

Kapitel 48

Microsoft SharePoint 2007- Webparts

In diesem Kapitel:

Ausführungen	1044
Architektur	1044
Erstellung von SharePoint-Webparts	1044
ASP.NET User Controls als SharePoint Webparts	1051

SharePoint ist der Name für ein Produkt der Firma Microsoft, das der Realisierung von Intranet-, Extranet- und Internet-Webportalen dient. Kernfunktion sind Datenverwaltung, Content Management und Workflows. Informationen, die standardmäßig in SharePoint verwaltet werden können, sind Kontakte, Aufgaben, Formulare, Diskussionsbeiträge, Kalender, Nachrichten, Dokumente und Kalkulationstabellen. SharePoint ist erweiterbar, sodass ein Zugriff auf beliebige Daten möglich ist.

Ausführungen

SharePoint 2007 gibt es in folgenden Ausführungen:

- Windows SharePoint Services 2007 (WSS): kostenfreie Erweiterung für Windows Server 2003, enthalten in Windows Server 2008.
- Microsoft Office SharePoint Server 2007 (MOSS): kommerzielles Serverprodukt, das mehr Funktionen bietet als WSS (Nachfolger des SharePoint Portal Server 2003).

Windows SharePoint Services (WSS) ist eine Teilmenge von Microsoft Office SharePoint Server (MOSS).

Architektur

SharePoint basiert auf den Internet Information Services (IIS) 6.0 / 7.0 und ASP.NET 2.0 / 3.5. Zum Betrieb der Grundfunktionen sind die IIS 6.0 und ASP.NET 2.0 ausreichend. Eigene Erweiterungen können optional AJAX verwenden; dann werden die ASP.NET 2.0 AJAX Extensions oder ASP.NET 3.5 benötigt.

SharePoint verwendet alle zentralen ASP.NET-Funktionen, insbesondere die Steuerelemente, die Webpart-Infrastruktur, die Vorlagenseiten (Masterpages), das Navigationskonzept und die Konfigurationsdateien (*web.config*).

Erstellung von SharePoint-Webparts

Ein SharePoint-Webpart ist eine erweiterte Form eines ASP.NET-Webparts. Zur einfachen Erstellung eines solchen Webparts sollte man die Visual Studio Extensions for Windows SharePoint Services 3.0 installieren, die nicht zum Lieferumfang von Visual Studio gehören.

ACHTUNG Zum Redaktionsschluss dieses Buchs gab es diese Erweiterung noch nicht für Visual Studio 2008, sondern nur für Visual Studio Version 2005.

Zur Erstellung eines Webparts sollte man die Projektvorlage *Web Part* verwenden. Die Projektvorlage legt ein leeres Webpart an. Ein Webpart ist eine C#-Klasse, die von `System.Web.UI.WebControls.WebParts.WebPart` abgeleitet wird. Die Klasse benötigt eine GUID, die über die Annotation `System.Runtime.InteropServices.Guid` zugewiesen werden muss.

ACHTUNG Ein SharePoint Webpart kann in Visual Studio leider nur als Custom Control entwickelt werden. Dies bedeutet, dass der Web Form-Designer nicht verwendet werden kann und man komplett codebasiert arbeiten muss.

Beispiel 1

Abbildung 48.1 zeigt ein »Hello World«-Webpart, das durch die einfache Ausgabe einer Zeichenkette in der überschriebenen `Render()`-Methode entsteht.

Durch die Kompilierung erstellt Visual Studio automatisch eine SharePoint-Lösung (.wsp-Datei), welche die Assembly, eine das Webpart beschreibende XML-Datei (.webpart) und die Definition eines SharePoint-Features (feature.xml) enthält. Außerdem wird eine Batchdatei zur Installation des Features (inklusive Webpart) erstellt.

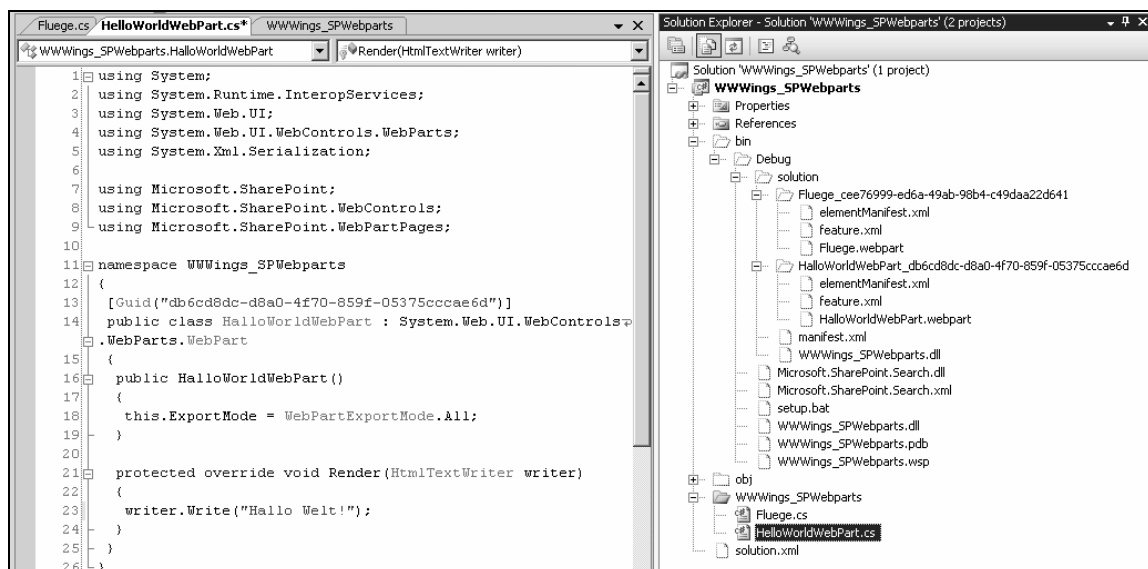


Abbildung 48.1 Beispiel für ein »Hello World«-Webpart

Beispiel 2

Das zweite Beispiel enthält ein Webpart, das durch Überschreiben der Methode `CreateChildControls()` entsteht. In der Methode werden zwei ASP.NET-Serversteuerelemente erzeugt.

```
using System;
using System.Runtime.InteropServices;
using System.Web.UI;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Serialization;

using Microsoft.SharePoint;
using Microsoft.SharePoint.WebControls;
using Microsoft.SharePoint.WebPartPages;

namespace WichtigeLinksWebPart
{
    [Guid("a6543d37-54e5-496a-8ee9-7aec7c47d9ee")]
    public class WichtigeLinksWebPart : System.Web.UI.WebControls.WebParts.WebPart
    {
        public WichtigeLinksWebPart()
        {
            this.ExportMode = WebPartExportMode.All;
        }

        protected override void Render(HtmlTextWriter writer)
        {
            base.Render(writer);
        }

        protected override void CreateChildControls()
        {
            base.CreateChildControls();

            System.Web.UI.WebControls.HyperLink h1 = new System.Web.UI.WebControls.HyperLink();
            h1.NavigateUrl = "http://www.world-wide-wings.de";
            h1.Text = "World Wide Wings-Website";

            System.Web.UI.WebControls.Literal br = new System.Web.UI.WebControls.Literal();
            br.Text = "<br>";

            System.Web.UI.WebControls.HyperLink h2 = new System.Web.UI.WebControls.HyperLink();
            h2.NavigateUrl = "http://www.HolgerSchwichtenberg.de";
            h2.Text = "Website des Autors";

            this.Controls.Add(h1);
            this.Controls.Add(br);
            this.Controls.Add(h2);
        }
    }
}
```

Listing 48.1 Beispiel für ein Webpart

Konfiguration

Eigenschaften wie *Name*, *Beschreibungstext* und *Versionsnummer* zu *Webpart* und *Feature* setzt man in den Projekteigenschaften.

Da SharePoint-Erweiterungen signiert sein sollten, erhält ein SharePoint-Projekt automatisch einen Schlüssel (*Temporary.snk*) und einen entsprechenden Eintrag in der AssemblyInfo-Datei.

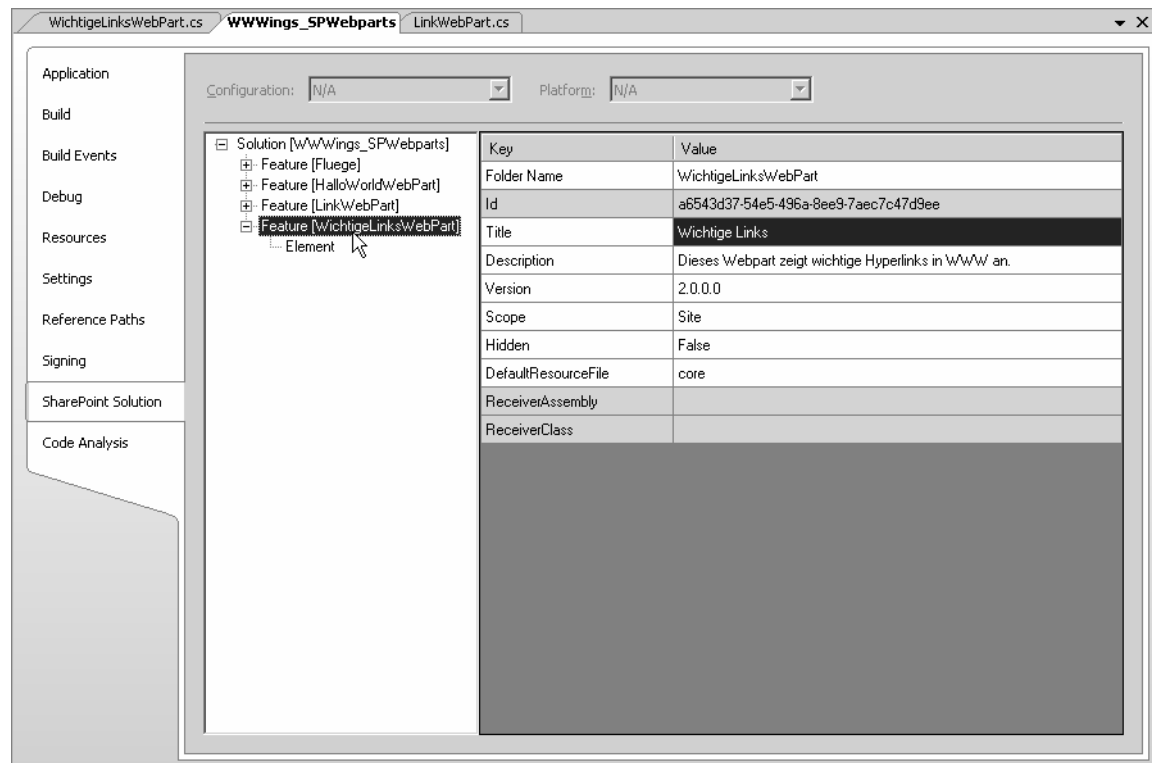


Abbildung 48.2 Festlegung der Eigenschaften für das SharePoint-Feature

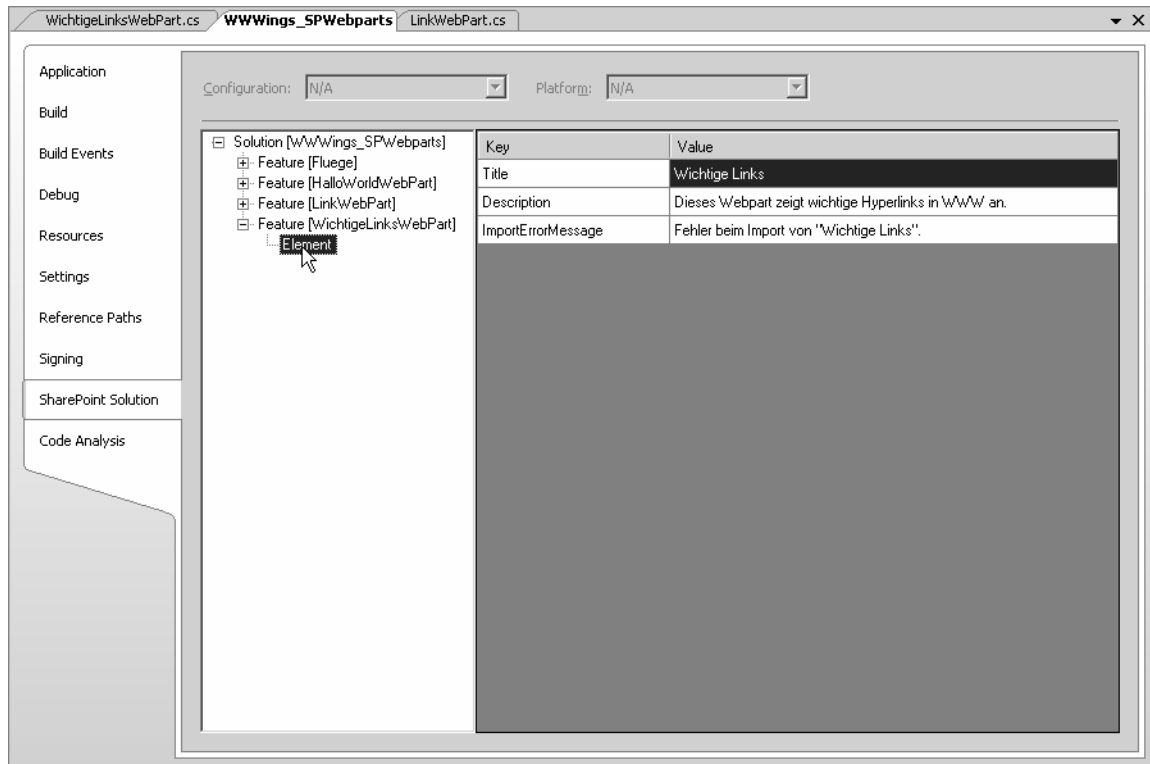


Abbildung 48.3 Festlegung der Eigenschaften für das Webpart

Installation

Die Installation kann der Entwickler über die Batchdatei oder einfach durch den Menüpunkt *Build/Deploy Solution* anstoßen. Die Installation einer WSP-Datei in SharePoint besteht aus dem Hinzufügen der Datei zur Web Form (*Adding Solution* in Abbildung 48.4) und dann der Zuweisung zu einem Webserver (*Deploying Solution* in der gleichen Abbildung). Danach muss man die Features aktivieren.

```
----- Build started: Project: WWWings_SPWebparts, Configuration: Debug Any CPU -----
WWWings_SPWebparts -> H:\demo\WWWings_SPWebparts\bin\Debug\WWWings_SPWebparts.dll
----- Deploy started: Project: WWWings_SPWebparts, Configuration: Debug Any CPU -----
----- Generate WWWings_SPWebparts.wsp file and setup batch file-----
Creating solution ...
Operation completed successfully.

Creating setup batch file ...
Operation completed successfully.

----- Add and deploy WWWings_SPWebparts.wsp to the SharePoint -----
Adding solution ...
Operation completed successfully.

Deploying solution ...
Operation completed successfully.

----- Activate features in solution if necessary -----
Activating feature Fluege ...
Operation completed successfully.

Activating feature HalloWorldWebPart ...
Operation completed successfully.

Activating feature LinkWebPart ...
Operation completed successfully.

Restarting IIS ...
Operation completed successfully.

===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
===== Deploy: 1 succeeded, 0 failed, 0 skipped =====
|
```

Abbildung 48.4 Installationsschritte für Webparts

Verwendung der Lösung

Die erstellten Webparts erscheinen danach beim Anpassen einer SharePoint-Seite (*Websiteaktionen/Seite bearbeiten*) in der Liste der verfügbaren Webparts.

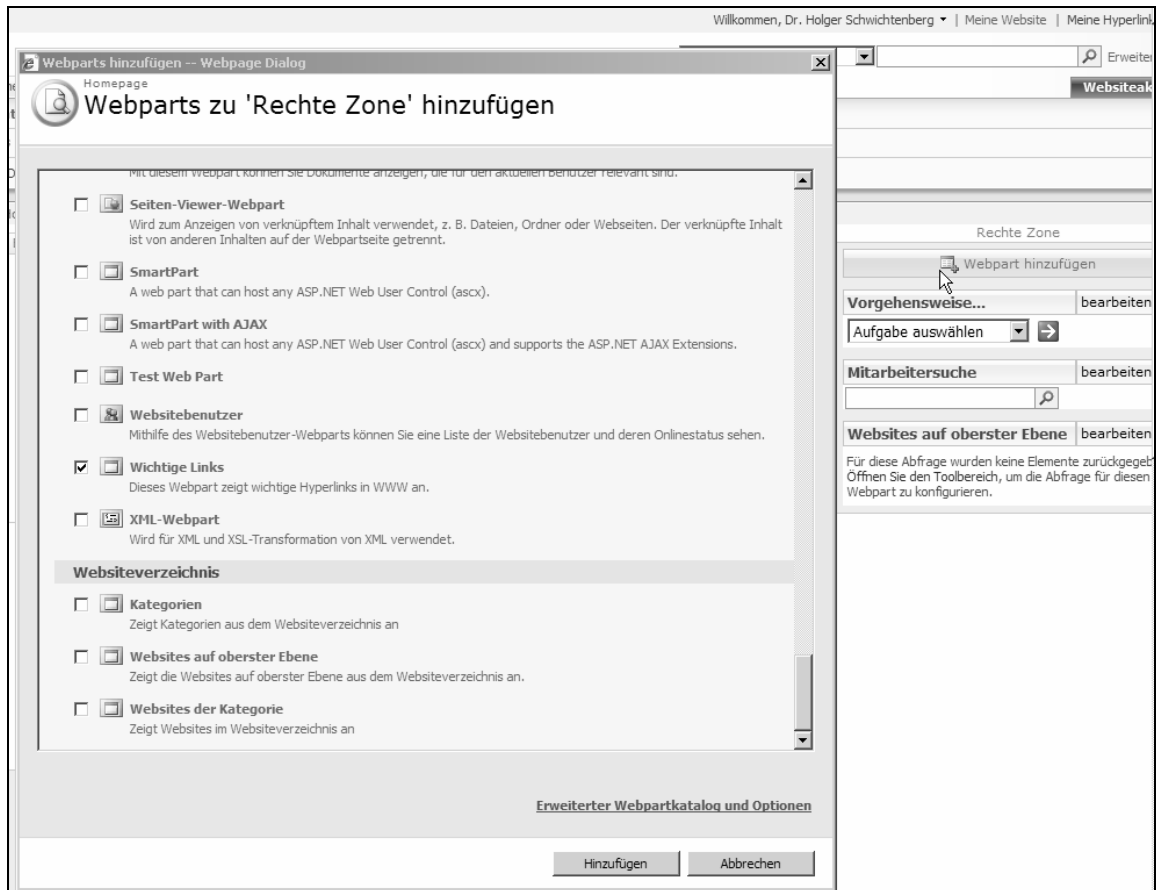


Abbildung 48.5 Auswahl des Webparts



Abbildung 48.6 Das Webpart in Betrieb

ASP.NET User Controls als SharePoint Webparts

Die rein codebasierte Entwicklung von SharePoint Webparts ist nicht komfortabel. Leider bietet Microsoft selbst keine Lösung zur Nutzung von User Controls in SharePoint. Eine solche Lösung gibt es aber als Open Source-Projekt (<http://www.codeplex.com/smartpart>) unter dem Namen *Return of the SmartPart*. Bei Return of the SmartPart handelt es sich um ein Webpart, das ein beliebiges User Control aufnehmen kann.

Diese Lösung besteht aus

- einer .NET-Assembly (DLL)
- zwei Konfigurationsdateien (.wsp und .dsp)
- einer SharePoint-Lösungsdatei (.wsp). Eine WSP-Datei ist ein Archiv, das die DLL und die zugehörigen Konfigurationsdateien enthält.

Der einfachste Weg zur Installation führt über die WSP-Datei, die mithilfe des in SharePoint mitgelieferten Kommandozeilenwerkzeugs *stsadm.exe* installiert werden kann:

```
stsadm.exe -o addSolution -filename ReturnOfSmartPartv1 2.wsp  
stsadm.exe -o deploySolution -name ReturnOfSmartPartv1_2.wsp -allcontenturls -local -allowGacDeployment
```

Danach ist das Webpart für eine Website zu aktivieren:

```
stsadm.exe -o activatefeature -n ReturnOfSmartPart -url http://server
```

Innerhalb der Website muss man dann im Dateisystem einen Ordner */UserControls* (Name ohne Leerzeichen!) im Wurzelordner des SharePoint-Webs anlegen und dort die gewünschten User Controls zusammen mit ihren Hintergrundcodedateien hineinkopieren.

Anschließend kann man in SharePoint das Webpart *SmartPart* auswählen (Abbildung 48.7).

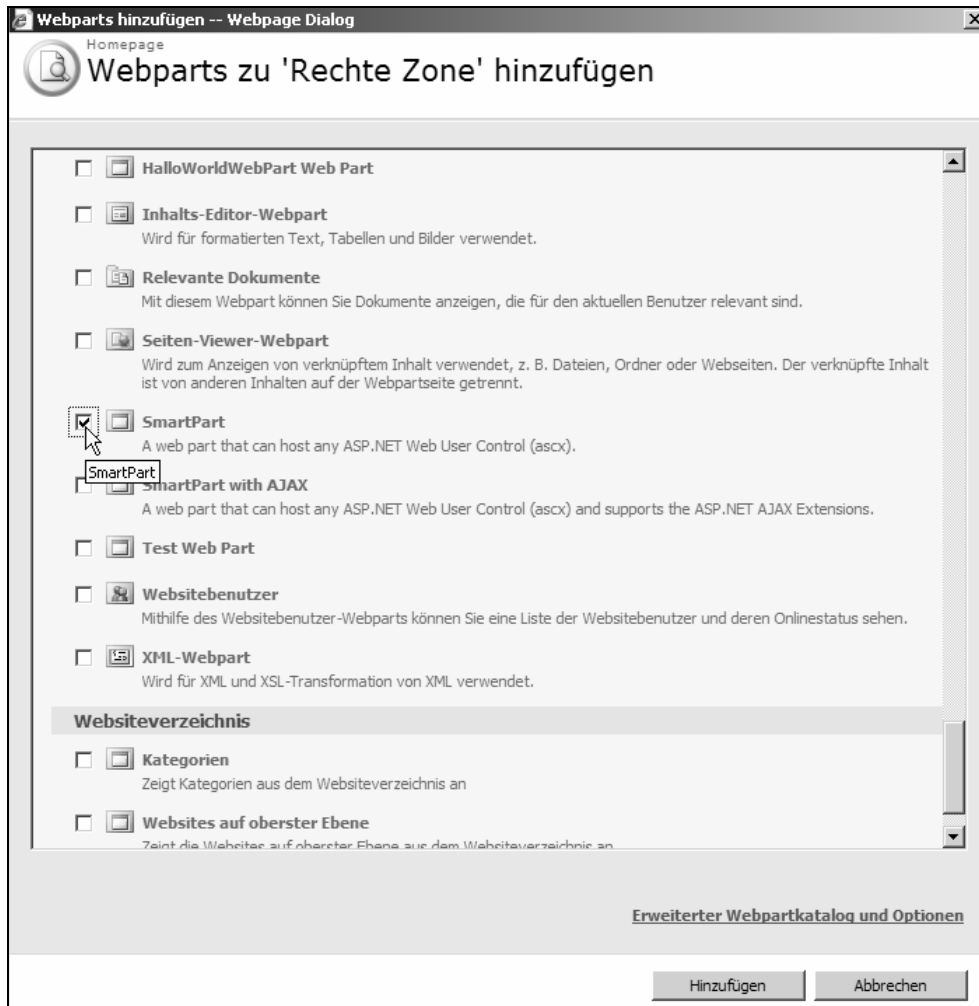


Abbildung 48.7 Auswahl des SmartPart-Webparts

In der Konfiguration des Webparts kann man dann eines der User Controls aus dem `/UserControls`-Ordner auswählen und einen Titel vergeben. Wenn es ein Problem mit dem User Control gibt, wird dies häufig schon hier angezeigt.

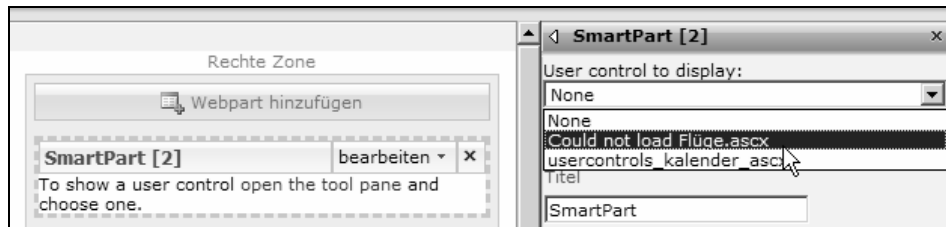


Abbildung 48.8 Festlegung des darzustellenden User Controls

Abbildung 48.9 zeigt ein User Control für den World Wide Wings-Flugplan in einer SharePoint-Seite.



Abbildung 48.9 Ein in eine SharePoint-Seite eingebettetes ASP.NET User Control

