using System;

using System.Web.UI;

using System.Collections.Specialized;

namespace Wintellect

{

public class AutoCounter : Control, IPostBackDataHandler,

IPostBackEventHandler

{

public event EventHandler Decrement;

public event EventHandler Increment;

public event EventHandler CountChanged;

public int Count

{

get

{

int count = 0;

if (ViewState["Count"] != null)

count = (int) ViewState["Count"];

return count;

}

set { ViewState["Count"] = value; }

}

public bool LoadPostData (string postDataKey,

NameValueCollection postCollection)

{

int temp = Count;

try {

Count = Convert.ToInt32 (postCollection[postDataKey]);

}

catch (FormatException) {

Count = 0;

}

return (temp != Count);

}

public void RaisePostDataChangedEvent ()

{

if (CountChanged != null)

CountChanged (this, new EventArgs ());

}

public void RaisePostBackEvent (string eventArgument)

{

if (eventArgument == "dec") {

Count--;

if (Decrement != null)

Decrement (this, new EventArgs ());

}

else if (eventArgument == "inc") {

Count++;

if (Increment != null)

Increment (this, new EventArgs ());

}

}

protected override void Render (HtmlTextWriter writer)

{

// Output an <a> tag

writer.WriteBeginTag ("a");

writer.WriteAttribute ("href", "javascript:" +

Page.GetPostBackEventReference (this, "dec"));

writer.Write (HtmlTextWriter.TagRightChar);

// Output a less-than sign

writer.Write ("&lt;");

// Output a </a> tag

writer.WriteEndTag ("a");

// Output an <input> tag

writer.Write (" ");

writer.WriteBeginTag ("input");

writer.WriteAttribute ("type", "text");

writer.WriteAttribute ("name", UniqueID);

if (ID != null)

writer.WriteAttribute ("id", ClientID);

writer.WriteAttribute ("value", Count.ToString ());

writer.WriteAttribute ("size", "3");

writer.Write (HtmlTextWriter.TagRightChar);

writer.Write (" ");

// Output another <a> tag

writer.WriteBeginTag ("a");

writer.WriteAttribute ("href", "javascript:" +

Page.GetPostBackEventReference (this, "inc"));

writer.Write (HtmlTextWriter.TagRightChar);

// Output a greater-than sign

writer.Write ("&gt;");

// Output a </a> tag

writer.WriteEndTag ("a");

}

}

}