWebPart.StatusBarWebPart();  statusBarWebPart.Message = "Item Updated";  statusBarWebPart.Hidden = true;  coll.AddWebPart(statusBarWebPart, "Left", 1);  coll.SaveChanges(statusBarWebPart);  }  }  }  --C# code snippet end--  --VB code snippet start--  Public Class PageStatusEventReceiver  Inherits SPItemEventReceiver  Private statusBarWebPart As StatusBarWebPart  ''' <summary>  ''' An item was added.  ''' </summary>  Public Overrides Sub ItemAdded(properties As SPItemEventProperties)  MyBase.ItemAdded(properties)  Dim web As SPWeb = properties.Web  Dim oUrl As String = properties.List.DefaultViewUrl  Dim coll As Microsoft.SharePoint.WebPartPages.SPLimitedWebPartManager = web.GetLimitedWebPartManager(oUrl, System.Web.UI.WebControls.WebParts.PersonalizationScope.[Shared])  If coll.WebParts.Count > 1 Then  statusBarWebPart = DirectCast(coll.WebParts(1), StatusBarWebPart)  If statusBarWebPart IsNot Nothing Then  statusBarWebPart.Message = "Item Added-vb"  coll.SaveChanges(statusBarWebPart)  End If  Else  statusBarWebPart = New StatusBarWebPart()  statusBarWebPart.Message = "Item Added-vb"  statusBarWebPart.Hidden = True  coll.AddWebPart(statusBarWebPart, "Left", 1)  coll.SaveChanges(statusBarWebPart)  End If  End Sub  ''' <summary>  ''' An item was updated.  ''' </summary>  Public Overrides Sub ItemUpdated(properties As SPItemEventProperties)  MyBase.ItemUpdated(properties)  Dim web As SPWeb = properties.Web  Dim oUrl As String = properties.List.DefaultViewUrl  Dim coll As Microsoft.SharePoint.WebPartPages.SPLimitedWebPartManager = web.GetLimitedWebPartManager(oUrl, System.Web.UI.WebControls.WebParts.PersonalizationScope.[Shared])  If coll.WebParts.Count > 1 Then  statusBarWebPart = DirectCast(coll.WebParts(1), StatusBarWebPart)  If statusBarWebPart IsNot Nothing Then  statusBarWebPart.Message = "Item Updated-vb"  coll.SaveChanges(statusBarWebPart)  End If  Else  statusBarWebPart = New StatusBarWebPart()  statusBarWebPart.Message = "Item Updated-vb"  statusBarWebPart.Hidden = True  coll.AddWebPart(statusBarWebPart, "Left", 1)  coll.SaveChanges(statusBarWebPart)  End If  End Sub  End Class  --VB code snippet end--  -Code block end- |

1. Now you can build and debug it.

## More Information

SPPageStatusSetter class  
<http://msdn.microsoft.com/en-us/library/office/microsoft.sharepoint.webcontrols.sppagestatussetter(v=office.15).aspx>  
SPLimitedWebPartManager class  
<http://msdn.microsoft.com/en-us/library/office/microsoft.sharepoint.webpartpages.splimitedwebpartmanager(v=office.15).aspx>  
SPItemEventReceiver class  
<http://msdn.microsoft.com/en-us/library/office/microsoft.sharepoint.spitemeventreceiver(v=office.15).aspx>  
Using SPPageStatusSetter to Show Status SharePoint 2010  
<http://www.learningsharepoint.com/2010/10/03/using-sppagestatussetter-to-show-status-sharepoint-2010/>