

Uvod u Web programiranje

Lekcija 1: Tipovi aplikacija	2
Lekcija 2: Korišćenje ASP.NET-a	7
Lekcija 3: Korišćenje Visual Studio .NET-a	18

O ovom poglavlju

U ovom poglavlju naučićete sve o aplikacijama za Internet i alatima koje koristite da biste ih napravili. Upoznaćete, takođe, koncepte i pojmove koji se koriste u celom udžbeniku, kao i način na koji ASP.NET veb aplikacije funkcionišu, delove veb aplikacije, način na koji je organizovano radno okruženje *Microsoft .NET Framework* i kako se koristi programsko okruženje *Microsoft Visual Studio .NET*. U odeljcima ovog poglavlja dobićete uopšten pregled pojmova koji se detaljnije onjavaju u narednim poglavljima.

Pre nego što počnete

Da biste prešli lekcije iz ovog poglavlja, morate da imate:

- Visual Studio .NET instaliran na svom računaru;
- pristup Internetu preko lokalne računarske mreže (LAN-a), širokopojasne mreže ili modemske veze.

Lekcija 1: Tipovi aplikacija

Postoje četiri različita tipa Internet aplikacija koje možete da pravite koristeći Visual Studio .NET. U ovoj lekciji naučićete pomalo o svakom tipu Internet aplikacija i steći opštu sliku o tome kako ASP.NET veb aplikacija funkcioniše.

Po završetku ove lekcije, moći ćete da:

- opišete četiri različita tipa Internet aplikacija i saznate gde da potražite gradivo o razvoju svakog pojedinog tipa;
- objasnite kako se veb aplikacija izvršava na Internetu i po čemu se ona razlikuje od tradicionalne, statičke veb lokacije;
- razumete ulogu koju ASP.NET igra u izradi veb aplikacija;
- navedete delove iz kojih je sastavljen ASP.NET i opišete neke prednosti koje ovo okruženje ima u odnosu na druge tehnologije za izradu veb aplikacija, kao što je zajednički interfejs mrežnog prolaza.

Vreme potrebno za lekciju: 5 minuta

Šta možete da napravite?

Strogo govoreći, Internet aplikacija je svaka aplikacija koja, na neki način, koristi Internet. To znači da se i aplikacije koje od korisnika zahtevaju da se registruju na Internet ili koje pružaju pomoć korišćenjem Interneta, u izvesnom smislu svrstavaju u grupu aplikacija za Internet.

Ta definicija je preširoka da bi mogla da stane u jednu knjigu. Da bismo barem malo suzili oblast koja nas zanima, najpre ćemo identifikovati četiri tipa aplikacija za Internet:

- **Veb aplikacije** To su aplikacije koje obezbeđuju sadržaj od servera do računara klijenta preko Interneta. Korisnici pristupaju veb aplikaciji preko veb pretraživača.
- **Veb servisi** To su komponente kojima se obezbeđuju usluge obrade podataka od jednog servera do drugih aplikacija na Internetu.
- **Aplikacije koje koriste Internet** To su samostalne aplikacije koje objedinjuju razne aspekte Interneta kako bi obezbedile onlajn registraciju, pomoć, ažuriranje podataka ili druge usluge korisniku na Internetu.
- **Aplikacije za ravnopravne računare** To su samostalne aplikacije koje koriste Internet za komuniciranje sa drugim korisnicima na čijim računarima se izvršavaju odgovarajuće instance date aplikacije.

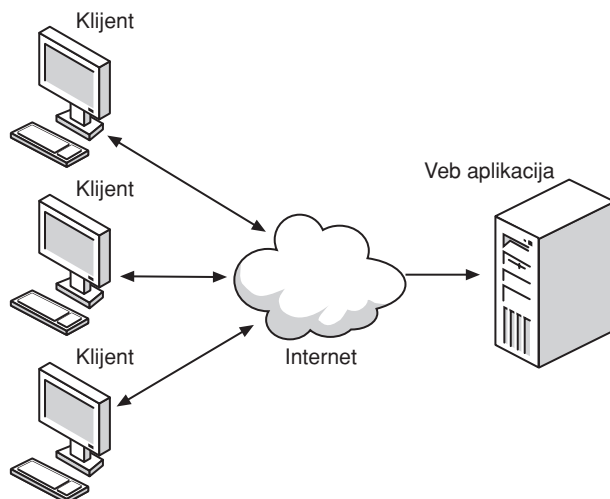
Možete da koristite Visual Studio .NET za pravljenje bilo kog od navedenih tipova aplikacija. Predmet proučavanja ove knjige je prvi tip, veb aplikacije. U tabeli 1-1 prikazane su razne teme za pomoć okruženja Visual Studio .NET i udžbenici iz ove serije u kojima je obrađen svaki tip Internet aplikacije.

Tabela 1-1 Izvori informacija o Internet aplikacijama

Tip aplikacije	Korišćenje ovih tema u onlajn pomoći	Priprema za ispit MCSD korišćenjem
vab aplikacije	ASP.NET, veb obrasci, prostor imena <i>System.Web</i>	Ove knjige
Web servisi	ASP.NET, XML veb servisi, prostor imena <i>System.Web.Services</i>	<i>MCAD/MCSD Udžbenik za pripremu ispita individualnim tempom: Razvoj XML veb servisa i serverskih komponenti pomoću Microsoft Visual Basic .NET-a i Microsoft Visual C# .NET-a</i>
Aplikacije koje koriste Internet	Microsoft Windowsovi obrasci, HTML pomoć, WebBrowser kontrola, prostor imena <i>System.Net</i>	<i>MCAD/MCSD Udžbenik za pripremu ispita individualnim tempom: Razvoj aplikacija zasnovanih na Windowsu alatima Microsoft Visual Basic .NET i Microsoft Visual C++</i>
Aplikacije za ravnopravne računare	Pristupanje Internetu, zamjenljivi protokoli, prostori imena <i>System.Net</i> i <i>System.Net.Sockets</i>	Teme u okviru onlajn pomoći za ove prostore imena i MSDN članci o aplikacijama za ravnopravne računare i klijentsko-serverske aplikacije sa okruženjem .NET Framework

Kako veb aplikacije funkcionišu

Veb aplikacije koriste arhitekturu klijent-server. Veb aplikacija se nalazi na serveru i reaguje na zahteve koje upućuje veći broj klijenata preko Interneta, kao što je prikazano na slici 1-1.

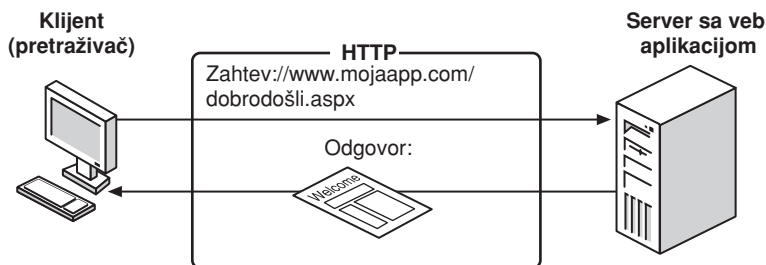


Slika 1-1 Arhitektura ASP.NET veb aplikacije

Na strani klijenta, veb aplikacija se smešta u pretraživač. Korisnički interfejs aplikacije poprima oblik stranica hipertekstualnog markerskog jezika (HTML) koje interpretira i prikazuje klijentov pretraživač.

4 Razvoj Web aplikacija: Microsoft Visual Basic .NET i Visual C# .NET

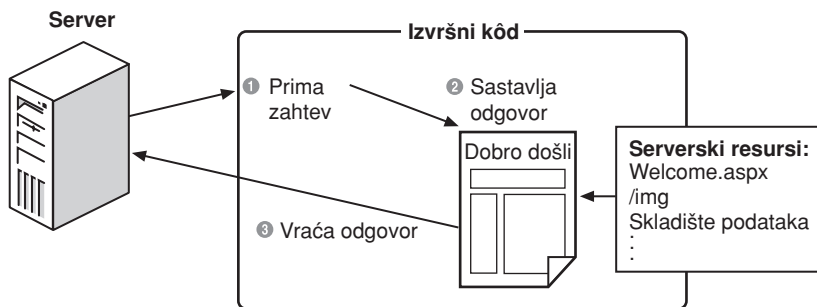
Na strani servera, veb aplikaciju izvršavaju Microsoftovi servisi za informacije na Internetu (*Internet Information Services – IIS*). IIS upravlja aplikacijom, prenosi zahteve od klijenata do aplikacije i vraća odgovore aplikacije klijentu. Ovi zahtevi i odgovori prenose se preko Interneta korišćenjem protokola za prenos hiperteksta (*Hypertext Transport Protocol – HTTP*). *Protokol* je skup pravila kojima se opisuje način na koji dva ili više lica komuniciraju putem medija, kao što je Internet. Na slici 1-2 prikazuje se kako se vrši interakcija između klijenta i servera na Internetu.



Slika 1-2 Interakcija između klijenta i servera u veb aplikaciji

Veb aplikacija sastavlja odgovore na zahteve od resursa koje pronalazi na serveru. U ove resurse spadaju: izvršni kôd koji se izvršava na serveru (a koji, po tradiciji, inače smatramo aplikacijom u Microsoft *Windows* programiranju), veb obrasci, HTML stranice, datoteke sa slikama i ostali mediji koji ulaze u sastav aplikacije.

Veb aplikacije umnogome liče na tradicionalne veb lokacije, a razlikuju se od njih jedino po tome što se sadržaj, koji se predstavlja korisniku, zapravo sastavlja dinamički, izvršavanjem koda, umesto da se servira sa statičke stranice uskladištene na serveru. Na slici 1-3, prikazano je kako veb aplikacija sastavlja odgovor u vidu HTML-a koji vraća korisniku.



Slika 1-3 Odgovor ASP.NET veb aplikacije iz serverskih resursa

Izvršna datoteka veb aplikacije omogućava vam da uradite mnoge stvari koje ne možete da obavite pomoću statičke veb lokacije, i to:

- prikupljanje informacija od korisnika i skladištenje tih informacija na serveru;

- obavljanje zadataka za korisnika, kao što su: dostava porudžbenice za neki proizvod, vršenje složenih računskih operacija, ili dobijanje informacija iz baze podataka;
- identifikovanje konkretnog korisnika i predstavljanje interfejsa koji je njemu prilagođen;
- predstavljanje sadržaja koji se menja velikom brzinom, kao što su: popis inventara, porudžbina čija je realizacija u toku i informacije o otpremanju robe.

Ovaj spisak je samo delimičan. U suštini, sa veb aplikacijom možete da radite sve što bi vam palo na pamet da može da se obavi pomoću bilo koje aplikacije tipa klijent-server. Ono što veb aplikaciju čini posebnom jeste činjenica da se interakcija između klijenta i servera odigrava na Internetu.

Šta obezbeđuje ASP.NET

ASP.NET je platforma koja se koristi za izradu veb aplikacija i veb servisa koji se izvršavaju u IIS-u. ASP.NET nije jedino sredstvo pomoću kojeg se pravi veb aplikacija. Ima i drugih tehnologija, pre svega CGI, koje vam, takođe, omogućavaju da pravite veb aplikacije. Međutim, ASP.NET je specifičan po tome što se izuzetno čvrsto integriše u Microsoftov server, programiranje, pristup podacima i alatke za bezbednost.

ASP.NET obezbeđuje visok nivo doslednosti tokom razvoja veb aplikacije. Ova doslednost, na neki način, dostiže nivo doslednosti koji je Microsoft Office uneo u aplikacije namenjene upotrebi na stonim računarima. ASP.NET je deo radnog okruženja .NET Framework i sastoji se od nekoliko komponenti:

- **Visual Studio .NET-ove razvojne alatke za Web** Tu spadaju vizuelne alatke za dizajniranje veb stranica i aplikativni šabloni, upravljanje projektom i uvođenje alatki za veb aplikacije.
- **Prostori imena *System.Web*** To su delovi radnog okruženja .NET Framework koji obuhvataju programske klase koje se tiču stavki svojstvenih Webu, kao što su HTTP zahtevi i odgovori, pretraživači i e-pošta.
- **Serverske i HTML kontrole** Ovo su komponente korisničkog interfejsa koje upotrebljavate da biste od korisnika prikupljali informacije kao i da biste im vraćali odgovore.

Osim navedenih komponenti, ASP.NET koristi i neke druge, uopštenije, programske komponente i Windowsove alatke, koje, same po sebi, nisu sastavni deo ASP.NET-a. Međutim, one su izuzetno značajne za ASP.NET programiranje, a obuhvataju:

- **Microsoft Internet Information Services (IIS)** Kao što smo napomenuli u prethodnom odeljku, IIS smešta veb aplikacije na Windows server.
- **Programski jezici Microsoft *Visual Basic .NET*, Microsoft *Visual C#* i Microsoft *Visual J#*** Ova tri jezika imaju integralnu podršku u okruženju Visual Studio .NET za izradu veb aplikacija.
- **Radno okruženje .NET Framework** Ono se sastoji od čitavog niza Windowsovih programskih klasa, uključujući ASP.NET klase, ali i klase za druge programske zadatke, kao što su: pristup datotekama, konverzija tipova podataka, rad sa nizovima i stringovima i drugo.

- **Klase i alatke Microsoft ADO.NET za baze podataka** To su komponente koje omogućuju pristup Microsoft SQL serveru i ODBC bazama podataka. Pristup podacima je često ključna komponenta veb aplikacija.
- **Microsoft Application Center Test (ACT)** Ovom komponentom Visual Studio .NET-a obezbeđuje se automatizovano testiranje naprezanja veb aplikacija.

ASP.NET je najpotpunija platforma za razvoj veb aplikacija koje se izvršavaju u IIS-u. Međutim, važno je da zapamtite da ASP.NET nije nezavisan od platforme. S obzirom na to da se smešta pod IIS-om, ASP.NET mora da se izvršava na Windows serverima. Da biste napravili veb aplikacije koje rade na serverima, tipa drugačijeg od Windows/IIS, recimo Linux/Apache, morate da se služite drugim alatkama – uglavnom CGI-em.

Prednosti ASP.NET-a

ASP.NET ima mnoge prednosti u odnosu na druge platforme kada je u pitanju pravljenje veb aplikacija. Najznačajniju prednost, verovatno, predstavlja njegovo integrisanje u Windows servere i alatke za programiranje. Veb aplikacije napravljene pomoću ASP.NET-a lakše se izrađuju, iz njih se brže otklanjanju programске greške i njihovo uvođenje u upotrebu je lakše zato što se svi ovi poslovi obavljaju u samo jednom razvojnom okruženju – Visual Studio .NET-u.

Razvijajocima veb aplikacija, ASP.NET nudi i sledeće pogodnosti:

- izvršne datoteke veb aplikacije su kompajlirane, tako da se izvršavaju brže od interpretiranih skriptova;
- ažuriranje uvedenih veb aplikacija obavlja se u hodu, bez ponovnog pokretanja servera;
- pristup radnom okruženju .NET Framework, koje pojednostavljuje mnoge aspekte Windows programiranja;
- koristi Visual Basic, programski jezik poznat ogromnom broju korisnika, a koji je poboljšan kako bi potpuno podržao objektno orijentisano programiranje;
- uvođenje novog programskog jezika, Visual C#-a, koji predstavlja bezbedno tipiziranu objektno orijentisanu verziju programskog jezika C-a;
- automatsko upravljanje stanjem kontrola na veb stranici (poznatim pod nazivom **serverske kontrole**), tako da se one ponašaju mnogo više nalik Windowsovim kontrolama;
- mogućnost pravljenja novih, namenskih serverskih kontrola na osnovu postojećih kontrola;
- ugrađenu bezbednost preko Windows servera ili pomoću drugih metoda provere autentičnosti i ovlašćivanja;
- integrisanost sa ADO.NET-om kako bi se obezbedio pristup bazi podataka i alati za dizajniranje baze podataka u okviru samog Visual Studio .NET-a;
- puna podrška za proširivi markerski jezik (*Extensible Markup Language – XML*), kaskadne stilske listove (*Cascading Style Sheet – CSS*) i druge nove i ustanovljene veb standarde;
- ugrađene mogućnosti za keširanje često traženih veb stranica na serveru, lokalizovanje sadržaja za konkretne jezike i kulturološke celine i utvrđivanje sposobnosti pretraživača.

Lekcija 2: Korišćenje ASP.NET-a

U ovoj lekciji naučićete kako ASP.NET organizuje veb aplikacije po pojedinim delovima i saznati koje su uloge tih delova i kako se oni nazivaju. Upoznaćete se sa veb obrascima koji su centralni element korisničkog interfejsa veb aplikacija.

ASP.NET je deo većeg radnog okruženja .NET Framework, pa će se u ovoj lekciji takođe govoriti o tome kako je radno okruženje .NET Framework organizovano i po čemu se .NET aplikacije razlikuju od tradicionalnih Windowsovih aplikacija na koje ste možda navikli.

Konačno, lekciju završavamo gradivom o programskim jezicima koje možete da koristite za stvaranje veb aplikacija. ASP.NET nije vezan ni za jedan konkretan programski jezik, pa ćemo, stoga, pri kraju lekcije navesti i druge raspoložive jezike i pojasniti neke važnije razlike između dva jezika (to su Visual Basic .NET i Visual C#) kojima se služimo u ovom udžbeniku.

Po završetku ove lekcije, moći ćete da:

- navedete delove veb aplikacije i opišete kako oni funkcionišu na serveru;
- objasnite po čemu se veb obrazac razlikuje od HTML stranice i Windowsovog obrasca i u čemu im je sličan;
- opišete neke različite komponente koje možete postaviti na veb obrazac;
- date objašnjenje u vezi sa delovima radnog okruženja .NET Framework i sa načinom na koji zajedničko jezičko izvršno okruženje (*Common Language Runtime* – CLR) izvršava .NET aplikacije;
- shvatite kako je radno okruženje .NET Framework organizovano i saznate gde da pronađete klase koje upravljaju uobičajenim zadacima programiranja aplikacija;
- napravite poređenje između programskih jezika Visual Basic .NET i Visual C#.

Vreme potrebno za lekciju: 10 minuta

Delovi veb aplikacije

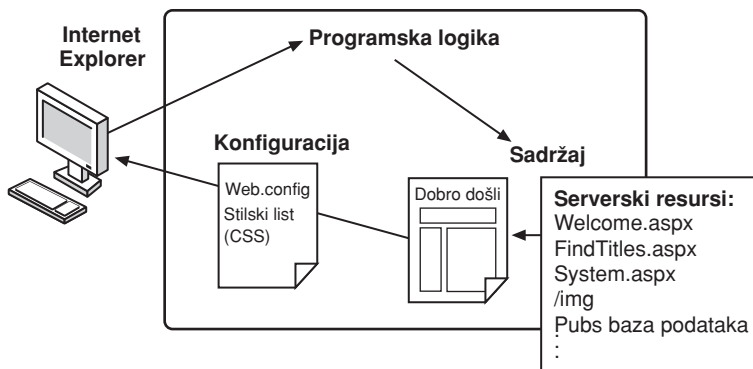
Veb aplikacija se sastoji iz tri dela: sadržaja, programske logike i informacija o veb konfiguraciji. U tabeli 1-2 dat je sažet prikaz ovih delova i navedeni su primeri na osnovu kojih se može utvrditi mesto na kojem se oni nalaze u ASP.NET veb aplikaciji.

Tabela 1-2 Delovi ASP.NET veb aplikacije

Deo	Tipovi datoteka	Opis
Sadržaj	veb obrasci, HTML, slike, audio-zapisi, video-zapisi, ostali podaci	Datoteke sa sadržajem određuju izgled veb aplikacije. One mogu da sadrže statički tekst i slike, kao i elemente koje u hodu sastavlja programska logika (kao što je slučaj sa upitom nad bazom podataka).
Programska logika	izvršne datoteke, skriptovi	Programska logika utvrđuje kako aplikacija reaguje na radnje korisnika. ASP.NET veb aplikacije imaju biblioteku za dinamičko povezivanje (<i>Dynamic-Link Library – DLL</i>), datoteku koja se izvršava na serveru, a koja može da uključi i skriptove koji se izvršavaju na računaru klijenta.
Konfiguracija	datoteka veb konfiguracije, stilski listovi, parametri IIS-a	Konfiguracione datoteke i parametri određuju kako se aplikacija izvršava na serveru, ko ima pristup, kako se otklanjaju greške, i druge pojedinosti.

Veb obrazac je ključni element veb aplikacije. Može se reći da je **veb obrazac** (*Web form*) „hibrid” nastao ukrštanjem obične HTML stranice i Windowsovog obrasca – on ima isti izgled i slično ponašanje kao HTML stranica, ali pri tom ima i kontrole koje reaguju na događaje i izvršavaju kôd, baš kao Windowsov obrazac.

Kod završene veb aplikacije, izvršni deo veb obrasca uskladišten je u sklopu (.dll) koji se izvršava na serveru pod kontrolom ASP.NET radnog procesa (asp_wp.exe) koji se odvija zajedno sa IIS-om. Deo sa sadržajem veb obrasca čuva se u direkto-rijumu sadržaja veb servera, kao što je prikazano na slici 1-4.



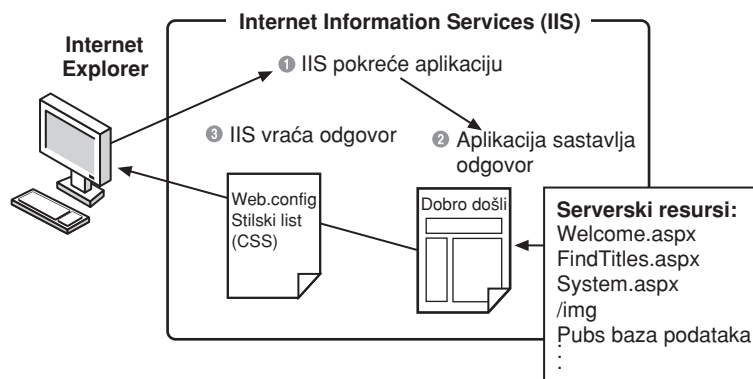
Slika 1-4 Delovi ASP.NET veb aplikacije na veb serveru

Kada korisnik, preko vlastitog pretraživača, dođe do jednog veb obrasca, operacije se odvijaju sledećim redosledom:

1. IIS pokreće ASP.NET radni proces u slučaju da njegovo izvršavanje nije upravo u toku. ASP.NET radni proces učitava sklop koji je u vezi sa tim veb obrascem.

2. Sklop sastavlja odgovor korisniku na osnovu sadržaja veb obrasca kojeg je korisnik tražio i bilo koje programske logike koja nudi dinamički sadržaj.
3. IIS vraća korisniku odgovor u obliku HTML-a.

Čim dobije traženi veb obrazac, korisnik može da upiše podatke, izabere opcije, pritisne dugmad i koristi bilo koju kontrolu među onima koje se pojavljuju na stranici. Neke kontrole, kao što su dugmad, imaju za cilj da stranicu ponovo vrate serveru radi obrade događaja i sam redosled operacija obnavlja se kao što je ilustrovano na slici 1-5.



Slika 1-5 Kako delovi aplikacije međusobno deluju

Ovaj ciklus događaja detaljnije je opisan u lekciji 2 iz poglavlja 2.

Komponente veb obrasca

Veb obrasci mogu da sadrže nekoliko različitih tipova komponenti, čiji je sažet prikaz dat u tabeli 1-3.

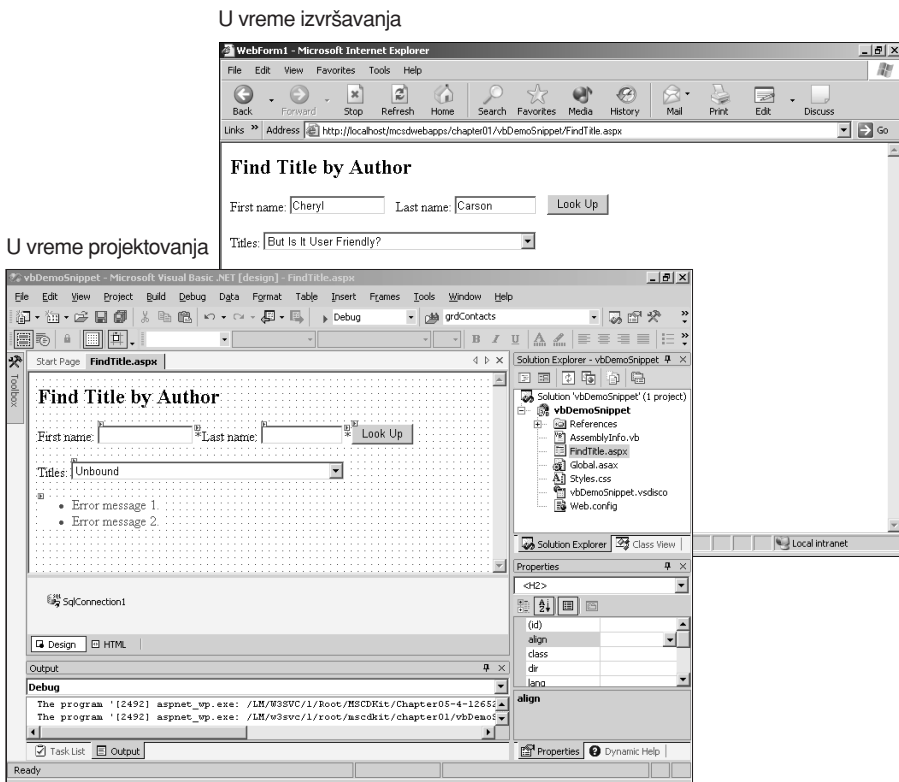
Tabela 1-3 Komponente veb obrasca

Komponenta	Primeri	Opis
Serverske kontrole	TextBox, Label, Button, ListBox, DropDownList, DataGrid	Ove kontrole reaguju na korisničke događaje pokretanjem procedura događaja na serveru. Serverske kontrole imaju ugrađene mogućnosti za čuvanje upisanih podataka između prikaza stranica. Serverske kontrole služe vam da definišete korisnički interfejs veb obrasca.
HTML kontrole	Text Area, Table, Image, Submit Button, Reset Button	To su kontrole koje predstavljaju standardne vizuelne elemente koji se vide u HTML-u. HTML kontrole su korisne kada nije neophodan čitav niz istovremenih mogućnosti koje nude serverske kontrole.

Tabela 1-3 Komponente veb obrasca

Komponenta	Primeri	Opis
Kontrole za podatke	SqlConnection, SqlCommand, OleDbConnection, OleDbCommand, DataSet	Kontrole za podatke su sredstvo za povezivanje, izvršavanje komandi i izdvajanje podataka iz SQL i OLE baza podataka i XML datoteka.
Sistemske komponente	FileSystemWatcher, EventLog, MessageQueue	Ovo su komponente koje obezbeđuju pristup različitim događajima na nivou sistema koji se dešavaju na serveru.

Serverske i HTML kontrole koriste se za pravljenje korisničkog interfejsa na veb obrascu. Kontrole za podatke i sistemske komponente pojavljuju se na veb obrascu samo u vreme projektovanja da bi vam poslužile kao vizuelno sredstvo za zadavanje njihovih svojstava i rukovanje njihovim događajima. U vreme izvršavanja, kontrole za podatke i sistemske komponente nisu vizuelno prikazane. Na slici 1-6 ilustrovan je veb obrazac koji sadrži različite komponente.



Slika 1-6 Veb obrazac sa različitim komponentama

U poglavlju 4 možete da pročitate više pojedinosti o korišćenju serverskih i HTML kontrola na veb obrascu.

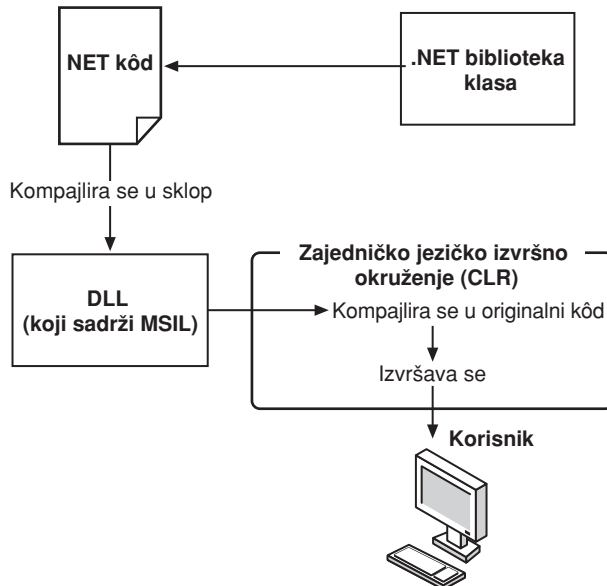
.NET Framework

ASP.NET je značajan deo radnog okruženja .NET Framework, ali on je, ipak, samo jedan deo. Saznanje o tome šta još .NET Framework može da obezbedi, pomoći će vam da ASP.NET aplikaciju efikasnije isprogramirate, a da pri tom ne morate da pišete nov kôd za obavljanje zadataka koji su već implementirani u okviru radnog okruženja .NET Framework.

Podsetićemo se, najpre, na neke važnije činjenice. Radno okruženje .NET (Framework) je nova Microsoftova programska platforma za razvoj Windowsovog i veb softvera. Sastoji se od dva dela; to su:

- izvršna softverska mašina koja se zove zajedničko jezičko izvršno okruženje (CLR);
- biblioteka klasa koja obezbeđuje osnovne programske funkcije, kao što su funkcije koje su ranije bile dostupne samo preko Windows API-ja i funkcije na nivou aplikacije koje se koriste za razvoj Weba (ASP.NET), pristup podacima (ADO.NET), bezbednost i upravljanje sa udaljenog računara.

.NET aplikacije se ne izvršavaju na isti način kao tradicionalne Windowsove aplikacije na čije pravljenje ste možda navikli. Umesto da se kompajlira u izvršnu datoteku koja sadrži matični kôd, .NET aplikativni kôd se kompajlira u Microsoftov posrednički jezik (*Microsoft Intermediate Language – MSIL*) i skladišti u datoteku koja se naziva sklopom (*assembly*). U vreme izvršavanja, sklop se kompajlira do svog konačnog stanja pomoću CLR-a. Tokom izvršavanja, CLR obezbeđuje upravljanje memorijom, provere bezbednosti tipa podataka i druge poslove za aplikaciju u vreme njenog izvršavanja. Slika 1-7 pokazuje kako sve to radi.



Slika 1-7 Kako se izvršava .NET aplikacija.

Aplikacije koje se izvršavaju pod CLR-om nazivaju se **upravljanim kodom** (*managed code*) budući da se CLR stara o mnogobrojnim poslovima koje bi ranije obavila sama izvršna datoteka aplikacije. Upravljeni kôd rešava probleme Windows programiranja koji se tiču registracije i verziranja komponenti (ponekad ga zovu DLL pakao) pošto sklop sadrži sve informacije o broju verzije i tipovima podataka koje su potrebne CLR-u da bi pokrenuo aplikaciju. CLR obavlja registraciju dinamički u vreme izvršavanja, a ne statički preko sistemskog registra kao što se to radi kod aplikacija zasnovanih na Component Object Modelu (COM-u).

Biblioteka klasa .NET obezbeđuje pristup svim funkcijama CLR-a. Ova biblioteka klasa organizovana je po prostorima imena. Svaki prostor imena sadrži funkcionalno srodnu grupu klasa. U tabeli 1-4 sažeto su prikazani .NET prostori imena koji su najzanimljiviji za programere veb aplikacija.

Tabela 1-4 Prikaz biblioteke klasa radnog okruženja .NET Framework

Kategorija	Prostori imena	Obezbeđuje klase za
Zajednički tipovi	System	Svi zajednički tipovi podataka, uključujući stringove, nizove i numeričke tipove. Ove klase obuhvataju i metode za konvertovanje tipova, za rad sa stringovima i nizovima, kao i za matematičke zadatke i generisanje slučajnih brojeva.
Pristup podacima	<i>System.Data, System.Data.Common, System.Data.OleDb, System.Data.SqlClient, System.Data.SqlTypes</i>	Pristupanje bazama podataka. Ove klase uključuju metode za povezivanje sa bazama podataka, izvršavanje komandi, izdvajanje podataka i modifikovanje podataka.
Otklanjanje programskih grešaka	System.Diagnostics	Otklanjanje programskih grešaka i praćenje izvršavanja aplikacije.
Pristup datotekama	<i>System.IO, System.IO.IsolatedStorage, System.DirectoryServices</i>	Pristupanje sistemu datoteka. One obuhvataju metode za čitanje i pisanje datoteka i za dobijanje putanja i naziva datoteka.
Mrežna komunikacija	<i>System.Net, System.Net.Sockets</i>	Komuniciranje na Internetu korišćenjem protokola niskog nivoa, kao što je TCP/IP. Ove klase se koriste prilikom pravljenja aplikacija za ravnopravne računare.
Bezbednost	<i>Sytem.Security, System.Security.Cryptography, System.Security.Permissions, System.Security.Policy, System.Web.Security</i>	Obezbeđivanje provere autentičnosti korisnika, ovlašćivanja korisnika i šifrovanja podataka.

Tabela 1-4 Prikaz biblioteke klasa radnog okruženja .NET Framework

Kategorija	Prostori imena	Obezbeđuje klase za
Veb aplikacije	<i>System.Web, System.Web.Caching, System.Web.Configuration, System.Web.Hosting, System.Web.Mail, System.Web.SessionState, System.Web.UI, System.Web.UI.Design, System.Web.UI.WebControls, System.Web.UI.HtmlControls</i>	Pravljenje klijentsko-serverskih aplikacija koje se izvršavaju na Internetu. To su osnovne klase koje se koriste za izradu ASP.NET veb aplikacija.
Veb servisi	<i>System.Web.Services, System.Web.Services.Configuration, System.Web.Services.Description, System.Web.Services.Discovery, System.Web.Services.Protocols</i>	Pravljenje i objavljivanje komponenti koje se mogu koristiti na Internetu. Ovo su osnovne klase koje se koriste za izradu ASP.NET veb servisa.
Windowsove aplikacije	<i>System.Windows.Forms, System.Windows.Forms.Design</i>	Pravljenje aplikacija pomoću komponenti Windows korisničkog interfejsa. Ove klase obezbeđuju Windowsove obrasce i kontrole, kao i mogućnost pravljenja namenskih kontrola.
XML podaci	<i>System.Xml, System.Xml.Schema, System.Xml.Serialization, System.Xml.XPath, System.Xml.Xsl</i>	Pravljenje XML datoteka i pristup tim datotekama.

Pošto prostori imena .NET-a organizuju klase po funkciji, možete ih koristiti da biste lakše locirali klase i članove klase koji obezbeđuju one funkcije CLR-a koje želite da upotrebite. Primera radi, prostor imena *System* je jedan od najčešće korišćenih prostora imena s obzirom na to da sadrži klase za sve osnovne tipove podataka. Svaki put kada deklarišete promenljivu sa numeričkim tipom, tipom stringa ili niza podataka, zapravo koristite prostor imena *System*.

Ovakav pristup omogućuje radnom okruženju .NET Framework da obezbedi ugrađene metode za konvertovanje tipova podataka i za rukovanje stringovima i nizovima. Na primer, redovi koda koji su prikazani u nastavku koriste ugrađene metode klasa *String* i *Array* za sortiranje liste.

Visual Basic .NET

```
' Deklarisati i inicijalizovati string.
Dim strFruit As String =
"oranges apples peaches kumquats nectarines mangos"
' Deklarisati niz.
Dim arrFruit As String ( )
' Staviti svaku reč u element niza.
arrFruit = strFruit.Split(" ")
' Sortirati niz.
System.Array.Sort(arrFruit)
' Vratiti sortirane podatke nazad u string.
```

```
strFruit = String.Join(" ", arrFruit)
```

Visual C#

```
// Deklarisati i inicijalizovati string.  
string strFruit = "oranges apples peaches kumquats nectarines mangos";  
// Deklarisati niz.  
string [ ] arrFruit;  
// Staviti svaku reč u element niza.  
arrFruit = strFruit.Split(" ".ToCharArray ( ));  
// Sortirati niz.  
System.Array.Sort(arrFruit);  
// Vratiti sortirane podatke nazad u string.  
strFruit = System.String.Join(" ", arrFruit);
```

Mnogi metodi klase u prostoru imena *System* mogu se koristiti direktno, a da pri tome ne mora najpre da se napravi objekat iz te klase. Oni se nazivaju **zajedničkim članovima** (*shared members*) u Visual Basic .NET-u, a **statičkim članovima** (*static members*) u jeziku Visual C#. Zajednički i statički članovi mogu se pozivati iz samog naziva klase, kao što se vidi u redu *System.Array.Sort* gore navedenog koda. Kao drugi primer klase sa zajedničkim i statičkim članovima može se navesti klasa *Math*, kao što prikazuju sledeći metodi *Pi* i *Pow*:

Visual Basic .NET

```
' Izračunati površinu kruga.  
dblCircArea = System.Math.Pi * System.Math.Pow(intRadius, 2)
```

Visual C#

```
// Izračunati površinu kruga.  
dblCircArea = System.Math.Pi * System.Math.Pow(intRadius, 2);
```

Radno okruženje .NET Framework obezbeđuje 124 različita prostora imena. U tabeli 1-4 navedeno je svega 40-ak najčešće korišćenih prostora imena. Spisak prostora imena radnog okruženja .NET Framework možete naći u okviru teme „Class Library” u onlajn pomoći za Visual Studio .NET.

Programski jezici

Kao što je slučaj sa radnim okruženjem .NET Framework u celini, i ASP.NET je sasvim nezavisan od programskog jezika. To znači da možete odabrati svaki jezik koji je implementirao kompajler koji odgovara CLR-u. Osim što razvija sopstvene programske jezike, Microsoft se udružio sa mnogim prodavcima jezika kako bi obezbedio .NET podršku za Perl, Pascal, Eiffel, Cobol, Python, Smalltalk i druge programske jezike.

U ovoj knjizi govori se o pravljenju veb aplikacija pomoću programskih jezika Visual Basic .NET i Visual C#. Ova dva jezika su funkcionalno ekvivalentna, što znači da svaki od njih nudi podjednake mogućnosti za pravljenje veb aplikacija. Razlike između ova dva jezika su sintaksičke i stilske prirode.

Većina današnjih programera opredeliće se za jezik kojim najbolje vladaju. Današnji Visual Basic programeri lagodnije će se osećati da rade na razvoju veb aplikacija u jeziku Visual Basic .NET. S druge strane, C ili C++ programeri biće na „svom terenu” ako se služe jezikom Visual C#.

Ako ste početnik u programiranju, ili ukoliko ste voljni da proširite svoje veštine programiranja savladavanjem novih jezika, istovremeno učenje oba jezika, Visual Basic .NET-a i Visual C#-a, je cilj koji ćete izvanredno primeniti u praksi. Ovo posebno važi kada pravite veb aplikacije, pošto se većina poslova obavlja preko klasa radnog okruženja .NET Framework, što znači da Visual Basic .NET kôd i Visual C# kôd često izgledaju skoro identično.

U tabeli 1-5 rezimirane su neke značajne razlike između jezika Visual Basic .NET i Visual C#. Korisno je da zapamtite ove informacije ako se prvi put opredeljujete za jedan od ovih jezika ili ukoliko planirate da se preorijentišete sa jednog jezika na drugi.

Tabela 1-5 Razlike između Visual Basic .NET-a i Visual C#-a

Odlika	Visual Basic .NET	Visual C# .NET
Razlikovanje velikih i malih slova	Ne razlikuje velika i mala slova: <pre>response.write ("Yo") ' OK</pre>	Razlikuje velika i mala slova: <pre>response.write ("Yo"); // Error Response.Write("Yo"); // OK</pre>
Funkcionalni blokovi	Koristi početne i završne naredbe za deklarisanje funkcionalnih blokova koda: <pre>Sub Show(strX as Response.Write) End Sub</pre>	Koristi vitičaste zagrade za deklarisanje funkcionalnih blokova koda: <pre>void Show (string strX) { Response.Write(strX); }</pre>
Konverzija tipova	Podrazumeva se da su implicitne konverzije tipova dozvoljene: <pre>Dim X As Integer X = 3.14 ' </pre> <p>Konverzije možete da ograničite uključivanjem naredbe <i>Option Strict On</i> na početak modula.</p>	Implicitne konverzije tipova ograničene su na operacije koje garantuju da se informacije neće izgubiti, kao što je konvertovanje iz <i>int</i> u <i>float</i> : <pre>int X = 0; float Y = X; // OK</pre> <p>Druge konverzije tipova vrše se eksplicitno putem konvertovanja: <pre>Y = 3.14 F; X = (int)Y; // Konvertovanje, OK.</pre> <p>Ili, primenom metoda za konverziju tipova: <pre>string Z; Z = Y.ToString();</pre> </p> </p>

Tabela 1-5 Razlike između Visual Basic .NET-a i Visual C#-a (nastavak)

Odlika	Visual Basic .NET	Visual C# .NET
Komentari	Komentari obavezno počinju apostroфом ('): <pre>' Ovo je komentar.</pre>	Komentari mogu biti trojaki: u bloku (<i>/* */</i>), umetnuti (<i>//</i>) i dokumentacioni (<i>///</i>): <pre>/* Komentari u bloku mogu da se protežu na više redova ili da služe za razjašnjenje koda. */ // Umetnuti komentari nalaze se // desno od koda. /// <rezime>Opis /// klase.</rezime></pre> Detaljnije informacije o dokumentacionim komentarima pronaći ćete pod temom „XML Documentation” u onlajn pomoći za Visual Studio.
Nizovi	Elementi niza navode se između malih zagrada: <pre>arrFruit(1) = "Apple"</pre>	Elementi niza navode se između uglastih zagrada: <pre>arrFruit[1] = "Apple";</pre>
Metodi	Možete izostaviti male zagrade iza naziva metoda ukoliko su argumenti izostavljeni: <pre>strX = objX.ToString</pre>	Morate da stavite male zagrade iza svih metoda: <pre>strX = objX.ToString();</pre>
Završavanje naredbe	Naredbe se završavaju povratkom na početak novog reda: <pre>Response.Write ("Hello")</pre>	Naredbe se prekidaju znakom tačka-zapeta (;): <pre>Response.Write ("Hello");</pre>
Nastavak naredbe	Naredbe se nastavljaju korišćenjem podvlake (_): <pre>intX = System.Math.Pi * _ intRadius</pre>	Naredbe se nastavljaju sve do znaka tačka-zapeta (;) i, po potrebi, mogu da zauzmu više redova: <pre>intX = System.Math.Pi * intRadius;</pre>
Operator za nadovezivanje stringova	Upotrebite znak ampersand (&) ili znak plus (+) da biste spojili stringove: <pre>strFruit = "Apples" & _ " Oranges"</pre>	Upotrebite znak plus (+) da biste spojili stringove: <pre>strFruit = "Apples" + " Oranges";</pre>
Operatori poredjenja	Koristite znakove =, >, <, >=, <=, < > da biste uporedili vrednosti: <pre>If intX >= 5 Then</pre>	Koristite znakove ==, >, <, >=, <=, != da biste uporedili vrednosti: <pre>if (intX >= 5)</pre>

Tabela 1-5 Razlike između Visual Basic .NET-a i Visual C#-a (nastavak)

Odlika	Visual Basic .NET	Visual C# .NET
Negacija	Upotrebite ključnu reč <i>Not</i> da biste izrazili logičku negaciju: <code>If Not IsPostBack Then</code>	Upotrebite operator <i>!</i> da biste izrazili logičku negaciju: <code>If (!IsPostBack)</code>
Poređenje objekata	Koristite ključnu reč <i>Is</i> da biste uporedili promenljive objekta: <code>If objX Is objY Then</code>	Koristite znakove <code>==</code> da biste uporedili promenljive objekta: <code>if (objX ==objY)</code>
Postojanje objekta	Upotrebite ključnu reč <i>Nothing</i> ili funkciju <i>IsNothing</i> da biste proverili da li objekat postoji: <code>If IsNothing(objX) Then</code>	Upotrebite ključnu reč <i>null</i> da biste proverili da li objekat postoji: <code>if (objX == null)</code>

Pored razlika prikazanih u tabeli 1-5, ova dva jezika se značajno razlikuju i po ključnim rečima. Primeri sa kodovima koji se navode u celom udžbeniku najbolje govore o tim razlikama. Pod temom „Language Equivalents” u pomoći za Visual Studio .NET pronaći ćete sva poređenja između jezika Visual Basic .NET, Visual C# i drugih Microsoftovih jezika.

Lekcija 3: Korišćenje Visual Studio .NET-a

Programsko okruženje Visual Studio .NET uvodi nove tipove prozora, nov način upravljanja tim prozorima i novu integraciju sa sadržajima Interneta. Ova lekcija približiće vam te nove funkcije, a takođe dati opšti pregled nekih starijih mogućnosti za otklanjanje programskih grešaka i pružanje pomoći korisnicima Visual Studio .NET-a sa gledišta programiranja veb aplikacija.

Ukoliko ste se već bavili programiranjem služeći se starijim verzijama Visual Studija i radije biste preskočili ovu lekciju, ne zaboravite da nikakve izmene u aplikaciji ne možete više da obavite za vreme otklanjanja programskih grešaka bez ponovnog pokretanja aplikacije. Ta karakteristična funkcija Visual Studio .NET-a, poznata pod nazivom „uređivanje i nastavljnje” (*edit-and-continue*), nije više dostupna u jezicima Visual Basic .NET i Visual C#.

Po završetku ove lekcije, moći ćete da:

- koristite početnu stranicu da biste otvorili nove ili postojeće projekte, dobili aktuelne podatke o proizvodu i postavili prioritete okruženja;
- navedete dva tipa prozora Visual Studio .NET-a i koristite funkciju Auto Hide (*automatsko skrivanje*) kako biste maksimalno iskoristili prostor na ekranu za uređivanje dokumenata;
- isecate i prenosite stavke pomoću alatke Clipboard Ring u kutiji sa alatkama;
- uređujete veb obrasce i HTML stranice vizuelno ili u HTML-u;
- pišete kôd korišćenjem automatskih funkcija editora koda i menjate parametre Visual Studio .NET-a kako biste te funkcije uključivali i isključivali;
- izgrađujete i izvršavate aplikacije, kao i da otklanjate programske greške iz njih, koristeći Visual Studio .NET;
- dobijete pomoć i postavite filtre za pomoć za programski jezik kojem dajete prioritet.

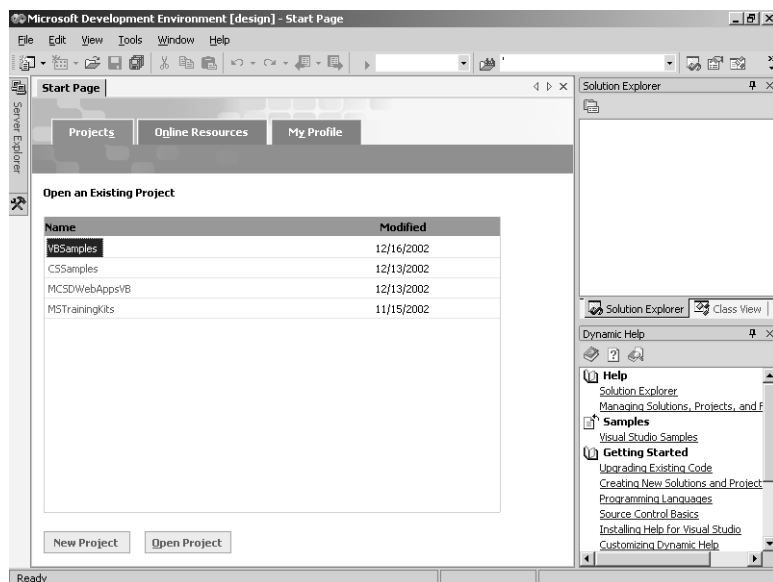
Vreme potrebno za lekciju: 30 minuta

Početna stranica

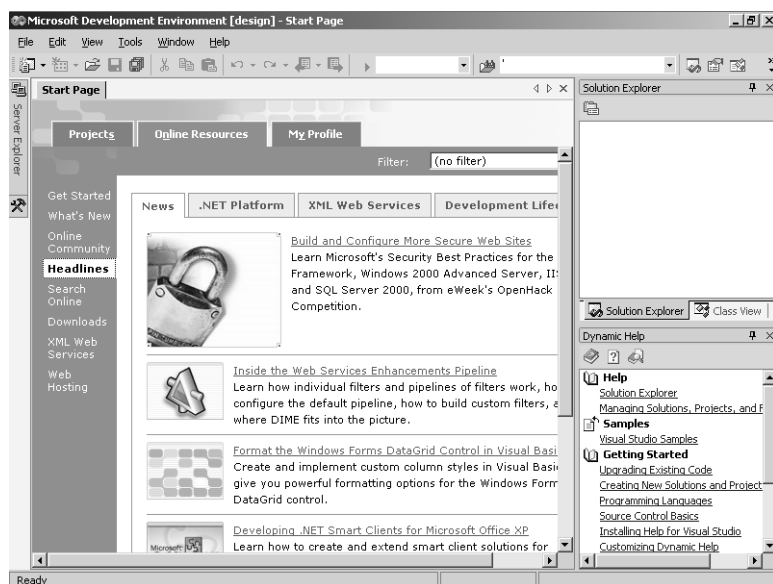
Prilikom pokretanja Visual Studio .NET-a, prvo što ćete uočiti jeste početna stranica, kao što je prikazano na slici 1-8. Početna stranica sadrži razne okvire zahvaljujući kojima ćete informacije lakše pronaći i pomoću kojih su pojednostavljeni neki uobičajeni poslovi, kao što je otvaranje neke nedavno korišćene datoteke.

Okvir Projects, ilustrovan na slici 1-8, prikazuje četiri poslednja sačuvana projekta u obliku hiperveza. Da biste otvorili neki od njih, izaberite naziv projekta. Da biste napravili nov projekat, ili otvorili neki postojeći koji nije naveden na spisku nedavnih projekata, izaberite odgovarajuće dugme na kartici Projects.

Na početnoj stranici levo nalazi se lista ostalih tema koje sadrže najnovija obaveštenja o Visual Studio .NET-u, drugim Microsoftovim proizvodima, programiranju, pružanju usluga smeštanja veb lokacije i drugim pitanjima. Ukoliko odaberete bilo koju od ovih tema, početna stranica će je prikazati, kao što se vidi na slici 1-9.



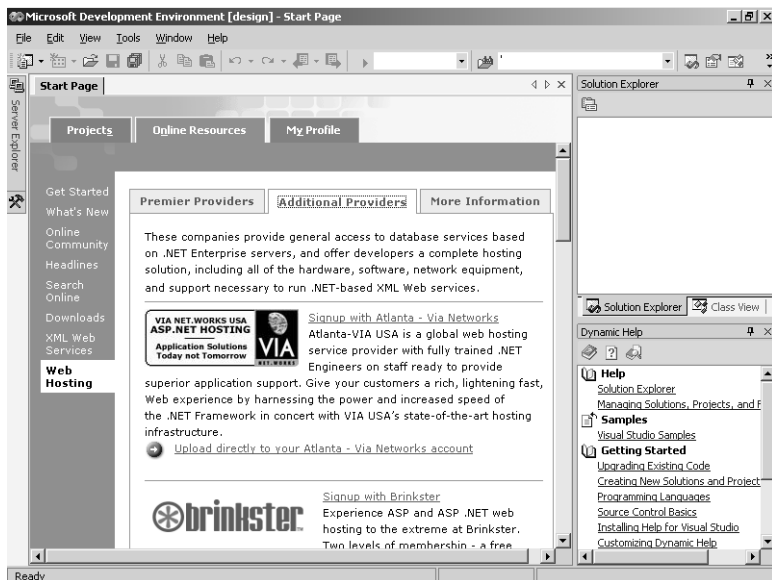
Slika 1-8 Početna stranica Visual Studio .NET-a



Slika 1-9 Naslovi aktuelnih tema Visual Studio .NET-a

Informacije koje se prikazuju na početnoj stranici su dinamičke i većina njih potiče sa Interneta. Upravo zahvaljujući tome, informacije su aktuelne. Ne bi bilo loše da povremeno proveravate okvire Headlines i Downloads kako biste dobili najnovije vesti.

Za razvijaoce veb aplikacija posebno je zanimljiv okvir Web Hosting, koji je prikazan na slici 1-10. On služi za povezivanje sa veb lokacijama na koje će se smestiti vaše ASP.NET veb aplikacije na Internetu.

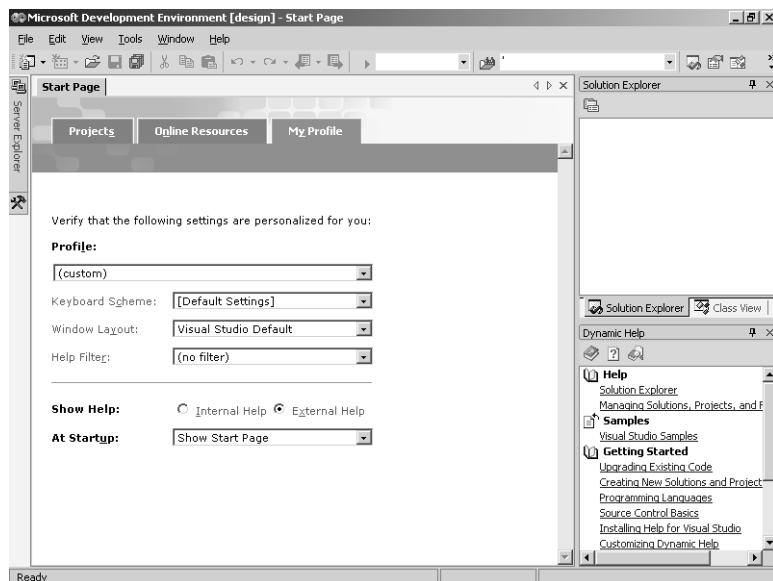


Slika 1-10 Pružanje usluga smeštanja ASP.NET aplikacija

Neke veb lokacije, poput one koju je postavila firma Brinkster, nude ograničene usluge besplatnog smeštanja aplikacija. Ali, ovi servisi su izuzetno korisni prilikom učenja ASP.NET-a, pošto vam omogućavaju da svoj rad delite sa ostalim ljudima u svetu, i to bez ikakvog napora i troškova koji se tiču postavljanja vašeg veb servera.

Na okviru My Profile na početnoj stranici možete da podešavate svoje prioritete za Visual Studio .NET, kao što je prikazano na slici 1-11.

Zahvaljujući ovim opcijama, možete da promenite podrazumevani raspored elemenata na prozoru za Visual Studio .NET, postavite onaj programski jezik kojim se najčešće služite i naznačite da li će komanda Help biti prikazana u okvirima za projektovanje Visual Studio .NET-a ili u jednom prozoru kao zasebna aplikacija. Prozor Help će možda biti prenatrpan ukoliko se prikaže u okviru Visual Studio .NET-a, pa stoga, ukoliko nemate monitor od 19 inča, dobro bi bilo da se opredelite za opciju External Help.

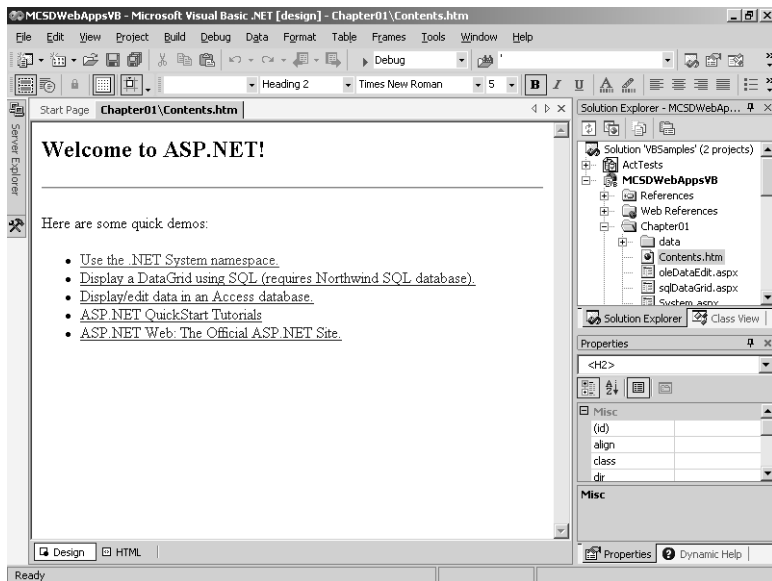


Slika 1-11 Podesite svoje prioritete u okviru My Profile

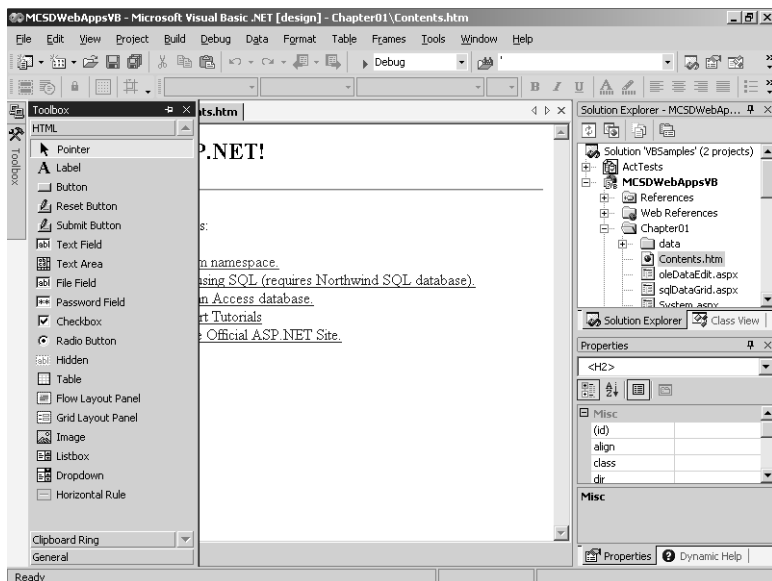
Prozori Visual Studio .NET-a

Visual Studio .NET ima dva tipa prozora: prozore za dokumente i prozore za alate. Prvi koji su navedeni prikazuju sadržinu vaše aplikacije: obrasce, veb stranice i kôd – sve to pojaviće se u prozorima za dokumente. Možete da imate više istovremeno otvorenih prozora za dokumente, a prelazićete sa jednog na drugi biranjem njihovih kartica pri vrhu ekrana, kao što se vidi na slici 1-12.

Prozori za alate prikazuju komponente koje koristite za pravljenje svoje aplikacije. U te komponente spadaju: kontrole, veze sa bazama podataka, klase i svojstva koja upotrebljavate u projektu. Prozori za alate otvaraju se levo i desno od prozora za dokument, a njihovo otvaranje i zatvaranje možete podešavati biranjem njihovih kartica, kao što je ilustrovano na slici 1-13.



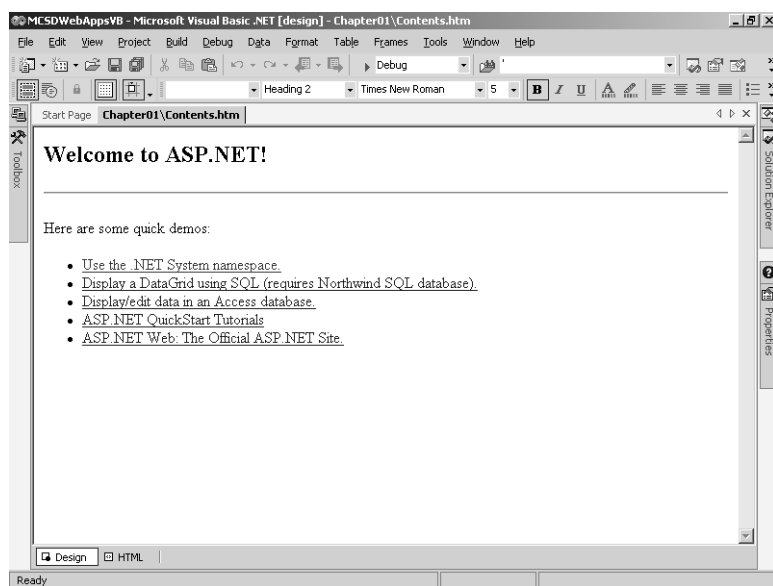
Slika 1-12 Prozor za dokument



Slika 1-13 Prozor za alatke

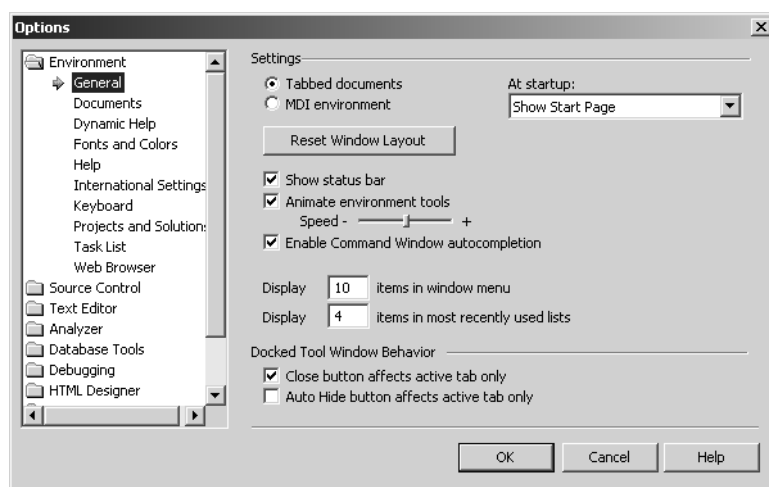
Da bi se prozor za alatke čiju ste karticu upravo izabrali zadržao na ekranu, aktivirajte dugme Auto Hide koje se nalazi u gornjem desnom uglu prozora za alatke. Dugme Auto Hide ima sličicu „pribadače”. Ponovo pritisnite „pribadaču” kako bi se prozor za alatke vratio u stanje prikaza sa karticama. Možete koristiti prikaz sa karticama da biste sakrili prozore za alatke s obe strane prozora za dokument i tako

dobili više prostora za uređivanje sadržine svoje aplikacije, kao što se vidi na slici 1-14.



Slika 1-14 Prozori za alatke sa karticama oko prozora za dokument

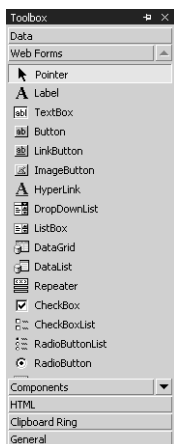
Prikaz sa karticama je podrazumevani prikaz prozora za alatke i prozora za dokumente u Visual Studio .NET-u. Ovu funkciju možete da isključite i preorijentirate se na tradicionalniji raspored prozora ukoliko izaberete najpre Options iz menija Tools, a potom svoje prioritete iz okvira za dijalog, kao što je prikazano na slici 1-15.



Slika 1-15 Okvir za dijalog Options

Kutija sa alatkama

Kutija sa alatkama (*Toolbox*) Visual Studio .NET-a prikazuje kontrole i komponente koje možete da dodajete u prozor za dokument. Sadržina kutije sa alatkama menja se zavisno od tipa dokumenta kojeg u datom trenutku uređujete. Primera radi, kada uređujete veb obrazac, kutija sa alatkama prikazuje serverske kontrole, HTML kontrole, kontrole za podatke i druge komponente koje možete ubacivati na veb obrazac, kao što se vidi na slici 1.16.

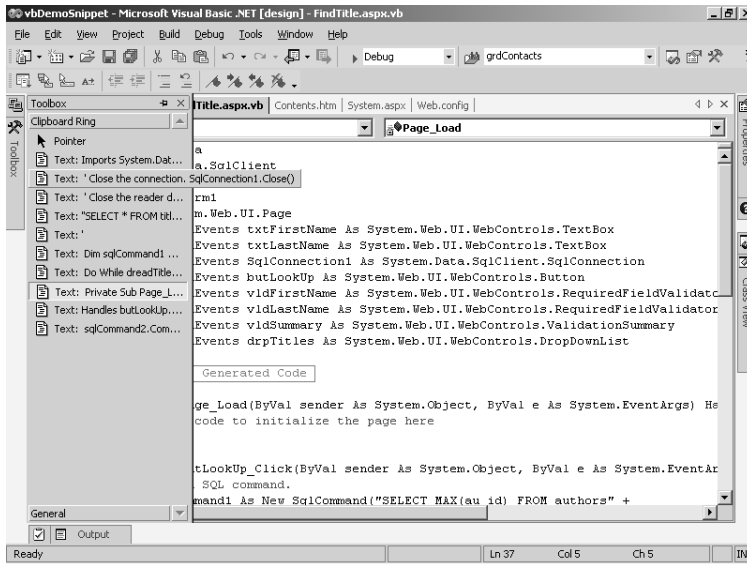


Slika 1-16 Prozor kutije sa alatkama

Komponente u kutiji sa alatkama kategorizovane su na način koji je prikazan na slici 1-16. Kada odaberete jednu od ovih kategorija, kutija sa alatkama prikazuje stavke u toj kategoriji. Možete razgledati sve stavke u kutiji sa alatkama pomoću strelica nagore ili nadole koje se nalaze na vrhu, odnosno pri dnu spiska komponenti.

Ukoliko je tekući dokument kôd, kutija sa alatkama sadrži samo karticu Clipboard Ring, koja je prikazana na slici 1-17. Clipboard Ring vodi evidenciju o poslednjih 20 stavki koje ste iskopirali (CTRL+C) ili isekli (CTRL+X) tako da ih možete preneti nazad u dokument.

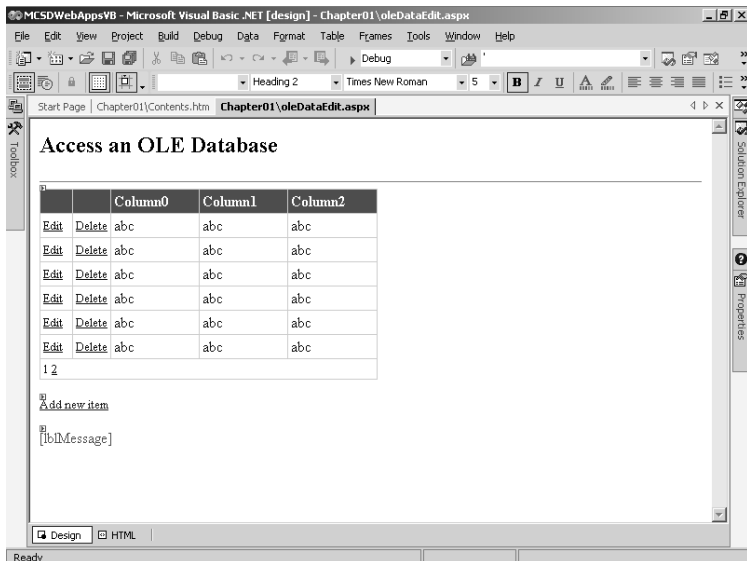
Da biste preneli neku stavku iz Clipboard Ringa, izaberite je i prevucite do mesta na koje hoćete da je umetnete. Prilikom povlačenja pokazivača miša preko stavke u Clipboard Ringu, Visual Studio otvara tu stavku kako biste mogli da vidite što više teksta sadržanog u njoj.



Slika 1-17 Clipboard Ring u prozoru kutije sa alatkama

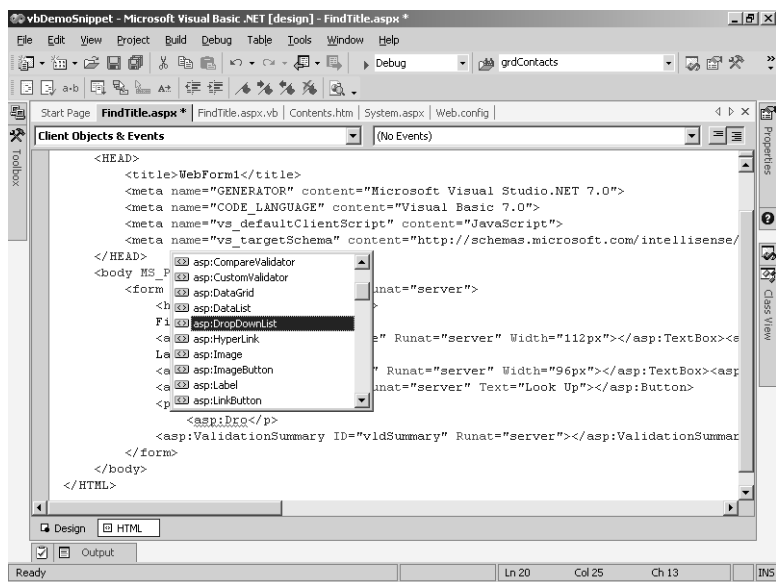
Uređivanje veb dokumenata

Veb obrasce i HTML dokumente možete uređivati vizuelno, koristeći iste tehnike povlačenja i puštanja kao prilikom uređivanja Windowsovih obrazaca, ili na isti način kao što uređujete tekstualne datoteke. Da biste prešli sa jednog režima uređivanja na drugi, izaberite kartice Design ili HTML koje se nalaze pri dnu prozora Document, kao što je prikazano na slici 1-18.



Slika 1-18 Web dokument u režimu Design

Neki poslovi ne mogu se uopšte uraditi vizuelno, pa ćete često biti primorani da veb dokument uređujete kao tekst. Osim toga, rad u režimu HTML može da bude prikladniji od korišćenja vizuelnih alatki ukoliko već dobro poznajete HTML. Tehnologija *IntelliSense* u Visual Studio .NET-u nudi vam pomoć u vezi sa popunjavanjem HTML elemenata, kao što je prikazano na slici 1-19.

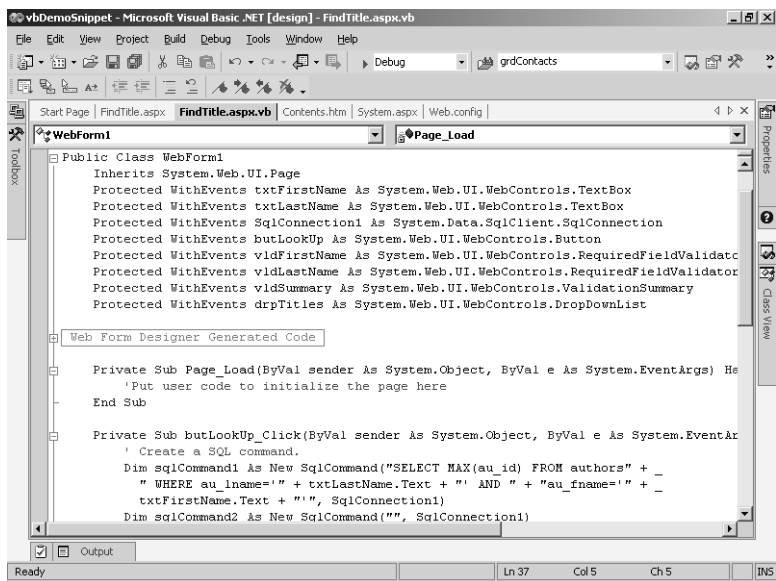


Slika 1-19 IntelliSense za HTML elemente u Visual Studio .NET-u

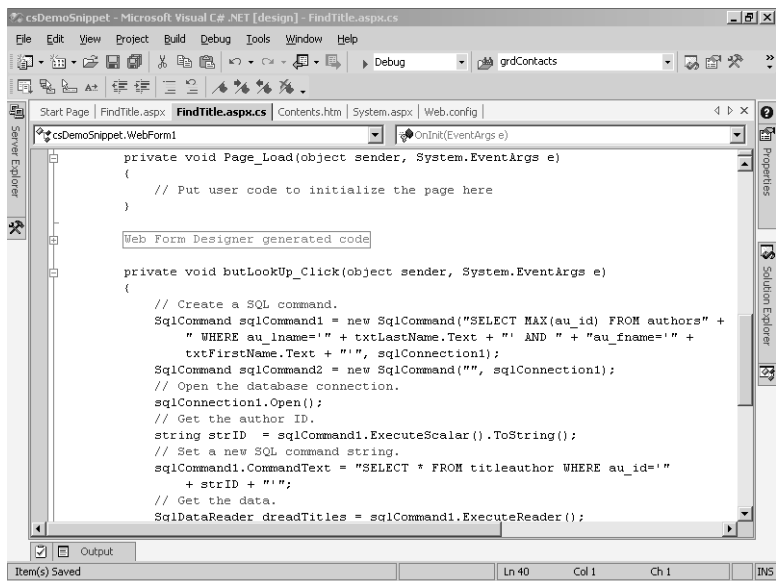
Možete da se ponovo vratite na režim rada Design kako biste pregledali promene koje ste uneli radeći u režimu HTML, biranjem kartice Design pri dnu prozora za dokument.

Uređivanje datoteka sa kodom u pozadini

Veb obrasci imaju datoteke sa kodom u pozadini koje su sa njima povezane. Ove datoteke formiraju se automatski upravo u trenutku kada pravite nov veb obrazac i poznate su pod nazivom *datoteka sa kodom u pozadini (code-behind files)*. Njihovo osnovno ime isto je kao naziv veb obrasca, samo što je iza imena dodata još oznaka tipa datoteke .vb ili .cs, kao što je ilustrovano na slikama 1-20 i 1-21.



Slika 1-20 Datoteka sa kodom u pozadini povezana sa veb obrascem (Visual Basic .NET)



Slika 1-21 Datoteka sa kodom u pozadini povezana sa veb obrascem (Visual C#)

Veb obrazac se povezuje sa svojom kodnom datotekom pomoću @ *Pagedirective* koja se nalazi u HTML-u veb obrasca i izgleda ovako:

Visual Basic .NET

```
<%@ Page Language="vb" AutoEventWireup="false"  
Codebehind="Form1.aspx.vb" Inherits="WebApplication1.Webform1"%>
```

Visual C#

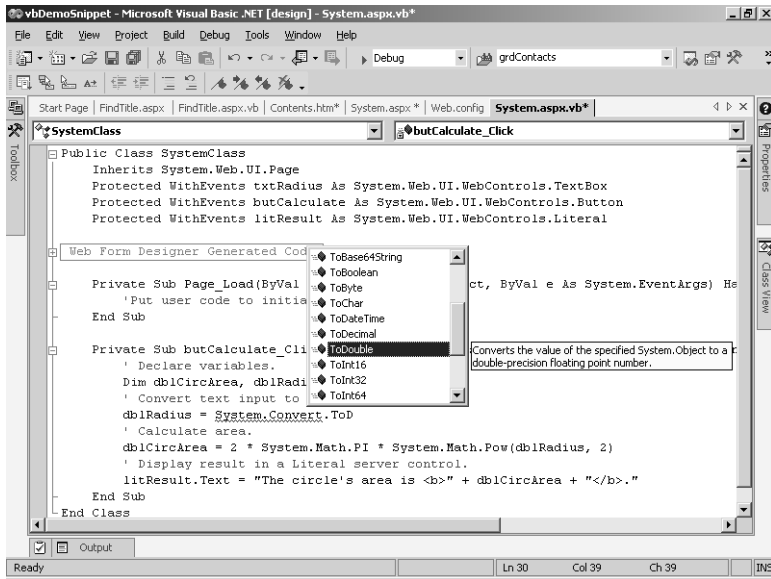
```
<%@ Page language="c#" Codebehind="WebForm1.aspx.cs"  
AutoEventWireup="false" Inherits="WebApplication1.Webform1"%>
```

Visual Studio automatski održava informacije o datoteci u ovoj direktivi Page, pa, ukoliko veb obrazac snimate pod drugačijim nazivom datoteke, atribut CodeBehind se automatski ažurira. Međutim, Visual Studio ne održava automatski informacije u atributu Inherits direktive Page. Ako promenite koreni prostor imena projekta ili naziva klase veb obrasca, moraćete ručno da ažurirate informacije u direktivi Page veb obrasca.

Visual Studio .NET generiše definiciju klase, proceduru za inicijalizaciju i proceduru događaja *Page_Load* za svaku datoteku sa kodom u pozadini. Ne smete da menjate kôd u oblastima sa naznakom *Web Form Designer Generated Code*, pošto bi ga Visual Studio .NET kasnije možda modifikovao, pa bi vaše promene bile poništene.

Možete da sakrijete generisani kôd biranjem znaka minus (-) koji se nalazi levo od direktive #Region. Biranjem ovog znaka minus, oblast se sažima u jedan red i znak se pretvara u plus (+), koji možete da izaberete kako biste ponovo otvorili oblast. Ovu istu funkciju možete da upotrebite za sažimanje ili otvaranje ostalih blokova koda, kao što su definicije klase i procedure.

Editor koda Visual Studio .NET-a takođe obezbeđuje, preko tehnologije IntelliSense, popunjavanje ključnih reči i članova klase koje koristite u kodu, kao što je prikazano na slici 1-22.

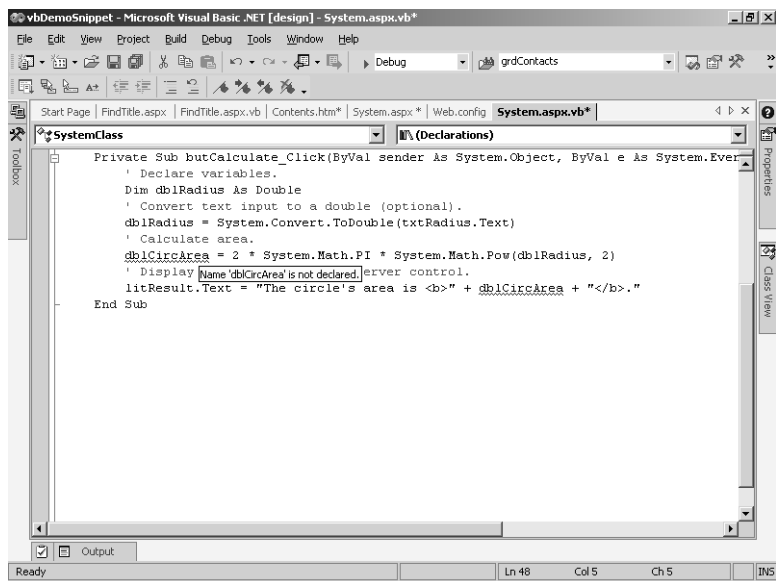


Slika 1-22 Funkcija automatskog popunjavanja

Ukoliko programirate u Visual Basicu, funkcija automatskog popunjavanja takođe će ispraviti veličinu prvog slova ključnih reči i naziva članova kada završite jedan red. Međutim, ako koristite Visual C#, Visual Studio .NET neće prepoznati ključnu reč ili naziv člana u slučaju da veličina njihovog prvog slova nije ispravna. Razlog leži u tome što Visual Basic .NET ne razlikuje veličinu slova, dok je Visual C# razlikuje.

Editor koda Visual Studio .NET-a ističe sintaksne greške i nedeklarisane promenljive čim završite jedan red. Ove greške su podvučene talasastom linijom, pa ukoliko pokazivačem miša pređete preko date greške, prikazaće se njen opis, kao što se vidi na slici 1-23.

Većinu automatskih funkcija editora koda možete uključivati i isključivati menjanjem parametara u okviru za dijalog Options, koji je prikazan na slici 1-15. Služeći se okvirom za dijalog Options možete da menjate još i automatsko uvlačenje pasusa, popunjavanje bloka koda i druge parametre svojstvene jeziku.



Slika 1-23 Otkrivanje greške u editoru koda

Uređivanje veb obrazaca koji se sastoje od jedne izvorne datoteke

ASP.NET takođe podržava veb obrasce koji se sastoje od jedne izvorne datoteke (*single-source Web forms*). Kao što samo ime nagoveštava, i kôd i HTML veb obrasca iz jednog izvora čuvaju se zajedno, u jednoj datoteci. Mnogi primeri ASP.NET koda i mnogi priručnici koji se na Web prosleđuju koriste samo jednu datoteku sa izvornim kodom, jer se one lakše distribuiraju i prikazuju. Kao primer navodimo sledeći veb obrazac koji se sastoji od jedne izvorne datoteke za izračunavanje površine kruga:

Visual Basic .NET

```

<%@ Page Language="VB" %>
<script runat="server">
    Private Sub butCalculate_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs)
        ' Deklarisati promenljive.
        Dim dblCircArea, dblRadius As Double
        ' Konvertovati upisani tekst u tip double (opciono).
        If txtRadius.Text <> "" Then _
            dblRadius = System.Convert.ToDouble(txtRadius.Text)
        ' Izračunati površinu.
        dblCircArea = System.Math.PI * System.Math.Pow(dblRadius, 2)
        ' Prikazati rezultat.
        ShowResult(dblCircArea)
    End Sub

```

```

Sub ShowResult(ByVal Result As Double)
    litResult.Text = "<h3>Results</h3>"
    litResult.Text += "<p>The circle's area is: <b>" + Result.ToString()
-
    + "</b>"
End Sub
</script>
<html>
<head>
    <title>Calculate Area</title>
</head>
<body>
    <form runat="server">
        <h2>Calculate Area
        </h2>
        <hr />
        Circle radius:
        <asp:TextBox id="txtRadius" Runat="server"></asp:TextBox>
        <asp:Button id="butCalculate" onclick="butCalculate_Click"
        Runat="server"
            Text="Calculate"></asp:Button>
        <p>
            <asp:Literal id="litResult" Runat="server"></asp:Literal>
        </p>
    </form>
</body>
</html>

```

Visual C#

```

<%@ Page Language="C#" %>
<script runat="server">
    private void butCalculate_Click(object sender, EventArgs e)
    {
        // Deklarisati promenljive.
        double dblCircArea, dblRadius;
        // Konvertovati upisani tekst u tip double (opciono).
        if (txtRadius.Text != "")
        {
            dblRadius = Convert.ToDouble(txtRadius.Text);
            // Izračunati površinu.
            dblCircArea = 2 * Math.PI * Math.Pow(dblRadius, 2);
            // Prikazati rezultat.
            ShowResult(dblCircArea);
        }
    }

    void ShowResult(double Result)
    {
        litResult.Text = "<h3>Results</h3>";
        litResult.Text += "<p>The circle's area is: <b>"
        + Result.ToString() +
            "</b>";
    }

```

```

</script>
<html>
<head>
<title>Calculate Area</title>
</head>
<body>
  <form runat="server">
    <h2>Calculate Area
    </h2>
    <hr />
    Circle radius:
    <asp:TextBox id="txtRadius" Runat="server"></asp:TextBox>
    <asp:Button id="butCalculate" onclick="butCalculate_Click"
    Runat="server"
      Text="Calculate"></asp:Button>
    <p>
      <asp:Literal id="litResult" Runat="server"></asp:Literal>
    </p>
  </form>
</body>
</html>

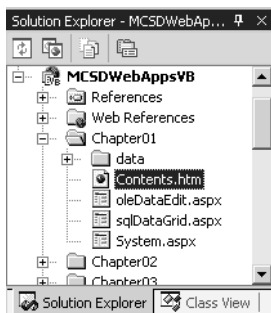
```

Visual Studio može da uređuje, pa čak i da izvršava ove veb obrasce koji se sastoje od jedne izvorne datoteke. Međutim, napredne funkcije, kao što je automatsko popunjavanje, nisu omogućene za kodove koji se upisuju između elemenata `<script>` i `</script>` na stranici. Upravo je to razlog zašto su primeri sa kodom u ovoj knjizi prikazani u datotekama sa kodom u pozadini.

Solution Explorer

Visual Studio .NET organizuje aplikacije u projekte i rešenja. **Projekat** je kolekcija datoteka od kojih će se na kraju napraviti samo jedna izvršna datoteka.

Rešenje (*solution*) je grupa projekata koji obrazuju jednu funkcionalnu jedinicu. Datoteke u rešenju pregledaćete pomoću Solution Explorera, čiji je prikaz dat na slici 1-24.



Slika 1-24 Solution Explorer

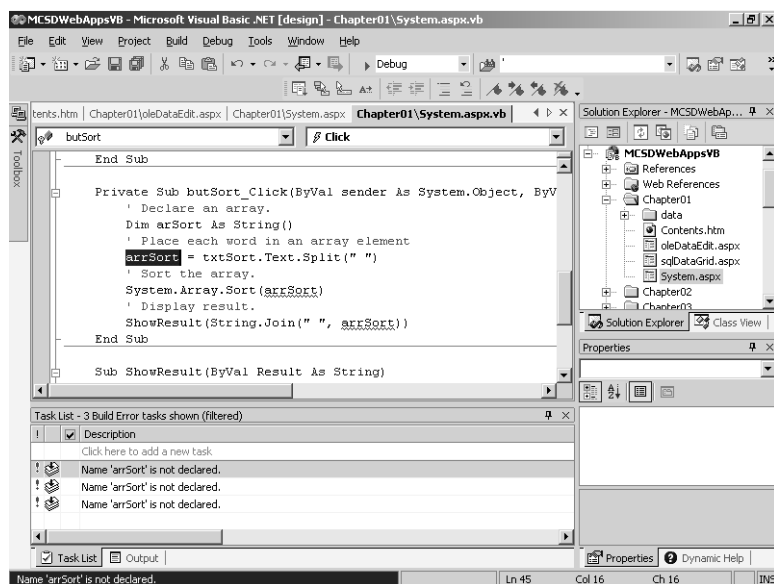
Projekat prikazan **polucrnim** slovima je početni projekat. **Početni projekat** (*start-up project*) je projekat koji se pokreće kada izaberete komandu Start u Visual Stu-

dio .NET-u. Kada razvijate više projekata koji čine deo jednog rešenja, početni projekat uglavnom poziva i druge projekte iz rešenja.

Informacije o rešenju čuvaju se u datoteci sa rešenjem (.sln), a podrazumeva se da je ona smeštena u vašoj omotnici My Documents. Rešenje možete da otvorite korišćenjem ove datoteke, a projekte otvarate direktno koristeći njihove datoteke sa projektom (.vproj ili .cproj), koje se nalaze u omotnicama za projekte. Ako otvorite datoteku sa projektom, Visual Studio .NET pravi novu datoteku sa rešenjem kada sačuvate projekat.

Pokretanje projekta

Projekat u okviru Visual Studio .NET-a možete pokrenuti biranjem komande Start na paleti alatki, biranjem opcije Start iz menija Debug, ili pritiskom na taster F5. Kada pokrenete projekat, Visual Studio .NET gradi projektne datoteke i prikazuje eventualne greške koje se pojavljuju u prozoru Task List, kao što je ilustrovano na slici 1.25.

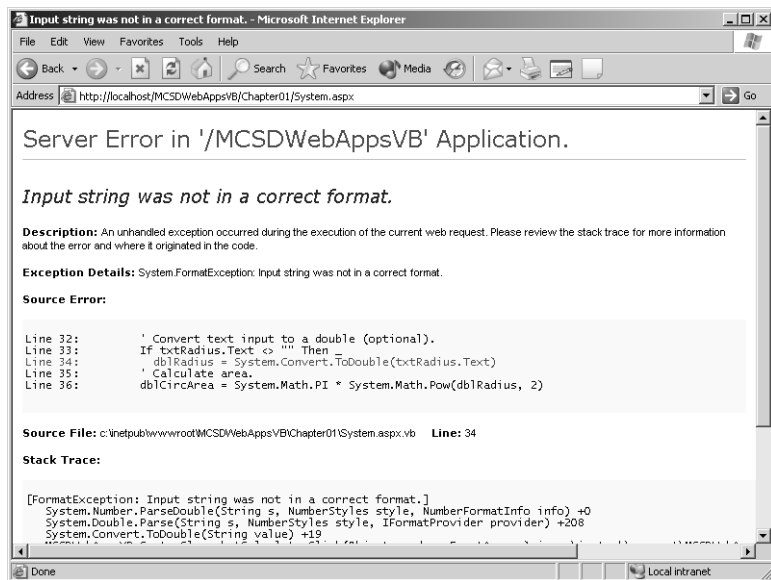


Slika 1-25 Pokretanje projekta sa greškama tokom gradnje

Postavite pokazivač na opis greške u prozoru Task List i pritisnite dva puta uzastopno taster miša da biste označili red sa greškom u prozoru za dokument. Tako ćete znati gde se nalazi greška koju treba da ispravite. Task List takođe prikazuje zadatke koje ste u svom kodu prokomentarisali, kao što su 'TODO, //TODO, 'UNDONE, //UNDONE, 'HACK ili //HACK. Simbole koje ćete upotrebiti za identifikovanje zadataka možete sami dodavati i menjati konfigurisanjem parametara Environment u okviru za dijalog Options Visual Studija.

Ukoliko tokom gradnje nema grešaka, Visual Studio .NET pokreće aplikaciju u režimu Debug (za pronalaženje i otklanjanje grešaka), a u slučaju veb aplikacije,

pokreće Internet Explorer i prikazuje početnu stranicu aplikacije. Ako nastane greška za vreme izvršavanja aplikacije u režimu Debug, Visual Studio .NET prikazuje grešku u pretraživaču, kao što je ilustrovano na slici 1-26.



Slika 1-26 Projekat veb aplikacije koji sadži greške u vreme izvršavanja

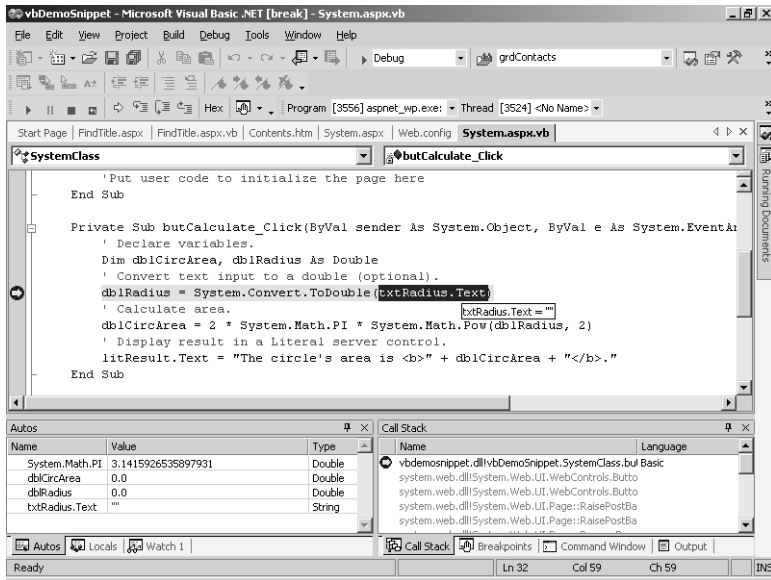
U ovom trenutku možete da birate između dve mogućnosti:

- Ukoliko vam je poznato šta je grešku prouzrokovalo, možete da zaustavite aplikaciju zatvaranjem prozora pretraživača i vratite se u Visual Studio .NET kako biste potom prikazanu grešku ispravili.
- Ako niste sigurni šta je uzrok greške, možete izabrati komandu Back u pretraživaču, prebaciti se u Visual Studio .NET kako biste postavili prelomnu tačku na mesto u kodu pre pojave greške i nakon toga ponovo se vratiti na pretraživač nastojeći da još jednom pokušate da uradite posao. Visual Studio .NET će zaustaviti aplikaciju na prelomnoj tački koju ste postavili, pa kôd možete natenane iščitavati dok ne pronađete izvor greške.

Čim grešku pronađete, morate zaustaviti aplikaciju da biste grešku ispravili. U starijim verzijama Visual Studio .NET-a, greške ste mogli da ispravljate u režimu Debug, nastavljajući izvršavanje aplikacije.

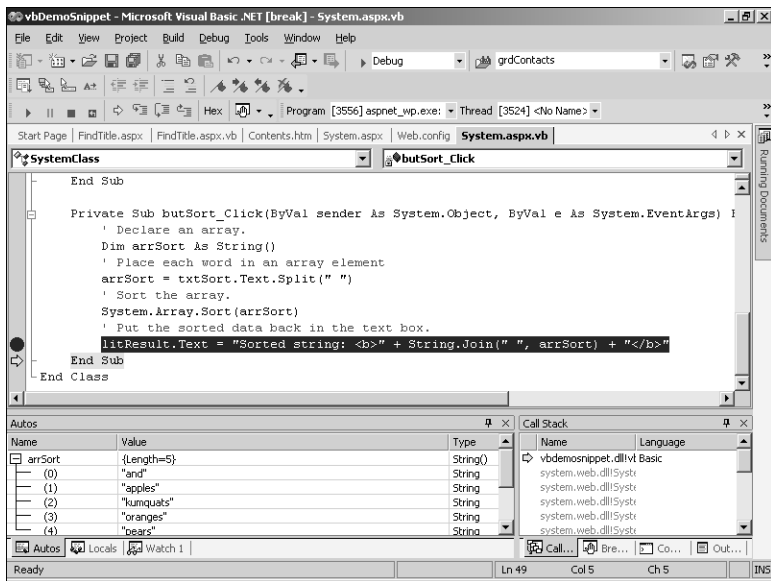
Postavljanje prelomnih tačaka i provera promjenljivih

Projekat možete da zaustavite u nekom konkretnom redu koda ako u njega postavite prelomnu tačku. Tokom izvršavanja projekta, Visual Studio .NET će zaustaviti projekat i u editoru koda prikazati red sa prelomnom tačkom pre nego što se taj red izvrši, kao što je ilustrovano na slici 1-27.



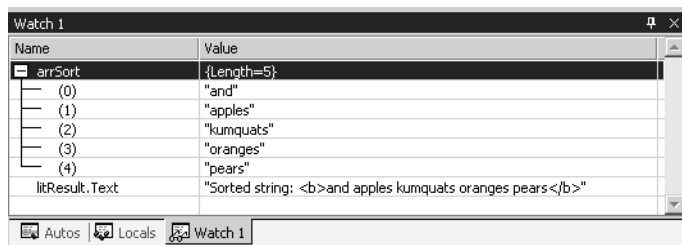
Slika 1-27 Projekat zaustavljen u prelomnoj tački

Prelomnu tačku postavljate pritiskom pomoću tastera miša na sivu marginu sa leve strane reda u koji želite da umetnete prekid, ili biranjem datog reda, nakon čega pritisnete taster F9. Na slici 1-28, vidi se prelomna tačka koju smo postavili.



Slika 1-28 Postavljanje prelomne tačke

Kada Visual Studio .NET jednom zastane na prelomnoj tački, možete pregledati vrednost aktivnih promenljivih pomeranjem pokazivača miša preko promenljive. Ukoliko je promenljiva složenog tipa, kao što je objekat ili niz, podatke sadržane u njoj možete pregledati ako promenljivu dodate u prozor Watch, kao što je prikazano na slici 1-29.



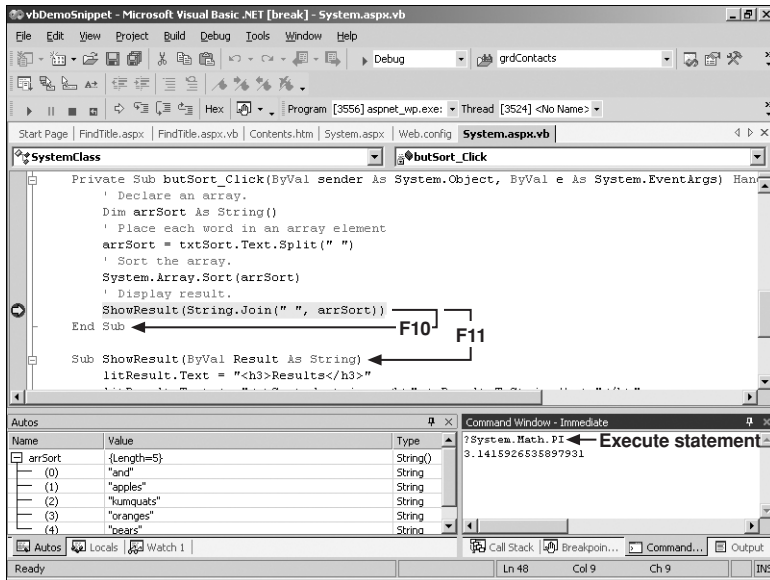
Slika 1-29 Prozor Watch

Da biste neku stavku stavili u prozor Watch, izaberite je i prevucite na prozor Watch. Pritisnite pomoću miša znak plus (+) koji stoji ispred stavke u prozoru Watch kako biste pregledali podstavke, kao što su elementi niza ili svojstva objekta.

Izvršavanje naredbi

Nakon zaustavljanja na prelomnoj tački, možete da nastavite izvršavanje aplikacije ako tasterom miša izaberete komandu Continue na paleti alatki, ili pritisnete taster F5. Umesto toga, možete da izvršavate jedan po jedan red pritiskom na taster F10 ili F11.

Tasterom F10 izvršava se pozivanje svake procedure kao pojedinačne naredbe. Drugim rečima, komandom zadatom preko tastera F10 procedura se **prelazi** (*steps over*) tako što se izvršava i zaustavlja na sledećem redu u aktivnoj proceduri. Taster F11 funkcioniše tako što izvršava pozive procedure **upadanjem u** (*stepping in*) proceduru i zaustavlja se na prvom redu u pozvanoj proceduri. Da biste izvršili samo jedan red koda izvan konteksta projekta, upišite kôd u prozor Command. Na slici 1-30 prikazane su ove različite tehnike.



Slika 1-30 Tehnike izvršavanja naredbi

Rezultati naredbi koje ste upisali u prozor Command usmeravaju se na sledeći red istog prozora. Na primer, naredba `?System.Math.PI` prikazuje rezultat 3.1415926535897931 u sledećem redu.

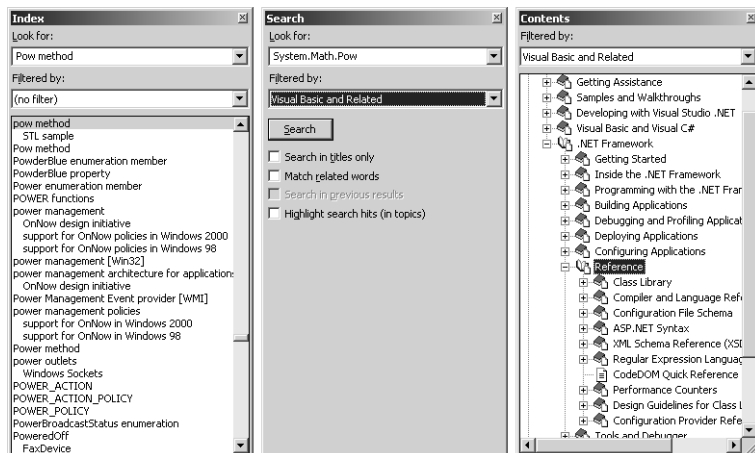
Traženje pomoći

Visual Studio .NET sadrži objedinjenu kolekciju pomoćnih datoteka u okviru sistema Help koje se bave pitanjima programskog okruženja, jezika, radnog okruženja .NET Framework, tehničke podrške i članaka iz mreže za razvijaoce. Pomoć se prikazuje bilo unutar prozora za dokument ili u zasebnom prozoru izvan Visual Studio .NET-a, zavisno od toga koje prioritete ste postavili na početnoj stranici ili u okviru za dijalog Options.

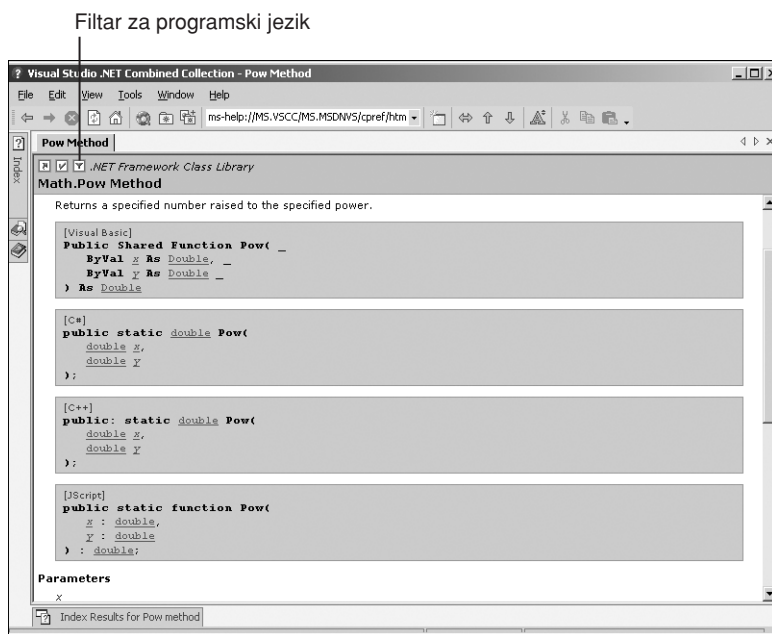
Iz sistema Help teme možete da pronalazite na tri načina: preko prozora Contents, prozora Index i prozora Search. Svi oni ponašaju se kao prozori za alate u Visual Studio .NET-u – mogu da se „prikluče”, a potom da se sakriju ili prikažu pomoću kartica, kao što je prikazano na slici 1-31.

Svaki prozor za pretraživanje nudi padajuću listu Filter iz koje možete da birate konkretan programski jezik ili temu koja vam treba. Ova funkcija je naročito korisna u prozorima Search i Index, budući da je objedinjena kolekcija Help velika. Filtri sa natpisom *Visual Basic And Related* i *Visual C# And Related* sadrže skoro sve teme koje obrađujemo u ovom udžbeniku.

Teme koje sadrže primere sintakse ili koda imaju ikonu jezičkog filtra na vrhu svake stranice sa sličicom levka. Izaberite ikonu filtra da biste promenili programski jezik koji je prikazan u datoj temi, ili da biste pogledali sve primere koji se tiču jezika, kao što je ilustrovano na slici 1-32.



Slika 1-31 Prozori za traženje pomoći



Slika 1-32 Postavljanje filtra za programski jezik

Pored sistema Help sadržanog u Visual Studio .NET-u, Microsoft pruža usluge davanja pomoći na veb lokaciji Got-DotNet, na adresi www.gotdotnet.com. Na njoj možete pronaći priručnike o korišćenju ASP.NET-a i veze sa velikim brojem veb lokacija koje se bave sličnim pitanjima.

Rezime

- Postoje četiri tipa Internet aplikacija: veb aplikacije, veb servisi, aplikacije koje koriste Internet i aplikacije za ravnopravne računare.
- Veb aplikacije izvršavaju se na serveru, obrađuju korisničke zahteve za stranice i sastavljaju te stranice korišćenjem izvršnog koda i statičkih resursa na serveru.
- Veb aplikacije mogu da obezbede dinamički sadržaj na osnovu dinamičkih serverskih resursa, kao što je baza podataka, i na osnovu korisničkih unosa, kao što je izrada tabele o otplaćivanju hipoteke na osnovu podataka koje je korisnik dao o visini zajma.
- ASP.NET je platforma za pravljenje veb aplikacija koje se izvršavaju na Windowsovim serverima koristeći IIS i radno okruženje .NET Framework.
- Veb aplikacije se sastoje od sadržaja, izvršne datoteke i konfiguracionih datoteka.
- Sadržaj veb aplikacije prikazuje se preko veb obrazaca. Veb obrasci koriste HTML komponente, kao što su konvencionalne HTML stranice. Međutim, slično Windowsovim obrascima, i oni mogu da reaguju na korisničke događaje, kao što je pritisak tasterom miša.
- Izvršna datoteka veb aplikacije smeštena je u datoteci tipa .dll koja se naziva sklopom. Sklopovi su kompajlirani samo delimično, a konačno kompajliranje obavlja CLR uoči samog pokretanja aplikacije.
- Radno okruženje .NET Framework sastoji se od CLR-a i biblioteke klasa .NET. Biblioteka klasa .NET programerima stavlja na raspolaganje poslove koje CLR obavlja u vremenu izvršavanja.
- Klase .NET grupisane su prema programskom zadatku u prostore imena. Zahvaljujući ovakvom načinu grupisanja, lako ćete pronaći klasu, metod, ili svojstvo koji su vam potrebni za obavljanje zadatka.
- Koristite početnu stranicu Visual Studio .NET-a da biste pročitali aktuelna obaveštenja o proizvodu, da biste otvorili nov projekat ili već postojeće projekte, da biste postavili korisničke prioritete okruženja i prijavili se za usluge smeštanja aplikacije na Webu.
- Uredite veb obrasce i HTML stranice vizuelno, koristeći režim uređivanja Design u prozoru za dokument. Nakon toga ih uredite kao tekst, služeći se režimom HTML u prozoru za dokument.
- Postavite jezički filter u sistemu Help kako biste pregledali primere sa kodom u jednom programskom jeziku ili u više njih.
- Promenite funkcije i mogućnosti okruženja Visual Studio .NET koristeći okvir za dijalog Options.

Laboratorija: Početak rada sa Visual Studio .NET-om

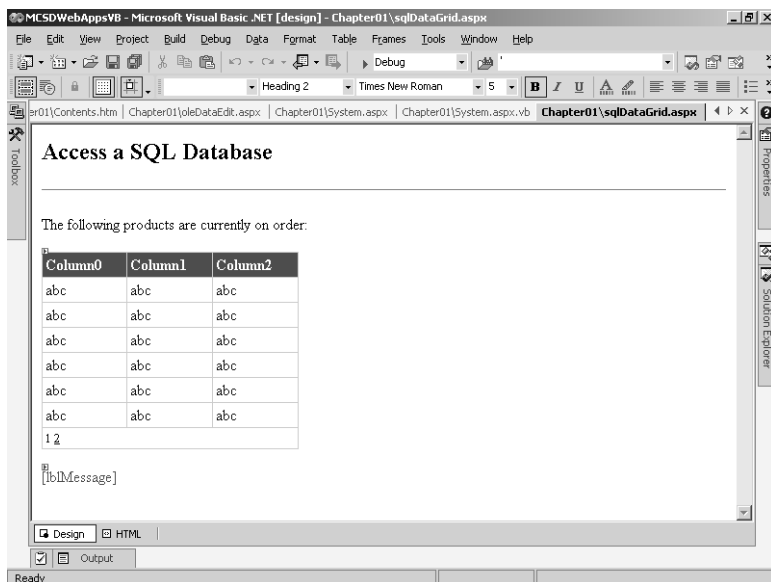
U ovoj laboratoriji upoznaćete se sa programskim okruženjem Visual Studio .NET, prijaviti se za smeštanje aplikacije na Webu i razmotriti priručnike za brzo učenje ASP.NET-a. Ova vežbanja čine temelj za konkretne veštine programiranja o kojima ćete više naučiti u narednim poglavljima.

Vreme potrebno za lekciju: 30 minuta



Vežba 1: Prilagođavanje okruženja Visual Studio .NET

U ovoj vežbi prilagodit ćete raspored elemenata u prozoru Visual Studio .NET-a kako biste dobili maksimalnu površinu za projektovanje. Takođe, promenit ćete parametre Visual Studio .NET-a da biste sistem Help prikazali u zasebnom, spoljnom prozoru, a potom ćete u okviru sistema Help naznačiti jezički filtar. Kada sve završite, okruženje Visual Studio .NET izgledaće kao što je prikazano na slici 1-33.



Slika 1-33 Prilagođeno okruženje Visual Studio .NET

Kada prvi put pokrećete Visual Studio .NET, podrazumevanim rasporedom elemenata u prozoru predviđeno je da se prozori Server Explorer i Toolbox prikažu u obliku kartica na levoj strani ekrana, a prozori Solution Explorer i Properties na desnoj.

► **Da biste maksimalno povećali prostor na ekranu za uređivanje veb dokumenata i koda, uradite sledeće:**

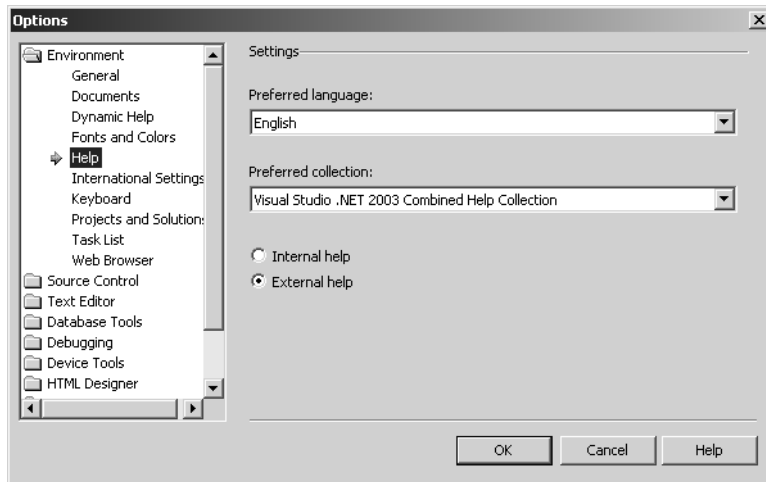
- Iz menija Window izaberite opciju Auto Hide All.
Ili
- Odaberite komandu Auto Hide koja se nalazi u gornjem desnom uglu prozora Solution Explorer i prozora Properties. Dugme Auto Hide liči na „pribadaču”.

Kada je funkcija Auto Hide uključena, prozor nestaje kada se pokazivač miša skloni sa njega. Prozor se ponovo otvara kada pokazivačem miša pređete preko kartice ovog prozora koja se nalazi na ivici ekrana.

Sledeće što se u Visual Studio .NET-u podrazumeva, jeste prikaz sistema Help unutar prozora za dokument. Budući da i sam sistem Help ima veći broj prozora, njihovo ređanje, jednog do drugog, moglo bi da zatrpa ekran.

► **Da biste sistem Help prikazali u zasebnom prozoru, uradite sledeće:**

1. Iz menija Tools izaberite Options. Sada Visual Studio .NET otvara okvir za dijalog Options.
2. U okviru za dijalog Options odaberite odrednicu Help koja se nalazi u sastavu omotnice Environment. Sada Visual Studio .NET prikazuje opcije za sistem Help, kao što je ilustrovano na slici 1-34.



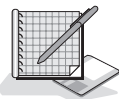
Slika 1-34 Podešavanje opcija Help

3. Izaberite najpre opciju External Help, a potom OK. Visual Studio prikazuje poruku u kojoj se navodi da promene neće stupiti na snagu dok se okruženje ponovo ne pokrene.

Ukoliko, eksperimentišući sa rasporedom elemenata u prozoru Visual Studio .NET, ne budete zadovoljni postignutim rezultatima, možete se vratiti na podrazumevane parametre prozora.

► **Da biste se vratili na podrazumevane parametre prozora, uradite sledeće:**

1. Iz menija Tools izaberite Options. Sada Visual Studio .NET otvara okvir za dijalog Options.
2. U okviru za dijalog Options, pod omotnicom Environment odaberite General.
3. Pritisnite dugme Reset Window Layout. Sada Visual Studio .NET prikazuje upozorenje, tražeći od vas da potvrdite da li želite da organizaciju prozora vratite na podrazumevanu.
4. Izaberite OK da bi ovo upozorenje nestalo, pa nakon toga ponovo odaberite OK kako bi se okvir za dijalog Options zatvorio. Visual Studio .NET vraća podrazumevani raspored elemenata u prozoru.

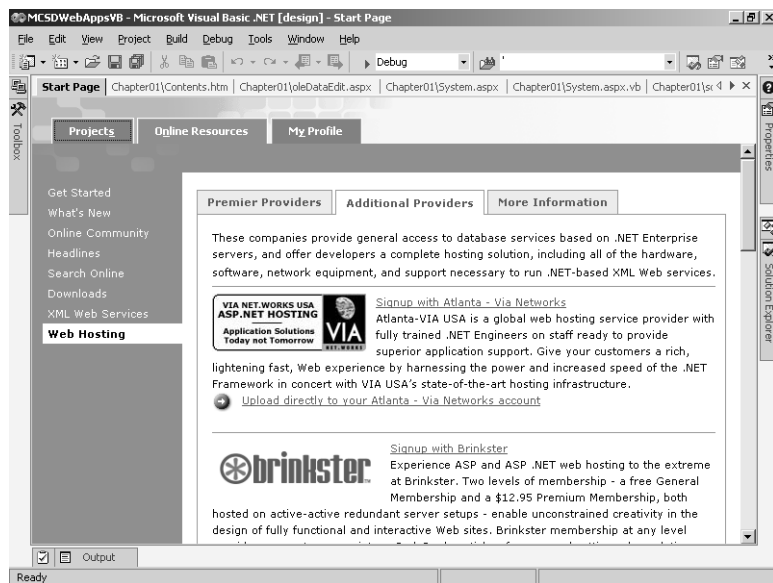


Vežba 2: Podešavanje naloga za smeštanje aplikacije na Webu

U ovoj vežbi podesićete nalog za smeštanje aplikacije na Webu, zahvaljujući kome će vam biti dozvoljeno da svoje veb aplikacije uvedete na Internet radi javnog testiranja i procenjivanja. Doduše, da biste prešli vežbe u ovom udžbeniku, nije neophodno da imate nalog za smeštanje na Webu, pošto veb aplikacije možete da izvršavate i da iz njih otklanjate programske greške lokalno, u svojoj radnoj stanici. Ipak, posedovanje naloga za smeštanje aplikacije na Webu omogućiće vam da ispitajte kako se vaša veb aplikacija ponaša kada je istovremeno traži veliki broj korisnika, a osim toga, i da svoja dostignuća na planu programiranja podelite sa drugima, što je više nego zabavno.

► **Da biste aplikaciju smestili na Web server, uradite sledeće:**

1. Opreделите se za dobavljača usluga sedišta.
2. Registrujte se kod tog servisa.
3. Prijavite se servisu i svoju aplikaciju mu dostavite preko Interneta.
4. Da bi se otvorio okvir Web Hosting u prozoru Visual Studio .NET, izaberite najpre početnu stranicu u prozoru za dokument, pa odaberite Web Hosting i, konačno, karticu Hosting Services. Otvara se okvir Web Hosting, kao što je prikazano na slici 1-35.



Slika 1-35 Dobavljači usluga sedišta za ASP.NET veb aplikacije

U odeljcima koji slede detaljnije se opisuju koraci koje treba da preduzmete radi podešavanja naloga za smeštanje aplikacije na Webu.



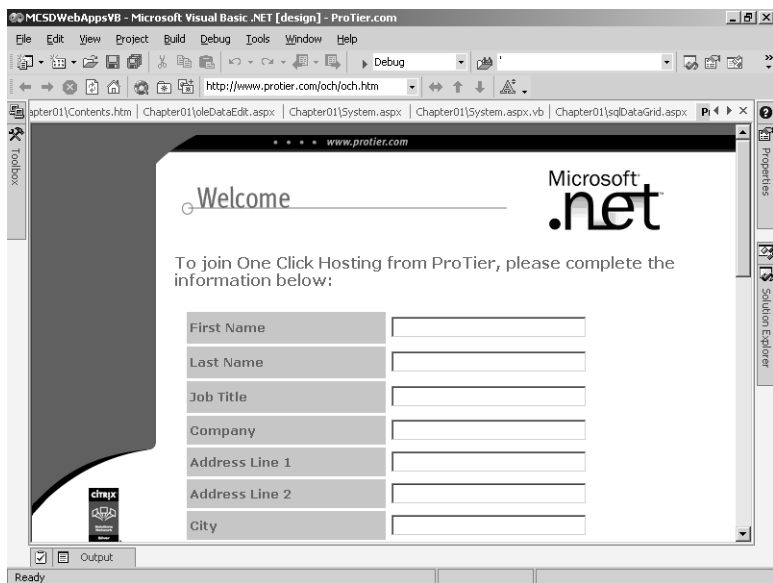
Vežba 3: Izbor dobavljača usluga na Internetu

U okviru Web Hosting navodi se veći broj dobavljača usluga sedišta koji podržavaju ASP.NET veb aplikacije. Svaki od tih dobavljača nudi razne kombinacije usluga koje pruža besplatno i onih uz nadoknadu. Da biste procenili koji bi, među svima njima, bio pravi za vas, posetite veb lokacije biranjem njihovih veza na početnoj stranici Visual Studio .NET-a. Pored cene, u obzir treba da uzmete i sledeće:

- **Nivo podrške** Da li dobavljač usluga sedišta ima forum korisnika ili neku oblast u koju se mogu slati odgovori na vaša pitanja? Ukoliko ima, proverite kakvo mišljenje o njemu imaju drugi korisnici.
- **Mogućnost kretanja** Da li ćete moći lako da prebacite aplikacije iz faze testiranja u fazu punog uvođenja? Da li dobavljač usluga sedišta obezbeđuje skladištenje i performanse koje će vam biti potrebne?
- **Podrška bazi podataka** Većina servisa pruža usluge smeštanja SQL baze podataka, ali svaki naplaćuje različitu visinu nadoknade za te usluge.

► **Da biste se registrovali kod dobavljača usluga na Internetu, uradite sledeće:**

1. U okviru Web Hosting izaberite vezu Sign Up With za dobavljača usluga za kojeg ste se opredelili. Visual Studio .NET prikazuje postupak registracije kod dobavljača usluga u prozoru za dokument, kao što se vidi na slici 1-36.



Slika 1-36 Stranica dobrodošlice dobavljača usluga ProTier

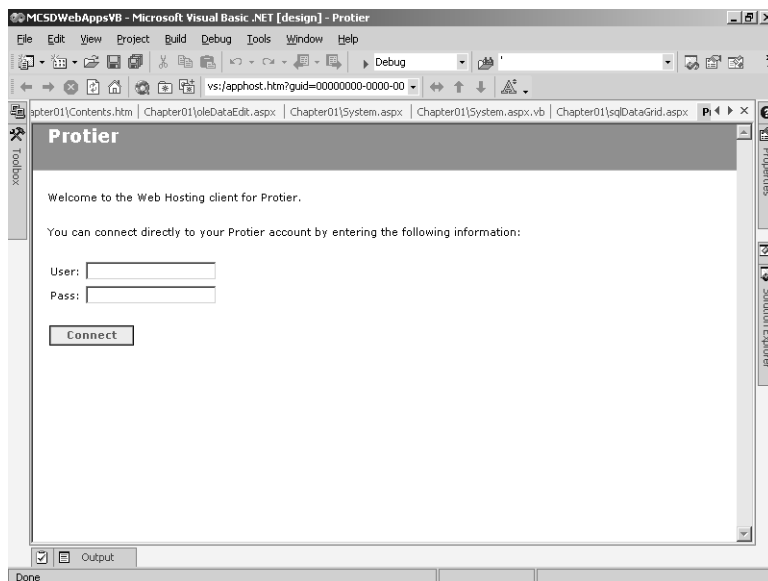
- Postupite prema uputstvima dobavljača usluga, kojima se od vas najčešće zahteva da date podatke o svom imenu, adresi i e-pošti. Kada popunite sve što treba, dobavljač usluga će vam e-poštom poslati korisničko ime i lozinku kojima vam se omogućuje da pristupite svom novom nalogu.

Napomena Registrujte se kod dobavljača usluga da biste otvorili vlastiti nalog. Tačan postupak razlikuje se od jednog do drugog dobavljača, ali se svi oni pridržavaju osnovnih koraka koji su opisani u nastavku.

Svi dobavljači usluga koji su prikazani na okviru Web Hosting nude **usluge tipa One-Click Hosting**, što znači da im svoje završene veb aplikacije možete dostaviti preko Interneta direktno iz okvira Web Hosting u prozoru Visual Studio .NET.

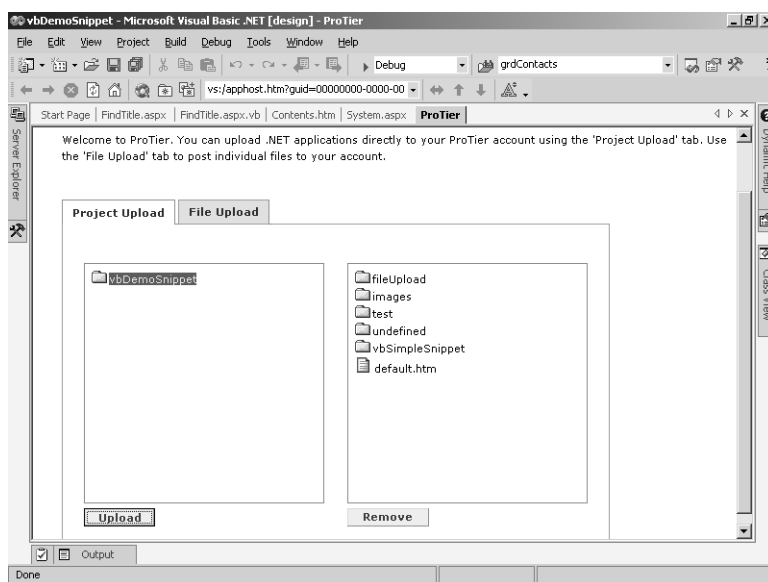
► **Da biste aplikaciju dostavili preko Interneta dobavljaču usluga sedišta, uradite sledeće:**

- Iz okvira Web Hosting u prozoru Visual Studio .NET izaberite vezu Upload Directly To Your Account za dobavljača usluga kod kojeg ste se prijavili. Visual Studio .NET prikazuje, u prozoru za dokument, stranicu pomoću koje dobavljač usluga proverava vaš identitet, kao što je ilustrovano na slici 1-37.



Slika 1-37 Stranica kojom dobavljač ProTier vrši proveru identiteta

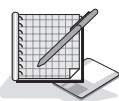
2. Na stranicu za proveru identiteta upišite korisničko ime i lozinku koji su predviđeni za vaš nalog. (Njih obično dobijate e-poštom odmah po prijavi za otvaranje naloga.) Pošto ste upisali potrebne podatke, dobavljač usluga prikazuje, u prozoru za dokument, stranicu za dostavu aplikacije preko Interneta, kao što se vidi na slici 1-38.



Slika 1-38 Stranica za dostavu aplikacije dobavljaču usluga ProTier

3. Iz spiska na levoj strani odaberite omotnicu koja sadrži aplikaciju koju hoćete da dostavite preko Interneta. Iz spiska na desnoj strani izaberite omotnicu na serveru u koju dostavljate aplikaciju. Izaberite Upload kako biste poslali projekat dobavljaču usluga sedišta.

Pošto ste aplikaciju poslali, možete je videti i vi, kao i bilo ko drugi, kretanjem do njene lokacije na Internetu. Dobavljači usluga smeštanja na Webu ne zahtevaju od korisnika da se prijavljuju za gledanje aplikacija, već samo za dostavljanje aplikacija.

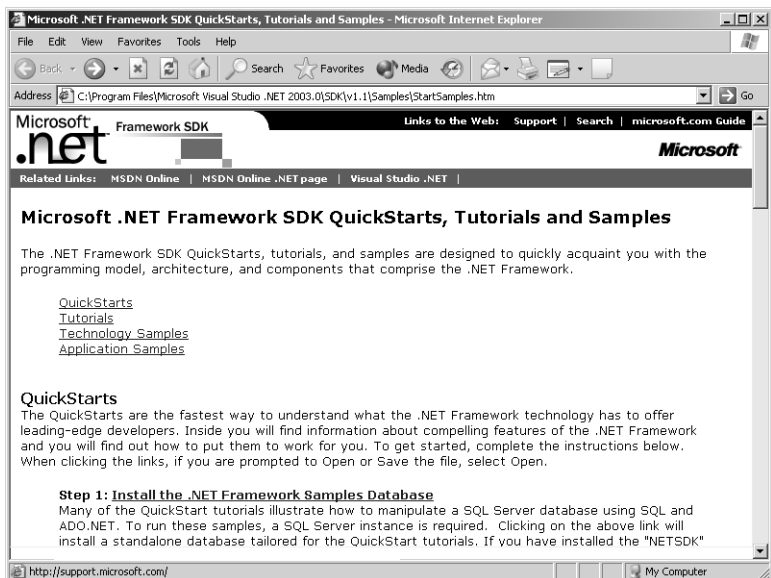


Vežba 4: Istraživanje priručnika za brzo učenje ASP.NET-a

U ovoj vežbi instaliraćete priručnike za brzo učenje koji se nalaze u sklopu *.NET SDK primera* i pregledaćete priručnike za brzo učenje *za ASP.NET*. Ovi priručnici za „brzi start” sadrže obilje informacija o različitim aspektima radnog okruženja .NET Framework i spadaju među najbolje resurse za učenje programiranja korišćenjem ASP.NET-a.

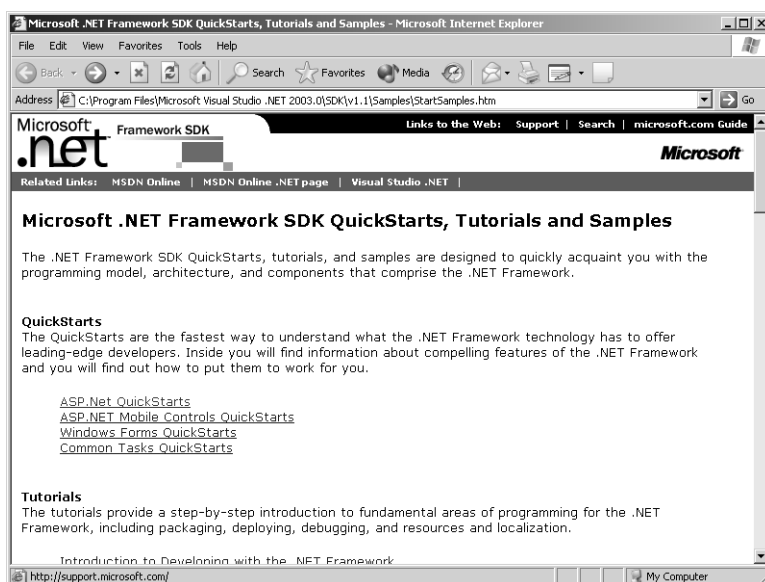
- **Da biste instalirali i pregledali priručnik za brzo učenje ASP.NET-a, uradite sledeće:**

1. Iz Windowsovog menija Start izaberite najpre All Programs, pa potom Microsoft .NET Framework SDK i, konačno, Samples And QuickStart Tutorials. Windows prikazuje stranicu sa uputstvima za instaliranje priručnika Microsoft .NET Framework SDK QuickStarts, Tutorials and Samples, kao što je ilustrovano na slici 1-39.



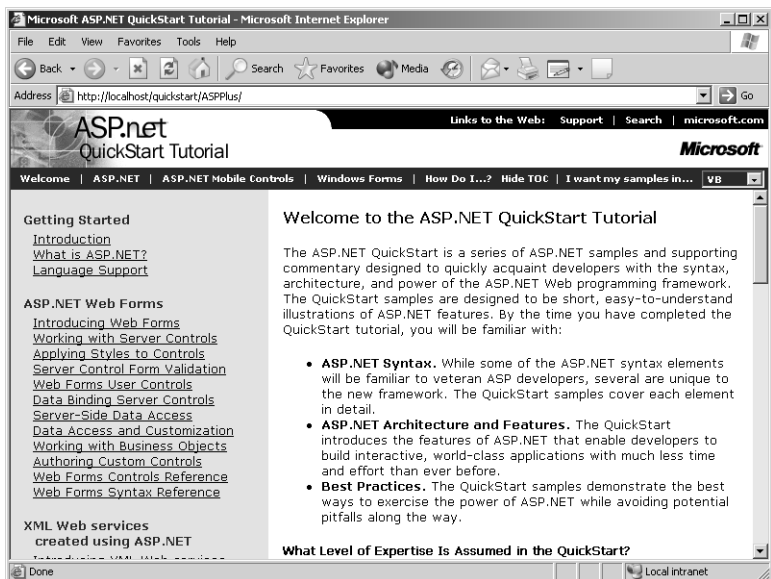
Slika 1-39 Stranica sa uputstvima za instaliranje priručnika i primera Microsoft .NET Framework SDK QuickStarts, Tutorials and Samples

- Izaberite vezu Step 1: Install The .NET Framework Samples Database kako biste instalirali i konfigurirali SQL bazu podataka kojom se služe primeri navedeni u priručniku za brzo učenje.
- Izaberite hipervezu Step 2: Set Up The QuickStarts da biste instalirali i konfigurirali veb lokacije koje se koriste u primerima navedenim u priručniku za brzo učenje.
- Kada sledeći put budete pristupili vezi Microsoft .NET Framework QuickStart Tutorials, iz menija Start, Internet Explorer preskače ovu stranicu koja se odnosi na instaliranje i odvodi vas direktno na stranicu Microsoft .NET Framework SDK QuickStarts, Tutorials and Samples (slika 1-40).



Slika 1-40 Stranica sa priručnicima QuickStart u okruženju .NET Framework

- Izaberite vezu ASP.NET QuickStarts kako biste proučili stranicu sa sadržajem priručnika, koja je nalik na ovu prikazanu na slici 1-41.



Slika 1-41 Stranica sa sadržajem priručnika za brzo učenje ASP.NET-a

Priručnici za brzo učenje ASP.NET-a sadrže teme koje se odnose na obavljanje raznih zadataka ASP.NET programiranja u jezicima Visual Basic .NET, Visual C# i JScript. Kod većine primera u priručnicima za brzo učenje serverski skript se postavlja u blokove skripta, a ne korišćenjem datoteka sa kodom u pozadini, kao što je slučaj u ovom udžbeniku. Uglavnom je bolje da se koriste datoteke sa kodom u pozadini, jer se na taj način korisnički interfejs razdvaja od programske logike.

Pregled poglavlja

Sledeća pitanja su namenjena utvrđivanju najvažnijih delova gradiva koje je u ovom poglavlju obrađeno. Ukoliko ne znate odgovor na neko pitanje, opet pročitajte odgovarajuću lekciju i pokušajte ponovo da na njega odgovorite. Tačni odgovori na ova pitanja nalaze se u dodatku.

1. Navedite dva primera po kojima se ASP.NET veb aplikacija razlikuje od tradicionalne Windows aplikacije.
2. Koja su dva glavna dela iz kojih se sastoji radno okruženje .NET Framework?
3. Kako se vraća na podrazumevani raspored elemenata u prozoru Visual Studio .NET?
4. Zašto editor koda Visual Studio .NET-a automatski ne popunjava sledeći delimično urađen red koda (važi samo za korisnike jezika Visual C#)?

```
int intX = system.math
```
5. Kada ASP.NET ne može da se koristi za pravljenje veb aplikacije?

