

## Kapitel 47

# Webparts

### **In diesem Kapitel:**

Bestandteile eines Webportals	1039
Inhalt einer WebPartZone	1039
Anzeigemodi	1040
Eigene Webparts erstellen	1040
Webpart-Verbindungen (Web Part Connections)	1040

Webparts sind Elemente auf Portalseiten (in Java wird dieses Konzept *Portlets* genannt). Gegenüber normalen Steuerelementen zeichnen sich Webparts dadurch aus, dass der Endbenutzer die Steuerelemente hinsichtlich Position, Gestaltung und Inhalt konfigurieren kann. Webparts ermöglichen die Erstellung von personalisierbaren Webportalen aus Elementen, die zur Laufzeit durch den Endbenutzer konfigurierbar sind. Insbesondere ist es möglich, die darzustellenden Elemente auszuwählen und die Anordnung der Elemente per *Ziehen und Ablegen* innerhalb des Internet Explorers zu verändern.

Die bereits aus dem Microsoft SharePoint Server bekannten Webparts sind in einer erweiterten Form auch in ASP.NET ab Version 2.0 integriert.

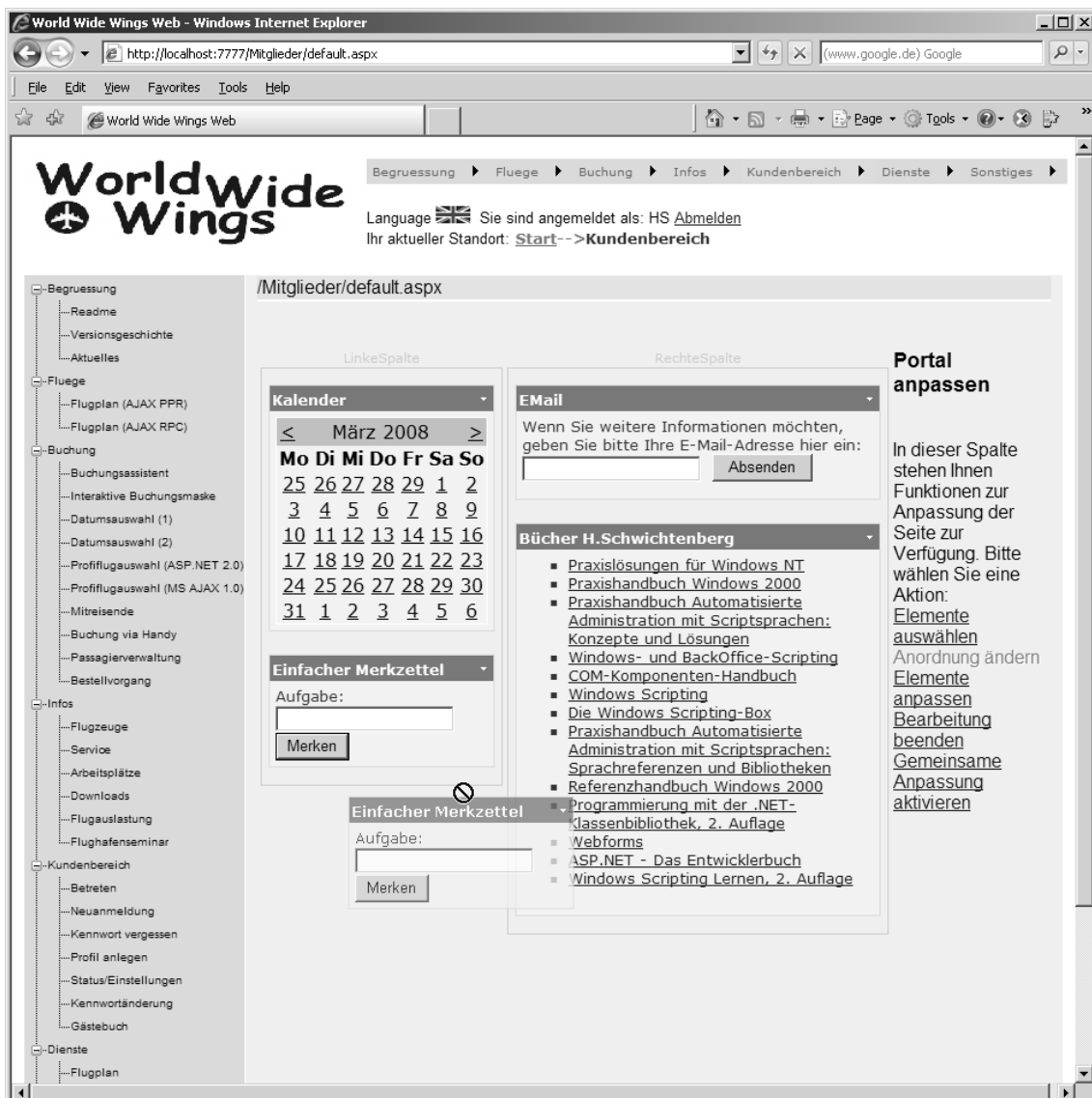


Abbildung 47.1 Ziehen und Ablegen eines Webparts [/Mitglieder/Default.aspx]

**HINWEIS** Die ASP.NET-Webparts bilden die Grundlage für die nächste Version von Microsoft SharePoint und anderen Produkten, die Webparts verwenden. Die Webparts in ASP.NET 2.0 entsprechen nicht den Webparts in SharePoint Portal Server 2003.

## Bestandteile eines Webportals

Eine Portalseite besteht aus folgenden Elementen:

- aus genau einem `<asp:WebPartManager>`-Element
- aus einem oder mehreren `<asp:WebPartZone>`-Elementen
- aus optional jeweils möglichst einem Element des Typs `<asp:CatalogZone>`, `<asp:EditorZone>` und `<asp:ConnectionsZone>`.

## Inhalt einer WebPartZone

Das `WebPartZone`-Element besitzt ein `<ZoneTemplate>`-Unterelement, in dem festgelegt wird, welche Elemente in der jeweiligen Zone angezeigt werden sollen. Mögliche Mitglieder können sein:

- jedes beliebige Serversteuerelement (z. B. `<asp:Calendar>`)
- normale User Controls
- User Controls, die `System.Web.UI.WebControls.WebParts.IWebPart` implementieren
- Klassen, die von `System.Web.UI.WebControls.WebParts.WebPart` erben oder die Schnittstelle `System.Web.UI.WebControls.WebParts.IWebPart` implementieren.

Das `WebPartZone`-Element kann über zahlreiche Stilelemente konfiguriert werden:

```
<asp:WebPartZone BorderColor="#CCCCCC" Font-Names="Verdana" ID="LinkeSpalte" Padding="6" runat="server">
  <PartChromeStyle BackColor="#EFF3FB" BorderColor="#D1DDF1" Font-Names="Verdana"
    ForeColor="#333333" />
  <MenuLabelHoverStyle ForeColor="#D1DDF1" />
  ...
  <ZoneTemplate>
    <asp:Calendar runat="server" ID="Calendar1" />
    <HS:Merkzettel runat="server" ID="TextInputControl1" />
    <HS:BuecherControl ID="BC1" runat="server" />
  </ZoneTemplate>
</asp:WebPartZone>
```

**Listing 47.1** Deklaration einer *WebPartZone*

## Anzeigemodi

Ein Webpart-basiertes Portal kennt fünf Anzeigemodi, die über das Attribut `DisplayMode` des `WebPartManager`-Steuerelements gesetzt werden können:

- `this.WebPartManager.DisplayMode = WebPartManager.BrowseDisplayMode;`  
normaler Anzeigemodus (Standardeinstellung)
- `this.WebPartManager.DisplayMode = WebPartManager.DesignDisplayMode;`  
Veränderung der Anordnung der Webparts per *Ziehen und Ablegen*
- `this.WebPartManager.DisplayMode = WebPartManager.CatalogDisplayMode;`  
Auswahl der anzuzeigenden Webparts
- `this.WebPartManager.DisplayMode = WebPartManager.EditDisplayMode;`  
Konfiguration der Webparts
- `this.WebPartManager.DisplayMode = WebPartManager.ConnectDisplayMode;`  
Konfiguration der Datenverbindung von Webparts

Die Modi `CatalogDisplayMode` und `EditDisplayMode` umfassen auch die Funktion des Modus `DesignDisplayMode`.

## Eigene Webparts erstellen

Jedes ASP.NET-Steuerelement kann als Webpart eingesetzt werden. Einige Möglichkeiten (z.B. Kontrolle über verfügbare Aktionen) stehen nur für Steuerelemente zur Verfügung, die von der Klasse `System.Web.UI.WebControls.WebParts.WebPart` abgeleitet wurden oder welche die Schnittstelle `System.Web.UI.WebControls.WebParts.IWebPart` implementieren.

In dem Fallbeispiel zu diesem Kapitel finden Sie folgende Anwendungsbeispiele:

- `Merkzettel.aspx` ist ein User Control, das über die Implementierung der Schnittstelle `System.Web.UI.WebControls.WebParts.IWebPart` zum Webpart aufgewertet wird.
- `MottoWebPart.vb` und `MottoWebPartEdit.vb` implementieren ein Webpart rein codebasiert.

## Webpart-Verbindungen (Web Part Connections)

Web Part Connections erlauben die automatische Datenübergabe zwischen Webparts. Ein *Provider-Webpart* ist dabei ein Webpart, das Daten für einen oder mehrere Webpart-Konsumenten bereitstellt. Es handelt sich um ein Pullmodell: Der Konsument holt Daten von dem Provider.

Der Datenaustausch erfolgt durch eine zu definierende Schnittstelle, welche die Klasse des Provider-Webparts implementieren muss. Das Provider-Webpart muss über die Methode `ProvideInterfaceForConnection()` einen Zeiger auf die Kommunikationsschnittstelle zurückliefern. Der Konsument muss eine Methode anbieten, welche diese definierte Austauschschnittstelle empfängt.

```
public interface ITextCommunication
{
    string Text {get;}
}
```

**Listing 47.2** Deklaration einer textbasierten Datenaustauschnittstelle für Webparts

```
[ConnectionProvider("TextProvider")]
public ITextCommunication ProvideInterfaceForConnection()
{
    return this;
    //INSTANT C# NOTE: Inserted the following 'return' since all code paths must return a value in
C#:
    return null;
}public string Text
{
    get
    {
        return this.TextBox1.Text;
    }
}
```

**Listing 47.3** Implementierung im Provider-Webpart

```
[ConnectionConsumer("TextConsumer1")]
public void GetITextProvider(de.ITVisions.Webparts.ITextCommunication myInterface)
{
    if (myInterface != null)
    {
        ContentText = myInterface.Text;
    }
}
```

**Listing 47.4** Implementierung im Konsumenten-Webpart

Webparts, welche die gleiche Austauschchnittstelle unterstützen, können verbunden werden. Die Verbindung kann statisch im Code definiert (Listing 47.5) oder durch den Benutzer konfigurierbar gehalten werden (hier aus Platzgründen nicht erklärt). Ein Provider kann mehrere Konsumenten besitzen.

```
<asp:WebPartManager ID="WebPartManager" runat="server">
    <StaticConnections>
        <asp:WebPartConnection
            ID="Conn1"
            ConsumerID="Motto1"
            ProviderID="Merkzettel1">
        </asp:WebPartConnection>
    </StaticConnections>
</asp:WebPartManager>
```

**Listing 47.5** Deklaration einer statischen Webpart-Verbindung

