

Kapitel 49

Webanwendungen für kleine Endgeräte mit den mobilen Steuerelementen

In diesem Kapitel:

| | |
|--|------|
| Funktionsumfang | 1056 |
| Geschichte der mobilen Steuerelemente | 1056 |
| Webseiten für mobile Anwendungen | 1057 |
| Mobile Steuerelemente | 1057 |
| Entwickeln und Testen mobiler Webanwendungen | 1060 |
| Praxisbeispiel | 1062 |
| Zustandsverwaltung | 1068 |
| Formatvorlagen | 1069 |
| Erkennen von Gerätefähigkeiten | 1070 |
| Gerätespezifische Ausgaben | 1070 |
| Erstellung selbst definierter mobiler Steuerelemente | 1074 |

Die meisten Webanwendungen richten sich an Benutzer mit Desktopcomputern, auf denen ein »vollständiger« Webbrowser läuft. Immer wichtiger werden aber kleinere (mobile) Endgeräte wie Mobiltelefone und Pocket PCs. Diese Endgeräte besitzen einen sehr viel kleineren Bildschirm und oft nur funktionsreduzierte Webbrowser. Diese Webbrowser können oft kein »normales« HTML verstehen, sondern nur *Compact HTML* (cHTML) oder die *Wireless Markup Language* (WML).

Für die Entwicklung von Webanwendungen für kleine Endgeräte bietet ASP.NET eine spezielle Unterstützung in Form der *mobilen Serversteuerelemente* (engl. Mobile Controls).

TIPP

Zum guten Design von Anwendungen für mobile Endgeräte gehört, dass man

- auf feste Positionierung von Steuerelementen verzichtet.
- Auswahlsteuerelemente wie Auswahlfelder anstelle von Textfeldern verwendet, in die Benutzer Werte eingeben müssen (die Texteingabe ist auf mobilen Endgeräten in der Regel weniger komfortabel als bei Desktopsystemen).
- Auswahlfelder statt Optionslisten oder Listfelder einsetzt, da Auswahlfelder platzsparender sind.
- mehrere kleinere Webseiten statt einer großen einsetzt, um dem Benutzer das lästige Scrollen zu ersparen.
- eine möglichst einfache Navigation verwendet, in welcher der Benutzer sich leicht vor und zurück bewegen kann.

Funktionsumfang

Für die Erstellung von Webanwendungen für mobile Endgeräte bietet ASP.NET einen speziellen Satz serverseitiger Steuerelemente für mobile Anwendungen im Namensraum `System.Web.UI.MobileControls` in `System.Web.Mobile.dll` an. Die Mobile Controls realisieren Adaptive Rendering und erstellen automatisch eine auf die Zielpattform zutreffende Ausgabe in der Wireless Markup Language 1.1 (WML), in Compact HTML 1.0 (cHTML), XHTML oder HTML 3.2. Alternativ kann der Entwickler Vorgaben für unterschiedliche Plattformen manuell definieren. Zur Erstellung mobiler Webanwendungen bietet VWD einen speziellen grafischen Designer an. Außerdem unterstützen mobile Web Forms die automatische Aufteilung von Inhalten in Seiten, wenn die Darstellungsgröße nicht ausreichend ist.

HINWEIS

Es gibt keine Neuerungen für mobile Steuerelemente in den ASP.NET-Versionen 2.0 und 3.5 gegenüber ASP.NET 1.1.

Geschichte der mobilen Steuerelemente

Die Mobile Controls waren in ASP.NET 1.0 nicht enthalten. Microsoft hat diese erst später als kostenloses Add-On angeboten unter dem Namen *Mobile Internet Toolkit* (MIT oder auch *MMIT* abgekürzt). Das MIT war in ASP.NET 1.1 dann schon standardmäßig im Lieferumfang des .NET Framework Redistributable enthalten. In ASP.NET ab Version 2.0 gilt es als Kernbestandteil von ASP.NET.

Ursprünglich war geplant, mit ASP.NET 2.0 einen Steuerelementsatz zu liefern, der Adaptive Rendering für HTML, XHTML, WML und cHTML gleichzeitig unterstützt. Leider hat Microsoft diesen Plan auf eine spätere Version von ASP.NET vertagt. Daher können die Standardserversteuerelemente weiterhin nur HTML ausgeben. Webseiten für kleine Endgeräte (z. B. PDAs, Handys, Smartphones) müssen mit dem speziellen Steuerelementsatz `System.Web.UI.MobileControls` erstellt werden. Diese *mobilen Steuerelemente* unterstützen die fallweise Ausgabe (Adaptive Rendering) von Compact HTML 1.0, HTML 3.2, WML 1.x oder XHTML.

Webseiten für mobile Anwendungen

Eine Website für mobile Anwendungen besitzt Ähnlichkeiten, aber auch Unterschiede zu einem normalen Web Form:

- Die ASPX-Datei beginnt mit einer @Page-Direktive.
- Der Namensraum System.Web.UI.MobileControls muss registriert werden.
- Die Hintergrundcodeklasse erbt von System.Web.UI.MobileControls.MobilePage.
- Die Seite besteht aus einem oder mehreren Formularelementen (<mobile:Form>).

ACHTUNG Es ist wegen der typischerweise beschränkten Darstellungsfläche auf mobilen Geräten bei mobilen Web Forms geboten, mehrere <Form>-Elemente innerhalb einer Seite an das Endgerät zu senden, und dann zwischen diesen Seiten zu wechseln. Dementsprechend unterstützt der Designer im VWD auch die Darstellung mehrerer Formulare.

Seitenwechsel

Eine Webseite kann aus mehreren <mobile:Form>-Elementen bestehen, aber es wird immer nur der Inhalt eines Elements zum Browser gesendet. Das aktive Formular wird durch das Attribut ActiveForm bestimmt, z.B. ActiveForm = Form3.

Seitenumbrüche

Eine weitere Möglichkeit zur Aufteilung der Seiteninhalte ist der automatische Seitenumbruch (Paginierung), der mit dem Attribut Paginate im <mobile:Form>-Element aktiviert und durch verschiedene PagerStyle-Elemente formatiert wird. Der Seitenumbruch kann dann im Programmcode durch die Attribute CurrentPage und PageCount gesteuert werden.

```
<mobile:Form ID="Form3" Runat="server"
Paginate="True"
PagerStyle-BackColor="Cyan"
PagerStyle-NextPageText=">"
PagerStyle-PreviousPageText="<">
...
</mobile:Form>
```

Listing 49.1 Rumpf für ein mobiles Web Form

Mobile Steuerelemente

Die VWD-Werkzeugleiste enthält für mobile Web Forms einen speziellen Steuerelementsatz. Normale Serversteuerelemente können nicht verwendet werden. Das grundsätzliche Programmiermodell ist gleich: Mobile Steuerelemente sind Serversteuerelemente, werden durch Tags mit dem Zusatz runat="server" deklariert und sind im Programmcode als Objekte ansprechbar.

Der Steuerelementsatz für mobile Web Forms ist wesentlich kleiner als der Steuerelementsatz für normale Web Forms. Allerdings verbirgt sich hinter einem mobilen Steuerelement die Funktionalität mehrerer normaler Steuerelemente, z. B. erzeugt das Steuerelement `SelectionList` wahlweise ein Auswahlfeld (`DropDown`), ein Listenfeld (`ListBox`), eine Menge von Kontrollkästchen (`CheckBox`) oder Optionsschaltflächen (`Radio`).

Der große Vorteil der mobilen Steuerelemente ist die fallweise Ausgabe von Compact HTML 1.0, HTML 3.2, WML 1.x oder XHTML. Mobile Steuerelemente besitzen aber insgesamt viel weniger Funktionen als ihre »großen« Pendanten, z. B. gibt es keinen Autopostback.

Überblick

Abbildung 49.1 zeigt die Werkzeugleiste für mobile Steuerelemente. Tabelle 49.1 erläutert die Steuerelemente.

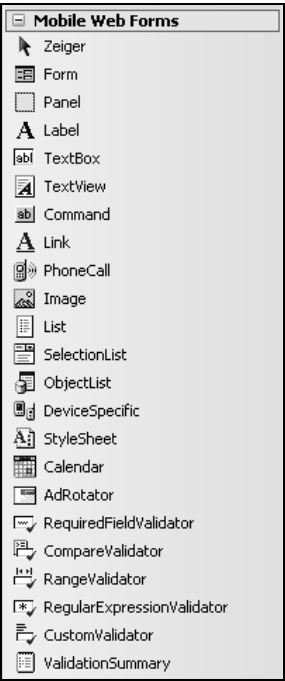


Abbildung 49.1 Die mobilen Serversteuerelemente in der VWD-Toolbox

| Steuerelement | Beschreibung |
|---------------|---|
| Form | Ein Formular, in dem die Steuerelemente eingebunden werden können. Im Gegensatz zu normalen Web Forms können mehrere <Form>-Elemente auf eine Seite gesetzt werden. |
| Panel | Gruppierungsmöglichkeit für weitere Steuerelemente |
| Label | Beschriftungselement. Kann auch zur programmgesteuerten Textausgabe verwendet werden. |
| TextBox | Eingabefeld für Text |
| TextView | Darstellung längerer Texte. Verfügt über grundlegende Formatierungsmöglichkeiten wie z. B. , <i>, etc. |

| Steuerelement | Beschreibung |
|----------------------------|---|
| Command | Ein Link oder eine Schaltfläche |
| Link | Ein Verweis auf eine andere Webseite oder eine andere Form innerhalb der aktuellen Webseite |
| PhoneCall | Zeigt ein Element zum Starten eines Telefonanrufs zu einer bestimmten Nummer. Endgeräte, die keine Telefonanrufe unterstützen, zeigen nur die Telefonnummer an. |
| Image | Grafiken |
| List | Liste mit selektierbaren Elementen (vergleichbar mit Repeater und DataList) |
| SelectionList | Auswahlliste verschiedener Alternativen |
| ObjectList | Anzeige von Daten. Vergleichbar mit GridView-Steuerelement; eine einzelne Spalte wird in einer Listenansicht gezeigt. Ein Link führt dann jeweils zu einer Detailansicht. |
| DeviceSpecific | Definition einer gerätespezifischen Darstellung. |
| StyleSheet | Formatierung der Darstellung |
| Calendar | Kalender |
| AdRotator | Wechselnde Werbeeinblendungen |
| RequiredFieldValidator | Prüfung, dass ein Feld nicht leer gelassen wurde |
| CompareValidator | Vergleich zweier eingegebener Werte |
| RangeValidator | Prüfen eines bestimmten Gültigkeitsbereichs |
| RegularExpressionValidator | Prüfung gegen einen regulären Ausdruck |
| CustomValidator | Benutzerdefinierte Prüfung |
| ValidationSummary | Zusammenfassung der Überprüfungsergebnisse |

Tabelle 49.1 Mobile Web Controls

Textdarstellung mit dem TextView-Steuerelement

Um Text innerhalb von mobilen Endgeräten darzustellen, gibt es mehrere einfache Möglichkeiten. Neben statischem Text und der Verwendung des Label-Steuerelements wird ein TextView-Steuerelement angeboten.

Mittels des TextView-Steuerelementes lassen sich größere Texte im HTML-Format darstellen. Das bedeutet genau genommen, dass der Text, der mit diesem Steuerelement dargestellt wird, mit HTML und nicht im reinen Klartext definiert werden kann. Dadurch können die folgenden Formatierungstags verwendet werden: <p>,
, , <i> und <a>.

```
<mobile:TextView id="C_TextView" runat="server">
<p>Dies ist ein Beispiel für die Verwendung von <b>Formatierungen</b><br> innerhalb eines
<i>TextView</i>-Steuerelements.</p>
</mobile:TextView>
```

Dies ist insofern praktisch, da ansonsten nur XML-konformer Text erstellt werden darf, das heißt der `
`-Tag würde nicht funktionieren, da dieser keinen schließenden Tag verwendet. Dieses Steuerelement sorgt auch automatisch dafür, dass sehr langer Text über mehrere Seiten dargestellt werden kann. Dem Benutzer wird bei sehr langen Texten die Möglichkeit geboten, sich seitenweise durch den Text zu bewegen (Pagination). Dieses Verhalten wird vom Steuerelement vollständig kontrolliert.

Entwickeln und Testen mobiler Webanwendungen

Anders als Visual Studio .NET 2002/2003 stehen im VWD keine speziellen Projektvorlagen für kleine Endgeräte zur Verfügung. Es gibt lediglich unter den Elementvorlagen folgende Einträge:

- *Mobiles Web Form*
- *Mobiles Web-Benutzersteuerelement* (Mobile Web User Control).

Bei einem *Mobilen Web-Benutzersteuerelement* (Mobile Web User Control) handelt es sich um das Pendant zu einem User Control, wie es von normalen Web Forms her bekannt ist. Dadurch lassen sich benutzerdefinierte Steuerelemente durch einfaches *Ziehen und Ablegen* erstellen und innerhalb eines Projektes beliebig oft verwenden.

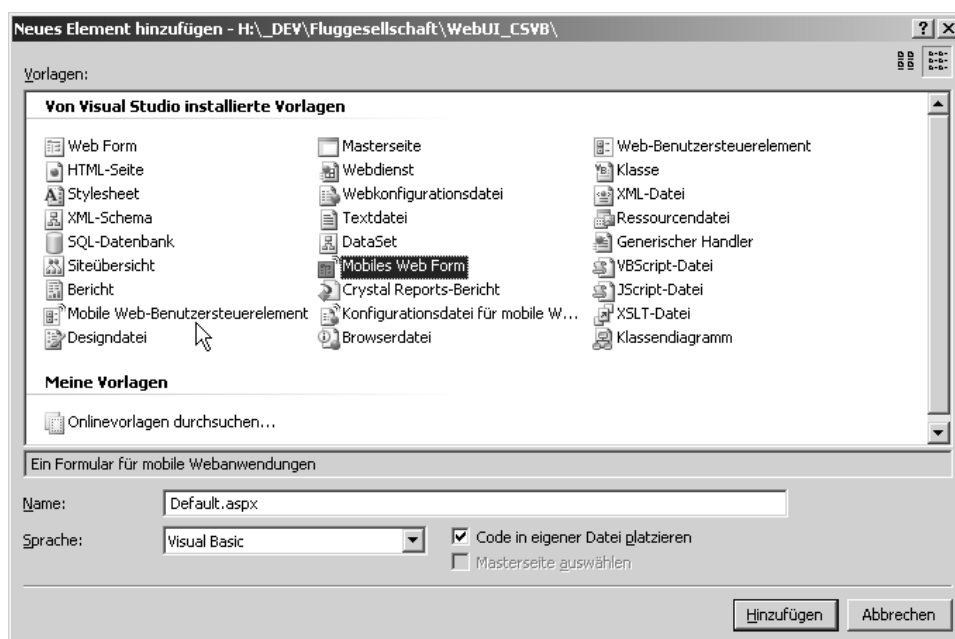


Abbildung 49.2 Elementvorlagen für mobile Webanwendungen

HINWEIS

Bei mobilen Web Forms können Programmiersprache und Codemodell frei gewählt werden. Vorlagenseiten (Masterpages) werden aber nicht unterstützt.

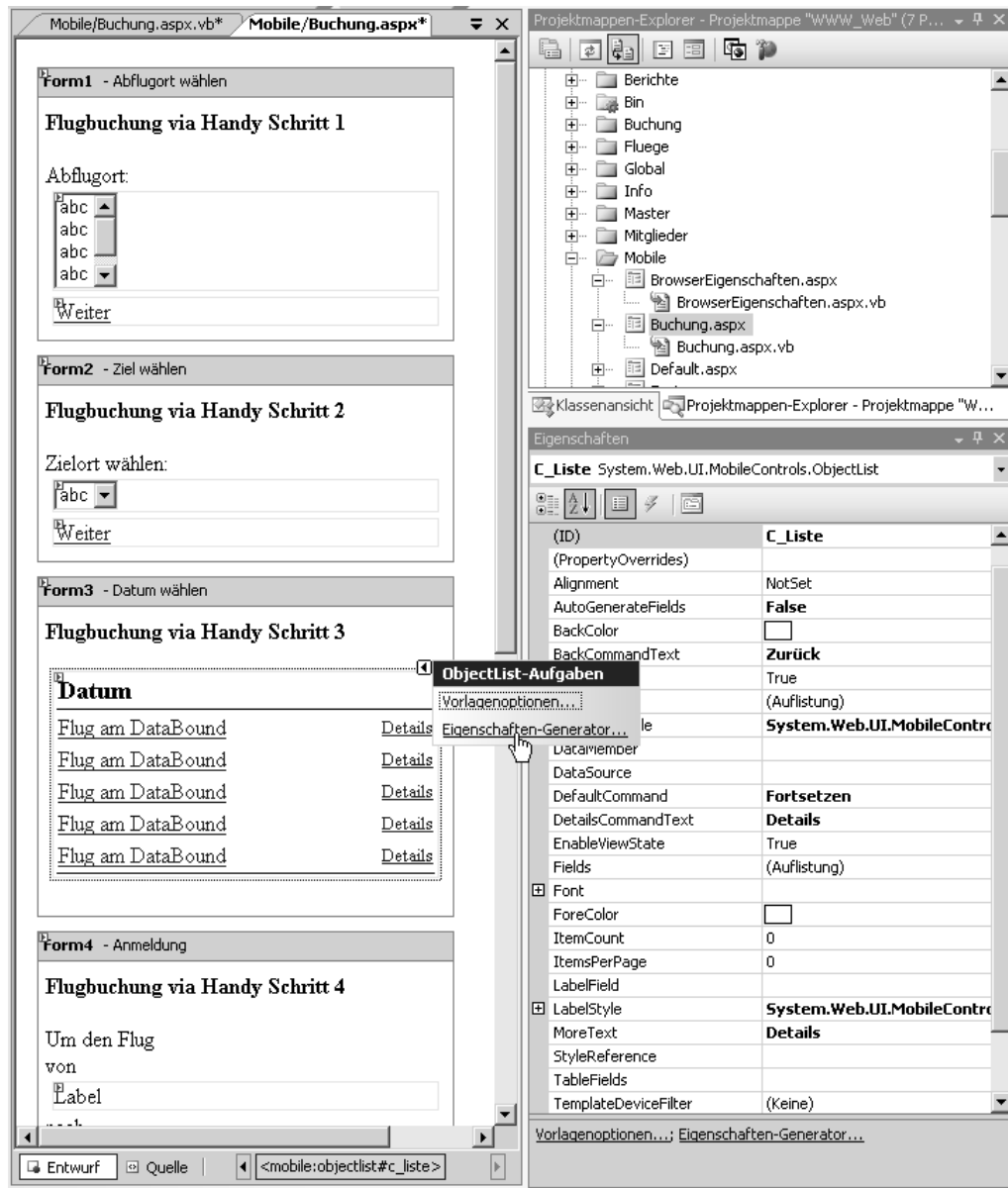


Abbildung 49.3 Eine mobile Website mit mehreren Web Forms

Testen mobiler Webanwendungen

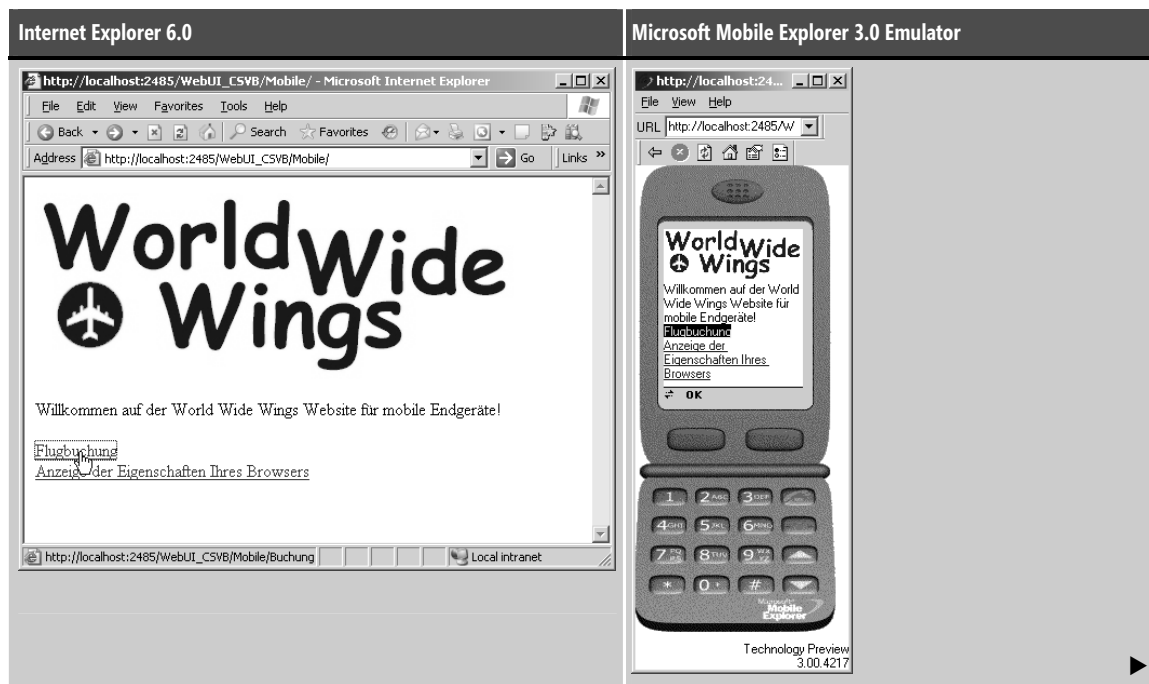
Zum Test mobiler Webanwendungen kann man Geräteemulatoren (z.B. Microsoft Mobile Explorer oder Openwave Phone Simulator) einsetzen. Mobile Web Forms lassen sich aber auch im Internet Explorer darstellen. Die Mobile Web Controls erzeugen dafür HTML 3.2. Während des Entwickelns und des Debuggens kann dies durchaus hilfreich sein. Dadurch muss nicht immer explizit auf Emulatoren zurückgegriffen werden.

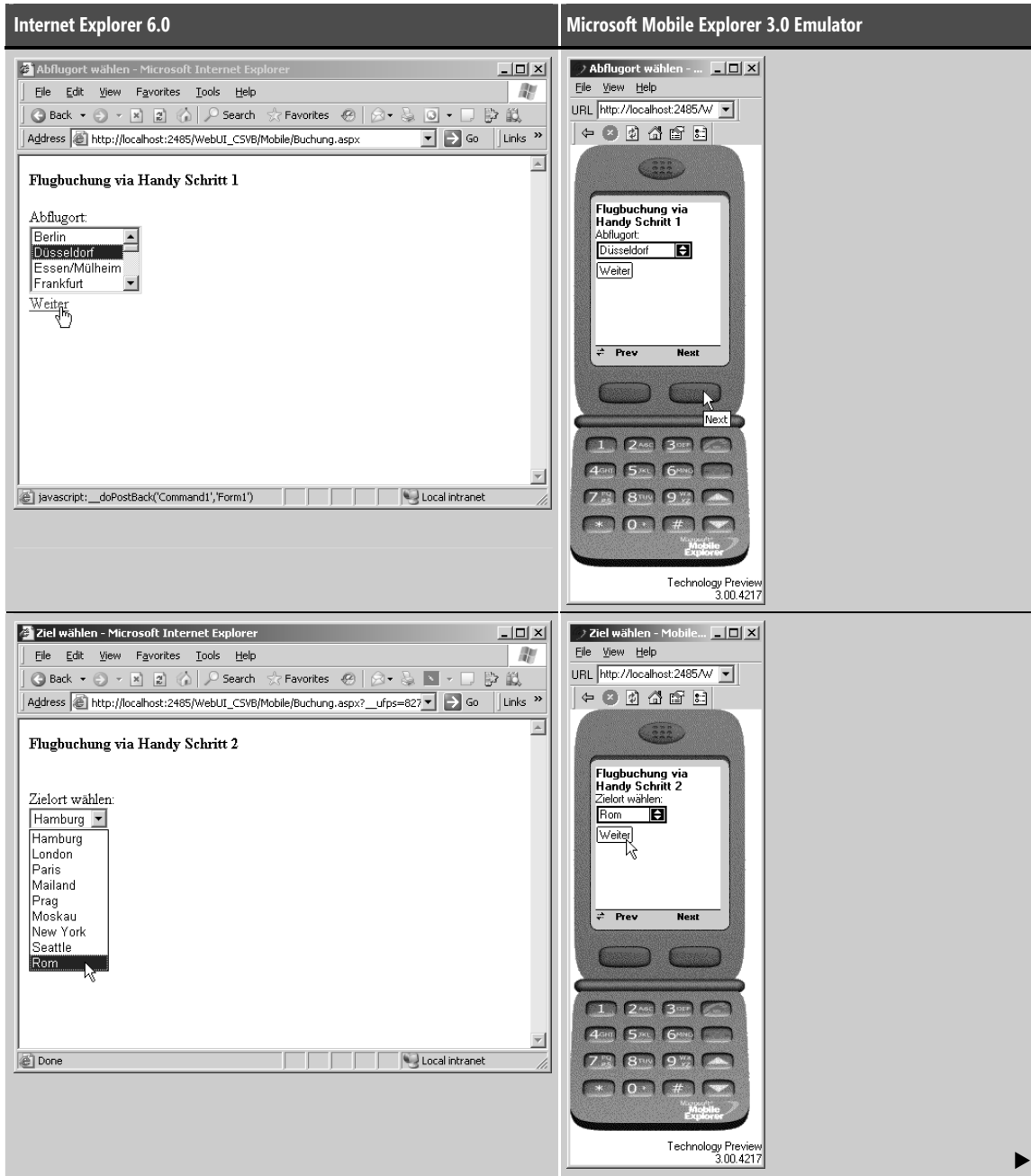
TIPP

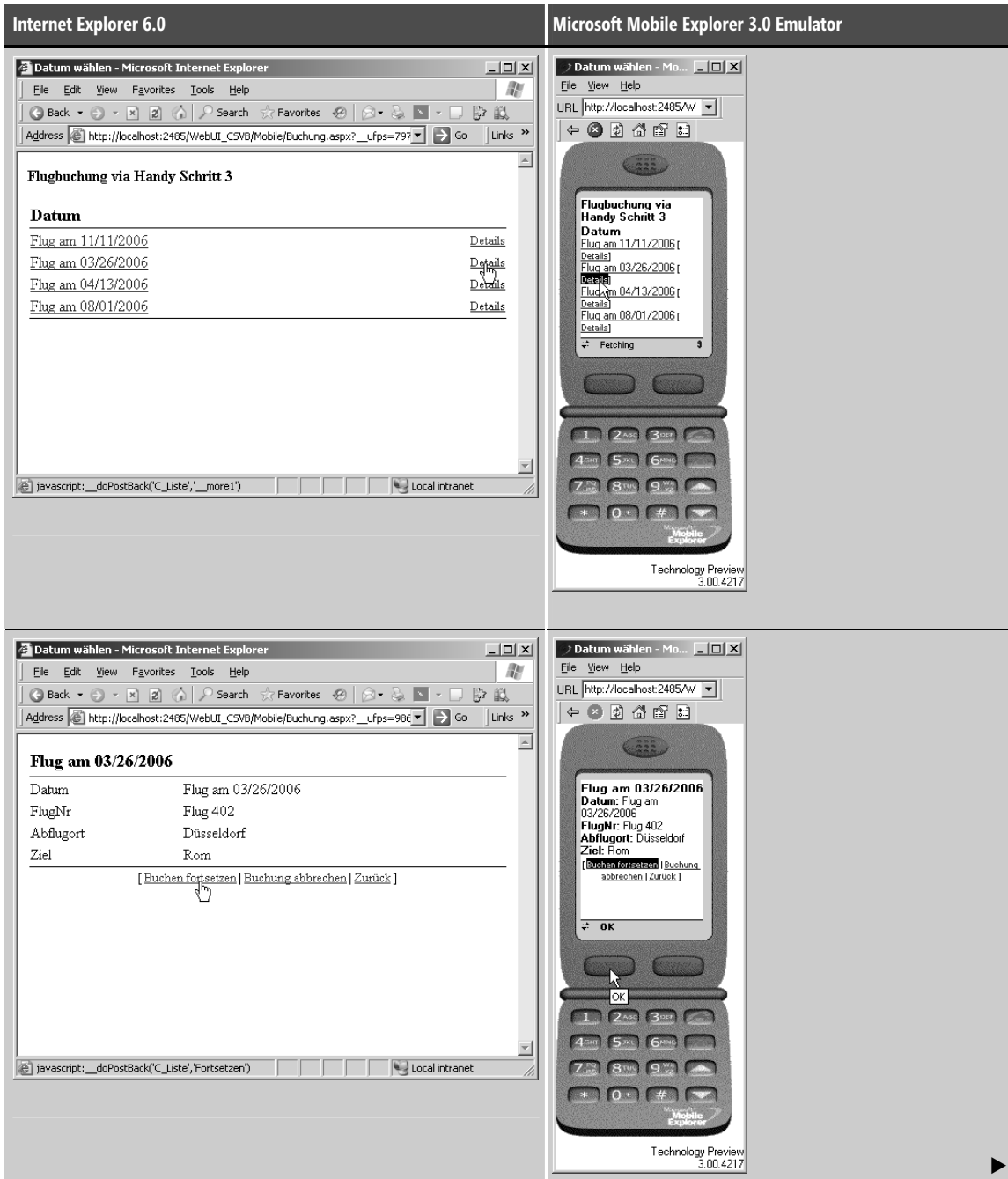
Mobile Webanwendungen sollten auch immer mit echten Endgeräten getestet werden.

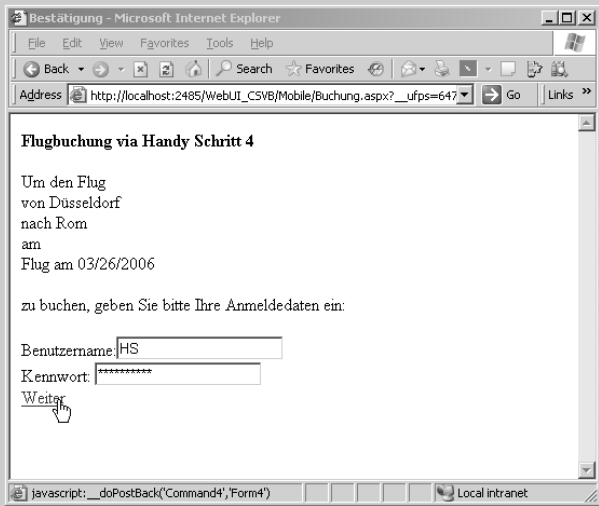


Praxisbeispiel

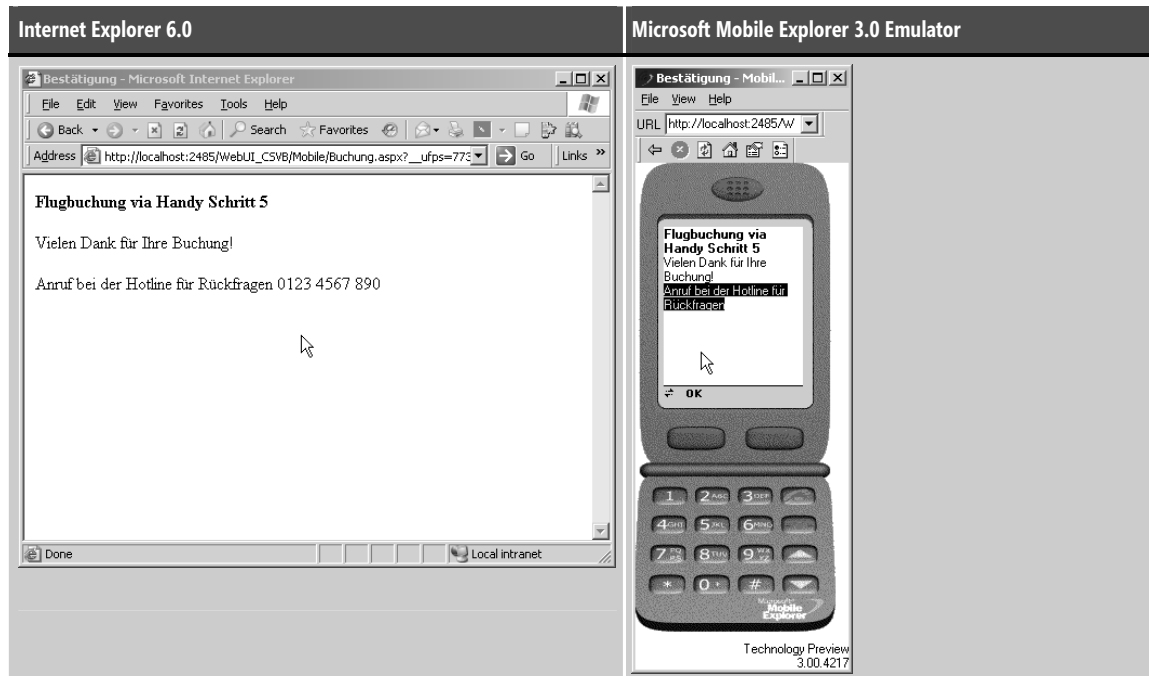
Als Beispiel für eine mobile Webanwendung kommt die Flugbuchung zum Einsatz. Die einzelnen Bildschirmabbildungen stellen jeweils die einzelnen Bedienungsschritte der Webanwendung auf dem Internet Explorer 6.0 und den Emulator für den Microsoft Mobile Explorer 3.0 gegenüber. Dies ermöglicht den detaillierten Vergleich der Darstellungen in beiden Browsern.







| Internet Explorer 6.0 | Microsoft Mobile Explorer 3.0 Emulator |
|---|---|
|  <p>Bestätigung - Microsoft Internet Explorer</p> <p>File Edit View Favorites Tools Help</p> <p>Back Forward Stop Reload Home Search Favorites</p> <p>Address http://localhost:2485/WebUI_CSVB/Mobile/Buchung.aspx?__ufps=647 Go Links</p> <p>Flugbuchung via Handy Schritt 4</p> <p>Um den Flug von Düsseldorf nach Rom am Flug am 03/26/2006</p> <p>zu buchen, geben Sie bitte Ihre Anmeldedaten ein:</p> <p>Benutzername: <input type="text" value="HS"/></p> <p>Kennwort: <input type="password" value="*****"/></p> <p>Weiter</p> <p>javascript: __doPostBack('Command4','Form4') Local intranet</p> |  <p>Anmeldung - Mobile...</p> <p>File View Help</p> <p>URL http://localhost:2485/w/</p> <p>Back Forward Stop Reload Home Search Favorites</p> <p>Flugbuchung via Handy Schritt 4</p> <p>Um den Flug von Düsseldorf nach Rom am Flug am 03/26/2006</p> <p>Weiter</p> <p>OK</p> <p>Technology Preview 3.00.4217</p> |
| |  <p>Anmeldung - Mobile...</p> <p>File View Help</p> <p>URL http://localhost:2485/w/</p> <p>Back Forward Stop Reload Home Search Favorites</p> <p>zu buchen, geben Sie bitte Ihre Anmeldedaten ein:</p> <p>Benutzername: <input type="text" value="HS"/></p> <p>Kennwort: <input type="password" value="*****"/></p> <p>Weiter</p> <p>Zurück</p> <p>Edit</p> <p>Technology Preview 3.00.4217</p> |



Quellcode

Die Webseite besteht aus fünf verschiedenen Formularen:

```
<%@ Page Language="..." AutoEventWireup="false" CodeFile="Buchung.aspx.vb" Inherits="Mobile_Default"
    UICulture="DE-de" Culture="DE-de" %>
<%@ Register TagPrefix="mobile" Namespace="System.Web.UI.MobileControls" Assembly="System.Web.Mobile" %>

<html xmlns="http://www.w3.org/1999/xhtml" >
<body>
    <mobile:Form Title="Abflugort wählen" id="Form1" runat="server" Paginate="True">
        <b>Flugbuchung via Handy Schritt 1</b><br /><br />Abflugort:<br /><mobile:SelectionList
            ID="C_Abflugort" Runat="server" SelectType="ListBox"></mobile:SelectionList>
        <mobile:Command ID="Command1" Runat="server" Format="Link">Weiter</mobile:Command></mobile:Form>
    <mobile:Form ID="Form2" Runat="server" Title="Ziel wählen"><b>Flugbuchung via Handy Schritt 2</b><br />
        <br><br><b>Zielort wählen:<br /></b><mobile:SelectionList
            ID="C_Zielort" Runat="server">
    </mobile:SelectionList> <mobile:Command ID="Command2" Runat="server"
    Format="Link">Weiter</mobile:Command></mobile:Form>
    <mobile:Form ID="Form3" Runat="server" Paginate="True" PagerStyle-BackColor="Cyan"
        PagerStyle-NextPageText=">" PagerStyle-PreviousPageText="<" Title="Datum wählen">
        <b>Flugbuchung via Handy Schritt 3</b><br /></b>
    <br />
    <mobile:ObjectList ID="C_Liste" Runat="server" CommandStyle-StyleReference="subcommand"
        LabelStyle-StyleReference="title" MoreText="Details" AutoGenerateFields="False"
        DetailsCommandText="Details" BackCommandText="Zurück" DefaultCommand="Fortsetzen">
        <Command Name="Fortsetzen" Text="Buchen fortsetzen" />
        <Command Name="Abbrechen" Text="Buchung abbrechen" />
    </mobile:ObjectList>
</body>
</html>
```

```

    <Field Name="Datum" DataField="FL_Datum" DataFormatString="Flug am {0:d}" Title="Datum" />
    <Field Name="FlugNr" DataField="FL_FlugNr" DataFormatString="Flug {0}" Title="FlugNr" />
    <Field Name="Abflugort" DataField="FL_Abflugort" Title="Abflugort" />
    <Field Name="Zielort" DataField="FL_Zielort" Title="Ziel" />

</mobile:ObjectList>
</mobile:Form><br>

<mobile:Form ID="Form4" Runat="server" Title="Anmeldung" Paginate="True"><b>Flugbuchung via Handy
Schritt 4</b><br /><br />Um den Flug<br />von <mobile:Label
ID="C_von" Runat="server">Label</mobile:Label><br />nach <mobile:Label ID="C_nach"
Runat="server">Label</mobile:Label><br />am <br /><mobile:Label ID="C_Datum"
Runat="server">Label</mobile:Label><br />zu buchen, geben Sie bitte Ihre Anmelde­daten ein:<br />
    <br />Benutzername:<mobile:TextBox
    ID="TextBox1" Runat="server">
</mobile:TextBox><br />Kennwort: <mobile:TextBox Password=True ID="TextBox2" Runat="server">
</mobile:TextBox><br /><mobile:Command ID="Command4" Runat="server"
    Format="Link">Weiter</mobile:Command><br /></mobile:Form><br>
</br>
<mobile:Form ID="Form5" Runat="server" Title="Bestätigung"><b>Flugbuchung via Handy Schritt 5<br />
    <br /></b>Vielen Dank für Ihre Buchung!<br /><br /><mobile:PhoneCall
    ID="PhoneCall1" Runat="server" PhoneNumber="0123 4567 890">Anruf bei der Hotline f&#252;r
R&#252;ckfragen</mobile:PhoneCall></mobile:Form><br>
</br>
</body>
</html>

```

Listing 49.2 Mobile Webseite für Flugbuchung [/Mobile/Buchung.aspx]

```

internal partial class Mobile_Default : System.Web.UI.MobileControls.MobilePage
{
    protected void Form1_Load(object sender, System.EventArgs e)
    {
        if (! Page.IsPostBack)
        {
            this.C_Abflugort.DataSource = DE.WWWings.FlugBLManager.Abflugorte;
            this.C_Abflugort.DataBind();
        }
    }

    protected void Form2_Activate(object sender, System.EventArgs e)
    {
        if (this.C_Abflugort.Selection != null)
        {
            this.C_Zielort.DataSource =
                DE.WWWings.FlugBLManager.Ziele_Fuer_Abflugort(this.C_Abflugort.Selection.Value);
            this.C_Zielort.DataBind();
        }
    }

    protected void Command1_Click(object sender, System.EventArgs e)
    {
        this.ActiveForm = Form2;
    }
}

```

```
protected void Command2_Click(object sender, System.EventArgs e)
{
    this.ActiveForm = Form3;
}

protected void Form3_Activate(object sender, System.EventArgs e)
{
    this.C_Liste.DataSource =
        DE.WWWings.DAL.Flug_DataManager.AlleFluegeRoute(this.C_Abflugort.Selection.Value,
            this.C_Zielort.Selection.Value);
    this.C_Liste.DataBind();
}

protected void Command4_Click(object sender, System.EventArgs e)
{
    this.ActiveForm = Form5;
}

protected void Form4_Activate(object sender, System.EventArgs e)
{
    this.C_von.Text = this.C_Abflugort.Selection.Value;
    this.C_nach.Text = this.C_Zielort.Selection.Value;
    this.C_Datum.Text = this.C_Liste.Selection["Datum"];
}

protected void C_Liste_ItemCommand(object sender,
System.Web.UI.MobileControls.ObjectListCommandEventArgs e)
{
    if (e.CommandName == "Abbrechen")
    {
        Response.Redirect("default.aspx");
    }
    this.ActiveForm = Form4;
}
}
```

Listing 49.3 Mobile Webseite für Flugbuchung [/Mobile/Buchung.aspx]

Zustandsverwaltung

Mobile Web Forms unterstützen die Zustandsverwaltung auf Seitenebene (Viewstate) und Sitzungsebene (Sessionstate). Anders als bei normalen Web Forms werden die Seitenzustandsdaten nicht in einem versteckten Feld zum Client gesendet, sondern in einer Sitzungsvariablen auf dem Server abgelegt.

Zur Reduzierung des Speicherbedarfs auf dem Server speichert ASP.NET aber nicht den Viewstate aller Anfragen, sondern nur eine bestimmte Anzahl (*die letzten x Anfragen*) pro Benutzer. Diese Anzahl ist in der *web.config*-Datei einstellbar.

```
<configuration>
  <system.web>
    <mobileControls sessionStateHistorySize="10" />
  </system.web>
</configuration>
```

Listing 49.4 Einstellung der Größe der Viewstate-Historie

ACHTUNG Im Unterschied zu normalen Web Forms kann der Viewstate in einem mobilen Webform ungültig werden, wenn die Sitzungsleerlaufdauer überschritten ist. In diesem Fall wird das Ereignis `ViewStateExpire()` bei der nächsten Anfrage ausgelöst.

Die Sitzungsverwaltung der mobilen Web Forms arbeitet standardmäßig mit Cookies. Dieses Verhalten kann wie bei den normalen Web Forms in der *web.config*-Datei geändert werden.

Formatvorlagen

Formatvorlagen sind in mobilen Webanwendungen unter Zuhilfenahme des `StyleSheet`-Steuerelements möglich, das mehrere `Style`-Steuerelemente enthalten darf. Ein mobiles Serversteuerelement kann im Attribut `StyleReference` auf den Namen eines `Style`-Steuerelements verweisen.

```
<%@ Page Language="CS" AutoEventWireup="false" CodeFile="Default.aspx.cs" Inherits="Mobile_Default" %>
<%@ Register TagPrefix="mobile" Namespace="System.Web.UI.MobileControls" Assembly="System.Web.Mobile" %>
<html xmlns="http://www.w3.org/1999/xhtml">
<body>
  <mobile:StyleSheet ID="C_StyleSheet" Runat="server">
    <mobile:Style Name="Style1" Wrapping="Wrap" Alignment="Center">
    </mobile:Style>
    <mobile:Style Name="Style2" Font-Size="Normal" Font-Bold="True" Font-Italic="True"
      Wrapping="NoWrap" Alignment="Right">
    </mobile:Style>
    <mobile:Style Font-Size="Large" Font-Bold="False" Font-Italic="True" Alignment="Left"
      Name="Style3">
    </mobile:Style>
  </mobile:StyleSheet>
  <mobile:Form ID="Form1" Runat="server">
    <mobile:Image ID="Image1" Runat="server" ImageUrl="~/App_Img/WorldWideWings_Logo.jpg">
  </mobile:Image><br /><br />Willkommen auf der WorldWideWings Website für mobile Endgeräte!<br />
    <br /><a
      href="~/Mobile/Buchung.aspx">Flugbuchung</a><br />
    <mobile:Link StyleReference="Style3" ID="Link1" NavigateUrl="BrowserEigenschaften.aspx"
      Runat="server">Anzeige der Eigenschaften Ihres Browsers</mobile:Link>
  </mobile:Form>
</body>
</html>
```

Listing 49.5 `StyleSheet`-Steuerelement [/mobile/default.aspx]

Erkennen von Gerätefähigkeiten

Um Zugriff auf die Fähigkeiten eines mobilen Endgerätes zu erhalten, kann man ähnlich wie in Verbindung mit normalen Web Forms die Browsereigenschaften über eine Klasse auslesen. Dadurch erhält die Anwendung die Möglichkeit, die Fähigkeiten des Endgerätes zu ermitteln und entsprechend darauf zu reagieren. Für Web Forms kann die Klasse `System.Web.HttpBrowserCapabilities` verwendet werden, für mobile Endgeräte steht die Klasse `System.Web.Mobile.MobileCapabilities` zur Verfügung. Eine kleine Auswahl der möglichen Eigenschaften ist im folgenden Beispiel enthalten:

```
protected void Page_Load(object sender, System.EventArgs e)
{
    System.Web.Mobile.MobileCapabilities b = null;
    b = (System.Web.Mobile.MobileCapabilities)Request.Browser;
    this.C_Text.Text += "Browser: " + b.Browser + "<br>";
    this.C_Text.Text += "Höhe: " + b.ScreenPixelsHeight + "<br>";
    this.C_Text.Text += "Breite: " + b.ScreenPixelsWidth + "<br>";
    this.C_Text.Text += "Fettschrift: " + b.SupportsBold + "<br>";
    this.C_Text.Text += "Schriftgrößen: " + b.SupportsFontSize + "<br>";
    this.C_Text.Text += "CSS: " + b.SupportsCss + "<br>";
    this.C_Text.Text += "XmlHttp: " + b.SupportsXmlHttp + "<br>";
}
```

Listing 49.6 Zugriff auf die Eigenschaften des Endgeräts

ACHTUNG Die Daten in dem `MobileCapabilities`-Objekt basieren zum großen Teil nicht auf vom Browser übermittelten Informationen, sondern auf in den Browserkonfigurationsdateien hinterlegten Eigenschaften.

TIPP Die Gerätefähigkeiten kann man auch verwenden, um Browser auf kleineren Endgeräten auf angepasste Seiten umzulenken, z. B.

```
protected void Page_Load(object sender, System.EventArgs e)
{
    if (Request.Browser.IsMobileDevice)
    {
        Response.Redirect("../mobile/Buchung.aspx");
    }
}
```

Gerätespezifische Ausgaben

Um gerätespezifische Ausgaben zu erzeugen, die nicht auf allen Endgeräten unterstützt werden, kann das `DeviceSpecific`-Steuerelement verwendet werden. Innerhalb dieses Steuerelements lassen sich für verschiedene unterstützte bzw. nicht unterstützte Fähigkeiten separate Darstellungen definieren (`<Choice>`). Dabei bezieht sich `<Choice>` auf einen Gerätefilter, der in der `web.config`-Datei definiert sein muss.

Der VWD bietet im Smarttagmenü des Steuerelements `DeviceSpecific` sowohl Hilfe bei der Erstellung (*Vorlagen bearbeiten*) von spezifischen Ausgaben als auch Gerätefiltern (*Vorlagenoptionen*). Leider ist die begriffliche Unterscheidung zwischen den beiden Konzepten – zumindest in der deutschen Version des VWD – wieder einmal sehr schlecht.

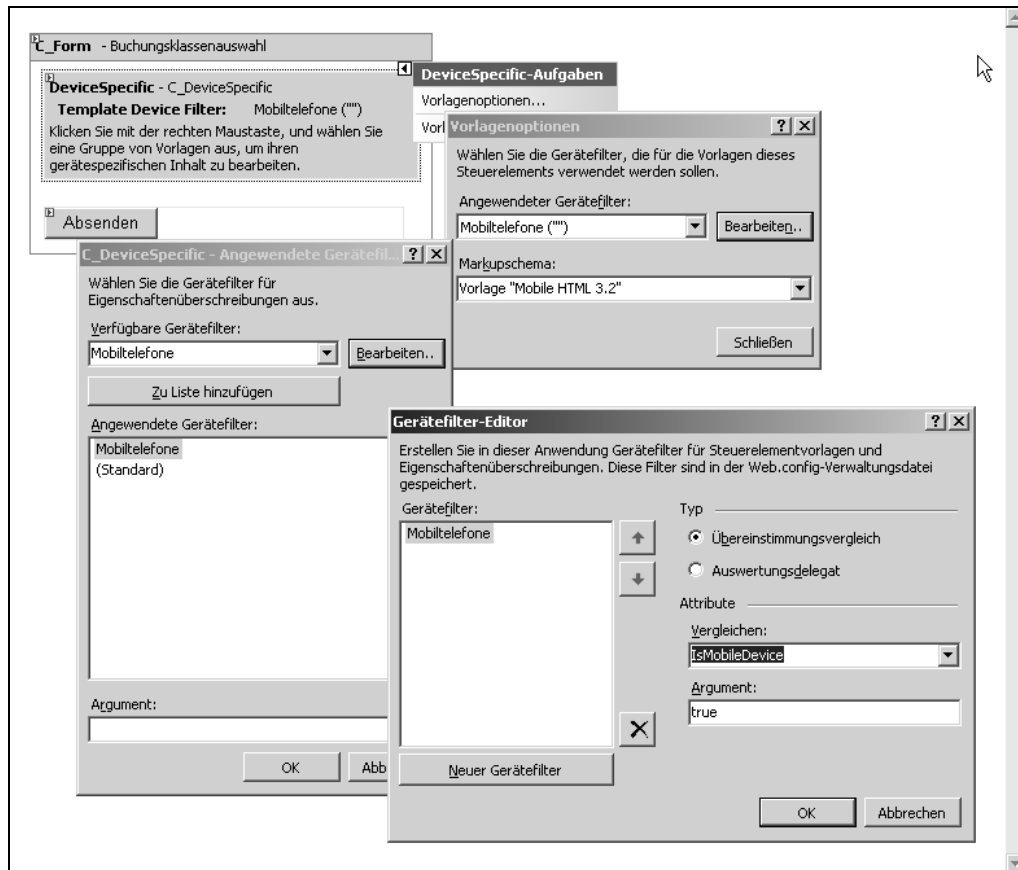


Abbildung 49.4 Erstellung eines Gerätefilters mit dem VWD

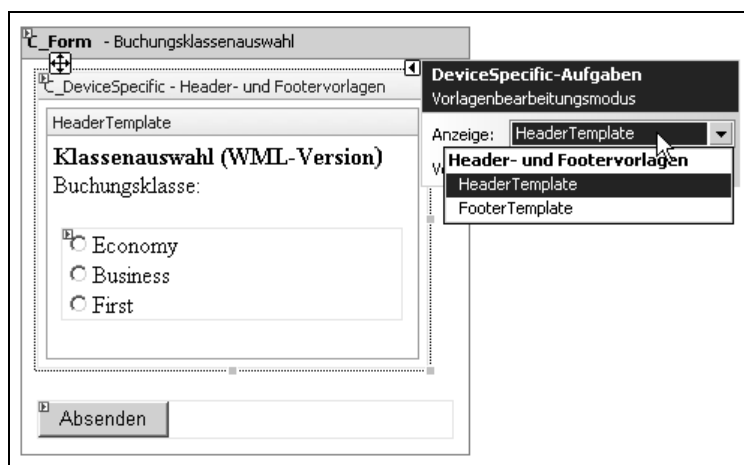


Abbildung 49.5 Erstellung einer gerätespezifischen Ausgabe

Beispiel

Das folgende Beispiel zeigt

- die Definition eines Gerätefilters, der zwischen mobilen und nicht mobilen Geräten unterscheidet.
- die Erstellung einer Seite, die für alle mobilen Geräte eine Liste mit Optionsschaltflächen und für die nicht mobilen Geräte ein Listenfeld zeigt. Die nicht mobilen Geräte erhalten zusätzlich ein Logo.

```
<deviceFilters>
  <filter name="Mobiltelefone" compare="IsMobileDevice" argument="true" />
</deviceFilters>
```

Listing 49.7 Definition eines Gerätefilters in der *web.config*-Datei

```
<%@ Page Language="C#" AutoEventWireup="false" CodeFile="GeraeteSpezifischeAusgabe.aspx.cs"
Inherits="Mobile_GeraeteSpezifischeAusgabe" %>

<% Register TagPrefix="mobile" Namespace="System.Web.UI.MobileControls" Assembly="System.Web.Mobile" %>
<html xmlns="http://www.w3.org/1999/xhtml">
<body>
  <mobile:Form Title="Buchungsklassenauswahl" ID="C_Form" Runat="server">
    <mobile:DeviceSpecific ID="C_DeviceSpecific" Runat="server">
      <Choice Filter="Mobiltelefone" Xmlns="http://schemas.microsoft.com/mobile/html32template">
        <HeaderTemplate>
          <b>Klassenauswahl (WML-Version)</b>
        </HeaderTemplate>
        <FooterTemplate>
          <hr/>
          <small>(C) WorldWideWings</small></FooterTemplate>
      </Choice>
      <Choice>
        <HeaderTemplate>
          <mobile:Image ID="Image1" Runat="server" ImageUrl="~/App_Img/WorldWideWings_Logo.jpg">
          </mobile:Image>
          <b>Klassenauswahl (HTML-Version)</b>
        </HeaderTemplate>
        <FooterTemplate>
          <hr/>
          <small>(C) WorldWideWings</small>
        </FooterTemplate>
      </Choice>
    </mobile:DeviceSpecific>
    <p>
      <mobile:Command ID="C_Absenden" Runat="server">Absenden</mobile:Command></p>
  </mobile:Form></body></html>
```

Listing 49.8 Einsatz eines Gerätefilters [/Mobile/GeraeteSpezifischeAusgabe.aspx]



Abbildung 49.6 Darstellung der Seite im Microsoft Internet Explorer 6.0



Abbildung 49.7 Darstellung der Seite im Microsoft Mobile Explorer 3.0

Erstellung selbst definierter mobiler Steuerelemente

Für die Erstellung eigener mobiler Steuerelemente gibt es zwei Optionen, die direkt vergleichbar sind mit den Optionen für normale Steuerelemente:

- Mobile User Control
- Mobile Custom Control

Ein Mobile User Control wird über die Elementvorlage *Mobiles Web-Benutzersteuerelement* erstellt. Das Steuerelement besteht aus einer *.ascx*-Datei und – optional – einer Hintergrundcodedatei. Die Seitenklasse ist von `System.Web.UI.MobileControls.MobileUserControl` abgeleitet. Bei der Erstellung und Verwendung gelten die gleichen Regeln wie bei den normalen User Controls.

Um ein neues mobiles Steuerelement als Custom Control von Grund auf zu entwickeln, kann man die grundlegenden Steuerelementfunktionalitäten von der abstrakten `System.MobileControl`-Klasse ableiten. Die `MobileControl`-Klasse wird direkt von `System.Web.UI.Control` abgeleitet und verfügt deswegen über dessen Eigenschaften über Inhalt und Style. Alle existierenden mobilen Steuerelemente sind ebenfalls von `MobileControl` abgeleitet. Nach dem Ableiten der Grundfunktionalität von einer Basisklasse kann man die Darstellung eines Steuerelements definieren, indem man die Ausgabe mittels der Methoden der Klasse `System.Web.UI.MobileControls.Adapters.HtmlMobileTextWriter` generiert. Diese ermöglichen die allgemeine Definition der Darstellung des späteren Steuerelements, wie dies auch für die benutzerdefinierten Steuerelemente geschehen kann.