
Stručný obsah

Úvod	15
Kapitola 1: Rámce aplikací a stránek	37
Kapitola 2: Serverové ovládací prvky ASP.NET a klientské skripty	93
Kapitola 3: Webové serverové ovládací prvky ASP.NET	133
Kapitola 4: Serverové ovládací prvky pro ověřování platnosti dat	215
Kapitola 5: Vzory stránek	247
Kapitola 6: Témata a vzhledy	279
Kapitola 7: Práce s daty v ASP.NET 3.5	301
Kapitola 8: Správa dat pomocí ADO.NET	385
Kapitola 9: Dotazování jazykem LINQ	459
Kapitola 10: Práce s jazykem XML a dotazy LINQ to XML	499
Kapitola 11: IIS7	559
Kapitola 12: Představení modelu poskytovatelů	587
Kapitola 13: Rozšíření modelu poskytovatelů	625
Kapitola 14: Navigace po webovém sídle	659
Kapitola 15: Personalizace (zosobnění)	713
Kapitola 16: Správa členství a správa rolí	745
Rejstřík	799

Obsah

Úvod	15
Trocha historie	15
Cíle ASP.NET	17
Produktivita vývoje	17
Výkonnost a škálovatelnost	19
Další funkce ASP.NET 3.5	20
Nové vývojářské infrastruktury	20
Kompilační systém ASP.NET	23
Sledování zdraví aplikací ASP.NET	24
Čtení a zápis konfiguračních nastavení	25
Lokalizace	26
Rozšíření rámce stránek	26
Objekty pro přístup k datům	28
Co potřebujete k ASP.NET 3.5	29
Kdo by si měl naši knihu přečíst	29
Čím se kniha zabývá	29
Konvence	34
Zdrojový kód	35
Opravy	36
Poznámka redakce českého vydání knihy	36
Kapitola 1: Rámce aplikací a stránek	37
Možnosti umístění aplikací	37
Vestavěný webový server	38
IIS	39
FTP	40
Webové sídlo vyžadující rozšíření FrontPage	40

Volby struktury stránek ASP.NET	41
Vložený kód	43
Model kódu v pozadí	45
Direktivy stránek ASP.NET 3.5	47
@Page	48
@Master	51
@Control	52
@Import	53
@Implements	55
@Register	55
@Assembly	56
@PreviousPageType	56
@MasterType	56
@OutputCache	57
@Reference	57
Události stránek ASP.NET	58
Práce se zpětným odesláním	59
Zpětné odesílání mezi stránkami	60
Aplikační složky ASP.NET	67
Složka \App_Code	67
Složka \App_Data	71
Složka \App_Themes	72
Složka \App_GlobalResources	72
Složka \App_LocalResources	72
Složka \App_WebReferences	73
Složka \App_Browsers	73
Kompilace	73
Poskytovatelé sestavení	77
Používání vestavěných poskytovatelů sestavení	78
Používání vlastních poskytovatelů sestavení	79
Global.asax	84
Práce s třídami ve VS2008	87
Shrnutí	92
Kapitola 2: Serverové ovládací prvky ASP.NET a klientské skripty	93
Serverové ovládací prvky ASP.NET	93
Typy serverových ovládacích prvků	94
Tvorba pomocí serverových ovládacích prvků	95
Práce s událostmi serverových ovládacích prvků	98
Aplikování stylů na serverové ovládací prvky	100
Pohled na běžné vlastnosti ovládacích prvků	100
Změna stylů pomocí kaskádových šablon	102
Serverové ovládací prvky HTML	105
Pohled na báзовou třídu HtmlControl	107
Pohled na třídu HtmlContainerControl	108
Pohled na všechny třídy HTML	108
Práce se třídou HtmlGenericControl	109
Práce se stránkami a serverovými ovládacími prvky v JavaScriptu	111
Práce s metodou Page.ClientScript.RegisterClientScriptBlock	112

Práce s metodou Page.ClientScript.RegisterStartupScript	114
Práce s metodou Page.ClientScript.RegisterClientScriptInclude	116
Klientské zpětné volání	116
Porovnání typického zpětného odeslání a zpětného volání	116
Použití funkce zpětného volání – jednoduchý přístup	118
Práce s funkcí zpětného volání s jedním parametrem	123
Složitější příklad použití funkce zpětného volání	126
Shrnutí	132
Kapitola 3: Webové serverové ovládací prvky ASP.NET	133
Přehled webových serverových ovládacích prvků	133
Serverový ovládací prvek Label	134
Serverový ovládací prvek Literal	136
Serverový ovládací prvek TextBox	137
Práce s metodou Focus()	138
Použití vlastnosti AutoPostBack	138
Práce s atributem AutoCompleteType	140
Serverový ovládací prvek Button	141
Vlastnost CausesValidation	141
Vlastnost CommandName	141
Tlačítka fungující s klientským JavaScriptem	143
Serverový ovládací prvek LinkButton	144
Serverový ovládací prvek ImageButton	145
Serverový ovládací prvek HyperLink	146
Serverový ovládací prvek DropDownList	147
Vizuální odstranění položek z kolekce	150
Serverový ovládací prvek ListBox	151
Jak uživatelům umožnit vícenásobný výběr	151
Příklad použití ovládacího prvku ListBox	152
Doplnění položek do kolekce	154
Serverový ovládací prvek CheckBox	155
Jak zjistit stav zaškrťovacích políček	156
Jak přiřadit zaškrťovacímu políčku hodnotu	156
Jak zarovnat text kolem zaškrťovacího políčka	157
Serverový ovládací prvek CheckBoxList	157
Serverový ovládací prvek RadioButton	160
Serverový ovládací prvek RadioButtonList	162
Serverový ovládací prvek Image	163
Serverový ovládací prvek Table	164
Serverový ovládací prvek Calendar	167
Volba data v ovládacím prvku Calendar	167
Volba formátu data předávaného prvkem Calendar	168
Výběr dne, týdne a měsíce	169
Práce s datovými rozsahy	169
Změna stylu a chování kalendáře	171
Serverový ovládací prvek AdRotator	175
Serverový ovládací prvek Xml	176

Serverový ovládací prvek Panel	177
Serverový ovládací prvek Placeholder	179
Serverový ovládací prvek BulletedList	180
Serverový ovládací prvek HiddenField	185
Serverový ovládací prvek FileUpload	187
Uploadování souborů pomocí ovládacího prvku FileUpload	187
Jak zajistit pro ASP.NET oprávnění k uploadování souborů	189
Omezení velikosti souborů	190
Uploadování více souborů z jedné stránky	192
Umístění odeslaného souboru do objektu proudu	195
Přesun obsahu souboru z objektu proudu do pole bajtů	195
Serverové ovládací prvky MultiView a View	197
Serverový ovládací prvek Wizard	200
Přizpůsobení boční navigace	202
Pohled na atribut AllowReturn	202
Práce s atributem StepType	202
Doplnění hlavičky do ovládacího prvku Wizard	203
Práce s navigačním systémem ovládacího prvku Wizard	204
Využití událostí ovládacího prvku Wizard	205
Zobrazování elementů formuláře ovládacím prvkem Wizard	206
Serverový ovládací prvek ImageMap	211
Shrnutí	213
Kapitola 4: Serverové ovládací prvky pro ověřování platnosti dat	215
Význam ověřování	215
Klientské versus serverové ověřování	216
Serverové ovládací prvky ASP.NET pro ověření	217
Příčiny ověřování	218
Serverový ovládací prvek RequiredFieldValidator	219
Serverový ovládací prvek CompareValidator	224
Serverový ovládací prvek RangeValidator	227
Serverový ovládací prvek RegularExpressionValidator	231
Serverový ovládací prvek CustomValidator	232
Serverový ovládací prvek ValidationSummary	237
Vypnutí klientského ověřování	240
Hlášení chyb obrázky a zvuky	241
Práce s ověřovacími skupinami	242
Shrnutí	246
Kapitola 5: Vzory stránek	247
Proč potřebujete vzory stránek?	247
Základy vzorů stránek	249
Vytváření kódu vzoru stránek	251
Vytváření kódu stránky obsahu	253
Kombinování typů stránek a jazyků	257
Zadání používaného vzoru stránek	258
Práce s titulkem stránky	259
Práce s ovládacími prvky a vlastnostmi ze vzoru stránek	260

Zadání výchozího obsahu ve vzoru stránek	267
Programové přiřazení vzoru stránek	269
Vnořování vzorů stránek	269
Vzory stránek specifické pro prohlížeče	273
Pořadí událostí	274
Kešování vzorů stránek	275
AJAX ASP.NET a vzory stránek	276
Shrnutí	278
Kapitola 6: Témata a vzhledy	279
Práce s tématy ASP.NET	279
Aplikování tématu na jednu stránku ASP.NET	279
Aplikování tématu na celou aplikaci	281
Odstranění témat ze serverových ovládacích prvků	281
Odstranění témat z webových stránek	282
Práce s tématy a stránkami vzoru	282
Seznámení s atributem StyleSheetTheme	283
Vytváření vlastních témat	283
Vytvoření potřebné struktury složek	283
Vytvoření vzhledu	284
Použití souborů CSS v tématu	286
Jak do témat zahrnout obrázky	289
Definování několika voleb vzhledů	292
Programová práce s tématy	294
Programové přiřazení tématu stránky	294
Programové přiřazení atributu SkinID ovládacího prvku	294
Témata, vzhledy a vlastní ovládací prvky	295
Shrnutí	299
Kapitola 7: Práce s daty v ASP.NET 3.5	301
Ovládací prvky datových zdrojů	301
Ovládací prvek SqlDataSource	303
Ovládací prvek LinqDataSource	313
Ovládací prvek AccessDataSource	318
Ovládací prvek XmlDataSource	319
Ovládací prvek ObjectDataSource	320
Ovládací prvek SiteMapDataSource	324
Konfigurace kešování ovládacího prvku datového zdroje	325
Ukládání informací o připojení	326
Užití připojených seznamových ovládacích prvků s ovládacími prvky datových zdrojů	329
GridView	329
Úprava dat řádků prvku GridView	344
Odstranění dat prvku GridView	351
DetailsView	353
Vkládání, aktualizování a odstraňování dat pomocí prvku DetailsView	359
ListView	361
FormView	369

Ostatní ovládací prvky pracující s daty	373
DropDownList, ListBox, RadioButtonList a CheckBoxList	373
TreeView	374
AdRotator	374
Menu	375
Přímá práce s daty	375
Změny syntaxe prvku DataBinder	376
Datové připojení na XML	377
Výrazy a tvůrci výrazů	377
Shrnutí	383
Kapitola 8: Správa dat pomocí ADO.NET	385
Základní funkce ADO.NET	386
Běžné činnosti ADO.NET	386
Základní obory názvů a třídy ADO.NET	391
Práce s objektem typu Connection	392
Práce s objektem typu Command	394
Práce s objektem typu DataReader	395
Práce s adaptérem dat	397
Práce s parametry	400
Seznámení se třídami DataSet a DataTable	403
Použití databáze Oracle v ASP.NET 3.5	408
Serverový ovládací prvek DataList	410
Seznámení s dostupnými šablonami	411
Práce s ItemTemplate	411
Práce s dalšími šablonami rozvržení	414
Práce s více sloupci	416
Serverový ovládací prvek ListView	417
Seznámení s dostupnými šablonami	417
Práce s šablonami	418
Vytvoření šablony rozvržení	419
Vytvoření šablony ItemTemplate	421
Vytvoření šablony EditItemTemplate	422
Vytvoření šablony EmptyItemTemplate	422
Vytvoření šablony InsertItemTemplate	423
Výsledky	423
Použití Visual Studia pro práci s ADO.NET	425
Vytvoření připojení ke zdroji dat	426
Práce s prvky DataSet	427
Použití prvku DataSet CustomerOrders	432
Asynchronní vykonávání příkazů	436
Asynchronní připojení	458
Shrnutí	458
Kapitola 9: Dotazování jazykem LINQ	459
LINQ to Objects	459
Tradiční dotazové metody	459
Náhrada tradičních dotazů dotazy LINQ	468
Seskupování dat	476
Další operátory LINQ	477

Spojení LINQ	477
Stránkování pomocí LINQ	479
LINQ to XML	480
Spojování dat XML	484
LINQ to SQL	485
Vkládací, aktualizací a odstraňovací dotazy pomocí LINQ	493
Rozšíření LINQ	497
Shrnutí	498
Kapitola 10: Práce s jazykem XML a dotazy LINQ to XML	499
Základy XML	500
InfoSet jazyka XML	502
Definice schématu XML – XSD	503
Úprava dokumentu XML a schématu XML ve Visual Studiu 2008	505
Třídy XmlReader a XmlWriter	507
Použití třídy XmlDocument namísto XmlReader	510
Použití schématu ve spojení s XmlTextReader	511
Ověřování podle schématu pomocí XmlDocument	514
Optimalizace NameTable	515
Získání typů CLR rámce .NET z XML	517
ReadSubtree a XmlSerialization	519
Vytvoření objektů CLR z XML pomocí LINQ to XML	521
Vytvoření XML pomocí XmlWriter	522
Vytvoření XML pomocí LINQ for XML	524
Vylepšení tříd XmlReader a XmlWriter ve verzi 2.0	527
XmlDocument a XPathDocument	528
Problémy modelu DOM	528
XPath, XPathDocument a XmlDocument	528
Prvky DataSet	533
Ukládání datových množin do XML	533
XmlDataDocument	534
Ovládací prvek XmlDataSource	536
XSLT	540
XslCompiledTransform	542
Webový serverový ovládací prvek XML	544
Ladění XSLT	546
Databáze a XML	546
FOR XML AUTO	547
SQL Server 2005 a datový typ XML	552
Shrnutí	558
Kapitola 11: IIS7	559
Modulární architektura IIS7	559
IIS-WebServer	560
IIS-WebServerManagementTools	563
IIS-FTPPublishingService	563
Rozšiřitelná architektura IIS7	564
Integrovaný kanál IIS7 a ASP.NET	564

Sestavení přizpůsobeného webového serveru	566
Závislosti aktualizací	566
Instalace IIS7 na Windows Vista	567
Instalace IIS7 na Windows Server 2008	567
Volby instalace z příkazového řádku	568
Možnost bezobslužné instalace	569
Aktualizace	570
Správce Internet Information Services (IIS)	570
Aplikační fondy	571
Webová sídla	575
Hierarchická konfigurace	576
Delegování	581
Přesun aplikace z IIS6 na IIS7	583
Shrnutí	585
Kapitola 12: Představení modelu poskytovatelů	587
Seznámení s poskytovateli	588
Model poskytovatelů v ASP.NET 3.5	588
Nastavení poskytovatele pro práci s Microsoft SQL Server 7.0, 2000, 2005 nebo 2008	591
Poskyvatelé členství	597
Poskyvatelé rolí	601
Poskyvatel zosobnění	605
Poskyvatel SiteMap	607
Poskyvatelé SessionState	608
Poskyvatelé webových událostí	611
Poskyvatelé konfigurace	619
Poskyvatel WebParts	622
Konfigurování poskytovatelů	623
Shrnutí	624
Kapitola 13: Rozšíření modelu poskytovatelů	625
Poskyvatelé jsou jednou vrstvou ve větší architektuře	625
Změna pomocí programování atributů	626
Jednodušší struktury hesel díky SqlMembershipProvider	626
Silnější struktury hesel díky SqlMembershipProvider	629
Seznámení s ProviderBase	630
Sestavení vlastních poskytovatelů	632
Vytvoření aplikace CustomProviders	632
Konstrukce potřebné kostry třídy	633
Vytvoření datového skladu uživatelů ve formátu XML	637
Definování instance poskytovatele v souboru web.config	638
Neimplementování metod a vlastností třídy MembershipProvider	639
Implementování metod a vlastností třídy MembershipProvider	640
Použití poskytovatele XmlMembershipProvider k přihlášení uživatelů	648
Rozšíření existujících poskytovatelů	649
Omezení schopností rolí novým poskytovatelem LimitedSqlRoleProvider	649
Použití nového poskytovatele LimitedSqlRoleProvider	653
Shrnutí	657

Kapitola 14: Navigace po webovém sídle	659
Mapy sídel ve formátu XML	660
Serverový ovládací prvek SiteMapPath	662
Vlastnost PathSeparator	664
Vlastnost PathDirection	665
Vlastnost ParentLevelsDisplayed	666
Vlastnost ShowToolTips	666
Dceřiné elementy ovládacího prvku SiteMapPath	667
Serverový ovládací prvek TreeView	667
Vestavěné styly ovládacího prvku TreeView	671
Součásti ovládacího prvku TreeView	671
Navázání ovládacího prvku TreeView na soubor XML	672
Výběr několika položek v prvku TreeView	675
Zadání vlastních ikon v ovládacím prvku TreeView	678
Zadání linií spojujících uzly	679
Programová práce s ovládacím prvkem TreeView	681
Serverový ovládací prvek Menu	686
Aplikování různých stylů na ovládací prvek Menu	688
Události ovládacího prvku Menu	692
Vázání ovládacího prvku Menu na soubor XML	693
Poskytovatel dat SiteMap	694
ShowStartingNode	695
StartFromCurrentNode	696
StartingNodeOffset	696
StartingNodeUrl	697
Rozhraní třídy SiteMap	698
Mapování adres URL	700
Lokalizace mapy sídla	701
Strukturování souboru Web.sitemap k lokalizaci	701
Úpravy souboru Web.config	702
Tvorba souborů prostředků sestavy (.resx)	703
Zkouška výsledků	704
Bezpečnostní omezování	705
Nastavení správy rolí pro správce	705
Nastavení oddílů správců	707
Aktivace bezpečnostního omezování	708
Vnořování souborů map sídel	709
Shrnutí	712
Kapitola 15: Personalizace (zosobnění)	713
Model personalizace	713
Vytvoření personalizačních vlastností	714
Přidání jednoduché personalizační vlastnosti	715
Využití personalizačních vlastností	716
Doplnění skupiny personalizačních vlastností	719
Práce se seskupenými personalizačními vlastnostmi	720
Definování typů personalizačních vlastností	721
Práce s vlastními typy	721
Zadání výchozích hodnot	724

Vytváření personalizačních vlastností pouze pro čtení	724
Anonymní personalizace	725
Aktivace anonymní identifikace koncových uživatelů	725
Práce s anonymní identifikací	728
Anonymní volby personalizačních vlastností	728
Varování ohledně úložiště profilu anonymních uživatelů	729
Programový přístup k personalizaci	729
Převedení anonymních uživatelů	730
Personalizace profilů	731
Jak stanovit, zda pokračovat v automatickém ukládání	732
Poskytovatelé personalizace	733
Práce s databází SQL Server Express	734
Práce s aplikací Microsoft SQL Server 7.0/2000/2005/2008	735
Použití více poskytovatelů	737
Správa aplikačních profilů	737
Vlastnosti třídy ProfileManager	738
Metody třídy ProfileManager	738
Tvorba stránky ProfileManager.aspx	739
Průzkum kódu stránky ProfileManager.aspx	742
Spuštění stránky ProfileManager.aspx	743
Shrnutí	744
Kapitola 16: Správa členství a správa rolí	745
Autentizace	745
Autorizace	746
Autentizace v ASP.NET 3.5	746
Nastavení členství na webu	746
Přidání uživatelů	749
Žádost o přihlašovací údaje	763
Práce s ověřenými uživateli	771
Zobrazení počtu připojených uživatelů	774
Práce s hesly	775
Autorizace v ASP.NET 3.5	780
Práce se serverovým ovládacím prvkem LoginView	780
Nastavení správy rolí na webovém sídle	783
Přidávání a přebírání aplikačních rolí	786
Odstraňování rolí	788
Přidávání uživatelů do rolí	789
Zjištění všech uživatelů v určité roli	789
Zjištění všech rolí určitého uživatele	791
Odstranění uživatelů z rolí	792
Kontrola uživatelů v rolích	793
Principy ukládání rolí do mezipaměti	794
Práce s nástrojem Web Site Administration Tool	795
Veřejné metody rozhraní Membership	796
Veřejné metody rozhraní Roles	796
Shrnutí	797
Rejstřík	799

Úvod

Jednoduše řečeno, sami zjistíte, že ASP.NET 3.5 je prostě úžasná technologie pro sestavování webových řešení! Když byla v roce 2000 uvedena technologie ASP.NET 1.0, řada lidí to považovala za revoluční skok vpřed v oblasti vývoje webových aplikací. Technologie ASP.NET 2.0 byla vzrušující a revoluční a ASP.NET 3.5 pokračuje v nastoleném směru a představuje dnes nejlepší rámec pro sestavování webových aplikací. Třebaže byly základy ASP.NET stanoveny již uvolněním ASP.NET 1.0, verze ASP.NET 3.5 stále staví na této bázi a více se zaměřuje na oblast zvýšení produktivity vývojáře.

Tato kniha popisuje ASP.NET kompletně. Nejenže představuje nové věci, ale také nabízí příklady všech technologií v akci. Tak se pohodlně posadte, přitáhněte si klávesnici a dopřejte si dobrou zábavu!

Trocha historie

Ještě než se začaly různé organizace zabývat vývojem aplikací pro Internet, většina vývoje se zaměřovala na „tlusté“ desktopové aplikace. Tyto aplikace „tlustých klientů“ se používaly ke všemu od domácí práce s počítačem přes hraní her k programům kancelářské produktivity a dalším. Zdálo se, že tento oblíbený aplikační model nemůže nic ohrozit.

V té době vyvíjela společnost Microsoft své klientské aplikace především pomocí Visual Basicu (VB).

Visual Basic nebyl jen programovací jazyk; byl navázán na prostředí IDE, které dovolovalo snadný vývoj tlustých klientských aplikací. Při práci v modelu Visual Basicu mohli vývojáři umísťovat ovládací prvky na formulář, nastavovat jejich vlastnosti a doplňovat kód potřebný ke zpracování jejich událostí. Když tak koncový uživatel kupříkladu stiskl nějaké tlačítko v jednom z formulářů Visual Basicu, tuto událost zpracoval kód ukrytý ve formuláři.

Přibližně v polovině 90. let pak na scénu vstoupil Internet. Společnost Microsoft nemohla posunout model Visual Basicu směrem k vývoji internetových aplikací. Internet nepochybně nabízel obrovské možnosti a ihned odhalil problémy, kterým musel čelit model aplikací tlustých klientů.

Internetové aplikace byly vlastně tvořeny jedinou instancí, k níž mohli přistupovat všichni uživatelé. Jelikož existovala jen jedna instance dané aplikace, její inovace nebo oprava znamenaly, že všechny změny byly ihned k dispozici úplně každému uživateli, který s danou aplikací pracoval prostřednictvím webového prohlížeče.

V rámci svého vstupu do světa webových aplikací vyvinula společnost Microsoft technologii Active Server Pages (ASP – aktivní serverové stránky). Technologie ASP nabízela rychlou a snadnou možnost vývoje webových stránek. Stránky ASP byly tvořeny jedinou stránkou, která obsahovala kombinaci různých značek a jazyků. Síla ASP spočívala v tom, že jste mohli zahrnout do stránky instrukce kódu VBScriptu nebo JScriptu, které se pak vykonaly na straně webového serveru ještě před odesláním stránky webovému prohlížeči uživatele. Jednalo se o jednoduchou možnost vytváření dynamických webových stránek přizpůsobovaných podle parametrů určených vývojářem.

K řízení chování na serveru používaly stránky ASP skripty vložené mezi lomené závorky a znaky procent – `<% %>`. Vývojář pak mohl sestavit stránku ASP, když začal sadou statického kódu HTML. Všechny dynamické elementy stránky se pak definovaly pomocí nějakého skriptového jazyka (např. VBScriptu nebo JScriptu). Když uživatel ve svém prohlížeči požadoval danou stránku od serveru, ujala se zpracování knihovna `asp.dll` (což byla aplikace ISAPI fungující jako most mezi skriptovým jazykem a používaným webovým serverem) a za běhu nadefinovala veškeré dynamické aspekty dané stránky v souladu s programovací logikou uloženou ve skriptech. Jakmile byly zpracovány dynamické složky stránky, byla výsledkem standardní stránka HTML odesílaná prohlížeči požadujícího klienta.

Jak se vyvíjel model webových aplikací, míchalo se do statického kódu HTML stále více jazyků, které pomáhaly manipulovat s chováním a vzhledem výstupní stránky. Časem bylo možné umístit do typické stránky ASP tolik různých jazyků, skriptů a prostého textu, že vývojáři začali označovat stránky využívající tyto prvky za *špagetový kód*. Bylo tak poměrně běžné mít stránku používající HTML, VBScript, JavaScript, CSS (Cascading Style Sheets – kaskádové šablony stylů), T-SQL i další věci. Za určitých okolností pak bylo takové sídlo z hlediska spravovatelnosti pravou noční můrou.

Technologie ASP se vyvíjela a objevovaly se nové verze. ASP 2.0 a 3.0 byly oblíbené, protože tyto technologie umožňovaly relativně přímočaré a jednoduché vytváření webových stránek. Jejich popularitu dále posílila skutečnost, že se objevily na konci 90. let, právě při nástupu éry „dot com“. V té době došlo k vývoji obrovského množství nových webových stránek a portálů a ASP byla jednou z hlavních technologií využívaných jednotlivci i společnostmi k jejich sestavení. I dnes ještě najdete na Internetu mnoho stránek `.asp` – některé i na sídle Microsoftu.

Už v době uvolnění poslední verze technologie Active Server Pages na konci roku 1998 však měli zaměstnanci Microsoftu Marc Anders a Scott Guthrie na mysli něco jiného. Jejich nápady se týkaly věci označované XSP (což byla zkratka bez jasného významu) – nového způsobu vytváření webových aplikací objektově orientovaným způsobem namísto procedurálního postupu využívaného v ASP 3.0. Své nápady ukázali mnoha různým skupinám v Microsoftu a setkali se s pochopením. V létě roku 2000 byla na konferenci profesionálních vývojářů (Professional Developers Conference) firmy Microsoft uvolněna věc nazvaná ASP+. Účastníci začali s touto technologií nadšeně pracovat. Jakmile byla celá technologie zpřístupněna (společně s finální verzí rámce .NET Framework 1.0), došlo k jejímu přejmenování na ASP.NET – obdržela vlastně příponu .NET stejně jako většina nových produktů společnosti Microsoft v té době.

Před uvedením .NET se model poskytovaný klasickým ASP a ten používaný ve Visual Basicu lišily tak zásadně, že jen málo vývojářů VB pracovalo také na webových aplikacích a jen minimum vývojářů webových aplikací pracovalo zároveň na tučných klientských aplikacích světa VB. Nacházel se tu velký předěl. Technologie ASP.NET tento předěl přemostila. Vývoji webových aplikací přinesla model událostí ve stylu Visual Basicu a nabídla tak velmi potřebné techniky správy stavu přes nestavový protokol HTTP. Její model se výrazně více podobá prvotnímu modelu Visual Basicu, kdy vývojář mohl umísťovat ovládací prvky na plochu návrháře neboli formulář, manipulovat s vlastnostmi ovládacích prvků, a dokonce pracovat s kódem skrývajícím se za těmito ovládacími prvky takovým způsobem, aby reagovaly na určité události, k nimž během jejich existence dochází. Technologie ASP.NET tak v zásadě zachovala to nejlepší z obou modelů, jak budete neustále odhalovat v celé naší knize.

Nepochybuji o tom, že se vám práce s nejnovější verzí ASP.NET 3.5 bude zamlouvat. Není nic lepšího, než si doslova „osahávat“ novou technologii a zjišťovat, co vše je možné. Následující oddíl popisuje cíle ASP.NET, abyste si dokázali udělat představu o tom, co je vlastně k máni.

Cíle ASP.NET

ASP.NET 3.5 je další důležitou verzí produktu, jež staví na rámci .NET Framework 2.0 a nabízí nové třídy i schopnosti. Tato verze rámce má interní kódové označení *Orcas*. Proto můžete někdy zaslechnout, jak se tato verze ASP.NET označuje jako *ASP.NET Orcas*. ASP.NET 3.5 pokračuje na cestě ke stanovenému cíli, učinit vývojáře ASP.NET těmi nejproduktivnějšími ve webovém prostředí.

Již od ASP.NET 2.0 se celá technologie zaměřovala na produktivitu vývojářů, správu, řízení i na výkonnost a škálovatelnost.

Produktivita vývojáře

Výrazná část zaměření ASP.NET 3.5 se týká produktivity. Už verze ASP.NET 1.x představovala zásadní navýšení produktivity – je možné ještě dále v tomto trendu pokračovat?

Cílem vývojového týmu ASP.NET bylo eliminovat většinu únavného vytváření kódu, jak to vyžadovala původní specifikace ASP.NET, a zjednodušit běžné úkoly ASP.NET. Nové možnosti zvýšení produktivity vývojáře jsou uváděny v celé naší knize. Nejprve se podívejme na starší technologii ASP.NET. Výpis Ú.1 představuje příklad používání technologie ASP.NET 1.0 k sestavení takové tabulky na webové stránce, která zajistí schopnost jednoduchého stránkování poskytovaných dat.

Výpis Ú.1: Zobrazení dat v serverovém ovládacím prvku DataGrid se zapnutým stránkováním (pouze VB)

```
<%@ Page Language="VB" AutoEventWireup="True" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SqlClient" %>

<script runat="server">

    Private Sub Page_Load(ByVal sender As System.Object, _
        ByVal e As System.EventArgs)
        If Not Page.IsPostBack Then
```

```
        BindData()
    End If
End Sub

Private Sub BindData()
    Dim prip As SqlConnection = New SqlConnection( _
        "server='localhost'; trusted_connection=true; Database='Northwind'")
    Dim prik As SqlCommand = New SqlCommand("Select * From Customers", prip)
    prip.Open()

    Dim da As SqlDataAdapter = New SqlDataAdapter(prik)
    Dim ds As New DataSet

    da.Fill(ds, "Customers")

    DataGrid1.DataSource = ds
    DataGrid1.DataBind()
End Sub

Private Sub DataGrid1_PageIndexChanged(ByVal source As Object, _
    ByVal e As System.Web.UI.WebControls.DataGridPageChangedEventArgs)
    DataGrid1.CurrentPageIndex = e.NewPageIndex
    BindData()
End Sub

</script>
<html>
<head>
</head>
<body>
    <form runat="server">
        <asp:DataGrid id="DataGrid1" runat="server" AllowPaging="True"
            OnPageIndexChanged="DataGrid1_PageIndexChanged"></asp:DataGrid>
    </form>
</body>
</html>
```

Přestože se zde používá značný objem kódu, jedná se o zásadní vylepšení množství kódu potřebného k zajištění stejné věci v klasických Active Server Pages 3.0. Nebudeme se zabývat detaily tohoto staršího kódu; jenom jsme chtěli ukázat, že při doplňování jakékoli dodatečné funkčnosti (jako je stránkování) dat zobrazených v tabulce musel vývojář vytvářet vlastní kód.

Jedná se o jednu z oblastí, kde jsou nové zisky na straně produktivity vývojáře nejzřejmější. ASP.NET 3.5 nyní nabízí nový serverový ovládací prvek nazvaný GridView. Tento ovládací prvek se do značné míry podobá serverovému ovládacímu prvku DataGrid, který už možná znáte a milujete. Serverový ovládací prvek GridView však (kromě mnoha jiných nových funkcí) zahrnuje vestavěnou schopnost aplikovat stránkování, řazení a editování dat, přičemž z vaší strany je zapotřebí relativně málo práce. Výpis Ú.2 zachycuje příklad použití serverového ovládacího prvku GridView. Tento příklad sestavuje takovou tabulku dat z tabulky Customers v databázi Northwind, která zahrnuje stránkování.

Výpis Ú.2: Zobrazení stránkované datové množiny v novém serverovém ovládacím prvku GridView

```

<%@ Page Language="VB" %>

<script runat="server">

</script>

<html xmlns=http://www.w3.org/1999/xhtml>
<head runat="server">
  <title>Ukázka GridView</title>
</head>
<body>
  <form runat="server">
    <asp:GridView ID="GridView1" Runat="server" AllowPaging="True"
      DataSourceID="SqlDataSource1" />
    <asp:SqlDataSource ID="SqlDataSource1" Runat="server"
      SelectCommand="Select * From Customers"
      ProviderName="System.Data.OleDb"
      ConnectionString="Provider=SQLOLEDB;Server=localhost;uid=sa;
      pwd=password;database=Northwind" />
  </form>
</body>
</html>

```

Toť vše! Stránkování lze aplikovat použitím několika nových serverových ovládacích prvků. Tu- to možnost zapínáte prostřednictvím určitého atributu serverového ovládacího prvku, kterým je atribut AllowPaging prvku GridView:

```

<asp:GridView ID="GridView1" Runat="server" AllowPaging="True"
  DataSourceID="SqlDataSource1" />

```

K další zajímavé události dochází v sekci kódu dokumentu:

```

<script runat="server">
</script>

```

Tyto dva řádky kódu ve skutečnosti nejsou ke spuštění souboru zapotřebí. Jsou tu jenom proto, aby vám ukázaly jednu zásadní věc: *Aby vše fungovalo, nemusíte vytvářet vůbec žádný serverový kód!* Stačí jen zahrnout určité serverové ovládací prvky – jeden prvek k získání dat a druhý k jejich zobrazení. Tyto ovládací prvky se pak vzájemně propojí.

Výkonnost a škálovatelnost

Jedním z cílů návrhu ASP.NET, jak si je stanovil tým Microsoftu, bylo nabídnout nejrychlejší server webových aplikací na světě. Tato kniha se zabývá řadou výkonnostních vylepšení, jak je obsahuje ASP.NET 3.5.

Jedním z nejzajímavějších výkonnostních vylepšení je nová možnost kešování (ukládání do mezi- paměti) zaměřená na práci s Microsoft SQL Serverem. Technologie ASP.NET 3.5 zahrnuje funkci označovanou za *zrušení platnosti mezipaměti SQL*. Před nástupem ASP.NET 2.0 bylo možné ke- šovat výsledky přicházející od SQL Serveru a aktualizovat mezipaměť na základě určitého časové-

ho intervalu – kupříkladu každých 15 sekund nebo podobně. To znamenalo, že pokud došlo během této 15sekundové periody ke změně uložených dat, mohl uživatel pozorovat zastaralá data.

V některých případech jsou uložená data závislá na časovém intervalu nepřijatelná. V ideálním případě dojde ke zrušení dat uložených v mezipaměti, pokud se nějakým způsobem změní základní data ve zdroji, z něž se výsledky přejímají – tím je v našem případě SQL Server. Při práci s ASP.NET 3.5 to můžete zajistit funkcí označení mezipaměti SQL za neplatnou. Když se pak změní výsledná data SQL Serveru, změní se také výstupní mezipaměť a koncový uživatel vždy pozoruje aktuální data. Nabídnutá data nejsou nikdy zastaralá.

ASP.NET 3.5 podporuje 64bitové systémy. To znamená, že máte možnost provozovat aplikace ASP.NET na 64bitových procesorech firem Intel nebo AMD.

Protože je ASP.NET 3.5 plně zpětně kompatibilní s ASP.NET 1.0, 1.1 a 2.0, můžete vzít jakoukoli dříve vytvořenou aplikaci ASP.NET, překompilovat ji na .NET Framework 3.5 a provozovat na 64bitovém procesoru.

Další funkce ASP.NET 3.5

Právě jste se seznámili s některými z hlavních cílů týmu ASP.NET, který vytvořil ASP.NET 3.5. Aby mohl vytyčených cílů dosáhnout, musel tento tým vložit do ASP.NET celou horu nových funkcí. Některé z nich popisují následující oddíly.

Nové vývojářské infrastruktury

Významným aspektem v ASP.NET 3.5 je skutečnost, že máte k dispozici nové infrastruktury použitelné v aplikacích. Tým ASP.NET vybral některé z nejběžnějších programovacích operací prováděných ve webových aplikacích a vestavěl je přímo do ASP.NET. Tím vám ušetřil mnoho času a kódu.

Správa členství a rolí

Pokud jste před ASP.NET 2.0 vyvíjeli nějaký portál, který požadoval po uživatelích přihlášení do aplikace, čímž získali privilegovaný přístup, pak jste tuto věc museli vždy vytvořit úplně sami. Přitom vytváření aplikací s oblastmi přístupnými pouze vybraným jednotlivcům může být záludné.

V případě ASP.NET 3.5 je tato možnost již vestavěna. Nyní máte možnost ověřovat uživatele kupříkladu způsobem uvedeným ve výpisu Ú.3.

Výpis Ú.3: Ověření uživatele v kódu

VB

```
If (Membership.ValidateUser (Username.Text, Password.Text)) Then  
    ' Sem patří kód umožnění přístupu  
End If
```

C#

```
if (Membership.ValidateUser (Username.Text, Password.Text)) {  
    // Sem patří kód umožnění přístupu  
}
```

Nová řada rozhraní API, ovládacích prvků a poskytovatelů v ASP.NET 3.5 vám dovoluje řídit správu členství a rolí uživatelů aplikací. Pomocí těchto API můžete snadno spravovat uživatele a jejich komplexní role – vytvářet je, odstraňovat a upravovat. Všechny tyto schopnosti získáte prostřednictvím zmíněných API nebo vestavěného webového nástroje nazvaného Web Site Administration Tool.

Pokud se jedná o ukládání uživatelů a jejich rolí, používá ASP.NET 2.0 soubor .mdf (což je typ souboru nástroje SQL Server Express Edition) sloužící k ukládání všech uživatelů a jejich rolí. V žádném případě však nejste omezeni pouze na tento datový sklad. Můžete rozšířit vše, co vám nabízí ASP.NET, a sestavit si své vlastní poskytovatele s využitím jakéhokoli vámi upřednostňovaného datového skladu. Chcete-li tak kupříkladu vytvořit sklad uživatelů v LDAP nebo v databázi Oracle, můžete to udělat poměrně snadno.

Zosobnění (personalisation)

Jedním z pokročilých prvků, jež portály často nabízejí své členské základně, je možnost zosobnění nabídky – koncoví uživatelé si mohou přizpůsobit vzhled a funkčnost sídla podle svého. Schopnost zosobnit aplikaci a uložit osobní nastavení je nyní kompletně vestavěná do rámce ASP.NET.

Protože zosobnění se obvykle točí kolem uživatele a možná ještě rolí, kterých se tento uživatel účastní, lze architekturu zosobnění těsně navázat na infrastrukturu členství a rolí. V souvislosti s ukládáním vytvořených osobních nastavení se vám nabízí několik možností. Schopnost ukládat tato nastavení do Microsoft Accessu nebo SQL Serveru je vestavěná do ASP.NET 3.5. Podobně jako v případě schopností rozhraní API členství a rolí máte i zde možnost využívat flexibilní model poskytovatelů a následně buď změnit způsob, jakým takový vestavěný poskytovatel využívá dostupný datový sklad, nebo si sestavit svého vlastního poskytovatele dat a pracovat s úplně novým datovým skladem. API zosobnění také podporuje sjednocování datových skladů, takže v případě potřeby můžete používat i více datových skladů.

Protože lze pomocí těchto nových API velmi snadno vytvořit sídlo s možností vlastního nastavení, je tento prvek významným doplňkem pro všechny vytvářené aplikace.

Portál rámce ASP.NET

Ve dnech ASP.NET 1.0 mohli vývojáři přejít na webové sídlo týmu ASP.NET (na adrese www.asp.net) a stáhnout si ukázky webových aplikací, například IBuySpy. Tyto ukázky se označovaly za Developer Solution Kits (sady vývojářských řešení) a jsou základem mnoha dnešních internetových sídel. Některé byly dokonce rozšířeny do rámců otevřeného zdrojového kódu (Open Source), jako je tomu v případě DotNetNuke.

Na IBuySpy bylo výborné to, že jste mohli použít poskytnutý kód jako základ stavby svého webového skladu nebo portálu. Stačilo vzít základní kód a rozšířit jej. Mohli jste kupříkladu změnit vzhled a práci prezentační části kódu nebo do této modulární architektury zavést nějakou pokročilou funkčnost. Sady Developer Solution Kits byly poměrně oblíbené, protože výrazně zjednodušovaly provádění uvedených typů operací.

Díky popularitě systémů jako IBuySpy nabízí ASP.NET 3.5 vestavěnou schopnost používání modulů Web Parts ke snadnému sestavování portálů. Možnosti, kterých lze dosáhnout s využitím nového systému pro vytváření portálů (Portal Framework), jsou prostě úžasné. Síla stavby pomocí modulů Web Parts spočívá v tom, že umožňuje koncovým uživatelům plně si přizpůsobit portál svým požadavkům.

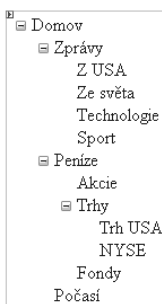
Pohyb po sídle

Členové týmu ASP.NET si uvědomují, že koncoví uživatelé požadují jednoduchou navigaci v celé aplikaci. Mechanika zajištění této funkčnosti logickým způsobem se však občas vytváří v kódu jen obtížně. Tým vyřešil tento problém v ASP.NET řadou navigačních serverových ovládacích prvků.

Zaprvé máte možnost vytvořit mapu sídla své aplikace v určitém souboru XML, s nímž mohou konkrétní ovládací prvky inherentně pracovat. Výpis Ú.4 představuje ukázkou takového souboru mapy sídla.

Výpis Ú.4: Příklad použití souboru mapy sídla

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0">
  <siteMapNode title="Domov" description="Domovská stránka" url="default.aspx">
    <siteMapNode title="Zprávy" description="Nejnovější zprávy" url="News.aspx">
      <siteMapNode title="Z USA" description="Zprávy z USA" url="News.aspx?cat=us" />
      <siteMapNode title="Ze světa" description="Zprávy ze světa" url="News.aspx?cat=world"/>
      <siteMapNode title="Technologie" description="Zprávy z techniky"
        url="News.aspx?cat=tech" />
      <siteMapNode title="Sport" description="Zprávy ze sportu" url="News.aspx?cat=sport" />
    </siteMapNode>
    <siteMapNode title="Penize" description="Nejnovější finanční informace"
      url="Finance.aspx">
      <siteMapNode title="Akcie" description="Poslední údaje" url="Quotes.aspx" />
      <siteMapNode title="Trhy" description="Poslední novinky z trhu" url="Markets.aspx">
      <siteMapNode title="Trh USA" description="Pohled na trh USA" url="MarketsUS.aspx" />
      <siteMapNode title="NYSE" description="New York Stock Exchange" url="NYSE.aspx" />
    </siteMapNode>
    <siteMapNode title="Fondy" description="Fondy" url="Funds.aspx" />
  </siteMapNode>
  <siteMapNode title="Počasí" description="Aktuální počasí" url="Weather.aspx" />
</siteMapNode>
</siteMap>
```



Obrázek Ú.1

Domov > Penize > Trhy > NYSE

Obrázek Ú.2

Jakmile máte vytvořenu mapu sídla, můžete tento soubor používat jako zdroj dat v několika nových serverových ovládacích prvcích navigace na sídle, jako jsou TreeView a SiteMapPath. Serverový ovládací prvek TreeView vám dovoluje umístit do aplikace „rozbalitelný“ systém pohybu po

sídle. Obrázek Ú.1 zachycuje příklad jednoho z mnoha vzhledů, jaké můžete takovému serverovému ovládacímu prvku TreeView přiřadit.

SiteMapPath je ovládací prvek nabízející možnost umístit do své aplikace takové navigační prvky, které koncovému uživateli ukazují zatím prošlou cestu a umožňují mu jednoduše se vrátit do vyšších úrovní. Obrázek Ú.2 je příkladem serverového ovládacího prvku SiteMapPath v akci.

Zmíněné nové možnosti navigace na sídle nabízejí výbornou možnost programového přístupu k rozvržení sídla, a dokonce i zobrazování jen těch částí sídla, které odpovídají roli koncových uživatelů.

Kompilační systém ASP.NET

Kompilace byla v ASP.NET 1.0 vždy trochu záluďná. Mohli jste totiž pomocí ASP.NET a Visual Studia sestavit aplikační soubory s kódem na pozadí, zavést je a pak sledovat kompilování souborů .aspx jednotlivých stránek, jak byly požadovány. Když jste pak v ASP.NET 1.0 zadali nějaké změny v souborech s kódem na pozadí, neodrazily se v aplikaci až do okamžiku jejího kompletního opakovaného sestavení. To znamenalo, že tentýž požadavek na jednotlivé stránky musel být vydán znovu, než došlo postupně k překompilování celé aplikace.

V ASP.NET 3.5 se s třídami a kompilací pracuje úplně jinak než v ASP.NET 1.0. Mechanika nového kompilačního systému ve skutečnosti začíná tím, jak je v ASP.NET 3.5 strukturovaná stránka. V ASP.NET 1.0 jste vytvořili své stránky buď pomocí modelu s kódem na pozadí, nebo umístěním veškerého serverového kódu přímo mezi značky <script> své stránky .aspx. Většina stránek se konstruovala s využitím modelu kódu na pozadí, protože to bylo výchozí nastavení při práci s Visual Studií .NET 2002 a 2003. V těchto IDE bylo poměrně obtížné vytvořit stránku ve stylu vloženého kódu. Pokud jste tak postupovali, nemohli jste využívat funkci IntelliSense, která může být při práci s tak obrovskou kolekcí tříd, jakou nabízí rámec .NET Framework, skutečnou spásou.

ASP.NET 3.5 nabízí nový model kódu na pozadí, protože .NET Framework 3.5 umožňuje pracovat s *částečnými třídami* (označovanými také za částečné typy). Při kompilaci se samostatné soubory zkombinují do jediného celku. Tím získáváte mnohem čistší stránky s kódem na pozadí. Kód, který byl součástí sekce Web Form Designer Generated vašich tříd, je separován od tříd s kódem na pozadí, jež vytváříte sami. Zkuste si to porovnat se skutečností, že v ASP.NET 1.0 musel být soubor .aspx odvozen od svého vlastního souboru s kódem na pozadí, aby představoval jedinou logickou stránku.

Aplikace ASP.NET 3.5 mohou zahrnovat adresář \App_Code, kam umísťujete zdroj své třídy. Všechny sem vložené třídy se dynamicky kompilují a ihned odrážejí v aplikaci. Při provádění změn nepoužíváte nějaký oddělený proces sestavení, jako to bylo v ASP.NET 1.0. Jedná se vlastně o model *uložení a okamžitého aplikování*, jak se používal v klasickém ASP 3.0. Visual Studio 2008 rovněž automaticky nabízí IntelliSense u všech objektů umístěných v adresáři \App_Code, ať už pracujete v modelu s kódem na pozadí nebo vloženým kódem.

ASP.NET 3.5 vám rovněž poskytuje nástroje, jež vám dovolují překompilovat aplikace ASP.NET, a to jak stránky .aspx, tak i kód na pozadí, takže s prvním převzetím stránek vaší aplikace nebude souviset žádné zpoždění. Jedná se rovněž o vynikající možnost zjištění, zda jste ve stránkách neudělali nějaké chyby, aniž byste museli všechny stránky sami vyvolávat. Předkompilování aplikací ASP.NET 3.5 je jednoduché – stačí použít `aspnet_compiler.exe` a využít některé z podporovaných příznaků. Pokud se někde v aplikaci objeví chyby, obdržíte upozornění. Je rovněž možné překompilovat si aplikaci a na server dodat pouze vytvořenou sestavu, čímž budete chránit svůj kód

před zkoumáním, změnami a jinými úpravami po jeho zavedení. Dále v knize si ukážeme scénáře obou těchto přístupů.

Sledování zdraví aplikací ASP.NET

Vestavěné schopnosti sledovat zdraví jsou dosti významnými prvky zjednodušujícími správu zavedených aplikací ASP.NET. Sledování zdraví je přesně tím, co již použitý termín naznačuje – schopností monitorovat zdraví a výkonnost zavedených aplikací ASP.NET.

Monitorování ASP.NET je postaveno na různých událostech sledování zdraví (jež se označují za webové události), k nimž v aplikaci dochází. Používání systému sledování zdraví vám dovoluje protokolovat takové webové události, jako jsou nesprávná přihlášení, spuštění a zastavení aplikací a všechny nezpracované výjimky. K protokolování (záznamu) událostí může docházet na více místech; proto můžete zaznamenávat do protokolu událostí nebo i do databáze. Kromě zajišťování takového diskového protokolování můžete celý systém využít k odesílání údajů o zdraví aplikace elektronickou poštou.

Nejenže můžete ve své aplikaci pracovat s konkrétními událostmi, ale máte také možnost využívat systém sledování zdraví k zachytávání aktuálního „zdravotního stavu“ běžící aplikace. Stejně jako většinu ostatních systémů vestavěných do ASP.NET 3.5 lze rozšiřovat také systém sledování zdraví a vytvářet si vlastní události zaznamenávající aplikační údaje.

Sledování zdraví je již standardně v souborech `.config` zapnuté. Výchozí nastavení sledování zdraví protokoluje všechny chyby a auditu selhání. Když tak kupříkladu vaše aplikace vyvolá nějakou chybu, pak se objeví příslušné upozornění v aplikačním protokolu.

Základní chování sledování zdraví můžete upravit menšími změnami v souboru `web.config` své aplikace. Předpokládejme kupříkladu, že chcete ukládat údaje o chybové události do souboru SQL Express zahrnutého v aplikaci. Těto změny dosáhnete přidáním uzlu `<healthMonitoring>` do souboru `web.config` způsobem zachyceným ve výpisu Ú.5.

Výpis Ú.5: Definování sledování zdraví v souboru `web.config`

```
<healthMonitoring enabled="true">
  <providers>
    <clear />
    <add name="SqlWebEventProvider" connectionStringName="LocalSqlServer"
      maxEventDetailsLength="1073741823" buffer="false" bufferMode="Notification"
      type="System.Web.Management.SqlWebEventProvider,
        System.Web, Version=2.0.0.0, Culture=neutral,
        PublicKeyToken=b03f5f7f11d50a3a"/>
  </providers>
  <rules>
    <clear />
    <add name="All Errors Default" eventName="All Errors"
      provider="SqlWebEventProvider"
      profile="Default" minInstances="1" maxLimit="Infinite"
      minInterval="00:01:00" custom="" />
    <add name="Failure Audits Default" eventName="Failure Audits"
      provider="SqlWebEventProvider" profile="Default" minInstances="1"
      maxLimit="Infinite" minInterval="00:01:00" custom="" />
  </rules>
</healthMonitoring>
```

```
</rules>
</healthMonitoring>
```

Po této změně se budou události ukládat do souboru ASPNETDB.MDF, jenž se automaticky vaším jménem vytvoří, pokud v projektu zatím neexistuje.

Po otevření zmíněného souboru SQL Express objevíte tabulku aspnet_WebEvent_Events, kam se všechny informace ukládají.

Mnohem více se o schopnostech sledování zdraví, jež je součástí ASP.NET 3.5, dozvíte v kapitole 32.

Čtení a zápis konfiguračních nastavení

Pomocí třídy `WebConfigurationManager` máte možnost číst a zapisovat serverové nebo aplikační konfigurační soubory. To znamená, že lze zapisovat a číst nastavení v souborech `machine.config` nebo `web.config`, jak je vaše aplikace používá.

Schopnost číst a zapisovat konfigurační soubory se neomezuje jen na práci s místním počítačem, na němž se vaše aplikace nachází. Zmíněné operace lze provádět i na vzdálených serverech a aplikacích.

K dispozici je také samozřejmě grafické rozhraní (GUI), v němž můžete kdykoli provádět operace čtení a změn konfiguračních souborů. Vynikající je ovšem to, že vestavěné nástroje GUI zajišťující tuto funkčnost (jako je doplněk MMC ASP.NET ve Windows XP nebo nové rozhraní IIS v případě práce ve Windows Vista) používají třídu `WebConfigurationManager`, takže máte možnost sestavovat si vlastní nástroje pro správu.

Výpis Ú.6 je příkladem čtení přípojovacího řetězce z aplikačního souboru `Web.config`.

Výpis Ú.6: Čtení řetězce připojení ze souboru `Web.config` aplikace

VB

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
    Try
        Dim retezecPripojeni As String = _
            ConfigurationManager.ConnectionStrings("Northwind").
                ConnectionString.ToString()
        Label1.Text = retezecPripojeni
    Catch ex As Exception
        Label1.Text = "Nebyl nalezen řetězec připojení."
    End Try
End Sub
```

C#

```
protected void Page_Load(object sender, EventArgs e)
{
    try
    {
        string retezecPripojeni =
            ConfigurationManager.ConnectionStrings["Northwind"].
                ConnectionString.ToString();
        Label1.Text = retezecPripojeni;
    }
}
```

```
catch (Exception)
{
    Label1.Text = "Nebyl nalezen řetězec připojení.";
}
}
```

Tento minimální kód zobrazí řetězec připojení k databázi Northwind, jak se nachází v souboru `web.config`, na obrazovku pomocí ovládacího prvku `Label`. Jak je zřejmé, položky lze z konfiguračních souborů přebírat docela jednoduše.

Lokalizace

ASP.NET dále zjednodušuje lokalizování aplikací. Nemusíte využívat jen Visual Studio, ale můžete vytvořit soubory prostředků (`.resx`) umožňující dynamicky měnit vytvořené stránky podle místních nastavení žadatele.

ASP.NET 3.5 zajišťuje schopnost poskytovat prostředky celé aplikaci nebo jen jednotlivým stránkám aplikace prostřednictvím dvou nových aplikačních složek: `App_GlobalResources` a `App_LocalResources`.

Položky definované v kterémkoli z vámi vytvořených souborů `.resx` jsou pak přímo přístupné v serverových ovládacích prvcích ASP.NET nebo programově pomocí výrazů podobných následujícímu:

```
<%= Resources.Resource.Question %>
```

Celý systém je přímočarý a jednoduše implementovatelný. Toto téma je podrobněji popsáno v kapitole 30.

Rozšíření rámce stránek

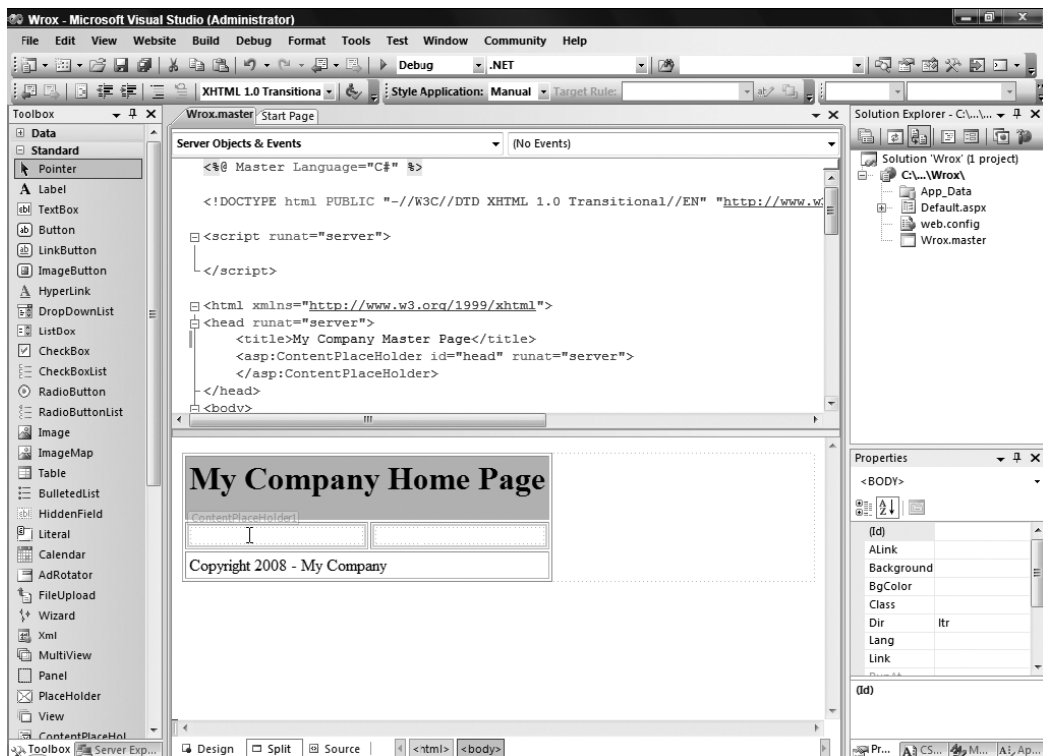
Stránky ASP.NET lze sestavovat na základě vizuální dědičnosti. To bylo možné ve světě formulářů Windows (Windows Forms), obtížněji, ale přece jen toho šlo dosáhnout také v ASP.NET. Rovněž získáváte schopnost jednoduše aplikovat konzistentní vzhled a chování stránek aplikace pomocí témat (motivů). Řada minulých potíží práce s ADO.NET byla nyní odstraněna doplněním nových sad ovládacích prvků datových zdrojů, které se dokážou postarat o přístup k datům a jejich přebírání z velké kolekce datových skladů.

Vzorové stránky (master pages)

Díky podpoře *vzorových stránek* můžete nyní v aplikacích ASP.NET využívat vizuální dědičnost. Protože má řada aplikací ASP.NET podobnou strukturu všech svých stránek, je rozumné sestavit šablonu stránky jen jednou a tu pak využívat v celé aplikaci.

V ASP.NET toho dosahujete vytvořením stránky `.master`, jak to ukazuje obrázek Ú.3.

Ukázková vzorová stránka může zahrnovat záhlaví, zápatí a další elementy, které mohou sdílet všechny stránky. Kromě těchto základních elementů, které budete požadovat na všech stránkách, jež dědí od této šablony a využívají ji, můžete ještě na samotnou vzorovou stránku umisťovat serverové ovládací prvky `<asp:ContentPlaceHolder>`, jež podřízené stránky (neboli stránky obsahu) dále používají ke změnám specifických oblastí šablony vzorové stránky. Editaci podstránky zachycuje obrázek Ú.4.



Obrázek Ú.3

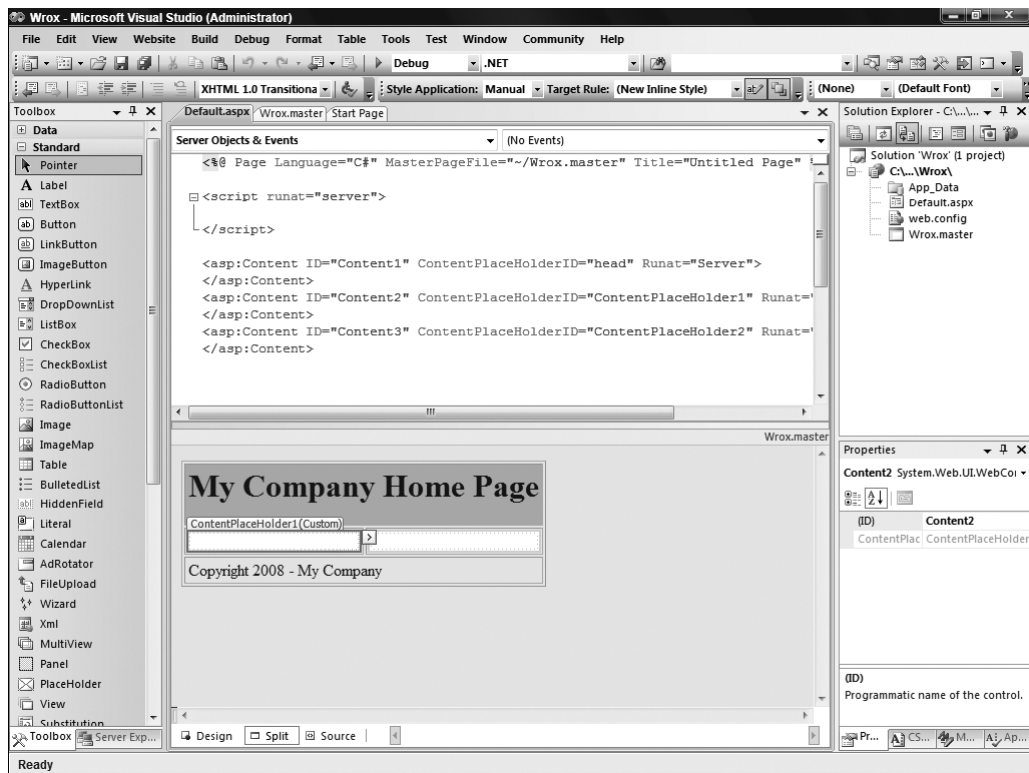
Když koncový uživatel vyvolá jednu z podstránek, pak se ve skutečnosti dívá na jedinou stránku zkompileovanou jak z příslušné podstránky, tak i vzorové stránky, od níž daná podřízená stránka dědí. To také znamená, že na nové jediné stránce je aktivní serverový i klientský kód z obou stránek.

Na vzorových stránkách je výborné to, že nyní máte jediné místo, kde lze zadávat změny ovlivňující celé sídlo. Není již tedy zapotřebí měnit úplně každou stránku v celé aplikaci.

Témata

Po zavedení témat (motivů) v ASP.NET 2.0 je velmi jednoduché zajistit konzistentní vzhled a fungování celého sídla. Témata jsou jednoduché textové soubory, v nichž definujete vzhled serverových ovládacích prvků, který lze dále aplikovat na celé sídlo, na jedinou stránku nebo na určitý serverový ovládací prvek. Jako doplněk definic serverových ovládacích prvků lze také snadno využívat grafiku a kaskádové šablony stylů (Cascading Style Sheets).

Témata jsou uložena v adresáři /App_Theme v kořenu aplikace, odkud se pak používají. Výbornou funkcí témat je to, že lze aplikovat dynamicky na základě nastavení využívajících novou službu zosobnění poskytovanou rámcem ASP.NET. Každý jedinečný uživatel vašeho portálu nebo aplikace si může z nabídky zvolit svůj vlastní vzhled a chování.



Obrázek Ú.4

Objekty pro přístup k datům

Jednou z kódově náročnějších činností v ASP.NET 1.0 bylo přebírání dat. To v mnoha případech znamenalo práci s řadou objektů. Jestliže jste někdy pracovali s ASP.NET, pak jistě víte, že proces zobrazení dat z tabulky Microsoft SQL Serveru v serverovém ovládacím prvku DataGrid byl dosti komplikovaný. Museli jste nejprve vytvořit množství nových objektů. Ty zahrnovaly objekt typu SqlConnection a za ním následující objekt typu SqlCommand. Jakmile jste měli uvedené objekty, vytvořili jste SqlDataReader naplňující prvek DataGrid vázáním výsledků. Nakonec se objevila tabulka obsahující přejímaná data (kupříkladu tabulka Customers z databáze Northwind).

ASP.NET dnes tuto náročnou proceduru eliminuje zavedením nové sady objektů pracujících specificky s přístupem k datům a jejich získáváním. Tyto nové datové ovládací prvky se používají tak snadno, že máte možnost přistupovat k datům a získávat je při vyplňování serverových ovládacích prvků ASP.NET, aniž byste museli vytvářet doplňkový kód. Příklad tohoto přístupu jste viděli ve výpisu Ú.2, kde serverový ovládací prvek <asp:SqlDataSource> získával řádky údajů z tabulky Customers v databázi Northwind ze SQL Serveru. Serverový ovládací prvek SqlDataSource byl následně navázan na nový serverový ovládací prvek GridView, a to pomocí jednoduchých atributů v samotném prvku GridView. To už opravdu nemůže být jednodušší!

Na této nové funkčnosti je vynikající ta skutečnost, že se neomezuje jen na SQL Server společnosti Microsoft. Ve skutečnosti máte k dispozici několik serverových ovládacích prvků datových zdrojů. Navíc si můžete vytvořit své vlastní. Kromě serverového ovládacího prvku `SqlDataSource` zahrnuje ASP.NET 3.5 prvky `AccessDataSource`, `XmlDataSource`, `ObjectDataSource`, `SiteMapDataSource` a nový serverový ovládací prvek `LinqDataSource`. Všechny tyto nové datové ovládací prvky budeme dále v knize používat.

Co potřebujete k ASP.NET 3.5

Chcete-li si projít příklady uvedené v knize, bude nejlepší nainstalovat si Visual Studio 2008. Stejně tak ale můžete používat Poznámkový blok a kompilátory příkazového řádku obsažené v rámci .NET Framework. Chcete-li si však vyzkoušet *všechny* příklady v knize, potřebujete následující:

- Windows Server 2003, Windows Server 2008, Windows 2000, Windows XP nebo Windows Vista,
- Visual Studio 2008 (tím se nainstaluje .NET Framework 3.5),
- SQL Server 2000, 2005 nebo 2008,
- Microsoft Access nebo SQL Server Express.

Je hezké, že pro práci s ASP.NET 3.5 nemusíte mít instalovanou službu Microsoft Internet Information Services (IIS), protože tato verze ASP.NET zahrnuje vestavěný webový server vycházející z již dříve vyvinuté technologie Microsoft Cassini. Obavy nemusíte mít ani v situaci, kdy nemáte k dispozici plnou verzi SQL Serveru. Mnoho příkladů využívajících tuto databázi lze změnit tak, aby pracovaly s nástrojem Microsoft SQL Server Express, který je k dispozici ke stažení zdarma.

Kdo by si měl naši knihu přečíst

Tato kniha byla napsána proto, aby vám představila nové funkce a schopnosti nabízené ASP.NET 3.5 a také vysvětlila základy poskytované technologií ASP.NET. Předpokládáme, že rozumíte webovým technologiím obecně a že jste pracovali s předchozí verzí ASP.NET, technologií Active Server Pages 2.0/3.0 nebo JavaServer Pages. Znáte-li základy webového programování, nebudete mít potíže porozumět obsahu knihy.

Jestliže s ASP.NET teprve začínáte, určitě si přečtěte knihu *Beginning ASP.NET 3.5: In C# and VB* od Imara Spaanjaarse (Wiley Publishing, Inc., 2008), protože ta vás seznámí s potřebnými základy.

Kromě práce s webovými technologiemi rovněž předpokládáme, že rozumíte základním programovacím konstrukcím, jako jsou proměnné, cykly `For Each` a objektově orientované programování.

Možná vás zajímá, zda je tato kniha určena vývojářům pracujícím v jazyku Visual Basic nebo C#. S potěšením prohlašujeme, že poslouží oběma! Pokud se kód výrazně odlišuje, najdete v knize příklady uvedené ve VB i C#.

Čím se kniha zabývá

Tato kniha se zabývá popisem technologie ASP.NET v její verzi 3.5. Jsou tu podrobně probrány všechny funkce zahrnuté do ASP.NET 3.5. Následující seznam přibližuje obsah jednotlivých kapitol.

- **Kapitola 1, „Rámce aplikací a stránek“.** První kapitola se zabývá rámci aplikací ASP.NET a také strukturou a rámci poskytovanými jednotlivým stránkám ASP.NET. Ukazuje, jak sestavovat aplikace ASP.NET využívající služby IIS nebo vestavěného webového serveru dodávaného s Visual Studiem 2008. Kapitola vám rovněž ukáže nové složky a soubory, které jsou součástí ASP.NET. Probírá nové možnosti kompilace kódu a ukazuje, jak docílit odesílání dat mezi stránkami. Kapitola končí ukázkou jednoduché práce se třídami ve Visual Studiu 2008.
- **Kapitoly 2, 3 a 4.** Tyto tři kapitoly jsou tu seskupeny, jelikož se všechny zabývají serverovými ovládacími prvky. Celá skupina kapitol začíná prozkoumáním principu serverových ovládacích prvků a jejich zásadní role ve vývoji ASP.NET. Kromě pohledu na rámec serverových ovládacích prvků se uvedené kapitoly pouštějí také hlouběji do popisu velkého množství prvků, které máte ve vyvíjených projektech ASP.NET k dispozici. Kapitola 2, „Serverové ovládací prvky ASP.NET a klientské skripty“, pojednává o základní práci se serverovými ovládacími prvky. Kapitola 3, „Webové serverové ovládací prvky ASP.NET“, probírá ovládací prvky, jež byly součástí technologie ASP.NET již od jejího prvního uvedení, i prvky přidávané v následujících verzích. Kapitola 4, „Serverové ovládací prvky pro ověřování platnosti dat“, popisuje speciální skupinu serverových ovládacích prvků sloužících k ověřování. Tyto ovládací prvky lze používat k základnímu i pokročilemu ověřování formulářů.
- **Kapitola 5, „Vzory stránek“.** Vzorové stránky jsou výborným rozšířením technologie ASP.NET. Umožňují vytvářet stránky šablon, s nimiž můžete pracovat v celé aplikaci a nikoli jen v jednotlivých stránkách. Tato kapitola zkoumá vytváření šablon a jejich aplikování na stránky obsahu v celé aplikaci ASP.NET.
- **Kapitola 6, „Témata a vzhledy“.** Soubory CSS (Cascading Style Sheet), s nimiž můžete pracovat v ASP.NET 1.0/1.1, nejsou v mnoha ohledech vyhovující. To se týká především oblasti serverových ovládacích prvků. Při práci s předchozími verzemi si vývojář nikdy nemohl být jistý tím, jaký výstup HTML mohou zmíněné soubory vygenerovat. Tato kapitola zkoumá, jak pracovat se styly vyžadovanými vaší aplikací, a ukazuje, jak vytvářet centrálně spravovaný vzhled a jednotné chování všech stránek aplikace pomocí témat a souborů vzhledů, jež jsou součástí tématu.
- **Kapitola 7, „Vázání dat v ASP.NET 3.5“.** Jedním z důležitějších úkolů ASP.NET je prezentování dat a tato kapitola ukazuje, jak to zajistit. ASP.NET nabízí řadu ovládacích prvků, k nimž můžete připojit data a nabídnout je koncovému uživateli. Tato kapitola zkoumá skryté schopnosti, jež vám dovolují pracovat s daty programově ještě před předáním nějakému ovládacímu prvku.
- **Kapitola 8, „Správa dat pomocí ADO.NET“.** Tato kapitola představuje datový model ADO.NET poskytovaný ASP.NET, který vám umožňuje rychle a logicky zpracovávat přejímání, aktualizování a odstraňování dat. Nový datový model dovoluje dostat se k datům uloženým v čemkoli od SQL Serveru po soubory XML, a to jen jedním nebo dvěma řádky kódu.
- **Kapitola 9, „Dotazování pomocí LINQ“.** Doplněkem rámce .NET Framework 3.5 je toužebně očekávaný LINQ. Jedná se o sadu rozšíření rámce .NET Framework zahrnujících operace dotazů, nastavení a transformace integrované v jazyku. Tato kapitola představuje LINQ a ukazuje, jak efektivně využívat tento nový prvek v dnešních webových aplikacích.
- **Kapitola 10, „Práce XML a LINQ na XML“.** Není pochyb o tom, že se jazyk XML stal hlavní technologií používanou k prezentaci dat. Z toho důvodu mají .NET Framework a ASP.NET 3.5

do svých rámců vestavěno mnoho funkcí, jež vám umožňují jednoduše extrahovat, vytvářet, zpracovávat a ukládat XML. Tato kapitola se blíže dívá na technologie XML vestavěné do ASP.NET a základního rámce .NET Framework.

- **Kapitola 11, „Vývoj pro IIS 7.0“.** Aktuální verze IIS je zřejmě nejzásadnější v celé historii tohoto serveru a mění způsob hostování aplikací ASP.NET i práci s nimi. IIS 7.0 je součástí operačního systému Windows Server 2008 a také Windows Vista.
- **Kapitola 12, „Úvod do modelu poskytovatelů“.** Řada systémů vestavěných do ASP.NET výrazně usnadňuje život programátorům a zvyšuje jejich produktivitu. Tyto systémy staví na architektuře označované za *model poskytovatelů* (provider model), který je značně rozšiřitelný. Tato kapitola uvádí přehled modelu poskytovatelů a jeho používání v ASP.NET 3.5.
- **Kapitola 13, „Rozšíření modelu poskytovatelů“.** Po představení modelu poskytovatelů ukazuje tato kapitola některé z možností jeho rozšíření, jak je podporuje ASP.NET 3.5. Kapitola rovněž nabízí několik ukázkových rozšíření modelu poskytovatelů.
- **Kapitola 14, „Navigace na sídle“.** Je naprosto zjevné, že mnoho vývojářů nepracuje jen s jednotlivými stránkami, ale že sestavuje aplikace. Proto potřebují mechaniku zajišťující určitou funkčnost v celé aplikaci a nejen na stránkách. Jednou z nových aplikačních schopností zajišťovaných ASP.NET 3.5 je systém navigace na sídle, jak jej popisuje tato kapitola. Základní navigační systém vám dovoluje definovat pomocí souboru XML navigační strukturu aplikace a zavádí celou řadu nových navigačních serverových ovládacích prvků pracujících s daty z takových souborů XML.
- **Kapitola 15, „Zosobnění“.** Vývojáři vždy hledají možnosti ukládání informací souvisejících s koncovými uživateli. Po uložení musí zosobňovací data zůstat trvale k dispozici pro následující návštěvy uživatele nebo získání dalších stránek v téže aplikaci. Tým ASP.NET vyvinul způsob ukládání takových údajů – zosobňovací systém ASP.NET. Na tomto systému je skvělé, že celé jeho chování nastavujete v souboru `web.config`.
- **Kapitola 16 „Správa členství a rolí“** Tato kapitola pojednává o novém systému správy členství a rolí, vyvinutému kvůli zjednodušení doplňování ověřování a autorizace do aplikací ASP.NET. Oba nové systémy jsou rozsáhlé – naprosto zastiňují některé komplikovanější ověřovací a opravňovací implementace z minulosti. Tato kapitola se zaměřuje na použití souboru `web.config` k řízení aplikování uvedených systémů a také na nové serverové ovládací prvky, jež na těchto systémech staví.
- **Kapitola 17, „Rámce portálu a moduly Web Parts“.** Tato kapitola vysvětluje moduly Web Parts – nový způsob zapouzdřování stránek do menších a spravovatelnějších objektů. Na těchto modulech je výborné to, že mohou být vytvořeny z většího portálového rámce, který může následně koncovým uživatelům povolit zcela změnit způsob konstrukce modulů Web Parts na stránce – včetně jejich vzhledu a rozvržení.
- **Kapitola 18, „Návrh HTML a CSS pomocí ASP.NET“.** Ve Visual Studiu 2008 byla věnována velká pozornost sestavování webových aplikací využívajících CSS. Tato kapitola blíže zkoumá, jak můžete efektivně pracovat s návrhem HTML a CSS ve svých aplikacích ASP.NET.
- **Kapitola 19, „ASP.NET a AJAX“.** AJAX je ve světě webových aplikacích horkou novinkou. AJAX je akronymem *Asynchronous JavaScript and XML* (asynchronní JavaScript a XML) a ve webovém vývoji představuje schopnost vytvářet aplikace využívající objekt XMLHttpRequest.

Novinkou Visual Studia 2008 je možnost ihned vytvářet aplikace ASP.NET podporující AJAX. Tato kapitola popisuje nový způsob sestavování takových aplikací.

- **Kapitola 20, „Sada nástrojů AJAX ASP.NET“.** Společně s novými možnostmi sestavování aplikací ASP.NET využívajících technologii AJAX přichází řada nových ovládacích prvků, které vše velmi zjednodušují. Tato kapitola blíže zkoumá sadu nástrojů AJAX ASP.NET a její použití v aplikacích.
- **Kapitola 21, „Zabezpečení“.** Kapitola 21 popisuje zabezpečení za hranicemi funkcí správy členství a rolí zajišťovaných v ASP.NET 3.5. Tato kapitola nabízí detailní pohled na mechaniku ověřování a autorizace vestavěnou do technologie ASP.NET a rovněž na typy přístupu HTTP a zosobnění.
- **Kapitola 22, „Správa stavu“.** Protože ASP.NET je technologie vycházející z principů požadavku a odpovědi, správa stavu a výkonnost požadavků a reakcí má zvláštní význam. Tato kapitola představuje tyto dvě samostatné, ale důležité oblasti vývoje ASP.NET.
- **Kapitola 23, „Kešování“.** Díky podstatě požadavku a následné odpovědi technologie ASP.NET je kešování (ukládání dříve vygenerovaných výsledků, obrázků a stránek do mezipaměti) na serveru poměrně důležitým aspektem výkonnosti aplikací ASP.NET. Tato kapitola sleduje některé pokročilejší kešovací možnosti nabízené ASP.NET a to včetně funkce zrušení platnosti mezipaměti SQL, jak ji obsahuje ASP.NET 3.5.
- **Kapitola 24, „Techniky ladění a zpracování chyb“.** Každá vámi sestavená aplikace ASP.NET by měla být schopna zpracovávat neočekávané chyby. Tato kapitola vám řekne, jak ve svých aplikacích správně strukturovat zpracování chyb. Ukazuje také použití různých ladicích technik k vyhledání chyb, jež může vaše aplikace obsahovat.
- **Kapitola 25, „Souborové operace I/O a proudy“.** Velmi často budou muset vaše aplikace ASP.NET pracovat s položkami mimo svůj základ. Mezi příklady patří soubory a proudy. Tato kapitola blíže zkoumá práci s různými typy souborů a proudů, které může vaše aplikace ASP.NET přijímat.
- **Kapitola 26, „Uživatelské a serverové ovládací prvky“.** Nejenže můžete aplikovat velké množství serverových ovládacích prvků obsažených v ASP.NET, ale tentýž rámec můžete využít k sestavení svých vlastních. Tato kapitola popisuje sestavování vlastních serverových ovládacích prvků a jejich používání v aplikacích.
- **Kapitola 27, „Moduly a handlers“.** Občas vám (jako vývojáři) prostě vytváření dynamických webových stránek nejnovějšími jazyky a s posledními databázemi neposkytne dostatečnou kontrolu nad aplikací. Někdy budete potřebovat ponořit se hlouběji a vytvářet aplikace interagující se samotným webovým serverem. Budete chtít pracovat s nízkourovňovými procesy, jako je zpracování příchozích a odchozích požadavků HTTP webovým serverem. Tato kapitola popisuje dvě metody úpravy způsobu, jakým ASP.NET zpracovává požadavky HTTP: `httpModule` a `HttpHandler`. Obě metody nabízejí zvláštní úroveň přístupu k základnímu zpracovávání ASP.NET a mohou tak představovat mocné nástroje pro tvorbu webových aplikací.
- **Kapitola 28, „Práce s existujícími objekty“.** Docela jistě budete mít nějaké komponenty vytvořené pomocí dřívějších technologií, které nebudete chtít přepracovávat, ale integrovat do nových aplikací ASP.NET. Rámec .NET Framework umožňuje v takových situacích poměrně jednoduše a přímočaře integrovat do nových aplikací dříve vytvořené komponenty COM. Tato kapitola vám ukáže nejen to, jak zahrnout komponenty COM do aplikací, ale také jak sestavit komponenty .NET novějšího stylu, nahrazující dřívější architekturu komponent COM.

- **Kapitola 29, „Sestavování a spotřebovávání služeb“.** Webové služby XML na sebe v posledních letech přitáhly velkou pozornost a jeden ze zásadních aspektů modelů webových služeb v .NET je také součástí ASP.NET. Tato kapitola odhaluje jednoduchost nejen sestavování webových služeb XML, ale také jejich spotřebovávání v aplikacích ASP.NET. Kapitola se pak pouští ještě dále a popisuje, jak vystavět webové služby XML využívající hlavičky SOAP a jak tento konkrétní typ služby využívat.
- **Kapitola 30, „Lokalizace“.** Vývojáři obvykle sestavují webové aplikace v angličtině, ovšem jak se rozšiřuje základna jejich uživatelů, uvědomují si potřebu globalizace. Ideální je pochopitelně sestavovat už od začátku webovou aplikaci tak, aby byla mezinárodní, to ovšem v mnoha případech nemusí být možné kvůli objemu dodatečné práce. ASP.NET poskytuje úžasnou možnost sestavování mezinárodních webových aplikací. Rychle zjistíte, že změny rozhraní API, doplnění schopností serverovým ovládacím prvkům a dokonce i vylepšení Visual Studia samotného vám dovolují vykonat všechnu dodatečnou práci snáze a nabídnout tak aplikaci celému světu. Tato kapitola zkoumá některé z důležitých okolností, jež je zapotřebí zvážit při sestavování celosvětově používaných webových aplikací.
- **Kapitola 31, „Konfigurace“.** Konfigurace může být v ASP.NET rozsáhlým tématem, protože tým ASP.NET nechtěl vytvořit nepřístupné a neproniknutelné nástrojové bloky. Místo toho sestavuje základní schopnosti ASP.NET způsobem, který lze později snadno rozšiřovat. Tato kapitola vás naučí změnit možnosti a chování ASP.NET pomocí různých konfiguračních souborů, jež máte k dispozici.
- **Kapitola 32, „Instrumentace“.** ASP.NET 3.5 rozšiřuje možnosti aplikování instrumentačních technik na aplikace. Rámec ASP.NET zahrnuje výkonnostní čítače, schopnost pracovat se systémem Windows Event Tracing (sledování událostí Windows), možnosti trasování aplikace (jak je popisuje kapitola 24) i to nejzajímavější – systém sledování zdraví dovolující protokolovat řadu různých událostí za celou dobu běhu aplikace. Tato kapitola detailně zkoumá právě tento systém sledování zdraví.
- **Kapitola 33, „Správa a řízení“.** Tým ASP.NET se nesnažil život zjednodušit jen vývojářům a těm umožnit dosáhnout vyšší produktivity při sestavování aplikací ASP.NET, ale také se zabýval možnostmi zjednodušení správy aplikací. V minulosti jste při práci s ASP.NET 1.0/1.1 spravovali aplikace ASP.NET změnami hodnot v nějakém konfiguračním souboru XML. Tato kapitola nabízí přehled nových grafických nástrojů obsažených v poslední verzi, jež vám dovolují spravovat webové aplikace jednoduše a efektivně.
- **Kapitola 34, „Zabalení a zavedení aplikací ASP.NET“.** Vytvořili jste tedy nějakou aplikaci ASP.NET. Co teď? Tato kapitola činí další krok za proces sestavení a ukazuje, jak takové aplikace ASP.NET zabalit a jednoduše zavádět. Výsledný produkt předávaný zákazníkovi můžete výrazně změnit mnoha volbami, které jsou k dispozici při práci s instalátory a kompilačním modelem.
- **Příloha A, „Migrování starších projektů ASP.NET“.** V některých případech budete sestavovat své aplikace ASP.NET 3.5 od začátku a vše bude nové. V jiných situacích to ovšem nebude možné. Budete muset vzít nějakou aplikaci ASP.NET určenou pro verzi 1.0, 1.1 nebo 2.0 rámce .NET Framework a migrovat ji tak, aby běžela na .NET Framework 3.5. Tato příloha se zaměřuje na migrování aplikací ASP.NET 1.x nebo 2.0 na rámec verze 3.5.

- **Příloha B, „Dokonalé nástroje ASP.NET“.** Tato kapitola blíže popisuje nástroje dostupné vývojářům pracujícím v ASP.NET. Mnoho z nich vám pomůže zrychlit vývojový proces a v řadě případů z vás dokonce udělá lepšího vývojáře.
- **Příloha C, „Silverlight“.** Technologie označovaná za WPF/E během svého vývoje nyní nese název Silverlight. Jedná se o prostředek sestavování aplikací jazykem XAML. Tato nová technologie dovoluje vývojářům sestavovat skutečně bohaté vektorové aplikace.
- **Příloha D, „Prostředky ASP.NET“.** Tato malá příloha vás nasměruje na hodnotné prostředky online, jejichž prostřednictvím můžete dále vylepšit své znalosti ASP.NET.

Konvence

Kniha využívá různé textové a odstavcové styly, kterými rozlišuje odlišné typy informací. Zde jsou příklady používaných stylů a vysvětlení jejich významu:

- Nově definovaná slova jsou znázorněna *kurzívou*.
- Klávesy, jako jsou Ctrl a Enter, jsou zapsány s velkým prvním písmenem a dále pak podle svého označení na klávesnici.
- Názvy souborů a složek, přípony souborů, adresy URL a kód se v běžném textu uvádějí neproporcionálním písmem.

Když představujeme nějaký blok kódu, který můžete zadat jako program a spustit, uvádíme jej na samostatných řádcích ve formě

```
public static void Main()  
{  
    AFunc(1,2,"abc");  
}
```

nebo tímto způsobem:

```
public static void Main()  
{  
    AFunc(1,2,"abc");  
}
```

Občas se setkáte také s kódem zapsaným kombinací stylů:

```
// Pokud jsme nedosáhli konce, vrátíme true, jinak  
// nastavíme pozici na neplatnou a vrátíme false.  
pos++;  
if (pos < 4)  
    return true;  
else {  
    pos = -1;  
    return false;  
}
```

Když je nějaký kód uvedený tímto způsobem, pak kód bez pozadí představuje zadání, které již bylo uvedeno dříve a jímž se nemusíte znovu zabývat. V aktuálním příkladu byste se měli zaměřit na kód s šedým pozadím.

Syntaktické použití metod, vlastností atd. demonstrujeme s využitím tohoto formátu:

```
SqlDependency="databáze:tabulka"
```


Slova zapsaná kurzivou představují *zástupný text*: odkazy na objekty, proměnné nebo parametrické hodnoty, které musíte zadat.

Většina ukázek kódu v celé knize je uváděna pod číslovanými výpisy s popisným označením, jako například:

Výpis Ú.7: Zacílení zařízení WML ve stránkách ASP.NET

Každý výpis má své označení (kupříkladu *1.3*), kde první číslo představuje kapitolu a číslo za tečkou je pak pořadové označení představující umístění výpisu v dané kapitole. Kód, který si můžete stáhnout z webového sídla společnosti Wrox (www.wrox.com), používá tentýž číslovací systém, takže si lze snadno vyhledat požadované příklady.

Když to má smysl, je veškerý kód zachycen ve verzi VB i C#. Výjimkou je kód, v němž je jedinou odlišností např. hodnota zadaná atributu `Language` v direktivě `Page`. V takových případech neopakuje kód ve verzi pro jazyk C#; kód uvádíme jen jednou, jako je tomu v následující ukázce:

```
<%@ Page Language="VB"%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>DataSetDataSource</title>
</head>
<body>
    <form id="form1" runat="server">
        <asp:DropDownList ID="Dropdownlist1" Runat="server" DataTextField="name"
            DataSourceID="XmlDataSource1">
        </asp:DropDownList>

        <asp:XmlDataSource ID="XmlDataSource1" Runat="server"
            DataFile="~/Painters.xml">
        </asp:DataSetDataSource>
    </form>
</body>
</html>
```

Zdrojový kód

Při práci s příklady v knize můžete zadávat veškerý kód manuálně, nebo použít doprovodné soubory zdrojového kódu. Veškerý **originální zdrojový kód** naší knihy je k dispozici ke stažení na adrese www.wrox.com. Jakmile se dostanete na toto sídlo, vyhledejte si název knihy (Professional ASP.NET 3.5 – buď pomocí políčka hledání, nebo ze seznamu témat) a klepněte na odkaz `Download Code`. Pak si můžete vybrat mezi stažením kompletních kódů v jednom velkém zabaleném souboru nebo jen kódem určité kapitoly.

Lokalizovaný zdrojový kód použitý v překladu této knihy si můžete stáhnout z webové stránky knihy na adrese <http://knihy.cpress.cz/k1603>.

Jakmile máte kód stažený, rozbalte jej nějakým svým oblíbeným dekomprimačním nástrojem. Můžete také přejít na hlavní stránku stahování kódu nakladatelství Wrox na adrese www.wrox.com/dynamic/books/download.aspx a pracovat s veškerým kódem této knihy i dalších. Pamatujte, že

potřebný kód odhalíte pomocí jeho čísla v knize, jak je uvedeno např. v označení „Výpis 1.3“. Stejná čísla výpisů jsme použili k pojmenování stáhnutelných souborů zdrojového kódu.

Opravy

Snažíme se, aby se v textu knihy ani v kódech neobjevily žádné chyby. Nikdo ale není dokonalý a k chybám dochází. Najdete-li v některé z našich knih chybu, například překlep nebo nefungující kód, oceníme, když nám to oznámíte. Oznámením chyb můžete ušetřit jiným čtenářům hodiny frustrace; zároveň nám tím pomáháte nabízet ještě kvalitnější informace.

Seznam chyb v této knize najdete na sídle www.wrox.com, když si titul vyhledáte pomocí políčka hledání nebo v seznamu. Na stránce podrobností o knize pak klepněte na odkaz Book Errata. Pak již spatříte všechny oznámené opravy, jak je sem umístili redaktoři nakladatelství Wrox. Úplný seznam knih včetně jejich oprav je k dispozici také na www.wrox.com/misc-pages/booklist.shtml.

Jestliže chybu, kterou jste sami odhalili, nenajdete na stránce Book Errata, přejděte na www.wrox.com/contact/techsupport.shtml a vyplňte oznamovací formulář. Informace prověříme a budou-li potvrzeny, umístíme na stránku chyb knihy příslušnou připomínku a problém v dalších vydáních odstraníme.

Poznámka redakce českého vydání knihy

I nakladatelství Computer Press, které pro vás tuto knihu přeložilo, stojí o zpětnou vazbu a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

Computer Press
redakce PC literatury
Holandská 8
639 00 Brno

nebo

knihy@cpress.cz

Další informace a případné opravy českého vydání knihy najdete na internetové adrese <http://knihy.cpress.cz/k1603>. Prostřednictvím uvedené adresy můžete též naší redakci zaslat komentář nebo dotaz týkající se knihy. Na vaše reakce se srdečně těšíme.