



DEO II

**Active Server
komponente i
ADO**

— |

|

—

|

|



Scripting Object Model i SOM objekti

AKO STE VEĆ RADILI PROGRAMIRANJE APLIKACIJA, VEROVATNO STE NAVIKLI DA budete u mogućnosti da čitate fajlove sa drajvova i da upisujete fajlove na drajvove. Ako ste radili JavaScript i VBScript programiranje na web stranicama, možda ste primetili da ove mogućnosti nisu prisutne. Naravno, pošto se JavaScript i VBScript izvršavaju pri preuzimanju sadržaja sa Interneta (download), te mogućnosti su bile izostavljene da zlobni programeri ne bi mogli pisati stranice koje uništavaju mašine korisnika jednostavno pri preuzimanju web stranice.

Na serveru, ćete ipak želeći da povremeno imate mogućnost da čitate sa drajvova ili da vršite upis na drajvove. Šesto poglavlje objašnjava objekte koji su dostupni za ASP i koji omogućavaju da se ove funkcije obave. Ali pažljivo nastavite; ovo je vrlo bitno, da ne biste slučajno uništili svoju mašinu!

Scripting Object Model

Active Server Pages 3.0 podržava korišćenje objekata koji su instalirani na serveru da bi se poboljšale mogućnosti koje se nasleđuju u ASP-u. Request i Response objekti su ugrađeni (ili unutrašnji) objekti. Komponente su takođe objekti. Već smo koristili MyInfo komponentu za pamćenje podataka između stranica i sesija, i dostupno je još nekoliko drugih komponenti/objekata. U stvari, objekti postoje za manipulaciju sa fajl sistemom, za upravljanje uobičajenim zadacima na web sajtu i čak i za promenu osobina Registry i za postavljanje fajlova na web stranice. O objektima razmišljajte kao o komponentama aplikacionog programa koje se mogu nanovo i nanovo koristiti i koje obezbeđuju određenu funkcionalnost na taj način da se lako mogu koristiti u Vašim ASP aplikacijama, bez potrebe da se “ponovo izmišlja točak”.

Model objekta je jednostavno opis toga kako se dostupni objekti uklapaju zajedno i kako su vezani jedan za drugog po funkciji. U ASP 3.0, dostupna su tri objekta preko Scripting Runtime Library. To su

- **Dictionary objekat** Ovaj objekat može pamtit i ime/vrednost parove koje možete kreirati, podesiti i dobiti.
- **FileSystemObject** Ovaj objekat Vam daje pristup osnovnom fajl sistemu (preko dajva, foldera i objekata fajlova) na serveru, i takođe u IE 5.0 na klijentu.
- **TextStream objekat** Ovaj objekat Vam omogućava da kreirate, čitate i vršite upis u tekstualne fajlove.

Ovi objekti su u vezi jedni sa drugima (model objekta) gde je FileSystemObject vrsta roditeljskog objekta koji ima mogućnost da instancira Drive, Folder, File i TextStream objekte, i Dictionary objekat služi za oba ova cilja. Zajedno čine niz objekata koji su dostupni za podrazumevane pretprocesore za skriptovanje (VBScript i JScript) preko Scripting Runtime Library.

Kreiranje objekata

Svaki put kada treba da koristite komponentu, možete kreirati njenu instancu korišćenjem ili Server.CreateObject metoda ili <OBJECT> oznaka. U zavisnosti od toga kako će se koristiti, treba da imate na umu nekoliko pitanja. Ova pitanja utiču na doseg i performanse Vaših objekata.

Više o dosegu

Objekti koji su kreirani korišćenjem bilo kog od ova dva metoda imaju, po defaultu, doseg na nivou stranice. Oni su dostupni dok stranica izvršava ASP, i oni se eliminišu jednom kada se završi izvršavanje stranice i kada se rezultati pošalju klijentu. Da bi se doseg podesio da bude na nivou sesije ili aplikacije, jednostavno možete umetnuti atribut SCOPE="APPLICATION" u Vaše <OBJECT> oznake (kad se kreiraju instance tog objekta u okviru global.asa fajla). Primer ove prakse se može naći u 5. poglavlju.

SAVET

Postoje razlike u načinu rada Server.CreateObject metoda i <OBJECT> oznaka. Kada se objekat kreira pomoću Server.CreateObject metoda, on se odmah instancira. Kada se objekat kreira pomoću <OBJECT> oznaka, on se ne kreira dok se ne pozove prvi put u okviru skripta. Takođe, imate mogućnost da uklanjate objekte iz sesije ili aplikacije ako ste koristili Server.CreateObject metodu da ih kreirate, ali nemate tu mogućnost ako ste koristili <OBJECT> oznake.

Performanse i modeli sa nitima (threads)

Performanse i skalabilnost Vaše aplikacije zavisi, delimično, od toga kako instancirate komponente i takođe od njihovog *modela sa nitima*. Niti (*threads*) su osnovna jedinica instrukcija koje procesor (CPU) može izvesti da bi obavio bilo koji zadatak koji zahteva operativni sistem i aplikacioni programi koji se izvršavaju. Razmišljajte o svakoj niti kao o nizu instrukcija koji predstavlja zadatak ili program. Mnoge niti se mogu izvršavati u isto vreme, gde se svaka od

njih bori da dobije procesorsko vreme i nosi prioritet. Model sa nitima koji koristi komponenta može uticati na njenu sposobnost da obavi svoj posao.

U Windowsu 2000 postoji pet modela niti (threads):

- **Sa jednom niti** Samo jedan proces u jednom trenutku može koristiti komponentu.
- **Sa apartment model niti** Više procesa može koristiti komponentu, ali korišćenjem samo jedne niti.
- **Sa neutralnim nitima** Više procesa može koristiti komponentu, na bilo kojoj od njenih niti.
- **Sa višestrukim ili slobodnim nitima** Više procesa može koristiti komponente na različitim nitima.
- **Obostrani model sa nitima** Komponenta može biti simultano i sa nitima u apartment modelu i sa slobodnim nitima.

Sa dosegom na nivou stranice, komponente sa nitima u apartmanu i sa neutralnim nitima rade najbolje; sa dosegom na nivou sesije i sa dosegom na nivou aplikacije, komponente sa obostranim modelom sa nitima rade najbolje. Microsoft preporučuje korišćenje dosega na nivou stranice gdegod je to moguće, pa će se većina Vašeg posla odnositi na komponente sa nitima u apartmanu (nema mnogo komponenti sa neutralnim nitima za sada).

Kreiranje Scripting Run-Time Library objekata

U sledećem odeljku, naučićete kako da kreirate Scripting Run-Time Library objekte individualno i kako da demonstrirate tipove koda koje treba koristiti za rad sa njima. Kad sakupljate zajedno u jednu veliku ASP stranicu u Project odeljku ovog poglavlja; i nakon toga, dajete listu svih atributa <OBJECT> oznaka koje se mogu podesiti, samo radi dobre mere. Na slici 6.1 možete videti kako izgleda otvaranje ASP stranice.

Dictionary objekat

Dictionary objekat je dizajniran da Vam omogući da kreirate kontejner za ime/vrednost parove koje ste izabrali. Ako želite da omogućite korisniku da generiše svoj sopstveni set ime/vrednost parova, Dictionary objekat postaje vrlo zgodan, u suprotnosti sa ime/vrednosti koje se izvode iz Form elemenata, gde su imena u ime/vrednost parovima ručno kodirana.



Slika 6.1

Uvodna ASP File System stranica za menadžment

Osobine i metode Dictionary objekta

Dictionary objekat ima sledeće osobine:

- **CompareMode** Ova osobina radi samo za VBScript, i podešava ili vraća mod za poređenje stringova za imena (ključeve) koje kreirate.
- **Count** Ova osobina vraća broj imena koja se trenutno nalaze u rečniku.
- **Item(ključ)** Ova osobina podešava ili vraća vrednost stavke (vrednosti) za određeni ključ (ime).
- **Key (ključ)** Ova osobina podešava vrednost imena (ključa).
- **Dictionary** objekat ima sledeće metode:
- **Add (ključ.stavka)** Ovaj metod dodaje ključ (ime) i stavku (vrednost) u Dictionary.
- **Exists (ključ)** Ovaj metod vraća True ako određeni ključ (ime) postoji i False ako ne postoji.
- **Items** Ovaj metod vraća niz koji sadrži stavke iz Dictionary.
- **Keys** Ovaj metod vraća niz koji sadrži ključeve iz Dictionary.
- **Remove(ključ)** Ovaj metod uklanja individualni ime/vrednost par koji je označen pomoću ključa koji ste odredili.
- **RemoveAll** Ovaj metod uklanja sve ime/vrednost parove u Dictionary.

Kreiranje Dictionary komponente sa dosegom na nivou stranice

Instancu Dictionary objekta možete kreirati korišćenjem ili VBScript /JScript sa Server.CreateObject metodom ili korišćenjem <OBJECT> oznaka u global.asa fajlu. Na primer, ako želite da izgradite Dictionary unutar ASP stranice, mogli biste koristiti sledeći kod:

```
%><OBJECT RUNAT="SERVER" SCOPE="PAGE" ID="objDictionary"
PROGID="Scripting.Dictionary"></OBJECT>
```

Zapazite ID parametar. Ime u ovom kodu je objDictionary; ali naravno, možete koristiti bilo koje ime koje Vam se sviđa. Korišćenje <OBJECT> oznaka u Vašem ASP skriptu je dozvoljeno, ali ne možete podesiti doseg na viši nivo od nivoa stranice ili će biti generisana greška. Na slici 6.1 su prikazana demonstraciona dugmad na stranici za kreiranje Dictionary objekta, pristup tom objektu iz stranice i iz global.asa fajla. Na slici 6.2 je prikazana poruka koju primete kada kreirate Dictionary objekat u okviru ASP stranice. Zapazite da daje listu tekućeg broja ključeva.



Slika 6.2

Dictionary objekat koji je kreiran sa nula ključeva

Lista imena /vrednosti u Dictionary

Samo radi zabave, ime/vrednost parovi u Dictionary objektu se nazivaju ključevi i stavke, respektivno. U sledećem segmentu koda kreirate instancu Dictionary objekta sa dosegom na nivou sesije (tako da će zadržavati vrednosti duž sukcesivnih stranica u našoj aplikaciji) u global.asa fajlu:

```
<OBJECT ID="objDictionarySes" RUNAT="server" SCOPE="Session"
PROGID="Scripting.Dictionary">
</OBJECT>
```

Sledeće, sastavljate neki kod za pristup tom Dictionary objektu iz ASP skripta u Vašoj aplikaciji. Ovaj kod dimenzioniše nekoliko promenljivih (za čuvanje podataka o paru ime/vrednost, brojač i referenciju na Dictionary objekat), postavlja vrednosti za neke od tih promenljivih na vrednosti koje su primljene sa stranice za zahtevanje formulara, i iterativno prolazi kroz kreirane parove za ime/vrednost (kao što je prikazano na slici 6.3):

```
Dim vKey
Dim vItem
Dim arKeys
Dim arItems
Dim intI
Dim objDictionaryS
Set objDictionaryS = Session("SessionDictionary")
vKey = Request.Form("key")
vItem = Request.Form("item")
objDictionaryS.Add vKey, vItem
arKeys = objDictionaryS.Keys
arItems = objDictionaryS.Items
Response.Write "<B>Session Dictionary Object Created</B><BR>"
Response.Write "<B>The Dictionary Key/Item Pairs You Added Are:</B><P>"
    For intI = 0 to objDictionaryS.Count - 1
        Response.Write "<B>Key = </B>" & arKeys (intI)
        Response.Write " and <B>Item = </B>" & arItems (intI) & "<BR>"
    Next
```

Kao eksperiment koji možete isprobati sami, dodajte ime/vrednost par u Dictionary, i onda ponovo pokušajte da dodate isto ime u Dictionary. Trebalo bi da dobijete poruku o grešci koja govori da ključ već postoji.

**Slika 6.3**

Globalni Dictionary objekat koji je kreiran pomoću ključa/stavke

Uklanjanje parova ime/vrednost iz Dictionary

Da biste uklonili ime/vrednost parove iz Dictionary objekta, možete koristiti ugrađeni Remove metod. Na slici 6.4 je prikazano uklanjanje parova ključ/vrednost. Kada sa naše stranice referencirate Dictionary objekat na nivou sesije koji je kreiran u global.asa fajlu, takođe treba da podesite da Dictionary na nivou sesije bude jednak objektu na Vašoj stranici da bi promene imale efekta, kao što je prikazano u sledećem kodu u liniji 3:

```
Set objDictionaryS = Session("SessionDictionary")
objDictionaryS.RemoveAll
Set Session("SessionDictionary") = objDictionaryS
```

**Slika 6.4**

Uklanjanje ključ/vrednost parova iz Dictionary objekta

Izmena imena/vrednosti Dictionary objekta

Jednom kreirana imena i vrednosti Dictionary objekta se mogu menjati na nivou stranice. Ovo biste možda želeli da uradite da bi određeni korisnik mogao da igra igru sa drugim korisnicima onlajn-neku vrstu proste igre rečima, gde jedan korisnik kreira ime/vrednost par, drugi korisnik menja taj ime/vrednost par, i prvi korisnik pokušava da pogodi šta se izmenilo.

Ove vrste izmena se mogu napraviti pomoću koda koji je sličan sledećem kodu, kao što je prikazano na slici 6.5:

```
Set objDictionaryS = Session("SessionDictionary")
vKey = Request.Form("key")
vItem = Request.Form("item")
objDictionaryS.Item(vKey) = vItem
Set Session("SessionDictionary") = objDictionaryS
Response.Write "<B>Dictionary Key Item Value Has Been
Changed To:</B><P>"
arKeys = objDictionaryS.Keys
arItems = objDictionaryS.Items
For intI = 0 to objDictionaryS.Count - 1
    Response.Write "<B>Key = </B>" & arKeys (intI)
    Response.Write " and <B>Item = </B>" & arItems
(intI) & "<BR>"
Next
```

Ovaj kod referencira Dictionary objekat na nivou sesije, podešava vrednosti nekoliko promenljivih u skladu sa Form elementima u stranici koja zahteva taj objekat, menja vrednost postojećih ime/vrednost parova u Dictionary objektu, i onda daje listu parova ime/vrednost da bi prikazao izmene koje su napravljene.

JEDNOMINUTNA VEŽBA

- Dictionary objekat sadrži parove ime/vrednost. Koje je ime Dictionary objekta za ime?
- Koje je ime Dictionary objekta za vrednost?
 - *Ključ*
 - *Stavka*



Slika 6.5

Promena vrednosti stavki za Dictionary objekat

Scripting.FileSystemObject objekat

Upozorenje! Nepravilno korišćenje FileSystemObject objekta može potencijalno ugroziti Vaš sistem zato što FileSystemObject može pristupiti bilo kom folderu ili fajlu u Vašem sistemu i napraviti izmene u njemu.

Samim tim, FileSystemObject objekat i njegovi pomoćni objekti spadaju u najvrednije alate koji se mogu naći u Vašem polju sa alatima, baš zbog njihove moći. Demo ASP stranica sa dugmadima za kreiranje i prikazivanje informacija o FileSystemObject objektu je prikazana na slici 6.6.

Samo zapamtite, bilo koji pristup drajvovima, folderima i fajlovima servera koji obezbeđujete bi mogao da ugrozi Vaše operacije, zato pazite vrlo dobro šta radite. Mada je u ovim primerima korisniku dozvoljeno da izabere bilo koji folder ili fajl koji želi da modifikuje, ovo je jedan od momenata kada bi moglo biti najbolje da se koristi težak kod sa referencijama na foldere i fajlove, umesto da se obezbeđuje neograničeni pristup preko promenljivih.

Osobine FileSystemObject objekta

FileSystemObject objekat obezbeđuje samo jednu osobinu-Drives-i ova osobina vraća kolekciju svih dostupnih drajvova u sistemu, i takođe bilo kojih mrežnih drajvova koji su mapirani na njega. Možete podesiti da objekat po Vašem izboru bude jednak ovoj osobini, i onda se drajvovima može pristupiti preko osobina i metoda Drive objekta (u sledećem odeljku).

Metode FileSystemObject objekta

FileSystemObject ima nekoliko metoda, gde se neke odnose na drajvove, neke na foldere i neke na fajlove. Postoji veliki broj načina za navigaciju do foldera i fajlova i za njihovu izmenu, i oni su korisni pod različitim okolnostima. Na primer, ako imate celu putanju do fajla, možda ćete želeći da koristite FileSystemObject da biste ga dobili i modifikovali. Ili, ako se već nalazite u folderu, možda ćete želeći da koristite objekat foldera da biste referencirali i izmenili fajl. To samo zavisi od toga gde se nalazite u fajl sistemu i koje informacije su Vam dostupne. U bilo kom slučaju, evo metoda FileSystemObject objekta:

- **DriveExists(slovodrajva)** Ovaj metod vraća True ako slovo drajva koje je određeno postoji i False ako ne postoji.
- **GetDrive(slovodrajva)** Ovaj metod vraća Drive objekat koji je ekvivalentan slovu drajva koje je određeno.
- **GetDriveName(slovodrajva)** Ovaj metod vraća ime drajva određenog pomoću slova za drajv u vidu stringa.
- **BuildPath(putanja, ime)** Ovaj metod dodaje fajl ili folder koji je određen po imenu u tekuću putanju i dodaje separator (“\”) ako je neophodno.
- **CopyFolder(izvor, odredište, overwrite)** Ovaj metod kopira folder iz jedne lokacije (izvora) u drugu (odredište), i briše i zamenjuje stari fajl po defaultu osim ako overwrite parametar nije eksplicitno postavljen na False.
- **CreateFolder(imefoldera)** Ovaj metod kreira folder čije je ime imefoldera na putanji koja je navedena u imefoldera.
- **DeleteFolder(imeputanjefoldera, sila)** Ovaj metod briše folder koji se nalazi u imeputanjefoldera i može čak i da briše foldere koji se mogu samo čitati ako je overwrite parametar postavljen na True.
- **FolderExists(imeputanjefoldera)** Ovaj metod vraća True ako folder koji je naveden u imeputanjefoldera postoji i False ako ne postoji.
- **GetAbsolutePathName(putanja)** Ovaj metod vraća putanju na osnovu putanje tekućeg foldera i određene putanje.

- **GetFolder(folder)** Ovaj metod vraća objekat foldera koji je ekvivalentan određenom folderu.
- **GetParentFolderName(putanja)** Ovaj metod vraća ime roditeljskog foldera za folder ili fajl koji je određen u putanji.
- **GetSpecialFolder(folder)** Ovaj metod vraća objekat foldera koji je ekvivalentan specijalnom folderu koji je određen, kao što je Windows folder, System folder ili Temporary folder.
- **MoveFolder(izvor, odredište)** Ovaj metod pomera određeni folder iz izvora (kompletna putanja koja uključuje ime foldera) na lokaciju određenu kao odredište (kompletna putanja koja uključuje ime foldera).
- **CopyFile(izvor, odredište, overwrite)** Ovaj metod kopira fajl koji je određen u izvoru na lokaciju određenu kao odredište, pri čemu briše i zamenjuje bilo koji postojeći fajl sa istom putanjom i imenom osim ako overwrite parametar nije postavljen na False.
- **CreateTextFile(putanjafajla, overwrite, unicode)** Ovaj metod kreira tekstualni fajl koji je određen u putanji fajla, pri čemu briše i zamenjuje bilo koji postojeći fajl sa istom putanjom i imenom osim ako overwrite parametar nije postavljen na False, i kao ASCII osim ako opcioni Unicode parametar nije postavljen na True.
- **DeleteFile(putanjafajla, sila)** Ovaj metod briše fajl koji je određen pomoću putanje fajla, i čak i one fajlove koji se mogu samo čitati ako je parametar sile postavljen na True.
- **FileExists(putanjafajla)** Ovaj metod vraća True ako određeni fajl postoji
- **GetBaseName(putanjafajla)** Ovaj metod vraća ime fajla koji je određen u putanji fajla, i uklanja podatke o drajvu, folderima i ekstenziji.
- **GetExtensionName(putanjafajla)** Ovaj metod vraća ekstenziju za dati fajl.
- **GetFile(putanjafajla)** Ovaj metod vraća File objekat koji je ekvivalentan fajlu koji je određen u putanji.
- **GetFileName(putanja)** Ovaj metod vraća delove putanje i imena fajla koji su određeni pomoću putanje.
- **GetTempName** Ovaj metod generiše slučajno ime fajla za upotrebu kod generisanja privremenih fajlova ili foldera.
- **MoveFile(izvor, odredište)** Ovaj metod pomera fajl koji je određen pomoću izvora na lokaciju koja je određena kao odredište.
- **OpenTextFile(imefajla, iomod, create, format)** Ovaj metod ili kreira ili otvara tekstualni fajl sa imenom imefajla (u zavisnosti od toga da li je create parametar postavljen na True), koji koristi iomod za čitanje, upis ili pripajanje i formatiran je kao ASCII, Unicode ili sistemski podrazumevani format.

Kreiranje FileSystemObject objekta

Sledeći kod se može koristiti za kreiranje FileSystemObject objekta i za iterativni prolaz kroz drajvove koji su dostupni na Vašoj lokalnoj mašini:

```

Dim objFileSysOb
Dim collObj
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Response.Write "<B>The FileSystemObject Has Been Created and the Server
Drives Are:</B>"
    For Each collObj in objFileSysOb.Drives
        Response.Write "<BR>Drive = " & collObj
    Next

```

Kod za kreiranje objekta i njegovo postavljanje u promenljivu.

Kod jednostavno dimenzioniše nekoliko promenljivih, postavlja objFileSysOb na FileSystemObject objekat koji je kreiran na serveru, i onda vraća nazad iterativnu listu svih drajvova na serveru (videti sliku 6.7).

Drive objekat

Jednom kada kreirate FileSystemObject objekat, možete lako da dođete do Drive objekta, i onda možete početi da uzimate informacije o dostupnim drajvovima. Drive objekti imaju sledeće osobine:

- **AvailableSpace** Ova osobina vraća veličinu prostora koja je dostupna na drajvu.
- **DriveLetter** Ova osobina vraća slovo drajva.
- **DriveType** Ova osobina vraća tip drajva kao unknown (nepoznat), removable (uklonjiv), fixed (fiksirani), network (mrežni), CD-ROM ili RAMDisk.
- **FileSystem** Ova osobina vraća tip fajl sistema drajva kao FAT, NTFS ili CDFS pod uslovom da je tip prepoznatljiv.



Slika 6.7
Svi drajvovi na serveru

- **FreeSpace** Ova osobina vraća ukupan slobodan prostor koji je dostupan na drajvu.
- **IsReady** Ova osobina je True ako je drajv spreman i False ako nije.
- **Path** Ova osobina vraća specifikaciju slova drajva za drajv (C:, na primer).
- **RootFolder** Ova osobina vraća objekat foldera koji je ekvivalentan root folderu za drajv.
- **SerialNumber** Ova osobina vraća serijski broj koji je identifikovao volumen diska.
- **ShareName** Ova osobina vraća mrežno ime za drajv koji se deli.
- **TotalSize** Ova osobina vraća ukupnu veličinu drajva.
- **VolumeName** Ova osobina podešava ili vraća ime volumena za lokalne drajvove.

Iteracija kroz osobine drajva

Možete postaviti listing svih dostupnih drajvova i njihovih osobina pomoću Drive objekta, korišćenjem sledećeg koda:

```
Dim vDrive
vDrive = Request.Form("drive")
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Response.Write "<B>The Drive Info Is:</B><P>"
Response.Write "Drive Letter = " & objFileSysOb.GetDrive(vDrive).DriveLetter & "<BR>"
Response.Write "Drive Type = " & objFileSysOb.GetDrive(vDrive).DriveType & "<BR>"
Response.Write "File System = " & objFileSysOb.GetDrive(vDrive).FileSystem & "<BR>"
Response.Write "AvailableSpace = " & objFileSysOb.GetDrive(vDrive).AvailableSpace & "<BR>"
Response.Write "Free Space = " & objFileSysOb.GetDrive(vDrive).FreeSpace & "<BR>"
Response.Write "Is Ready = " & objFileSysOb.GetDrive(vDrive).IsReady & "<BR>"
Response.Write "Path = " & objFileSysOb.GetDrive(vDrive).Path & "<BR>"
Response.Write "Root Folder = " & objFileSysOb.GetDrive(vDrive).RootFolder & "<BR>"
Response.Write "Serial Number = " & objFileSysOb.GetDrive(vDrive).SerialNumber & "<BR>"
Response.Write "Share Name = " & objFileSysOb.GetDrive(vDrive).ShareName & "<BR>"
Response.Write "Total Size = " & objFileSysOb.GetDrive(vDrive).TotalSize & "<BR>"
Response.Write "Volume Name = " & objFileSysOb.GetDrive(vDrive).VolumeName & "<BR>"
```

Ovaj kod dimenzioniše promenljivu pod imenom vDrive da pamti specifikaciju slova drajva koju je obezbedio korisnik i postavlja je na tu vrednost korišćenjem Request objekta. On onda kreira FileSystemObject i uzima Drive objekat i odgovarajuće osobine drajva pomoću FileSystemObject.GetDrive(vDrive) referencija na osobinu. Na slici 6.8 je prikazana rezultujuća lista osobina Drive objekta.



Slika 6.8

Lista osobina drajva

JEDNOMINUTNA VEŽBA

- Navedite tri osobine Drive objekta.
- Navedite tri tipa Drive objekata.
 - Slobodan prostor, slovo drajva, tip drajva
 - Uklonjivi, fiksirani, CD-ROM

Folder objekat

Iako možete dobiti foldere i fajlove pomoću FileSystemObject, Folder objekat se takođe može koristiti za dobijanje fajlova i za rad sa njima. Možete koristiti Subfolders osobinu Folder objekta da dobijete Folders kolekciju i Files osobinu da dobijete Files kolekciju, i čak možete kreirati i tekstualne fajlove pomoću Folder objekta.

Osobine Folder objekta

Folder objekat ima sledeće osobine:

- **Attributes** Ova osobina vraća attribute kao što su normal (normalan), read-only (samo za čitanje), hidden (sakriveni), system (sistemski), volume (volumen), directory (direktorijum), archive (arhiva), alias (drugo ime) i compressed (kompresovani).
- **DateCreated** Ova osobina vraća datum i vreme kada je folder bio kreiran.
- **DateLastAccessed** Ova osobina vraća datum i vreme kada se zadnji put pristupilo folderu.
- **DateLastModified** Ova osobina vraća datum i vreme kada je folder bio zadnji put modifikovan.
- **Drive** Ova osobina vraća slovo drajva za folder.
- **Files** Ova osobina vraća Files kolekciju File objekata koji su ekvivalentni svim fajlovima u folderu.
- **IsRootFolder** Ova osobina vraća True ako je folder root folder drajva.
- **Name** Ova osobina podešava ili vraća ime foldera.
- **ParentFolder** Ova osobina vraća Folder objekat koji je ekvivalentan roditeljskom folderu za tekući folder.
- **Path** Ova osobina vraća apsolutnu putanju za tekući folder.
- **ShortName** Ova osobina vraća DOS verziju imena foldera.
- **ShortPath** Ova osobina vraća DOS verziju imena putanje za folder.
- **Size** Ova osobina vraća veličinu fajlova i foldera koji se nalaze u tekućem folderu.
- **Subfolders** Ova osobina vraća Folders kolekciju za sve foldere u tekućim folderima.
- **Type** Ova osobina vraća opis tekućeg foldera kao string, ako je opis dostupan.

Metode Folder objekta

Jednom kada budete obezbedili Folder objekat, možete direktno kopirati, pomerati, brisati i tako dalje, ali bez potrebe da navedete punu putanju pošto se već nalazite tamo. Takođe možete kreirati i tekstualni fajl. Metodi za ostvarivanje ovih akcija su

- **Copy(odredište, overwrite)** Ovaj metod kopira tekući folder na lokaciju koja je određena kao odredište, pri čemu se briše i zamenjuje postojeći folder sa istim imenom osim ako overwrite parametar nije postavljen na True.
- **Delete(force)** Ovaj metod briše tekući folder i njegov sadržaj; i, ako je parametar force stavljen na True, briše čak i foldere koji se mogu samo čitati.
- **Move(odredište)** Ovaj metod pomera tekući folder na lokaciju koja je određena kao odredište, ali generiše grešku ako je folder sa istim imenom već na toj lokaciji.

- **CreateTextFile(putanjafajla, overwrite, unicode)** Ovaj metod kreira tekstualni fajl korišćenjem određene putanje fajla i imena fajla, pri čemu briše i zamenjuje postojeće fajlove sa tim imenom ako je overwrite parametar podešen na True i u Unicode formatu ako je unicode parametar podešen na True.

Pronalaženje root foldera Mogućnost da se locira i vrati referenca na root folder je često korisna, i sledeći kod radi upravo to:

```
vDrive = Request.Form("drive")
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Response.Write "<B>The Root Folder Is:</B><P>"
Response.Write "Root Folder = " &
objFileSysOb.GetDrive(vDrive).RootFolder & "<BR>"
```

Kod koristi već dimezionisanu vDrive promenljivu da dobije i zapamti specifikaciju Drive objekta koju je obezbedio korisnik, i onda traži root folder (videti sliku 6.9) za određeni drajv korišćenjem RootFolder osobine za taj drajv iz FileSystemObject objekta.



Slika 6.9

Pronalaženje root foldera

Pronalaženje podfoldera za root folder Sledeći primer koda služi za dobijanje podfoldera iz root foldera za drajv, i onda daje listu tih podfoldera i to je još jedna dobra mogućnost. Zapazite u sledećem kodu da treba da postavite referencu na Root Folder objekat, i da onda dobijete podfoldere iz toga, umesto da referenciranje obavljate u jednom koraku, kao što ste to radili u prethodnim primerima:

```

Dim colFolder
Dim colSubFolders
Dim colSubFolderItem
vDrive = Request.Form("drive")
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Set colFolder = objFileSysOb.GetDrive(vDrive).RootFolder
Set colSubFolders = colFolder.SubFolders
Response.Write "<B>The Root Subfolders Are:</B>"
For Each colSubFolderItem in colSubFolders
    Response.Write "Root SubFolder = " & colSubFolderItem & "<BR>"
Next

```



Slika 6.10

Podfolderi root foldera E: drajva

Takođe, zapazite da je u kodu bilo dimenzionisano nekoliko dodatnih promenljivih za rad sa dodatnim kolekcijama koje se koriste, kao što su Folders i SubFolders. Na slici 6.10 je prikazana lista podfoldera.

Prikaz informacija o folderu Svaki folder u kolekciji ima svoj sopstveni niz vrednosti podataka, uključujući i datum kada je bio kreiran, datum kada je bio zadnji put modifikovan, veličinu, tip, putanju i tako dalje. Ove vrednosti se mogu prikazati korišćenjem reference na podfolder i uobičajene For Each. . . Next petlje, kao što je prikazano na slici 6.11 i u sledećem kodu:

```

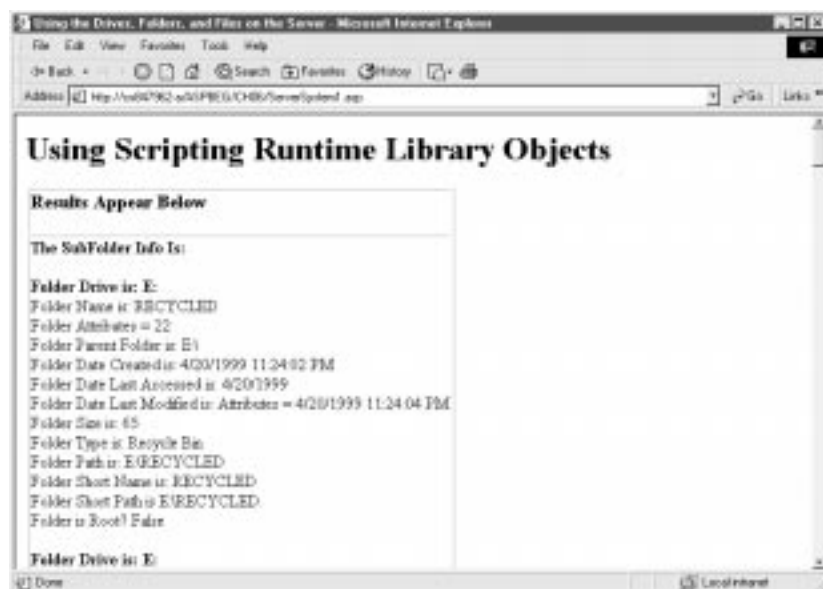
Dim vSubFolder
vDrive = Request.Form("drive")
vSubFolder = Request.Form("subfolder")

```

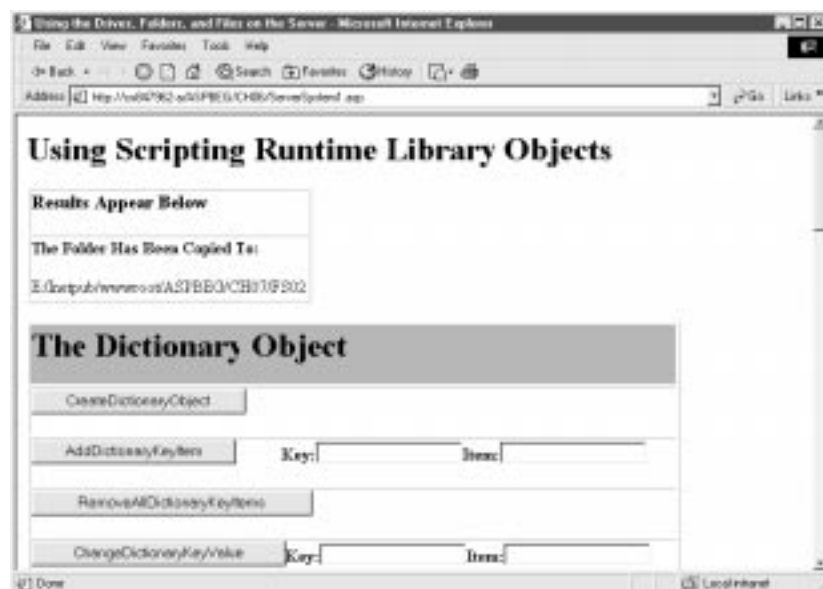
```
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Set colFolder = objFileSysOb.GetDrive(vDrive).RootFolder
Set colSubFolders = colFolder.SubFolders
Response.Write "<B>The SubFolder Info Is:</B><P>"
For Each colSubFolderItem in colSubFolders
    Response.Write "<B>Folder Drive is: " & colSubFolderItem.Drive &
"</B><BR>"
    Response.Write "Folder Name is: " & colSubFolderItem.Name & "<BR>"
    Response.Write "Folder Attributes = " & colSubFolderItem.Attributes &
"<BR>"
    Response.Write "Folder Parent Folder is: " &
colSubFolderItem.ParentFolder & "<BR>"
    Response.Write "Folder Date Created is: " &
colSubFolderItem.DateCreated & "<BR>"
    Response.Write "Folder Date Last Accessed is: " &
colSubFolderItem.DateLastAccessed & "<BR>"
    Response.Write "Folder Date Last Modified is: Attributes = " &
colSubFolderItem.DateLastModified & "<BR>"
    Response.Write "Folder Size is: " & colSubFolderItem.Size & "<BR>"
    Response.Write "Folder Type is: " & colSubFolderItem.Type & "<BR>"
    Response.Write "Folder Path is: " & colSubFolderItem.Path & "<BR>"
    Response.Write "Folder Short Name is: " &
colSubFolderItem.ShortName & "<BR>"
    Response.Write "Folder Short Path is " & colSubFolderItem.ShortPath
& "<BR>"
    Response.Write "Folder is Root? " & colSubFolderItem.IsRootFolder &
"<P>"
```

Kopiranje, pomeranje i brisanje foldera Kada posedujete referencu na određeni folder, možete da kopirate, pomerate i brišete foldere (pažljivo, naravno), kao što je prikazano u sledeća tri primera koda (i na slici 6.12):

```
Dim vSFolder
Dim vDFolder
Dim vOverwrite
Dim colFolderName
vSFolder = Request.Form("Sfoldername")
vDFolder = Request.Form("Dfoldername")
vOverwrite = Request.Form("overwrite")
Response.Write "<B>The Folder Has Been Copied To:</B><P>"
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Set colFolderName = objFileSysOb.GetFolder(vSFolder)
colFolderName.Copy vDFolder,vOverwrite
Response.Write vDFolder
```



Slika 6.11
Informacije o folderima za E: drajv



Slika 6.12
Kopiranje foldera

Zapazite da prvo dimenzionirate još neke promenljive za dobijanje izvornog foldera, odredišnog foldera i overwrite parametra od korisnika, i onda kreirate FileSystemObject objekat da biste stigli do korektnog objekta foldera (koji je određen u vSFolder). Poslednji korak je da se koristi Copy metod Folder objekta sa VDFolder i VOverwrite parametrima za obavljanje kopiranja i da se koristi Response objekat da se kaže korisniku šta je bilo iskopirano.

Kod za pomeranje foldera je sličan ovome:

```
vSFolder = Request.Form("Sfoldername")
vDFolder = Request.Form("Dfoldername")
Response.Write "<B>The Folder Has Been Moved To:</B><P>"
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Set colFolderName = objFileSysOb.GetFolder(vSFolder)
colFolderName.Move vDFolder
Response.Write vDFolder
```

Kod za brisanje foldera je takođe sličan ovome. Primarna razlika je dodatak force parametru (koji se takođe uzima od korisnika pomoću Request objekta i vForce promenljive):

```
Dim vForce
vSFolder = Request.Form("Sfoldername")
vForce = Request.Form("force")
Response.Write "<B>The Following Folder Has Been Deleted:</B><P>"
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Set colFolderName = objFileSysOb.GetFolder(vSFolder)
colFolderName.Delete vForce
Response.Write vSFolder
```

Prikazivanje fajlova u folderu Naravno, posedovanje referencije na folder je samo početna tačka. Sledeća očigledna stvar koju ćete želeći da uradite je da pristupite fajlovima u folderu. Jedan od prvih zahteva koji treba da budu ispunjeni da bi se ovo uradilo je da budete u mogućnosti da prikazete listu tih fajlova. Ovaj primer (koji je ilustrovan na slici 6.13) prikazuje kako možete prikazati listu svih fajlova u folderu, korišćenjem par promenljivih i standardne For Each ... Next petlje:

```
Dim vFileItem
Dim vFiles
vSFolder = Request.Form("Sfoldername")
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Set colFolderName = objFileSysOb.GetFolder(vSFolder)
Set vFiles = colFolderName.Files
Response.Write "<B>The Files In The Folder Are:</B><P>"
For Each vFileItem in vFiles
    Response.Write vFileItem & "<BR>"
Next
```

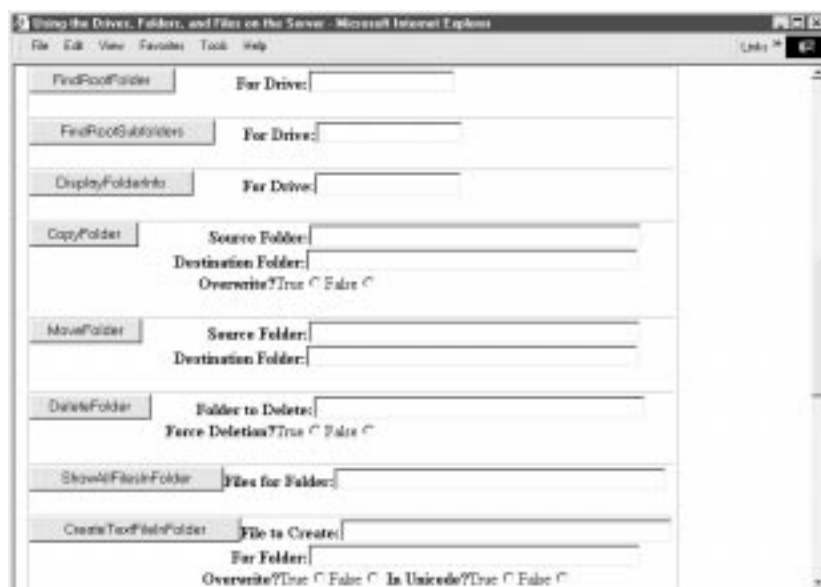
**Slika 6.13***Lista fajlova u folderu*

vFiles promenljiva je u stvari kolekcija fajlova koji se mogu navesti u listi korišćenjem For petlje, koja koristi Response objekat da bi ispisala ime svakog fajla.

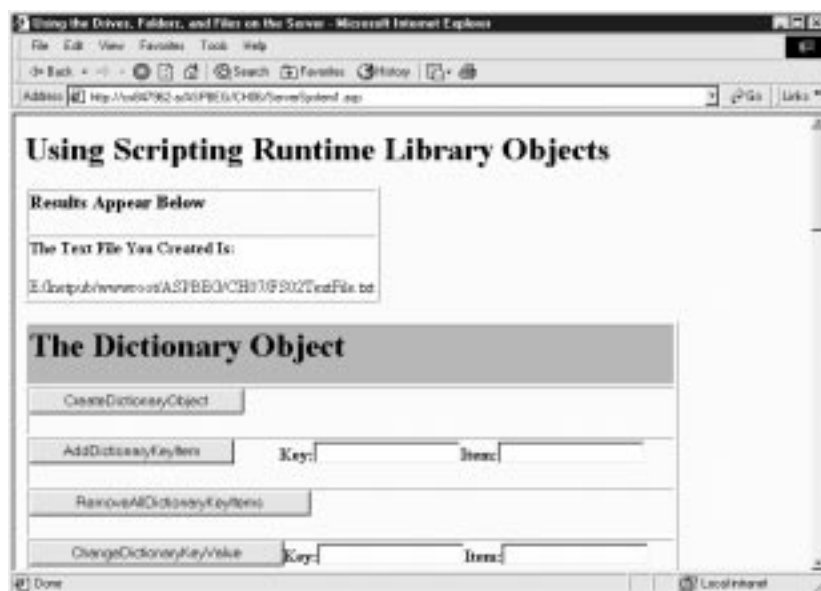
Kreiranje tekstualnih fajlova Na slici 6.14 je prikazan ostatak odeljka za foldere na ASP demonstracionoj stranici; i, kao što možete videti, obezbeđuje mogućnost za kreiranje tekstualnih fajlova. Da biste kreirali tekstualni fajl, treba da unesete korektno ime foldera u kome želite da kreirate fajl, i ime fajla koji želite da kreirate. Takođe, odlučite da li želite da izbrišete i zamenite postojeći fajl sa istim imenom, ili želite da fajl ima Unicode format.

Sledeći primer koda pokazuje kako se dobijaju te vrednosti od korisnika i kako se određuju za kreiranje tekstualnog fajla. Rezultati su prikazani na slici 6.15:

```
Dim vFilename
Dim vUnicode
vFilename = Request.Form("filename")
vSFolder = Request.Form("Sfoldername")
vOverwrite = Request.Form("overwrite")
vUnicode = Request.Form("unicode")
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Set colFolderName = objFileSysOb.GetFolder(vSFolder)
colFolderName.CreateTextFile vFilename,vOverwrite,vUnicode
Response.Write "<B>The Text File You Created Is:</B><P>"
Response.Write vSFolder & vFilename
```



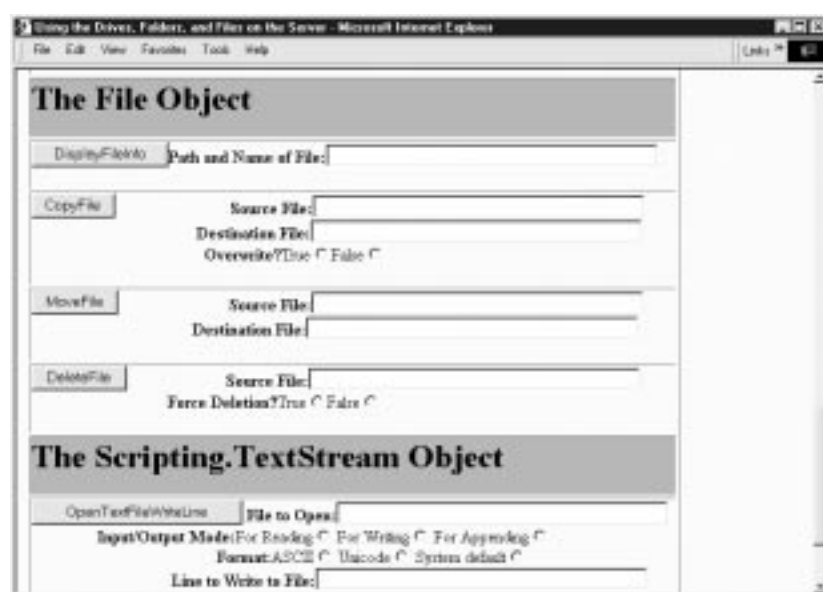
Slika 6.14
Ostatak kontrola za deo za foldere



Slika 6.15
Kreiranje tekstualnog fajla

File objekat

File objekat je pretposlednji objekat u hijerarhiji objekata fajl sistema, i ima osobine i metode koje su prilično slične osobinama i metodama *Folder* objekta. Oblast na ASP stranici koja služi za kreiranje i manipulaciju *File* objektom je prikazana na slici 6.16. Možete prikazati informacije o fajlu kao što su datum kreiranja i datum poslednje modifikacije, veličina i tako dalje; i možete kopirati, pomerati i brisati fajlove. Takođe možete kreirati tekstualne fajlove ili ih možete otvoriti kao *TextStream* objekte.



Slika 6.16

Odeljak za File objekat

Osobine File objekta

File objekat ima sledeće osobine:

- **Attributes** Ova osobina vraća attribute kao što su normal (normalan), read-only (samo za čitanje), hidden (sakriveni), system (sistemski), volume (volumen), directory (direktorijum), archive (arhiva), alias (drugo ime) i compressed (kompresovani).
- **DateCreated** Ova osobina vraća datum i vreme kada je fajl bio kreiran.
- **DateLastAccessed** Ova osobina vraća datum i vreme kada se zadnji put pristupilo fajlu.
- **DateLastModified** Ova osobina vraća datum i vreme kada je fajl bio zadnji put modifikovan.
- **Drive** Ova osobina vraća slovo drajva za fajl.
- **Drive** Ova osobina vraća slovo drajva za fajl.

- **Name** Ova osobina podešava ili vraća ime fajla.
- **ParentFolder** Ova osobina vraća Folder objekat koji je ekvivalentan roditeljskom folderu za tekući fajl.
- **Path** Ova osobina vraća apsolutnu putanju za tekući fajl.
- **ShortName** Ova osobina vraća DOS verziju imena fajla.
- **ShortPath** Ova osobina vraća DOS verziju imena putanje za fajl.
- **Size** Ova osobina vraća veličinu fajla u bajtovima.
- **Type** Ova osobina vraća opis tekućeg fajla kao string, ako je opis dostupan.

Metode File objekta

File objekat takođe ima metode koje su slične metodama Folder objekta, kao što je navedeno u sledećoj listi:

- **Copy(odredište, overwrite)** Ovaj metod kopira tekući fajl na lokaciju koja je određena kao odredište, pri čemu se briše i zamenjuje postojeći fajl sa istim imenom osim ako overwrite parametar nije postavljen na True.
- **Delete(force)** Ovaj metod briše tekući fajl i njegov sadržaj; i, ako je force parametar postavljen na True, briše čak i fajlove koji se mogu samo čitati.
- **Move(odredište)** Ovaj metod pomera tekući fajl na lokaciju koja je određena kao odredište, ali generiše grešku ako već postoji fajl sa istim imenom na toj lokaciji.
- **CreateTextFile(putanjafajla, overwrite, unicode)** Ovaj metod kreira tekstualni fajl korišćenjem određene putanje fajla i imena fajla, pri čemu briše i zamenjuje postojeće fajlove sa tim imenom ako je overwrite parametar podešen na True i u Unicode formatu ako je unicode parametar podešen na True.
- **OpenAsTextStream(iomod, format)** Ovaj metod otvara fajl u određenom modu (čitanje, upis, pripajanje) u zavisnosti od vrednosti iomod parametra, i kao ASCII, Unicode ili sistemski podrazumevani fajl, u zavisnosti od vrednosti format parametra.

Prikaz informacija o fajlu Jednom kada dobijete referencu na fajl, možete prikazati podatke o tom fajlu pomoću koda, koji izgleda ovako:

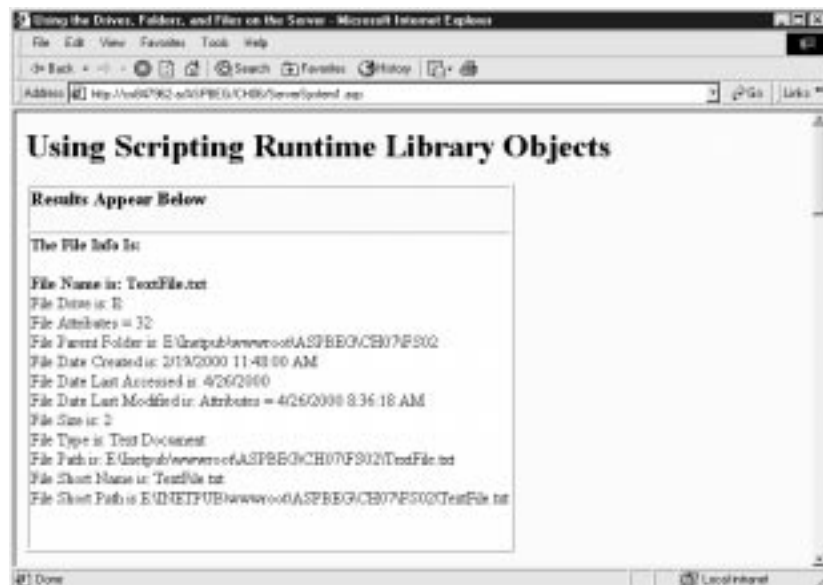
```
Dim vFilePathname
Dim colFileName
vFilePathname = Request.Form("filepathname")
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Set colFileName = objFileSysOb.GetFile(vFilePathname)
Response.Write "<B>The File Info Is:</B><P>"
Response.Write "<B>File Name is: " & colFileName.Name & "</B><BR>"
Response.Write "File Drive is: " & colFileName.Drive & "<BR>"
Response.Write "File Attributes = " & colFileName.Attributes & "<BR>"
Response.Write "File Parent Folder is: " & colFileName.ParentFolder & "<BR>"
Response.Write "File Date Created is: " & colFileName.DateCreated & "<BR>"
Response.Write "File Date Last Accessed is: " & colFileName.DateLastAccessed & "<BR>"
```

```

Response.Write "File Date Last Modified is: Attributes = " &
colFileName.DateLastModified & "<BR>"
Response.Write "File Size is: " & colFileName.Size & "<BR>"
Response.Write "File Type is: " & colFileName.Type & "<BR>"
Response.Write "File Path is: " & colFileName.Path & "<BR>"
Response.Write "File Short Name is: " & colFileName.ShortName & "<BR>"
Response.Write "File Short Path is " & colFileName.ShortPath & "<P>"

```

U ovom kodu, uzimate fajl tako što počinjete od FileSystemObject objekta, tako da korisnik mora obezbediti celokupnu putanju do fajla. Jednom kada budete imali referencu na fajl, možete koristiti Response objekat i osobine File objekta da prikazete svaku individualnu osobinu fajla, kao što je prikazano na slici 6.17.



Slika 6.17

Prikazane informacije o fajlu

Kopiranje, pomeranje i brisanje fajlova Fajlovi se mogu kopirati, pomerati i brisati baš kao i folderi, i sledeći primeri koda demonstriraju kako se to radi, počev od kopiranja fajla:

```

Dim vSFile
Dim vDFile
vSFile = Request.Form("Sfilename")
vDFile = Request.Form("Dfilename")
vOverwrite = Request.Form("overwrite")
Response.Write "<B>The File Has Been Copied To:</B><P>"
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")

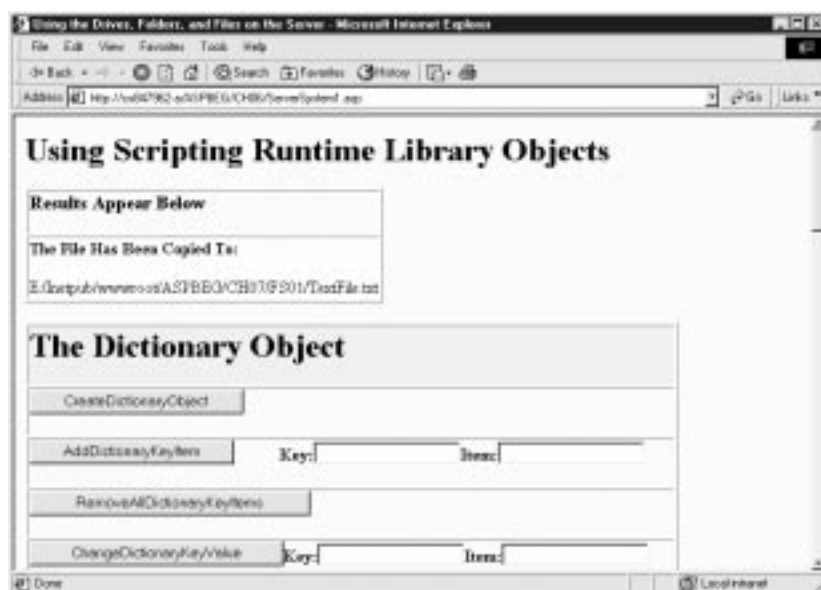
```

```

Set colFileName = objFileSysOb.GetFile(vSFile)
colFileName.Copy vDFile,vOverwrite
Response.Write vDFile

```

Treba da znate ime fajla koji treba iskopirati, gde ga treba iskopirati, i da li da se izbriše i zameni već postojeći fajl sa istim imenom; pa prvo, uzimate ove vrednosti od korisnika, i onda ih koristite sa File objektom i Copy metodom da biste obavili taj posao. Na slici 6.18 je prikazano kako ovo radi.



Slika 6.18
Kopiranje fajla

Za pomeranje fajla, kod je sličan:

```

vSFile = Request.Form("Sfilename")
vDFile = Request.Form("Dfilename")
Response.Write "<B>The File Has Been Moved To:</B><P>"
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Set colFileName = objFileSysOb.GetFile(vSFile)
colFileName.Move vDFile
Response.Write vDFile

```

Za brisanje fajlova, možete koristiti sledeći kod:

```

vSFile = Request.Form("Sfilename")
vForce = Request.Form("force")
Response.Write "<B>The Following File Has Been Deleted:</B><P>"
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Set colFileName = objFileSysOb.GetFile(vSFile)

```

```
colFileName.Delete vForce
Response.Write vSFile
```

Ovde dobijate File objekat iz FileSystemObject objekta i uzimate ime fajla koji treba izbrisati od korisnika, kao i vrednost sila parametra. Onda jednostavno obavite ovu akciju nad File objektom.

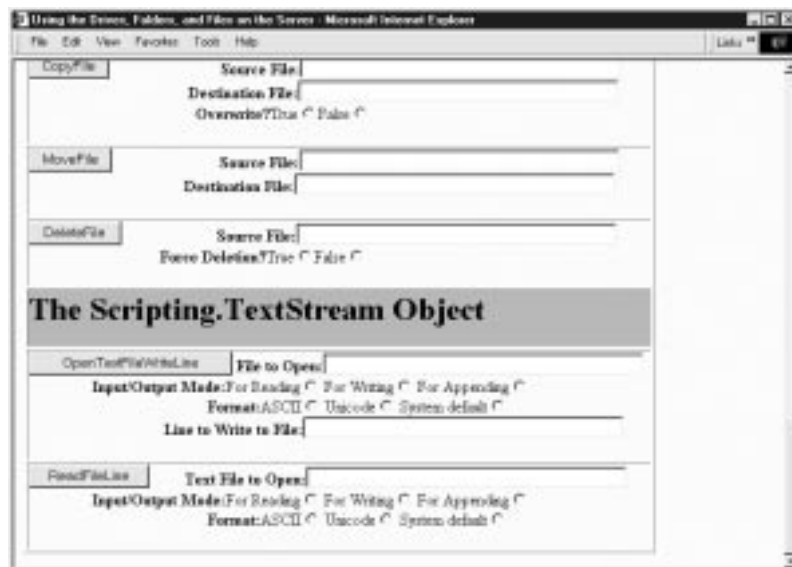
Scripting.TextStream objekat

Iako su TextStream objekti posebni entiteti, oni se mogu kreirati samo korišćenjem CreateTextFile, OpenTextFile i OpenAsTextStream metoda FileSystemObject objekta, Folder objekta i File objekta. Bilo koji od ovih objekata može koristiti CreateTextFile, dok jedino FileSystemObject objekat može koristiti OpenTextFile metod i samo File objekat može koristiti OpenAsTextStream metod. Na slici 6.19 je prikazan deo za TextStream objekat na Vašoj ASP stranici.

Osobine TextStream objekta

TextStream objekat predstavlja sledeće osobine kojima nam govori o sebi:

- **AtEndOfLine** Ova osobina vraća True ako je pokazivač na fajl na kraju linije u fajlu.
- **AtEndOfStream** Ova osobina vraća True ako je pokazivač na fajl na kraju fajla.



Slika 6.19

Odeljak za TextStream objekat

- **Column** Ova osobina vraća broj kolone tekućeg karaktera.
- **Line** Ova osobina vraća broj linije za tekuću liniju.

Metode TextStream objekta

TextStream objekat obezbeđuje metode za navigaciju fajlovima, kao i za čitanje i upis u fajlove. To su

- **Close** Ovaj metod zatvara tekući fajl.
- **Read (broj_karaktera)** Ovaj metod čita određeni broj karaktera iz fajla.
- **ReadAll** Ovaj metod čita sve karaktere u fajlu.
- **ReadLine** Ovaj metod čita liniju iz fajla u vidu stringa.
- **Skip(broj_karaktera)** Ovaj metod preskače određeni broj karaktera pri čitanju.
- **SkipLine** Ovaj metod preskače sledeću liniju pri čitanju.
- **Write(string)** Ovaj metod upisuje string u fajl.
- **WriteLine** Ovaj metod upisuje string (ako je određen) i karakter za novu liniju u fajl.
- **WriteBlankLines (broj)** Ovaj metod upisuje određeni broj karaktera za novu liniju u fajl.

Kreiranje fajla i upis u fajl File objekat se može koristiti za kreiranje novog fajla kao TextStream objekta, i onda se TextStream objektom može manipulirati pomoću WriteLine metoda za umetanje linije u okviru tog novog fajla, sve odjednom, kao što je prikazano u sledećem kodu (i na slici 6.20).

```
Dim vIomode
Dim vFormat
Dim vLinetowrite
Dim objTextStream
vSFile = Request.Form("Sfilename")
vIomode = Request.Form("iomode")
vFormat = Request.Form("format")
vLinetowrite = Request.Form("linetowrite")
Response.Write "<B>The Line You Have Written Is:</B><P>"
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Set colFileName = objFileSysOb.GetFile(vSFile)
Set objTextStream = colFileName.OpenAsTextStream(vIomode, vFormat)
objTextStream.WriteLine vLinetowrite
Response.Write vLinetowrite
```

I ovde uzimate ime fajla koji želite da kreirate, njegov io mod i format od korisnika preko nekih promenljivih koje dimenzionišete, i onda dobijate File objekat iz FileSystemObject objekta i kreirate novi tekstualni fajl kao TextStream objekat pomoću OpenAsTextStream metode. Konačno, upisujete liniju u novi fajl pomoću WriteLine metode.

Čitanje linije iz fajla Da biste pročitali liniju iz fajla, sve što treba da uradite je da dobijete referencu na taj fajl, i da onda koristite ReadLine metod za dobijanje linije i za prikazivanje te linije korisniku. Naravno, treba da pitate kako da se otvori fajl; ali, u praksi, to možda nećete ostaviti korisniku da odluči.

**Slika 6.20**

Kreiranje TextStream objekta i upis linije

Ovde, je radi demonstracije prikazano kako se to može uraditi. Na slici 6.21 su prikazani rezultati:

```
Dim vReadline
vSFile = Request.Form("Sfilename")
vIomode = Request.Form("iomode")
vFormat = Request.Form("format")
Response.Write "<B>The First Line in the File Is:</B><P>"
Set objFileSysOb = Server.CreateObject("Scripting.FileSystemObject")
Set colFileName = objFileSysOb.GetFile(vSFile)
Set objTextStream = colFileName.OpenAsTextStream(vIomode, vFormat)
vReadline = objTextStream.ReadLine
Response.Write vReadline
```



Slika 6.21

Čitanje linije iz fajla

Pitajte stručnjaka

Pitanje: Čuo sam za modele objekata i ranije, ali voleo bih da mi bolje objasnite šta su oni i za šta se primenjuju. Da li mogu kreirati svoj sopstveni model objekta? Pod kojim okolnostima su oni korisni i kako mi mogu pomoći?

Odgovor: Modeli objekata su prvi i najistaknutiji modeli, kao i bilo koja druga vrsta modela. Na najosnovnijem nivou, svi modeli su konceptualne reprezentacije stvari, gde su neki modeli realniji od drugih. Na primer, model železnice sadrži mala vozila za železnicu, ljude, šine i tako dalje, i svi oni su povezani zajedno u reprezentaciji prave železnice.

U programiranju, modeli se često koriste za predstavljanje toga kako su ideje i strukture povezane jedne sa drugima. Odnos podataka iz baze podataka u tabelama se može predstaviti preko linija koje idu od jedne tabele do druge (primere ovoga ćete videti u sledećim poglavljima). Za Scripting Library objekte, odnosi su relativno jednostavni-primarno se odnose na to koje objekte mogu kreirati i do kojih drugih objekata mogu izvršiti navigaciju, i koje kolekcije imaju.

Pitanje: FileSystemObject objekat izgleda da poseduje mnoge mogućnosti programa koji je istog tipa kao File Manager program. Da li bih mogao da razvijem svoj sopstveni FileSystemObject objekat pomoću ASP? Ako se ja ionako nalazim na serveru, zašto su mi potrebne te mogućnosti u ASP-u?

Odgovor: Razvijanje onlajn aplikacija za klijente je uobičajen posao, i takođe je uobičajeno da klijenti žele da budu automatizovane neke funkcije upravljačkog tipa. Ovo znači da ćete verovatno razviti niz funkcija za korisnike i takođe drugi niz funkcija za zaposlene klijenta koji će upravljati aplikacijom.

U okviru Vaših korisničkih funkcija, FileSystem mogućnosti će verovatno biti sakrivene ili čvrsto kontrolisane, da bi sprečile korisnike za unište Vaš server. U okviru upravljačkih funkcija za zaposlene klijenta, FileSystem mogućnosti će verovatno biti fleksibilniji, sa manje kontrole, mada ćete verovatno ugraditi sasvim dovoljnu kontrolu da sprečite uništenje Vašeg servera osim ako se zaposleni klijenta prvo ne upozori.

U bilo kom slučaju, veći deo korišćenja i upravljanja Vašom aplikacijom će se najverovatnije odvijati onlajn, a ne direktno na serveru. Zato Vam je potrebna mogućnost da uđete na server i da radite sa fajl sistemom pomoću pretraživača.

Projekat 6.1: Objedinjavanje ovih primera

U sledećem primeru, svi primeri koda koji su se koristili u ovom poglavlju se objedinjavaju i prikazuju kao jedan primer potpune stranice koja bi se mogla koristiti da sa daljine obavlja tipične funkcije za upravljanje fajlovima. Svaki objekat ili komponenta koji je kreiran i kojim se manipuliše ima svoj sopstveni odeljak na stranici; i za sve što se radi, stranica referencira samu sebe. Zapazite, da biste FileSystemObject koristili efikasno, morate obezbediti referencu na FileSystemObject biblioteku tipova koja se takođe nalazi blizu vrha Vašeg skripta, i to korišćenjem METADATA oznake. Korišćenje METADATA oznake za referenciranje biblioteka tipova za COM objekte se detaljnije diskutuje u kasnijim poglavljima.

Korak po korak

1. Prvi niz primera za ovu stranicu radi sa Dictionary objektom. Iako možete kreirati Dictionary objekat sa dosegom na nivou stranice, za nekoliko od ovih primera Vam je potreban Dictionary objekat sa dosegom na nivou sesije. Morate se setiti da dodate objekat pomoću global.asa fajla, kao što smo objasnili ranije u ovom poglavlju.
2. Uvek je dobro da detaljno dokumentujete i unesete komentare u Vaš kod, ali ovde sam izostavio komentare zato što su prilično dobri komentari obezbeđeni u tekstu poglavlja i dalji komentari bi napravili zbrku u kodu.
3. Celokupni kod za primere koje smo koristili u ovom poglavlju, kao i HTML koji proizvodi korisnički interfejs, izgleda ovako:

```
<%
Option Explicit
Dim btnClicked
%>
<!-- METADATA TYPE="typelib" FILE="E:\WINNT\System32\scrrun.dll" -->
<html><head>
<title>Using the Drives, Folders, and Files on the Server</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#77FF77">
<H2>Using Scripting Runtime Library Objects</H2>
```

```
<TABLE border=1>
<TR><TD><H3>Results Appear Below</H3></TD></TR><TR><TD bgcolor="white">
<%
btnClicked = Request.Form("btnval")
Select Case btnClicked
  Case "CreateDictionaryObject"
%><OBJECT RUNAT="SERVER" SCOPE="PAGE" ID="objDictionary"
PROGID="Scripting.Dictionary"></OBJECT>
<%
Response.Write "<B>The objDictionary Dictionary Object Has Been Created and
Contains " & objDictionary.Count & " Keys</B>"
Case "AddDictionaryKeyItem"
Dim vKey
    Dim vItem
    Dim arKeys
    Dim arItems
    Dim intI
    Dim objDictionaryS
    Set objDictionaryS = Session("SessionDictionary")
    vKey = Request.Form("key")
    vItem = Request.Form("item")
    objDictionaryS.Add vKey, vItem
    arKeys = objDictionaryS.Keys
    arItems = objDictionaryS.Items
    Response.Write "<B>Session Dictionary Object Created</B><BR>"
    Response.Write "<B>The Dictionary Key/Item Pairs You Added
Are:</B><P>"
        For intI = 0 to objDictionaryS.Count - 1
            Response.Write "<B>Key = </B>" & arKeys (intI)
            Response.Write " and <B>Item = </B>" & arItems (intI) &
"<BR>"
        Next
    Case "RemoveDictionaryKeyItem"
        Set objDictionaryS = Session("SessionDictionary")
        Response.Write "<B>Removing Dictionary Key/Items:</B>"
        objDictionaryS.RemoveAll
        Set Session("SessionDictionary") = objDictionaryS
    Case "ChangeDictionaryKeyValue"
        Set objDictionaryS = Session("SessionDictionary")
        vKey = Request.Form("key")
        vItem = Request.Form("item")
        objDictionaryS.Item(vKey) = vItem
        Set Session("SessionDictionary") = objDictionaryS
        Response.Write "<B>Dictionary Key Item Value Has Been Changed
To:</B><P>"
        arKeys = objDictionaryS.Keys
```

```

arItems = objDictionaryS.Items
For intI = 0 to objDictionaryS.Count - 1
    Response.Write "<B>Key = </B>" & arKeys (intI)
    Response.Write " and <B>Item = </B>" & arItems (intI) &
" <BR>"
Next
Case "CreateFileSystemObject"
    Dim objFileSysOb
    Dim collObj
    Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
    Response.Write "<B>The FileSystemObject Has Been Created and the
Server Drives Are:</B>"
    For Each collObj in objFileSysOb.Drives
        Response.Write "<BR>Drive = " & collObj
    Next
Case "DisplayDriveInfo"
    Dim vDrive
    vDrive = Request.Form("drive")
    Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
    Response.Write "<B>The Drive Info Is:</B><P>"
    Response.Write "Drive Letter = " &
objFileSysOb.GetDrive(vDrive).DriveLetter & "<BR>"
    Response.Write "Drive Type = " &
objFileSysOb.GetDrive(vDrive).DriveType & "<BR>"
    Response.Write "File System = " &
objFileSysOb.GetDrive(vDrive).FileSystem & "<BR>"
    Response.Write "AvailableSpace = " &
objFileSysOb.GetDrive(vDrive).AvailableSpace & "<BR>"
    Response.Write "Free Space = " &
objFileSysOb.GetDrive(vDrive).FreeSpace & "<BR>"
    Response.Write "Is Ready = " &
objFileSysOb.GetDrive(vDrive).IsReady & "<BR>"
    Response.Write "Path = " & objFileSysOb.GetDrive(vDrive).Path &
"<BR>"
    Response.Write "Root Folder = " &
objFileSysOb.GetDrive(vDrive).RootFolder & "<BR>"
    Response.Write "Serial Number = " &
objFileSysOb.GetDrive(vDrive).SerialNumber & "<BR>"
    Response.Write "Share Name = " &
objFileSysOb.GetDrive(vDrive).ShareName & "<BR>"
    Response.Write "Total Size = " &
objFileSysOb.GetDrive(vDrive).TotalSize & "<BR>"
    Response.Write "Volume Name = " &
objFileSysOb.GetDrive(vDrive).VolumeName & "<BR>"

```

```
Case "FindRootFolder"
    vDrive = Request.Form("drive")
    Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
    Response.Write "<B>The Root Folder Is:</B><P>"
    Response.Write "Root Folder = " &
objFileSysOb.GetDrive(vDrive).RootFolder & "<BR>"
    Case "FindRootSubfolders"
        Dim colFolder
        Dim colSubFolders
        Dim colSubFolderItem
        vDrive = Request.Form("drive")
        Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
        Set colFolder = objFileSysOb.GetDrive(vDrive).RootFolder
        Set colSubFolders = colFolder.SubFolders
        Response.Write "<B>The Root Subfolders Are:</B>"
        For Each colSubFolderItem in colSubFolders
            Response.Write "Root SubFolder = " & colSubFolderItem &
"<BR>"
        Next
    Case "DisplayFolderInfo"
        Dim vSubFolder
        vDrive = Request.Form("drive")
        vSubFolder = Request.Form("subfolder")
        Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
        Set colFolder = objFileSysOb.GetDrive(vDrive).RootFolder
        Set colSubFolders = colFolder.SubFolders
        Response.Write "<B>The SubFolder Info Is:</B><P>"
        For Each colSubFolderItem in colSubFolders
            Response.Write "<B>Folder Drive is: " &
colSubFolderItem.Drive & "</B><BR>"
            Response.Write "Folder Name is: " & colSubFolderItem.Name
& "<BR>"
            Response.Write "Folder Attributes = " &
colSubFolderItem.Attributes & "<BR>"
            Response.Write "Folder Parent Folder is: " &
colSubFolderItem.ParentFolder & "<BR>"
            Response.Write "Folder Date Created is: " &
colSubFolderItem.DateCreated & "<BR>"
            Response.Write "Folder Date Last Accessed is: " &
colSubFolderItem.DateLastAccessed & "<BR>"
            Response.Write "Folder Date Last Modified is: Attributes =
" & colSubFolderItem.DateLastModified & "<BR>"
            Response.Write "Folder Size is: " & colSubFolderItem.Size
```

```

& "<BR>"
        Response.Write "Folder Type is: " & colSubFolderItem.Type
& "<BR>"
        Response.Write "Folder Path is: " & colSubFolderItem.Path
& "<BR>"
        Response.Write "Folder Short Name is: " &
colSubFolderItem.ShortName & "<BR>"
        Response.Write "Folder Short Path is " &
colSubFolderItem.ShortPath & "<BR>"
        Response.Write "Folder is Root? " &
colSubFolderItem.IsRootFolder & "<P>"
    Next
    Case "CopyFolder"
        Dim vSFolder
        Dim vDFolder
        Dim vOverwrite
        Dim colFolderName
        vSFolder = Request.Form("Sfoldername")
        vDFolder = Request.Form("Dfoldername")
        vOverwrite = Request.Form("overwrite")
        Response.Write "<B>The Folder Has Been Copied To:</B><P>"
        Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
        Set colFolderName = objFileSysOb.GetFolder(vSFolder)
        colFolderName.Copy vDFolder,vOverwrite
        Response.Write vDFolder
    Case "MoveFolder"
        vSFolder = Request.Form("Sfoldername")
        vDFolder = Request.Form("Dfoldername")
        Response.Write "<B>The Folder Has Been Moved To:</B><P>"
        Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
        Set colFolderName = objFileSysOb.GetFolder(vSFolder)
        colFolderName.Move vDFolder
        Response.Write vDFolder
    Case "DeleteFolder"
        Dim vForce
        vSFolder = Request.Form("Sfoldername")
        vForce = Request.Form("force")
        Response.Write "<B>The Following Folder Has Been
Deleted:</B><P>"
        Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
        Set colFolderName = objFileSysOb.GetFolder(vSFolder)
        colFolderName.Delete vForce
        Response.Write vSFolder

```

```
Case "ShowAllFilesInFolder"
    Dim vFileItem
    Dim vFiles
    vSFolder = Request.Form("Sfoldername")
    Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
    Set colFolderName = objFileSysOb.GetFolder(vSFolder)
    Set vFiles = colFolderName.Files
    Response.Write "<B>The Files In The Folder Are:</B><P>"
    For Each vFileItem in vFiles
        Response.Write vFileItem & "<BR>"
    Next
Case "CreateTextFileInFolder"
    Dim vFilename
    Dim vUnicode
    vFilename = Request.Form("filename")
    vSFolder = Request.Form("Sfoldername")
    vOverwrite = Request.Form("overwrite")
    vUnicode = Request.Form("unicode")
    Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
    Set colFolderName = objFileSysOb.GetFolder(vSFolder)
    colFolderName.CreateTextFile vFilename,vOverwrite,vUnicode
    Response.Write "<B>The Text File You Created Is:</B><P>"
    Response.Write vSFolder & vFilename
Case "DisplayFileInfo"
    Dim vFilePathname
    Dim colFileName
    vFilePathname = Request.Form("filepathname")
    Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
    Set colFileName = objFileSysOb.GetFile(vFilePathname)
    Response.Write "<B>The File Info Is:</B><P>"
    Response.Write "<B>File Name is: " & colFileName.Name &
"</B><BR>"
    Response.Write "File Drive is: " & colFileName.Drive & "<BR>"
    Response.Write "File Attributes = " & colFileName.Attributes &
"<BR>"
    Response.Write "File Parent Folder is: " &
colFileName.ParentFolder & "<BR>"
    Response.Write "File Date Created is: " &
colFileName.DateCreated & "<BR>"
    Response.Write "File Date Last Accessed is: " &
colFileName.DateLastAccessed & "<BR>"
    Response.Write "File Date Last Modified is: Attributes = " &
colFileName.DateLastModified & "<BR>"
```

```

        Response.Write "File Size is: " & colFileName.Size & "<BR>"
        Response.Write "File Type is: " & colFileName.Type & "<BR>"
        Response.Write "File Path is: " & colFileName.Path & "<BR>"
        Response.Write "File Short Name is: " & colFileName.ShortName &
"<BR>"
        Response.Write "File Short Path is " & colFileName.ShortPath &
"<P>"
        Case "CopyFile"
            Dim vSFile
            Dim vDFile
            vSFile = Request.Form("Sfilename")
            vDFile = Request.Form("Dfilename")
            vOverwrite = Request.Form("overwrite")
            Response.Write "<B>The File Has Been Copied To:</B><P>"
            Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
            Set colFileName = objFileSysOb.GetFile(vSFile)
            colFileName.Copy vDFile,vOverwrite
            Response.Write vDFile
        Case "MoveFile"
            vSFile = Request.Form("Sfilename")
            vDFile = Request.Form("Dfilename")
            Response.Write "<B>The File Has Been Moved To:</B><P>"
            Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
            Set colFileName = objFileSysOb.GetFile(vSFile)
            colFileName.Move vDFile
            Response.Write vDFile
        Case "DeleteFile"
            vSFile = Request.Form("Sfilename")
            vForce = Request.Form("force")
            Response.Write "<B>The Following File Has Been Deleted:</B><P>"
            Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
            Set colFileName = objFileSysOb.GetFile(vSFile)
            colFileName.Delete vForce
            Response.Write vSFile
        Case "OpenTextFile"
            Dim vIomode
            Dim vFormat
            Dim vLinetowrite
            Dim objTextStream
            vSFile = Request.Form("Sfilename")
            vIomode = Request.Form("iomode")
            vFormat = Request.Form("format")
            vLinetowrite = Request.Form("linetowrite")

```

```

        Response.Write "<B>The Line You Have Written Is:</B><P>"
        Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
        Set colFileName = objFileSysOb.GetFile(vSFile)
        Set objTextStream = colFileName.OpenAsTextStream(vIomode,
vFormat)

        objTextStream.WriteLine vLinetowrite
        Response.Write vLinetowrite
    Case "ReadFileLine"
        Dim vReadline
        vSFile = Request.Form("Sfilename")
        vIomode = Request.Form("iomode")
        vFormat = Request.Form("format")
        Response.Write "<B>The First Line in the File Is:</B><P>"
        Set objFileSysOb =
Server.CreateObject("Scripting.FileSystemObject")
        Set colFileName = objFileSysOb.GetFile(vSFile)
        Set objTextStream = colFileName.OpenAsTextStream(vIomode,
vFormat)

        vReadline = objTextStream.ReadLine
        Response.Write vReadline
End Select
%>
<BR></TD></TR></TABLE>
<P>
<TABLE border=1><TR><TD>
<H3>The Dictionary Object</H3></TD></TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp"><INPUT TYPE="submit"
NAME="btnval" VALUE="CreateDictionaryObject">
<B>Creates a Dictionary object with page level scope.</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>Key:</B><INPUT TYPE="text" NAME="key"><B>Item:</B><INPUT TYPE="text"
NAME="item"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="AddDictionaryKeyItem">
<B> Adds a key (name) and item (value) to the Dictionary object and lists
them.</B>
</FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp"><INPUT TYPE="submit"
NAME="btnval" VALUE="RemoveDictionaryKeyItem">
<B>Removes a key/item pair.</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>Key:</B><INPUT TYPE="text" NAME="key"><B>Item:</B><INPUT TYPE="text"
NAME="item"><BR>

```

The HTML page actually
displayed begins here.


```

<INPUT TYPE="submit" NAME="btnval" VALUE="ChangeDictionaryKeyValue">
<B>Changes the value of an Item for a Key.</B></FORM></TD>
</TR>
<TR><TD>
<H3>The Scripting.FileSystemObject Object</H3></TD></TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp"><INPUT TYPE="submit"
NAME="btnval" VALUE="CreateFileSystemObject">
<B>Creates a FileSystemObject object with page level scope and lists the
Server Drives</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>For Drive:</B><INPUT TYPE="text" NAME="drive"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="DisplayDriveInfo">
<B>Displays information about a specific drive on the server</B></FORM></TD>
</TR>
<TR><TD>
<H3>The Folder Object</H3></TD></TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>For Drive:</B><INPUT TYPE="text" NAME="drive"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="FindRootFolder">
<B>Finds the root folder for a specific drive</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>For Drive:</B><INPUT TYPE="text" NAME="drive"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="FindRootSubfolders">
<B>Shows the subfolders of the root folder</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>For Drive:</B><INPUT TYPE="text" NAME="drive"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="DisplayFolderInfo">
<B>Displays information about subfolders</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>Source Folder:</B><INPUT TYPE="text" SIZE=50 NAME="Sfoldername"><BR>
<B>Destination Folder:</B><INPUT TYPE="text" SIZE=50 NAME="Dfoldername"><BR>
<B>Overwrite?</B><True<INPUT TYPE="radio" NAME="overwrite"
VALUE="True">False<INPUT TYPE="radio" NAME="overwrite" VALUE="False"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="CopyFolder">
<B>Copies a folder from one location to another</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>Source Folder:</B><INPUT TYPE="text" SIZE=50 NAME="Sfoldername"><BR>
<B>Destination Folder:</B><INPUT TYPE="text" SIZE=50 NAME="Dfoldername"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="MoveFolder">
<B>Moves a folder from one location to another</B></FORM></TD>
</TR><TR>

```

```
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>Folder to Delete:</B><INPUT TYPE="text" SIZE=50 NAME="Sfoldername"><BR>
<B>Force Deletion?</B><True<INPUT TYPE="radio" NAME="force"
VALUE="True">False<INPUT TYPE="radio" NAME="force" VALUE="False"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="DeleteFolder">
<B>Deletes a folder</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>Files for Folder:</B><INPUT TYPE="text" SIZE=50 NAME="Sfoldername"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="ShowAllFilesInFolder">
<B>Shows all the files in a folder</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>Name of File to Create:</B><INPUT TYPE="text" SIZE=50
NAME="filename"><BR>
<B>For Folder:</B><INPUT TYPE="text" SIZE=50 NAME="Sfoldername"><BR>
<B>Overwrite?</B><True<INPUT TYPE="radio" NAME="overwrite"
VALUE="True">False<INPUT TYPE="radio" NAME="overwrite" VALUE="False"><BR>
<B>In Unicode?</B><True<INPUT TYPE="radio" NAME="unicode"
VALUE="True">False<INPUT TYPE="radio" NAME="unicode" VALUE="False"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="CreateTextFileInFolder">
<B>Creates a text file within a specified folder</B></FORM></TD>
</TR>
<TR><TD>
<H3>The File Object</H3></TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>Path and Name of File:</B><INPUT TYPE="text" SIZE=50 NAME="filepath-
name"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="DisplayFileInfo">
<B>Displays information about a file</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>Source File:</B><INPUT TYPE="text" SIZE=50 NAME="Sfilename"><BR>
<B>Destination File:</B><INPUT TYPE="text" SIZE=50 NAME="Dfilename"><BR>
<B>Overwrite?</B><True<INPUT TYPE="radio" NAME="overwrite"
VALUE="True">False<INPUT TYPE="radio" NAME="overwrite" VALUE="False"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="CopyFile">
<B>Copies a file to another location</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>Source File:</B><INPUT TYPE="text" SIZE=50 NAME="Sfilename"><BR>
<B>Destination File:</B><INPUT TYPE="text" SIZE=50 NAME="Dfilename"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="MoveFile">
<B>Moves a file to another location</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
```

```

<B>Source File:</B><INPUT TYPE="text" SIZE=50 NAME="Sfilename"><BR>
<B>Force Deletion?</B><INPUT TYPE="radio" NAME="force"
VALUE="True">False<INPUT TYPE="radio" NAME="force" VALUE="False"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="DeleteFile">
<B>Deletes a file</B></FORM></TD>
</TR>
<TR><TD>
<H3>The Scripting.TextStream Object</H3></TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>Text File to Open:</B><INPUT TYPE="text" SIZE=50 NAME="Sfilename"><BR>
<B>Input/Output Mode:</B>For Reading<INPUT TYPE="radio" NAME="iomode"
VALUE="1">
For Writing<INPUT TYPE="radio" NAME="iomode" VALUE="2">
For Appending<INPUT TYPE="radio" NAME="iomode" VALUE="8"><BR>
<B>Format:</B>ASCII<INPUT TYPE="radio" NAME="format" VALUE="0">
Unicode<INPUT TYPE="radio" NAME="format" VALUE="-1">
System default<INPUT TYPE="radio" NAME="format" VALUE="-2">
<BR>
<B>Line to Write to File:</B><INPUT TYPE="text" SIZE=50 NAME="line-
towrite"><BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="OpenTextFile">
<B>Opens a text file and writes a line</B></FORM></TD>
</TR><TR>
<TD><FORM METHOD=POST ACTION="ServerSystem1.asp">
<B>Text File to Open:</B><INPUT TYPE="text" SIZE=50 NAME="Sfilename"><BR>
<B>Input/Output Mode:</B>For Reading<INPUT TYPE="radio" NAME="iomode"
VALUE="1">
For Writing<INPUT TYPE="radio" NAME="iomode" VALUE="2">
For Appending<INPUT TYPE="radio" NAME="iomode" VALUE="8"><BR>
<B>Format:</B>ASCII<INPUT TYPE="radio" NAME="format" VALUE="0">
Unicode<INPUT TYPE="radio" NAME="format" VALUE="-1">
System default<INPUT TYPE="radio" NAME="format" VALUE="-2">
<BR>
<INPUT TYPE="submit" NAME="btnval" VALUE="ReadFileLine">
<B>Reads a line from a file</B></FORM></TD>
</TR></TABLE>
</BODY>
</HTML>

```

PROVERA ZNANJA

1. Šta je model objekta?
 - A. Model funkcija koje svaki objekat izvodi u Vašoj aplikaciji
 - B. Model objekata pomoću kojih je korisnicima dozvoljeno da vrše interakciju
 - C. Opis ilustracije odnosa objekata jednih sa drugima
 - D. Ništa od gore navedenog
2. Šta sadrži Dictionary objekat, i kako se ti elementi koje sadrži mogu menjati?
 - A. Promenljive, i one se mogu menjati dodelom novih vrednosti.
 - B. Parove ime/vrednost, i oni se mogu menjati korišćenjem osobina Dictionary objekta.
 - C. Folder objekte, i oni se mogu menjati tako što se mogu uklanjati.
 - D. Dictionary objekat ne sadrži ništa što se može menjati.
 - E. U File objektu, koji atributi se mogu podešavati korišćenjem CreateTextFile metode?
 - F. Putanja fajla, overwrite, unicode
 - G. Ime fajla, veličina fajla, putanja fajla
 - H. Putanja fajla, ime fajla, veličina fajla
 - I. overwrite, Unicode, veličina fajla
3. Koji Scripting Library objekti se mogu koristiti za kreiranje tekstualnih fajlova?
 - A. Samo File objekat
 - B. Samo TextStream objekat
 - C. Svi ovi
 - D. Samo FileSystemObject objekat, Folder objekat, File objekat i TextStream objekat
4. Rad sa fajlovima ponekad zahteva navigaciju u okviru njih da bi se pronašli i editovali određeni delovi teksta. Da biste prešli sa jedne linije na sledeću liniju u okviru fajla, koristili biste _____metod _____ objekta. Da biste upisali praznu liniju u fajl, koristili biste _____metod _____ objekta. Da biste utvrdili u kojoj koloni se nalazite u okviru tekstualnog fajla, koristili biste _____ osobinu _____ objekta.