

KAPITOLA 10

Bohatě vybavené datové ovládací prvky

V předchozí kapitole jste viděli, jak se vykonávají dotazy pomocí ovládacích prvků pro zdroje dat, s asistencí vlastní třídy pro přístup k datům, nebo bez ní. V rámci těchto prací jste také použili některé bohatě vybavené datové ovládací prvky ASP.NET, jako je `GridView`. Nijak jste se ovšem nenořili do funkcionality, již tyto ovládací prvky poskytují.

V této kapitole se pěkně zblízka podíváte na tři nejmocnější datové ovládací prvky, které ASP.NET nabízí: `GridView`, `DetailsView` a `FormView`. Nezapomeneme na nový prvek `ListView`. Naučíte se doladit jejich formátování a pod svou kontrolu dostanete takové schopnosti, jako je výběr, řazení, filtrování a šablony. Seznámíte se také s pokročilými scénáři, mezi něž patří zobrazování obrázků, výpočet souhrnů, a vytváření seznamu typu hlavní záznam – sdružené záznamy podrobností (master-details) v jediném ovládacím prvku.

EVOLUCE BOHATĚ VYBAVENÝCH DATOVÝCH OVLÁDACÍCH PRVKŮ ASP.NET

Bohatě vybavené (rich) datové ovládací prvky se v ASP.NET změnily víc než jakákoliv jiná sada ovládacích prvků. Ve snaze poskytnout vývojářům co nejlepší ovládací prvky pro zobrazování dat, a přitom s minimem nezbytného kódu, byly ovládací prvky z ASP.NET 1.x (`DataGrid`, `DataList` a `Repeater`) nahrazeny v ASP.NET 2.0 mocnějšími nástroji (`GridView`, `DetailsView` a `FormView`), přičemž ASP.NET 3.5 ještě přišlo s `ListView`. Původní prvky z ASP.NET 1.x pořád existují, i když většina programátorů ASP.NET žádný z nich už nepoužívá, výjimkou jsou situace, kdy je to vhodné kvůli zpětné kompatibilitě. Následuje přehled všech datových ovládacích prvků ASP.NET.

- **`DataGrid`.** Byl zaveden v ASP.NET 1.0 a v ASP.2.0 kompletně nahrazen ovládacím prvkem `GridView`. `GridView` poskytuje stejnou sadu funkcí (a něco navíc) a zjednodušuje psaní kódu. Prvek `DataGrid` standardně není v *Toolboxu Visual Studia 2005*.
- **`DataList`.** Byl zaveden v ASP.NET 1.0 a většinu jeho funkcionality nahrazuje `GridView`, protože poskytuje obdobnou sadu šablon a mnohem jednodušší model kódu. `DataList` však má jednu funkci, kterou `GridView` nemá: možnost vytvořit víceloupcovou tabulku, ve které každá buňka tvoří separátní záznam. `GridView` tento neobvyklý design nepodporuje, protože si vynucuje, aby každý záznam zabíral oddělený řádek. Stejného výsledku ovšem docílíte s novým ovládacím prvkem `ListView`.
- **`Repeater`.** Byl zaveden v ASP.NET 1.0. Slouží jako holá kostra pro ovládací prvek založený na šabloně, který neposkytuje žádné funkce či parádičky. `Repeater` byl nahrazen prvkem `ListView`, který nabízí stejnou flexibilitu, a jež má navíc zabudované některé funkce, jako výběr, řazení a editování.

... pokračování z předchozí stránky.

- **GridView.** Nejmocnější mřížka v ASP.NET. Zobrazuje záznamy v řádcích tabulky, buď pomocí sloupců, nebo na základě šablony, nebo pomocí nějaké kombinace obojího. GridView byl zaveden v ASP.NET 2.0, v ASP.NET 3.5 se nezměnil.
- **ListView.** Flexibilní ovládací prvek založený na šabloně, který sice nenabízí všechny funkce, které má GridView, nicméně poskytuje možnost realizovat data bez značky <table>. ListView je jednou z novinek ASP.NET 3.5.
- **DetailsView.** DetailsView je nejmocnější ovládací prvek ASP.NET pro úlohy, ve kterých potřebujete v jednom okamžiku zobrazovat data z jediného záznamu. DetailsView používá tabulkové rozvržení a podporuje šablony. DetailsView byl zaveden v ASP.NET 2.0, v ASP.NET 3.5 se nezměnil.
- **FormView.** Je podobně jako DetailsView navržen k zobrazování dat z jediného záznamu. Na rozdíl od něj ovšem vyžaduje použití nějakých šablon. Jeho jediná skutečná výhoda spočívá v možnosti zobrazit data v designu, které není tabulkový. FormView byl zaveden v ASP.NET 2.0 a v ASP.NET 3.5 se nezměnil.

Tato kapitola se zaměřuje na prvky GridView, ListView, DetailsView a FormView (v uvedeném pořadí).

Ovládací prvek GridView

Jestliže jste programovali s ASP.NET 1.x, patrně jste používali původní ovládací prvek DataGrid. Když v týmu ASP.NET čelili výzvě rozšířit a zdokonalit prvek DataGrid tak, aby byla současně zachována zpětná kompatibilita, rozhodli se, že pro ASP.NET 2.0 vytvoří zcela nový ovládací prvek. Tímto ovládacím prvkem, který zůstal v ASP.NET 3.5 beze změny, je GridView.

GridView je neobyčejně flexibilní ovládací prvek pro zobrazování dat v mřížce skládající se z řádků a sloupců. Obsahuje širokou škálu interně zabudovaných schopností, jako jsou výběr, stránkování, řazení a editování, přičemž se dá rozšiřovat prostřednictvím šablon. Skvělou předností GridView oproti DataGrid je jeho podpora scénářů "bez kódu". Pomocí GridView se dá vyřešit mnoho běžně se vyskytujících úloh, jako jsou stránkování a výběr, aniž byste museli psát nějaký kód. S ovládacím prvkem DataGrid jste byli například nuceni obsluhovat události, pokud jste chtěli implementovat takové schopnosti.

GridView jste už v akci viděli v předchozí kapitole. Nicméně jste se vůbec nezabývali tím, jak ho přizpůsobit, aby zobrazoval data přesně tak, jak potřebujete.

Definice sloupců

V ukázkách GridView, s nimiž jste se setkali doposud, byla vlastnost GridView.AutoGenerateColumns nastavena na true. V takovém případě prozkoumá GridView datový objekt pomocí reflexe a vyhledá všechny sloupce (záznamu), nebo vlastnosti (vlastního objektu). Pak vytvoří své sloupce (každý z nich odpovídá jednomu sloupci, resp. vlastnosti) v tom pořadí, v jakém odpovídající sloupec, resp. vlastnost, našel.

Automatické generování sloupců je šikovné, když potřebujete rychle vytvořit nějaké testovací stránky, nicméně neposkytuje takovou flexibilitu, jakou obvykle potřebujete v praxi. Co když chcete některé sloupce skrýt, změnit jejich pořadí, nebo nakonfigurovat nějaký aspekt jejich zobrazení, jako je formátování nebo text v záhlaví? Ve všech těchto případech je potřeba nastavit AutoGenerateColumns na false a definovat sloupce v sekci <Columns> značky ovládacího prvku GridView.

TIP Je možné mít `AutoGenerateColumns` nastavenou na hodnotu `true`, a současně definovat sloupce v sekci `<Columns>`. V takovém případě se sloupce, které explicitně definujete, automaticky přidají před vygenerované sloupce. Tuto techniku jsme využili v předchozí kapitole, kde jsme vytvořili `GridView` s automaticky vygenerovanými sloupci vázanými na data, a současně jsme ručně specifikovali další sloupec s ovládacími prvky pro editovací akce. Nejvíce flexibility ovšem získáte, pokud všechny sloupce definujete explicitně.

Existuje několik typů sloupců, které naleznete v tabulce 10-1. Pořadí značek sloupců určuje pořadí sloupců v `GridView` (zprava doleva).

Tabulka 10-1. Typy sloupců.

Sloupec	Popis
<code>BoundField</code>	Zobrazí text z pole (field) ve zdroji dat.
<code>ButtonField</code>	Zobrazí tlačítko pro každý prvek v seznamu.
<code>CheckBoxField</code>	Zobrazí zaškrtnávací políčko pro každý prvek v seznamu. Použije se automaticky pro sloupec typu <code>true/false</code> . (V SQL Serveru jsou to sloupce s datovým typem <code>bit</code>).
<code>CommandField</code>	Poskytuje tlačítka pro výběr nebo editaci.
<code>HyperlinkField</code>	Tento sloupec zobrazí svůj obsah (pole ze zdroje dat nebo statický text) v podobě hypertextového odkazu.
<code>ImageField</code>	Tento sloupec zobrazí obrázek z binárního pole (za předpokladu, že se dá úspěšně interpretovat jako některý z podporovaných formátů obrázku).
<code>TemplateField</code>	Tento sloupec umožňuje pomocí vlastní šablony specifikovat vícenásobná pole (multiple fields), vlastní ovládací prvky, nebo libovolný HTML kód. Poskytuje nejvyšší úroveň kontroly nad zobrazením, nicméně vám dá také nejvíce práce.

Základním typem sloupce je `BoundField`, který se váže na jeden sloupec v datovém objektu. Takto např. vypadá definice pro jediný vázaný datový sloupec ze zdroje dat, v němž se zobrazí obsah pole `EmployeeID`:

```
<asp:BoundField DataField="EmployeeID" HeaderText="ID" />
```

Oproti automaticky generovanému sloupci už tu máme jedno vylepšení – záhlaví bylo změněno z `Employee-ID` na `ID`.

Když poprvé vytvoříte `GridView`, vlastnost `AutoGenerateColumns` nenastavujete, takže se použije výchozí hodnota `true`. Když ho svážete s nějakým ovládacím prvkem zdroje dat, nic se nezmění. Jakmile ale kliknete na odkaz `Refresh Schema` ovládacího prvku zdroje dat, přepne se vlastnost `AutoGenerateColumns` na `false`, a Visual Studio přidá značku `<asp: BoundField>` pro každé pole (fields), které najde ve zdroji dat. Tento přístup má několik výhod:

- Nastavením vlastností objektu sloupce si můžete snadno přizpůsobit pořadí sloupců, záhlaví sloupců a další detaily.
- Můžete skrýt sloupce, které nechcete zobrazovat. Jednoduše odstraníte odpovídající značky sloupců. (Tuto techniku byste ovšem neměli používat příliš často – pokud nechcete nějaká data zobrazit, je mnohem lepší zmenšit objem získávaných dat.)

TIP *Sloupce se dají také skrýt programátorsky. Sloupce se skrývají pomocí kolekce `Columns` ovládacího prvku `GridView`. Nastavíte-li například `GridView1.Columns[2].Visible` na `false`, skryjete třetí sloupec. Skryté sloupce se ve vygenerovaném kódu HTML vůbec neobjeví.*

- Explicitně definované sloupce jsou rychlejší než automaticky generované sloupce, protože automaticky generované sloupce nutí `GridView`, aby se při běhu obracel na zdroj dat.
- Můžete přidávat sloupce navíc, pokud chcete kombinovat výběr, editaci a další funkce.

TIP *Pokud modifikujete zdroj dat takovým způsobem, že bude vracet jinou sadu sloupců, je možné regenerovat sloupce `GridView`. Stačí vybrat `GridView` a v jeho inteligentní značce kliknout na odkaz `Refresh Schema`. Tím se vyhodí veškeré vaše vlastní sloupce, které jste přidali (jako například sloupec, co obsahuje ovládací prvky pro editaci).*

Podívejte se na kompletní deklaraci `GridView` s explicitně definovanými sloupci:

```
<asp:GridView ID="gridEmployees" runat="server" DataSourceID="sourceEmployees"
    AutoGenerateColumns="False">
    <Columns>
        <asp:BoundField DataField="EmployeeID" HeaderText="ID" />
        <asp:BoundField DataField="FirstName" HeaderText="First Name" />
        <asp:BoundField DataField="LastName" HeaderText="Last Name" />
        <asp:BoundField DataField="Title" HeaderText="Title" />
        <asp:BoundField DataField="City" HeaderText="City" />
    </Columns>
</asp:GridView>
<asp:SqlDataSource ID="sourceEmployees" runat="server"
    ConnectionString="<%$ ConnectionStrings:Northwind %>"
    ProviderName="System.Data.SqlClient"
    SelectCommand="SELECT EmployeeID, FirstName, LastName,
        BirthDate, Title, City FROM Employees">
</asp:SqlDataSource>
```

Když explicitně deklarujete vázaný sloupec, máte příležitost nastavit celou řadu jeho vlastností. Tyto vlastnosti jsou podrobně vypsány v tabulce 10-2.

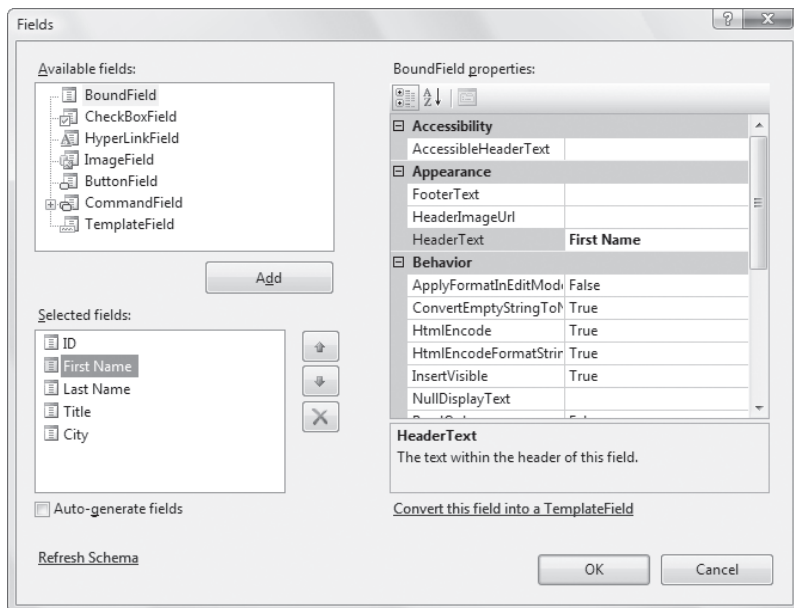
Tabulka 10-2. Vlastnosti sloupce typu `BoundField` (vázaný na data).

Vlastnost	Popis
<code>DataField</code>	Název pole (v řádce) nebo vlastnosti (objektu) prvku dat, který chcete v tomto sloupci zobrazit.
<code>DataFormatString</code>	Tato vlastnost formátuje hodnoty pole (field). Hodí se například, když potřebujete správně zobrazovat peněžní částky nebo datum. V předchozích verzích .NET bylo potřeba nastavit vlastnost <code>HtmlEncode</code> na <code>false</code> , jestliže jste chtěli použít vlastnost <code>DataFormatString</code> . V současnosti toto není požadováno.

Vlastnost	Popis
ApplyFormatInEditMode	<i>Je-li true, formátovací řetězec se bude aplikovat pouze tehdy, když se hodnota objeví v textovém boxu editačního režimu. Výchozí hodnota je false, což znamená, že se použije normální podkladový formát (částka se objeví ve tvaru 1143.02, nikoliv v naformátovaném tvaru \$1,143.02 či 1 143,02 Kč).</i>
FooterText HeaderText HeaderImageUrl	<i>První dvě vlastnosti nastavují text záhlaví a zápatí mřížky, pokud má mřížka záhlaví (ShowHeader je true) a zápatí (ShowFooter je true). Do záhlaví se obvykle dává vypovídající nadpis sloupce, zatímco v zápatí bývá nějaká dynamicky vypočítaná hodnota, například součet (tuto techniku předvedeme v oddílu "Souhrny v GridView" poblíž konce kapitoly). Chcete-li v záhlaví zobrazit místo textu obrázek, nastavte vlastnost HeaderImageUrl.</i>
ReadOnly	<i>Je-li true, není možné v režimu editace měnit hodnotu sloupce. Jednoduše se neposkytne žádný editační ovládací prvek. Pouze pro čtení obvykle bývají sloupce primárního klíče.</i>
InsertVisible	<i>Je-li false, nemůže se hodnota sloupce nastavit na režim vkládání. To se hodí tehdy, pokud chcete hodnotu daného sloupce nastavovat programátorsky, nebo ji založit na nějaké výchozí hodnotě definované v databázi.</i>
Visible	<i>Je-li false, sloupec nebude na stránce vidět (a nebude se pro něj generovat žádný kód HTML). Tato vlastnost poskytuje pohodlný způsob, jak programátorsky skrývat a zase odkrývat konkrétní sloupce, a tím měnit všeobecný vzhled dat.</i>
SortExpression	<i>Výraz, který se dá připojit na konec dotazu, aby se data seřadila podle daného sloupce. Používá se v součinnosti s řazením, viz oddíl "Řazení v GridView".</i>
HtmlEncode	<i>Je-li true (výchozí), pak se veškerý text HTML zakóduje, aby speciální znaky netropily na stránce nějakou neplechu. Kódování HTML můžete vypnout, chcete-li vložit nějakou funkční značku HTML (jako například hypertextový odkaz), nicméně tento přístup není bezpečný. Vždy je lepší použít kódování HTML na všechny hodnoty, a jinou funkcionalitu pak poskytovat přes reakci na události GridView související s výběrem.</i>
NullDisplayText	<i>Text, který se má zobrazit pro hodnoty null. Výchozí je prázdný řetězec, můžete ale uvést nějakou jinou explicitní výchozí hodnotu, jako je třeba "(hodnota chybí)".</i>
ConvertEmptyStringToNull	<i>Pokud je true, tak předtím, než bude potvrzena editace, budou nahrazeny všechny prázdné řetězce hodnotami null.</i>
ControlStyle HeaderStyle FooterStyle ItemStyle	<i>Konfigurují vzhled pouze pro tento sloupec, překrývají styly řádku. O stylech se více informací dozvíte v průběhu kapitoly.</i>

Nechcete-li sloupce konfigurovat ručně, vyberte GridView a klikněte v okně Properties na tři tečky (...) napravo od vlastnosti Columns. Uvidíte dialogové okno Fields, ve kterém můžete sloupce přidávat, od-

straňovat a vylepšovat (viz následující obrázek 10-1).



Obrázek 10-1. Konfigurace sloupců ve Visual Studiu.

I když jste se už seznámili s podpůrnými konstrukcemi `GridView`, stále jste na začátku dlouhého seznamu jeho schopností vyšší úrovně. V následujících oddílech se budeme zabývat těmito tématy:

- **Formátování.** Jak naformátovat řádky a hodnoty dat.
- **Výběr.** Jak umožnit uživateli vybrat řádek v `GridView`, a jak na to reagovat.
- **Řazení.** Jak dynamicky změnit uspořádání `GridView` (jakožto reakci na kliknutí v záhlaví sloupce).
- **Stránkování.** Jak rozdělit rozsáhlou sadu výsledků na několik stránek dat, ať už automaticky, nebo pomocí vlastního kódu pro stránkování.
- **Šablony.** Jak plně dostat pod kontrolu vzhled, formátování a editaci tím, že si nadefinujete šablony.

Formátování a GridView

Formátování se skládá z několika vzájemně provázaných úloh. Nejprve musíte zajistit, aby se řádně prezentovaly hodnoty vyjadřující datum, peněžní částky a jiné číselné hodnoty. To zařídíte pomocí vlastnosti `DataFormatString`. Pak budete chtít aplikovat na každý aspekt mřížky, počínaje záhlavím a konče hodnotami dat, perfektní směs voleb pro barvy, písmo, orámování a zarovnání. `GridView` tyto funkce podporuje prostřednictvím stylů. A nakonec můžete zachycovat události, prozkoumávat data řádků, a aplikovat formátování na konkrétní data programátorsky. V příštích oddílech se podíváme na všechny tyto techniky.

Prvek `GridView` sám vystavuje několik formátovacích vlastností, které samy sebe dostatečně vysvětlují svými názvy, a proto se zde jimi nezabýváme. Patří mezi ně vlastnost `GridLines` (přidání, nebo skrytí orámování tabulky), `CellPadding` a `CellSpacing` (rozestupy mezi buňkami), `Caption` a `CaptionAlign` (vztahují se k titulku celé mřížky).

TIP Chcete vytvořit `GridView`, který by roloval uvnitř webové stránky? Jde to velmi snadno. Jednoduše umístěte `GridView` do ovládacího prvku `Panel`, nastavte vhodnou velikost panelu a vlastnost `Panel.Scrollbars` na hodnotu `Auto`, `Vertical`, nebo `Both`.

Formátování sloupců vázaných na data

Každý sloupec typu `BoundField` poskytuje vlastnost `DataFormatString`. S ní můžete nakonfigurovat vzhled hodnot vyjadřujících čísla nebo datum, a to pomocí formátovacího řetězce.

Formátovací řetězce se obvykle skládají ze zástupných symbolů a indikátorů, které jsou uvnitř složených závorek. Typický formátovací řetězec vypadá takto:

```
{0:C}
```

Nula reprezentuje hodnotu, která se bude formátovat, písmeno vyjadřuje předdefinovaný formátovací styl. C zde znamená formát měny (z anglického slova *Currency*) a slouží k naformátování čísla jako částky se symbolem měny (podle místního nastavení), takže číslo 3400.34 se v USA zobrazí ve tvaru \$3,400.34. Počítač, který je nakonfigurován pro jiné místní nastavení, může zobrazit částku v jiném formátu a s jiným symbolem měny. U nás bude mít tato částka vzhled 3 400,34 Kč. Podívejte se na sloupec, ve kterém se používá uvedený formátovací řetězec:

```
<asp:BoundField DataField="UnitPrice" HeaderText="Price"
    DataFormatString="{0:C}" />
```

Chcete-li použít jiný formát měny, můžete změnit lokalizační nastavení webového serveru (v sekci Místní a jazyková nastavení (Regional and Language Options) ovládacích panelů (Control Panel)). Nebo můžete v kódu pro vaši webovou aplikaci nastavit jinou kulturu, nebo použít prvek `<globalization>` v souboru `web.config` (obě techniky se předvádějí v kapitole 18).

POZNÁMKA Potřebujete-li ve své aplikaci zobrazovat řetězce různých měn, zabudované formátování vám stačit nebude, takže to budete muset vyřešit nějak po svém. Typickou strategií je vytvořit strukturu `Money`, která bude obalovat peněžní částky a kulturu. Pak můžete překrýt `Money.ToString()` a vrátit takovou řetězcovou reprezentaci, jakou právě potřebujete.

Některé další volby pro číselné hodnoty jsou uvedeny v tabulce 10-3.

Tabulka 10-3. Formátovací řetězce pro čísla.

Typ	Formátovací řetězec	Ukázka
Peněžní částky	{0:C}	V USA \$1,234.50, u nás 1 234,50 Kč. Záporná čísla jsou v USA v závorkách (\$1,234.50), u nás se uvádí -1 234,50 Kč.
Vědecký (exponenciální)	{0:E}	1,234500E+003 (u nás).
Procenta	{0:P}	45.6%

Typ	Formátovací řetězec	Ukázka
Pevný počet desetinných míst	{0:F?}	Záleží na tom, kolik desetinných míst uvedete. {0:F3} vede na 123.400, {0:F0} pak na 123.

Další příklady naleznete v nápovědě Visual Studia. Pro hodnoty vyjadřující datum nebo čas je rovněž k dispozici rozsáhlý seznam voleb. Pokud například chcete zobrazit hodnotu `BirthDate` (datum narození) ve tvaru měsíc/den/rok (tzn. ve tvaru 12/30/05), mohli byste použít tento sloupec:

```
<asp:BoundField DataField="BirthDate" HeaderText="Birth Date"
    DataFormatString="{0:MM/dd/yy}" />
```

Další ukázky jsou uvedeny v tabulce 10-4.

Tabulka 10-4. Formátovací řetězce pro datum a čas.

Typ	Formátovací řetězec	Ukázka
Krátké datum	{0:d}	M/d/yyyy (například v USA 10/30/2005, u nás 30.10.2005).
Dlouhé datum	{0:D}	dddd, MMMM dd, yyyy (například v USA Sunday, January 30, 2005, u nás neděle 30. leden 2005).
Dlouhé datum a krátký čas	{0:f }	dddd, MMMM dd, yyyy HH:mm aa (například: Sunday, January 30, 2005 10:00 AM).
Dlouhé datum a dlouhý čas	{0:F}	dddd, MMMM dd, yyyy HH:mm:ss aa (například: Monday, January 30, 2005 10:00:23 AM).
Standard ISO, podle kterého lze řadit	{0:s}	yyyy-MM-dd HH:mm:ss (například: 2005-01-30 10:00:23).
Měsíc a den	{0:M}	MMMM dd (například v USA January 30, u nás 30. leden).
Všeobecný formát (General)	{0:G}	M/d/yyyy HH:mm:ss aa (závisí na místních nastaveních) (například: 10/30/2002 10:00:23 AM).

znaky pro formátovací řetězce nejsou specifické pro `GridView`. Můžete je také používat u jiných ovládacích prvků, ve výrazech vázání dat v šablonách (to uvidíte později v této kapitole), a také jako parametry mnoha různých metod. Například typy `Decimal` a `DateTime` vystavují své vlastní metody `ToString()`, které přebírají formátovací řetězec, takže můžete hodnoty formátovat ručně.

Styl

`GridView` vystavuje bohatý formátovací model založený na stylech. Celkem můžete pro `GridView` nastavit osm stylů, které se podrobněji popisují v tabulce 10-5.

Tabulka 10-5. Styly GridView.

Styl	Popis
HeaderStyle	Konfiguruje vzhled řádku záhlaví, který obsahuje nadpisy sloupců, pokud jste se rozhodli je zobrazovat (<code>ShowHeader</code> má hodnotu <code>true</code>).

Styl	Popis
RowStyle	Konfiguruje vzhled jednotlivých řádků.
AlternatingRowStyle	Pokud je tento styl nastaven, na každý druhý řádek se aplikuje dodatečné formátování. Toto formátování funguje jako dodatek ke stylu RowStyle. Pokud například nastavíte pomocí RowStyle nějaké písmo, použije se rovněž na každý druhý řádek, pokud explicitně nenastavíte prostřednictvím AlternatingRowStyle nějaké jiné písmo.
SelectedRowStyle	Konfiguruje vzhled aktuálně vybraného řádku. Funguje jako dodatek k formátování s RowStyle.
EditRowStyle	Konfiguruje vzhled řádku, který se nachází v režimu editace. Funguje jako dodatek k formátování s RowStyle.
EmptyDataRowStyle	Konfiguruje styl, který se použije pro jediný prázdný řádek v tom speciálním případě, kdy vázaný datový objekt neobsahuje žádné řádky.
FooterStyle	Konfiguruje vzhled řádku zápatí u spodní strany GridView, pokud jste se rozhodli je zobrazovat (ShowFooter má hodnotu true).
PagerStyle	Konfiguruje vzhled řádku s odkazy na další stránky, pokud máte zapnuto stránkování (AllowPaging je nastavena na true).

Popisované styly nejsou vlastnostmi s jedinou hodnotou. Každý styl vystavuje objekt Style, který obsahuje vlastnosti pro volbu barev (ForeColor a BackColor), orámování (BorderColor, BorderStyle a BorderWidth), rozměry řádku (Height a Width), zarovnání řádku (HorizontalAlign a VerticalAlign), a pro konfiguraci vzhledu textu (Font a Wrap). Tyto vlastnosti stylů umožňují upravit téměř každý aspekt vzhledu daného prvku. A pokud nechcete všechna nastavení týkající se vzhledu natvrdo nastavovat ve webové stránce, můžete nastavit vlastnost CssClass objektu stylu na nějakou CSS třídu, která je definována v propojeném stylovém předpisu (linked stylesheet, o stylech se více dozvíte v kapitole 16).

Definice stylů

Když se nastavují vlastnosti stylu, používají se dvě různé syntaxe (obě uvidíte v této kapitole). První je syntaxe s pomlčkou (anglicky se jí říká object-walker syntax), která vyjadřuje vlastnosti a podvlastnost ve formě atributů otevírající značky GridView. Ukázka:

```
<asp:GridView runat="server" ID="grid" RowStyle-ForeColor="DarkBlue" ... />
```

Alternativní syntaxe je založena na vnořených značkách, jako zde:

```
<asp:GridView runat="server" ID="grid" ...>
  <RowStyle ForeColor="DarkBlue" ... />
  ...
</asp:GridView>
```

Oba přístupy jsou ekvivalentní. Při nastavování vlastností stylu ovšem musíte učinit ještě jedno rozhodnutí. Můžete totiž specifikovat globální vlastnosti stylu, které ovlivňují všechny sloupce mřížky (jako v předchozích ukázkách), nebo definovat styly, jež budou specifické pro jednotlivé sloupce. Chcete-li vytvořit styl specifický

pro daný sloupec, musíte přidat atributy stylu (nebo vnořené značky) dovnitř značky patřičného sloupce, jehož vzhled chcete upravit, jak to vidíte zde:

```
<asp:GridView runat="server" ID="grid" ...>
  <Columns>
    <asp:BoundField DataField="EmployeeID"
      HeaderText="ID" ItemStyle-Width="30px" />
    ...
  </Columns>
</asp:GridView>
```

A takto vypadá ekvivalentní kód s vnořenou značkou:

```
<asp:GridView runat="server" ID="grid" ...>
  <Columns>
    <asp:BoundField DataField="EmployeeID" HeaderText="ID">
      <ItemStyle Width="30px" />
    </asp:BoundField>
    ...
  </Columns>
</asp:GridView>
```

Tato technika se používá hlavně tehdy, když potřebujete definovat konkrétní šířky sloupců. Pokud je neurčíte, ASP.NET udělá sloupce právě tak široké, aby se do nich vešla data, která obsahují (resp. pokud je zapnuté zalamování, aby se do nich vešel text, aniž aby se muselo rozdělit slovo na konci řádku). Jestliže jsou hodnoty v daném sloupci různě široké, šířka se určí podle nejdelší hodnoty, nebo podle délky záhlaví (rozhoduje, co je delší). Ale i když má mřížka sloupce dostatečně široké, možná byste rádi některý sloupec rozšířili ještě více, aby nevypadal přelplněný vzhledem ke stavu sousedních sloupců. V takovém případě budete muset explicitně specifikovat větší šířku.

Podívejte se na kompletně naformátovanou značku GridView:

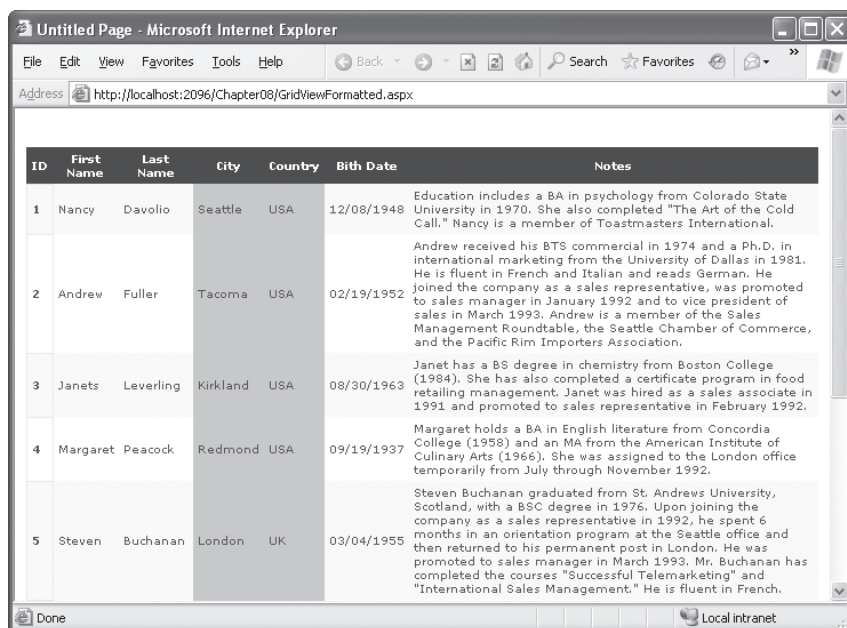
```
<asp:GridView ID="GridView1" runat="server" DataSourceID="sourceEmployees"
  Font-Names="Verdana" Font-Size="X-Small" ForeColor="#333333"
  CellPadding="4" GridLines="None" AutoGenerateColumns="False">
  <HeaderStyle BackColor="#990000" Font-Bold="True" ForeColor="White" />
  <RowStyle BackColor="#FFBDD6" ForeColor="#333333" />
  <AlternatingRowStyle BackColor="White" />
  <Columns>
    <asp:BoundField DataField="EmployeeID" HeaderText="ID">
      <ItemStyle Font-Bold="True" BorderWidth="1" />
    </asp:BoundField>
    <asp:BoundField DataField="FirstName" HeaderText="First Name" />
    <asp:BoundField DataField="LastName" HeaderText="Last Name" />
    <asp:BoundField DataField="City" HeaderText="City">
      <ItemStyle BackColor="LightSteelBlue" />
    </asp:BoundField>
    <asp:BoundField DataField="Country" HeaderText="Country">
      <ItemStyle BackColor="LightSteelBlue" />
    </asp:BoundField>
  </Columns>
</asp:GridView>
```

```

</asp:BoundField>
<asp:BoundField DataField="BirthDate" HeaderText="Birth Date"
    DataFormatString="{0:MM/dd/yyyy}" />
<asp:BoundField DataField="Notes" HeaderText="Notes">
    <ItemStyle Wrap="True" Width="400"/>
</asp:BoundField>
</Columns>
</asp:GridView>

```

V tomto příkladu se pomocí několika vlastností `GridView` nastavuje písmo, upravuje rozestup buněk a orámování buněk. Pomocí stylů se tučně zvýrazní záhlaví a zařídí se, aby se střídala barva pozadí lichých a sudých řádků. Styly specifické pro sloupec jsou použity pro zvýraznění informací o sídle (`City` a `Country`) pomocí jiné barvy pozadí, pro nastavení hodnot ID na tučný řez, a pro explicitní nastavení šířky sloupce `Notes`. Na závěr se prostřednictvím `DataFormatString` naformátují všechny hodnoty obsahující datum narození (sloupec `BirthDate`). Konečný výsledek (v místních nastaveních pro USA) vidíte na obrázku 10-2.



ID	First Name	Last Name	City	Country	Bith Date	Notes
1	Nancy	Davolio	Seattle	USA	12/08/1948	Education includes a BA in psychology from Colorado State University in 1970. She also completed "The Art of the Cold Call." Nancy is a member of Toastmasters International.
2	Andrew	Fuller	Tacoma	USA	02/19/1952	Andrew received his BTS commercial in 1974 and a Ph.D. in international marketing from the University of Dallas in 1981. He is fluent in French and Italian and reads German. He joined the company as a sales representative, was promoted to sales manager in January 1992 and to vice president of sales in March 1993. Andrew is a member of the Sales Management Roundtable, the Seattle Chamber of Commerce, and the Pacific Rim Importers Association.
3	Janets	Leverling	Kirkland	USA	08/30/1963	Janet has a BS degree in chemistry from Boston College (1984). She has also completed a certificate program in food retailing management. Janet was hired as a sales associate in 1991 and promoted to sales representative in February 1992.
4	Margaret	Peacock	Redmond	USA	09/19/1937	Margaret holds a BA in English literature from Concordia College (1958) and an MA from the American Institute of Culinary Arts (1966). She was assigned to the London office temporarily from July through November 1992.
5	Steven	Buchanan	London	UK	03/04/1955	Steven Buchanan graduated from St. Andrews University, Scotland, with a BSC degree in 1976. Upon joining the company as a sales representative in 1992, he spent 6 months in an orientation program at the Seattle office and then returned to his permanent post in London. He was promoted to sales manager in March 1993. Mr. Buchanan has completed the courses "Successful Telemarketing" and "International Sales Management." He is fluent in French.

Obrázek 10-2. Naformátovaný `GridView`.

Konfigurace stylů ve Visual Studiu

Není vůbec nutné kódovat vlastnosti stylu ovládacího prvku `GridView` ručně v jeho značce, protože má bohatě vybavenou podporu pro dobu návrhu. Vlastnosti stylu se dají upravovat v okně `Properties`. Pokud například chcete nakonfigurovat písmo pro záhlaví, rozbalte vlastnost `HeaderStyle`, kde uvidíte vnořenou vlastnost `Font`. Nastavte ji, jak potřebujete. Jediným omezením tohoto přístupu je, že se nedají nastavovat styly pro jednotlivé sloupce – pokud něco takového potřebujete, musíte nejprve otevřít dialog `Fields` (který jste viděli na obrázku 10-1), a v něm editovat vlastnost `Columns`. Pak můžete vybrat patřičný sloupec a nastavit jeho vlastnosti podle svých představ.