

به نام خالق برحق...

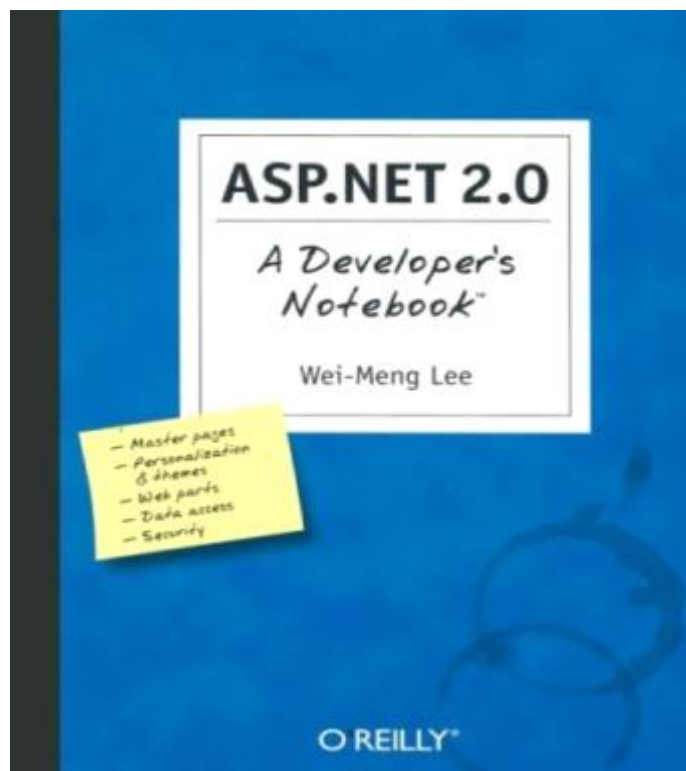
تقدیم به پویندگان راه علم

- این کتاب مدتها پیش ترجمه شد اما به دلایلی تا کنون به اشتراک گذاشته نشده است!
- مترجم ادعائی مبنی بر ترجمه ایده آل اثر ندارد اما امیدوار است در ارتقاء سطح علمی خواننده موثر باشد.
- این کتاب یک اثر رایگان بوده و هر گونه دخل و تصرف در آن از نظر مترجم بلامانع است.
- کتاب اصلی در ژانویه ۲۰۰۵ و بر اساس نسخه بتای ویژوال استودیو قالیف شده که پس از انتشار نسخه نهایی مولف در یک فایل متنی تغییرات اساسی را آورده است. نسخه پیش رو این تغییرات را نیز پوشش می دهد.
- کتاب توسط آقای علی دهقان نیز ترجمه و توسط انتشارات نص به چاپ رسیده است.

محسن قاسمپور

پاییز ۸۶





ASP.NET 2.0: A Developer's Notebook

By: Wei-Meng Lee

Publisher: O'Reilly

مترجم : محسن قا سمپور



با تشکر از مدیریت محترم شرکت مهندسی فرآیند گستر نیوشا

www.Niosha.Co.Ir



پیشگفتار

سالها پیش که ASP.Net چشم انداز جدیدی را به دنیا وب گشود، پدیده واقعا شگفت انگیزی بود که خیلی زود مورد توجه برنامه نویسان وب قرار گرفت. برنامه نویسان Asp کلاسیک خیلی زود به ASP.Net گرویدند و بسیاری دیگر ASP.Net را به سایر زبانها ترجیح دادند.

این ابزار کمپانی میکروسافت (Microsoft) خیلی زود به رقیبی جدی در زمینه ساخت برنامه های وب (Web Applications) و صفحات پویا (Dynamic Pages) تبدیل شد.

ASP.Net چیزی بیش از نسخه ارتقاء یافته (Asp (Upgrade) (Active Server Pages) کلاسیک به شمار می رود. در واقع ASP.Net یک پرش به جلو محسوب میشود.

نسخه نهایی ASP.Net 2.0 مدتهاست توسط میکروسافت در اختیار برنامه نویسان وب قرار گرفته است. در این کتاب نیز به بررسی همین نسخه از ASP.Net که تفاوتهای زیادی نسبت به نسخه های پیشین (1.0 و 1.1) دارد می پردازیم.

در این کتاب علاوه بر بررسی ASP.Net 2.0 از طریق بیش از ۵۰ مثال کاربردی با قابلیتها، مفاهیم و مطالب جدید ASP.Net 2.0 آشنا خواهید شد.

قابلیتهایی که برنامه نویسی را برای طراحان وب بسیار آسان و لذت بخش نموده است. از سوی دیگر این قابلیتها برنامه نویسان را در طراحی وب سایتهای کارآمدتر کمک می کنند.

بعنوان مثال Master Page ها، تمها (Themes) و اسکینها (Skins) شما را در ساخت صفحات سازگار، همگون و ایجاد ظاهر مناسب برای سایت کمک می کنند.

همچنین ASP.Net 2.0 امکانات جدیدی برای ساخت صفحات وب، مخصوص گوشیهایی همراه به ارمغان آورده است. (البته در این کتاب به این مبحث نمی پردازیم)

کنترلهای جدید و ویزاردها (Wizard) اعمال معمول و مهمی همچون دسترسی داده ها (Data Access) را بدون نوشتن حتی یک خط کد، فراهم آورده اند.

لینکهای آخر هر بخش راهنمای بسیاری خوبی در جهت تکمیل مطالب هر بخش به شمار می آیند و از طریق آنها می توانید به مطالب جامع تری دست یابید.

بخشهای مختلف این کتاب به همراه مثالهای گوناگون و متنوع شما را در دنیای جدید و هیجان انگیز ASP.Net 2.0 همراهی می کنند.



فهرست مطالب

۸.....	درباره مولف.....
۹.....	کتاب برای چه کسانی تالیف شده است؟.....
۱۰.....	نحوه سازماندهی کتاب.....
۱۳.....	۱. قابلیت‌های جدید Asp.Net 2.0.....
۱۶.....	۱.۱. ساخت پروژه جدید.....
۲۳.....	۲.۱. برنامه نویسی چند زبانه.....
۲۶.....	۳.۱. استفاده از قابلیت Focus در کنترلها.....
۳۰.....	۴.۱. تعریف اعتبار سنج های گروهی و چند گانه در یک صفحه.....
۳۴.....	۵.۱. قراردادن اسکریپت‌های سمت مشتری (Client Side) در صفحه.....
۳۶.....	۶.۱. ارسال اطلاعات از صفحه ای به صفحه دیگر.....
۴۳.....	۷.۱. نمایش گروهی کنترلها.....
۵۰.....	۸.۱. آپلود فایل به سایت.....
۵۴.....	۹.۱. ساخت Image Map برای سایت.....
۶۰.....	۲. Master Page ها و Site Navigations.....
۶۱.....	۱.۲ طراحی یک Master Page.....
۶۶.....	۲.۲ استفاده از Master Page بعنوان قالب صفحات.....

۳.۲. کار با Master Page در زمان اجرا (Run Time)	۷۴
۴.۲. ساخت Site Map برای سایت	۷۸
۵.۲. نمایش سلسله مراتبی داده ها توسط کنترل TreeView	۹۰
۶.۲. ساخت TreeView از طریق برنامه نویسی	۹۸
۷.۲. ساخت منو توسط کنترل Menu	۱۰۵
۳. Web Part ها	۱۰۹
۱.۳. آشنایی با Web Part ها	۱۱۱
۲.۳. اخت Web Part های سفارشی (Customize) شده	۱۱۹
۳.۳. جابجایی Web Part ها توسط کاربر	۱۲۹
۴.۳. افزودن Web Part توسط کاربر در زمان اجرا	۱۳۴
۵.۳. ویرایش Web Part توسط کاربر در زمان اجرا	۱۴۱
۶.۳. برقراری ارتباط میان Web Part ها	۱۴۶
۴. دسترسی به داده ها (Data Access)	۱۵۵
۱.۴. نمایش داده ها توسط GridView	۱۵۶
۲.۴. مرتب سازی و نمایش داده ها در چند صفحه	۱۷۰
۳.۴. ویرایش (Edit) و حذف (Delete) رکوردها	۱۷۴
۴.۴. نمایش یک رکورد در یک زمان	۱۸۰
۵.۴. کش (Cache) کردن منبع اطلاعاتی (Data Source)	۱۸۷

۱۹۰.....	۶.۴. کشینگ وابسته
۱۹۹.....	۷.۴. رمز کردن (Encrypt) رشته اتصال (Connection String)
۲۰۵.....	۸.۴. اتصال به یک Business Object
۲۱۸.....	۹.۴. اتصال به یک سند Xml
۲۲۵.....	۵. امنیت (Security)
۲۲۶.....	۱.۵. ساخت صفحه ورود (Login) توسط کنترل‌های امنیتی جدید
۲۳۴.....	۲.۵. اضافه کردن کاربر به سایت از طریق WAT
۲۴۲.....	۳.۵. جلوگیری از دسترسی غیر مجاز به صفحات
۲۴۴.....	۴.۵. بازیابی کلمات عبور کاربر
۲۴۸.....	۵.۵. تغییر کلمه عبور کاربر
۲۵۱.....	۶.۵. ساخت حساب کاربری (User Account) به کمک Create User Wizard
۲۵۵.....	۷.۵. گروه بندی کاربران در نقشها (Roles)
۲۶۱.....	۸.۵. مدیریت تنظیمات کاربر
۲۶۷.....	۶. کار آیی های جدید در Asp.Net 2.0
۲۶۸.....	۱.۶. ترجمه صفحات بصورت پویا
۲۷۳.....	۲.۶. تولید Proxy Class ها بصورت پویا
۲۷۸.....	۳.۶. پیش ترجمه (Precompile) سایت
۲۸۱.....	۴.۶. کش کردن بخشی از صفحه

- ۵.۶. ارتباط با سرور بدون انجام Post back ۲۸۵
- پروفایلها (Profiles) ۲۹۶
- ۱.۷. شخصی سازی (Personalization) سایت ۲۹۷
- ۲.۷. تأیید هویت کاربران توسط فرمها ۳۰۵
- ۳.۷. ذخیره پروفایل کاربران ناشناس (Anonymous Users) ۳۱۱
- ۴.۷. انتقال اطلاعات پروفایل ناشناس به پروفایل شناخته شده ۳۱۸
۷. تمها (Themes)، اسکین ها (Skins) و بومی سازی (Localization) ۳۲۳
- ۱.۸. ساخت تم و اسکین ۳۲۴
- ۲.۸. اعمال تم در زمان اجرا ۳۳۰
- ۳.۸. ذخیره تمها در پروفایل کاربران ۳۳۵
- ۴.۸. بومی سازی (Localization) برنامه های وب ۳۳۸

درباره مؤلف

Wei-Meng Lee طراح و توسعه دهنده وب فعالیتهای زیادی در زمینه آخرین تکنولوژیهای میکروسافت در کارنامه خود دارد.

او که در بسیاری کنفرانسهای بین المللی به عنوان سخنران حضور داشته و دارد، مؤلف کتابهای زیادی در زمینه های Net. ، Xml و تکنولوژیهای بیسیم نیز هست از جمله این کتابها می توان به Net Compact Framework و Windows Xp Unwired و ... اشاره کرد.

Wei-Meng Lee در حال حاضر مدیر منطقه ای میکروسافت در سنگاپور است.

مقالات و نوشته های او در سایت شخصی اش به آدرس <http://www.developerlearningsolutions.com> قابل دسترسی اند.

کتاب برای چه کسانی تألیف شده است ؟

فرض کتاب این است که خواننده با یکی از نسخه های قبلی Asp.Net (1.0 یا 1.1) تا حدودی آشنایی دارد و در محیط Visual Studio (2002 یا 2003) برنامه نویسی کرده و هم اکنون قصد دارد از Asp.Net 1.x به Asp.Net 2.0 کوچ کند.

در این کتاب سعی شده است تا از طریق بخشها و مثالهای متنوع برنامه نویسان با قابلیتهای جدید Asp.Net 2.0 در محیط Visual Studio 2005 آشنا شوند و از این قابلیتها در برنامه های خود بهره گیرند.

برای استفاده از این کتاب به نسخه Visual Studio 2005 که از Visual Basic پشتیبانی کند و یک نسخه Sql Server (ترجیحا Sql Server Express) نیاز دارید.

سورس (Source) کامل مثالهای کتاب را نیز می توانید از آدرس زیر دانلود کنید :

<http://www.oreilly.com/catalog/aspnetadn>

مطالب کتاب برای یک برنامه نویس با تجربه Asp.Net 1.x نیز مطالب جدیدی هستند چراکه دنیای Asp.Net 2.0 دنیای جدیدی است !

از آنجا که مثالهای کتاب به زبان Visual Basic 2005 پیاده سازی شده اند ، برنامه نویسان C# با اعمال تغییرات جزئی در مثالها می توانند از آنها بهره گیرند.

نحوه سازماندهی کتاب

این کتاب به هشت فصل تقسیم شده، که هر فصل بر روی یکی از قابلیت‌های جدید Asp.Net 2.0 متمرکز شده است.

برخی از فصلها را مجزا از دیگر فصول و به تنهایی می‌توانید مطالعه کنید، اما برای مطالعه برخی دیگر می‌بایست به فصلهای دیگر نیز مراجعه کنید.

در تمامی بخشهای هر فصل به مدد مثالهای گوناگون مطلب مورد نظر را بصورت گام به گام یاد خواهید گرفت.

در فصل اول، ابتداء با نحوه ایجاد یک برنامه وب Asp.Net 2.0 در محیط Visual Studio 2005 آشنا میشوید، سپس به بررسی برخی امکانات و قابلیت‌های جدید Asp.Net 2.0 خواهیم پرداخت. ادامه فصل نیز روشهای بهبود یافته ای از جمله **Cross Page Posting**، افزودن اسکریپتهای سمت مشتری (Client Side Script) به صفحه و... را پوشش میدهد.

در فصل دوم، به معرفی **Master Page** ها می‌پردازیم که از قابلیت‌های مهم و قدرتمند اضافه شده در Asp.Net 2.0 محسوب میشوند. با ساخت این صفحات و ارث‌بری سایر صفحات از آنها (که همانند ارث‌بری فرمهای ویندوز است) میتوانید برای وبتان جلوه منظم و یکنواختی فراهم کنید.

پس از Master Page ها در ادامه فصل با کنترل‌های جدید **Sitemap Path** و **Menu** آشنا میشوید که به شما امکان میدهند بدون نوشتن کدهای سنگین کاربران را به قسمتهای مختلف سایت هدایت کنید.

در فصل سوم، با **Web Part** ها آشنا میشوید و اینکه چگونه با استفاده از آنها پرتال (Web Portal) طراحی کنید.

امروزه وب سایتها حاوی اطلاعات گوناگون و فراوانی هستند که یک طراحی ضعیف می‌تواند باعث سردرگمی کاربر در سایت شود. به منظور جلوگیری از چنین گمراهی‌هایی پرتالها (مانند Msn) مطالبشان را در بخشهای جداگانه در اختیار کاربر قرار میدهند. سپس با استفاده از شخصی سازی (Personalization) کاربر این امکان را دارد تا بخشهای مورد علاقه خود را حفظ نموده و سایر بخشها را مخفی نماید. علاوه بر این کاربر می‌تواند تنظیمات مورد علاقه اش را ذخیره نماید و در مراجعات بعدی از آنها بهره گیرد.

همه آنچه گفته شد در Asp.Net 2.0 توسط **Web Part** ها امکان پذیر است.

در فصل چهارم، با کنترل‌هایی آشنا میشوید که با کمترین کد دسترسی داده‌ها (Data Access) را برایتان فراهم می‌آورند، دسترسی به داده‌ها از جمله معمولی‌ترین و در عین حال مهمترین اعمالی است که در برنامه نویسی وب بکار میرود. وجود کنترل‌های متنوع مختص این کار در Asp.Net بیانگر همین مطلب است.

یکی از مهمترین کنترل‌های جدید Asp.Net 2.0 کنترل **Gridview** است که نسخه بهبود یافته کنترل **Datagrid** که در نسخه‌های قبلی ارائه شده بود به شمار می‌آید. (البته Asp.Net 2.0 همچنان **Datagrid** را پشتیبانی میکند)

به جز Gridview که مهمترین کنترل دسترسی به داده ها در Asp.Net 2.0 است کنترل های دیگری نیز اضافه شده اند از جمله : **Datalist** و **DetailsView** که در زمان مناسب با آنها نیز آشنا میشوید.

Asp.Net 2.0 دسترسی به داده ها و کار با کنترل های مخصوص این کار را از طریق کنترل هایی چون **SqlDataSource**، **ObjectDataSource** و **XmlDataSource** فراهم میکند.

به یاری کنترل های معرفی شده در بالا مشکلی در دسترسی به داده های اسناد Xml ، پایگاه های داده (Data Bases) و سایر منابع اطلاعاتی نخواهید داشت.

فصل پنجم شما را با کنترل های امنیتی Asp.Net 2.0 آشنا میکند. با استفاده از این کنترل ها اعمال ورود کاربران (User Login) ، ثبت نام (Registration) ، تغییر کلمه عبور (Change PassWord) و ... را با قرار دادن کنترل مربوطه در صفحه (Drag And Drop) انجام خواهید داد. بعلاوه این کنترل ها شما را در ایمن (Secure) سازی وب تان کمک میکنند. قدرت این کنترل ها در استفاده از **API** های عضویتی (Membership APIs) است که راجع به آنها نیز در همین فصل بحث خواهیم کرد.

فصل ششم برخی بهبود های ساختاری Asp.Net 2.0 را بررسی می کند.

بعنوان مثال در Asp.Net 1.x از آنجا که صفحات بصورت پویا (Dynamic) ترجمه (Compile) می شوند و اولین باری که کاربر صفحه را بارگزاری (Load) میکند، صفحه کش (Cache) میشود، لذا بارگزاری صفحات در مرتبه اول به کندی صورت میگیرد. در Asp.Net 2.0 از طریق پیش ترجمه (PreCompile) سایت هنگام مراجعه کاربر به سایت ، سایت ترجمه شده و آماده را در اختیار کاربر قرار میدهید. بعلاوه در Asp.Net 2.0 به کمک **Fragment Caching** میتوانید بجای کش کردن تمام صفحه تنها بخشی از آنرا کش کنید. در ادامه این فصل نحوه تولید خود کار کلاس های پروکسی (Proxy) به کمک اسناد WSDL را خواهید آموخت.

در پایان فصل با **Client Callback Manager** آشنا میشوید که به شما امکان میدهد صفحات حاوی اطلاعات سمت سرور را بدون PostBack به روز (Update) نمائید.

در فصل هفتم ، می آموزید تا از طریق سرویس پروفایل (Profile Service) به سایتتان قابلیت شخصی سازی (Personalization) بدهید. از این طریق و با نگهداری اطلاعات مربوط به هر کاربر و بازیابی آنها جهت استفاده در مراجعات بعدی میزان استفاده کاربران از سایت را به مراتب افزایش میدهید. در Asp.Net 2.0 پروفایلها (Profiles) شما را با این دنیای جدید آشنا میکنند.

در فصل پایانی کتاب بیشتر به ظاهر سایت می پردازیم، در این فصل با تمها (Themes) و اسکین ها (Skins) و عملکرد آنها آشنا میشوید. به علاوه می آموزید که چگونه به کمک مکانیزم **Auto-Culture Handling** در برنامه هایتان از چندین زبان و فرهنگ (Culture) مختلف پشتیبانی کنید.

Asp.Net 2.0

فصل اول

قابلیتهای جدید ASP.NET 2.0

در کمتر از چند سال پس از خلق Asp.Net این ابزار به ابزاری مفید و مورد علاقه طراحان وب تبدیل شد.

در این کتاب راجع به آخرین نسخه این ابزار یعنی Asp.Net 2.0 بحث می کنیم.

شرکت میکروسافت بر اساس تجربیات میلیونی مشتریان و کمپانیهایش در سراسر جهان اقدام به بهبود محصولات خود میکند. بعلاوه که حفظ مشتری با ظهور ابزار متن بازی (Open Source) همچون Php و یا Macromedia FLASH رقابت را برای میکروسافت دشوارتر ساخته است.

با این وجود Asp.Net همچنان یکی از پر طرفدار ترین ابزار های طراحی وب در دنیای وب و اینترنت به شمار می آید.

در این فصل به معرفی اجمالی قابلیت های جدید Asp.Net 2.0 می پردازیم.

-Controls :

Asp.Net 2.0 با قابلیت ها و کنترل های جدید خود برنامه نویسی را برای طراحان وب بسیار آسان و لذت بخش نموده است.

اکنون شما با استفاده از کنترل های جدید ارائه شده در Asp.Net 2.0 اعمالی همچون دسترسی به داده ها (Data Access)، هدایت کاربران در سایت (Site Navigation)، ورود کاربران به سایت (User Login) و ... را به آسانی و بدون نوشتن کد های طولانی به سایتتان اضافه می کنید.

- Page Framework :

در این بخش Asp.Net 2.0 قابلیتهایی چون Master Page ها، تمها (Themes) و اسکینها (Skins) را به همراه دارد. علاوه بر اینها به کمک بومی سازی (Localization) میتوانید وب سایت های جهانی طراحی کنید.

- Services And APIs :

بسیاری از کنترل های جدید Asp.Net 2.0 توسط سرویسها و Api ها پشتیبانی میشوند. بعنوان مثال کنترل های مربوط به ورود (Login) کاربران توسط مجموعه ای از Api های عضویتی (Membership Apis) کنترل می شوند که اعمال تأیید هویت (Authentication)، ثبت نام (Registration) و ... را فراهم می آورند.

با استفاده از کنترل های جدید شما بطور مستقیم به این Api ها دسترسی دارید.

جدول ۱-۱ خلاصه مهمترین قابلیت های جدید Asp.Net 2.0 را نشان میدهد.

جدول ۱-۱. قابلیت‌های جدید Asp.Net 2.0

Data Controls	Page Framework	Services And APIs
-Data Controls شامل کنترل‌های دسترسی به داده از جمله GridView و DetailsView. در فصل چهارم به این موضوع می پردازیم.	- Master Pages ارث بری فرمهای وب از صفحات ویژوال. در فصل دوم به این موضوع میپردازیم.	- Membership APIs اساس کار مدیریت کاربران، از جمله افزودن کاربر جدید، حذف کاربر، شناسایی و تایید هویت، بازیابی کلمات عبور و... در فصل پنجم به این موضوع می پردازیم.
- Login Controls شامل کنترل‌های ساده و مفید جهت مدیریت و شناسایی کاربران سایت. در فصل پنجم به این موضوع می پردازیم.	- Themes And Skins ساخت وب سایتها با صفحات همگون و ظاهر زیبا به کمک اسکینهای گروهی توسط تمها. در فصل هشتم به این موضوع می پردازیم.	- Role Management (Roles) مدیریت نقشها افزودن کاربر به نقش، حذف کاربر از نقش و... در فصل پنجم به این موضوع می پردازیم.
-Web Parts شامل زیر بنای اصلی ساخت Web Part ها. در فصل سوم به این موضوع می پردازیم.	- Localization با این قابلیت تنها چند قدم با بومی سازی و عمومی سازی سایتان فاصله دارید. در فصل هشتم به این موضوع می پردازیم.	- Site Maps نمایش نقشه سایت، تولید راهنما برای کاربران سایت از طریق کنترل Sitemap. در فصل دوم به این موضوع می پردازیم.
- Navigation Controls شامل کنترل‌های نمایش دهنده اطلاعات سایت و منوها. در فصل دوم به این موضوع می پردازیم.	- Compilation ترجمه صفحات بصورت پویا بطوریکه پس از تغییر کد نیازی به ترجمه مجدد ندارند. همچنین تولید خودکار Proxy Class با استفاده از WSDI. در فصل ششم به این موضوع می پردازیم.	- Profile شخصی سازی (Personalization) وب سایت به کمک اشیاء پروفایل (Profile Objects). در فصل هفتم به این موضوع می پردازیم.
- سایر کنترل‌های استاندارد از جمله کنترل‌های File، ImageMap، Upload، MultiView و TreeView		

فصل اول را با تشریح مراحل ایجاد پروژه جدید آغاز و در ادامه با تعدادی کنترل جدید آشنا میشوید.

۱-۱ ایجاد پروژه جدید

مراحل ساخت پروژه جدید در VS 2005 نسبت به ورژن های قبلی کمی تغییر یافته است. در نسخه 2003 از طریق منوی File گزینه New project را انتخاب و سپس نوع پروژه ای را که می خواهید تعیین می کردید. (web, windows, windows services,...) در VS 2005 از طریق گزینه New Website در منوی فایل می توانید پروژه وب جدید ایجاد کنید.

تنظیمات ویرایشگر VS 2005 (IDE)

هنگامی که برای اولین بار VS 2005 را باز می کنید. از شما خواسته می شود تا تنظیمات پیش فرض مخصوص IDE را انتخاب نمایید. این تنظیمات به شرح زیر می باشند:

- Web Development Settings
- Visual basic Development setting
- Visual C# Development setting

بسته به نوع تنظیمی که انتخاب می کنید، منوی Setup شما نیز تغییراتی جزئی خواهد داشت.

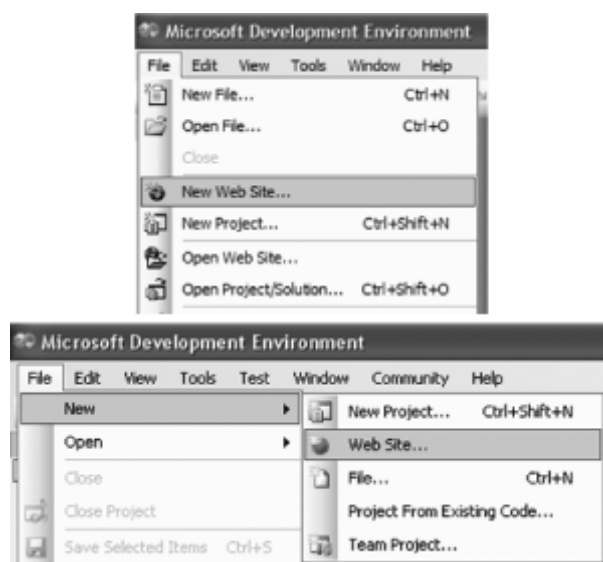
یادداشت: هم اکنون با استفاده از Visual Studio 2005 ساخت برنامه های وب با Asp.Net بسیار آسان شده است.

۱-۱-۱ مراحل انجام کار:

برای آشنایی با مراحل ساخت برنامه وب جدید مراحل زیر با قدم به قدم انجام دهید: ابتدا یک برنامه جدید باز می کنیم سپس فضایی را بر روی هاست (host) برایش در نظر می گیریم. در ادامه فرم وب دیگری به آن اضافه می کنیم. در نهایت شما نحوه Debug برنامه را خواهید دید.

1- می توانید تنظیمات مربوط به IDE را از طریق منوی Tools و گزینه ... Import and Export Settings تغییر دهید. از آنجاکه در سرتاسر کتاب راجع به برنامه های وب بحث می کنیم پیشنهاد می شود گزینه Web Development را انتخاب نمایید.

VS 2005 را باز کنید، چنانچه از قبل گزینه Web Development Setting را انتخاب کرده اید از منوی فایل گزینه new web site را انتخاب کنید. (شکل ۱-۱). چنانچه سایر تنظیمات را انتخاب کرده اید از منوی فایل گزینه New سپس web site را انتخاب کنید. (قسمت دوم شکل ۱-۱)



شکل ۱-۱. ایجاد وب سایت جدید در VS 2005

2- هنگامیکه کادر محاوره (Dialog box) مربوط به ساخت وب سایت جدید نمایش داده شد، از قسمت Templates قالبی را برای پروژه تان انتخاب کنید. از آنجا که می خواهیم یک پروژه خالی ایجاد کنیم گزینه ASP.Net Web site را انتخاب کنید.

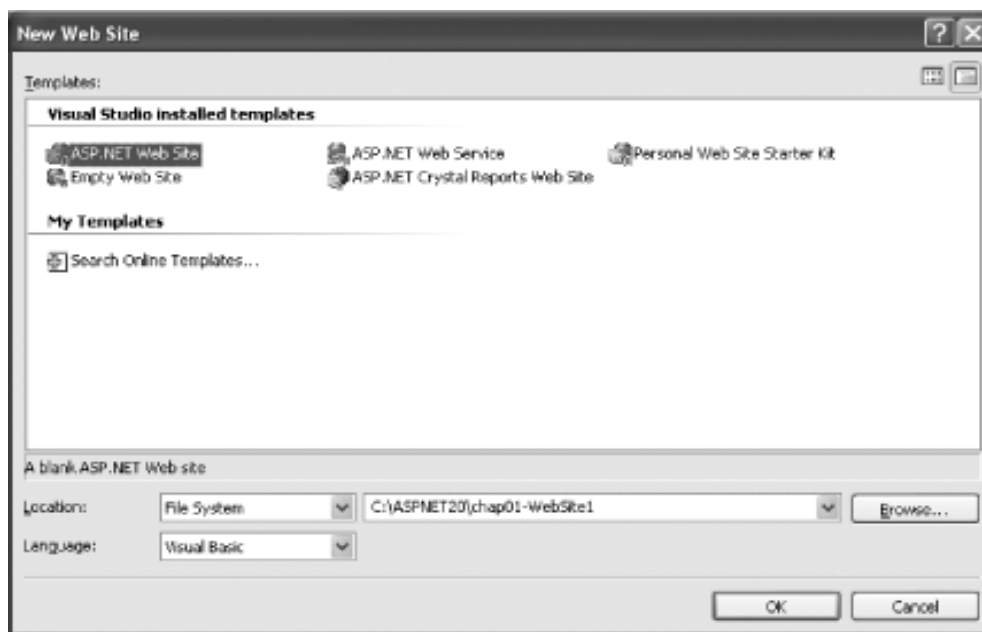
3- در پنجره ایجاد پروژه جدید بایستی زبانی جهت برنامه نویسی سمت سرور و محلی برای ذخیره سایتتان انتخاب کنید. از آنجا که در کتاب از زبان Visual Basic استفاده شده است، شما نیز از لیست باکس (list box) مقابل language گزینه Visual Basic را انتخاب نمایید.

VS 2005 چهار روش برای توسعه ویتان در اختیار شما قرار داده است. در این بخش شما گزینه File Sysem را از لیست باکس مقابل location انتخاب نمایید. این گزینه جدید ASP.NET 2.0 شما را از قید استفاده اجباری از IIS (Internet Information service) رها ساخته است. چرا که VS 2005 دارای وب سرور داخلی مخصوص خود است که در هنگام اجرای پروژه وب اجرا شده و مستقل از IIS وب شما را اجرا می کند.

پروژه تان را در آدرس C:\ASPNET20 و با نام Chap01-website1 ذخیره کنید. برای این کار در کمبو باکس (combo box) مقابل location آدرس C:\ASPNET20\Chap01_WebSite1 را تایپ کنید.

یادداشت : از این پس نگران ساخت Virtual Directory بر روی وب سرورتان نباشید. VS 2005 باوب سرور مبتنی بر فایل اجرای برنامه های وب را بسیار آسان نموده است. بعلاوه شما می توانید برنامه های وبتان را بر روی Windows XP Home Editon و بدون IIS نیز اجرا کنید.

شکل ۲-۱ مراحل تکمیل شده ساخت پروژه جدید را نشان می دهد.



شکل ۲-۱: انتخاب زبان، قالب و محل ذخیره پروژه

نکته: در تمامی مثال های کتاب از آدرس **C:\ASPNET20** جهت ذخیره پروژه هایتان استفاده کنید.

اگر می خواهید از IIS برای اجرای برنامه استفاده کنید از drop – down list مقابل گزینه location گزینه HTTP را انتخاب نمائید و آدرس URL را وارد نمائید. بعنوان مثال آدرس:

Http:// LocalHost / chap 01-web site 1

مراحل نصب IIS :

اگر می خواهید پروژه هایتان را بر روی IIS اجرا کنید. بایستی قبل از نصب VS 2005، IIS را نصب کنید. بطور پیش فرض ویندوز XP، IIS را نصب نمی کند و شما باید با استفاده از CD ویندوز آنرا بر روی سیستماتان نصب نمائید. برای این کار به آدرس زیر بروید:

ControlPanel → Add or Remove programs → Add / Remove windows components

در حالیکه VS 2005 را قبل از IIS نصب کرده باشید بایستی ASP.Net و IIS را به هم پیوند (associate) دهید. این کار را با استفاده از ابزار aspnet – regiis

(که در آدرس <Version>\Microsoft. Net\Framework\ windows \ C:\ قرار دارد) و با سوئیچ i – انجام دهید. مانند زیر:

aspnet_regiis -i

4- هنگامی که قالب، زبان و محل مربوط به پروژتان را انتخاب کردید بر روی دکمه OK کلیک کنید. VS پروژه را ایجاد کرده و فایل های پروژه در قسمت Solution Explorer نمایش داده می شوند. (شکل ۳-۱)

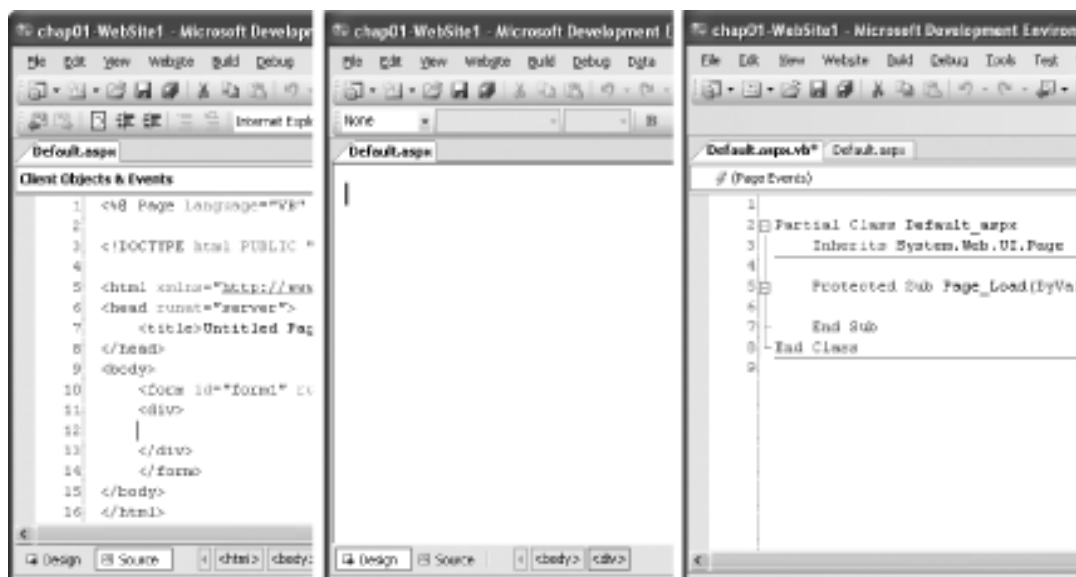


شکل ۳-۱ Solution Explorer

5- در ابتدای ساخت پروژه Asp.Net بطور پیش فرض شاخه ای به نام App_Data و یک فرم وب با نام Default.aspx به همراه صفحه مربوط به code_behind این صفحه (Default.aspx) و با نام Default.aspx.vb تولید می کند. (جهت نمایش فایل code_behind بر روی علامت + کنار Default.aspx در Solution Explorer کلیک کنید)

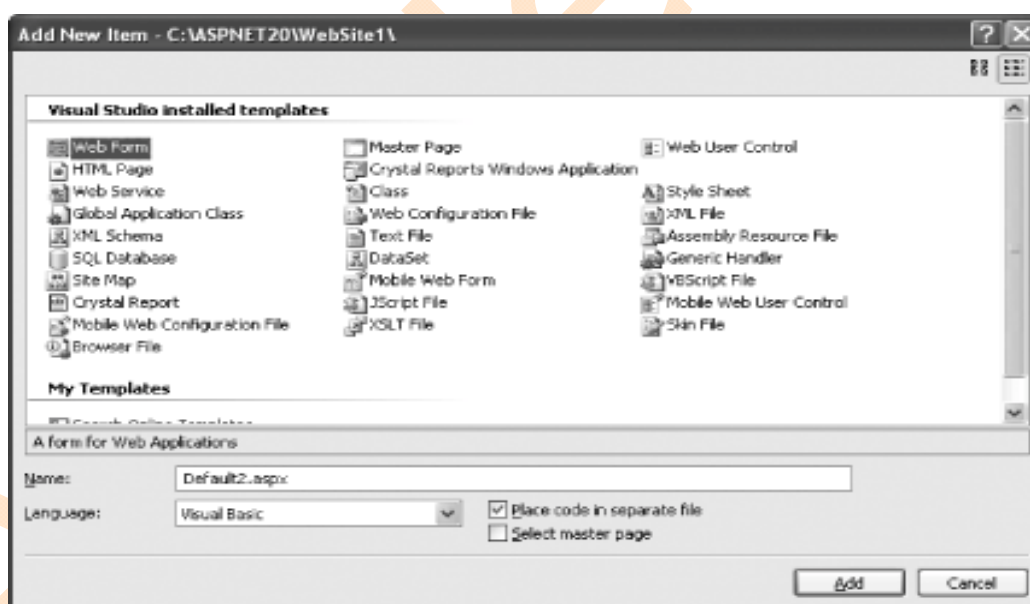
سوئچ کردن بین code_Behind View, Source view, Design view :

در VS 2005، به طور پیش فرض فرم در حالت Source view، نمایش داده می شود. در حالت Source view، شما می توانید انواع خصوصیات مربوط به فرم و کنترل های موجود در آنرا تغییر دهید و یا اصلاح کنید. برای رفتن به Design view دکمه Design گوشه پائین صفحه را کلیک کنید. در Design view شما تصویر صفحه را می بینید و قادرید کنترلها را بر روی صفحه قرار دهید. (Drag and drop) جهت دیدن code – behind تنها کافیست بر روی فرم کلیک مضاعف (Double click) کنید. در code view اعمالی را که صفحه و یا کنترل های موجود در آن بایستی انجام دهند کدنویسی می کنید. شکل ۴-۱ سه حالت صفحه را نشان می دهد.



شکل ۴-۱. سوئیچ بین حالات مختلف فرم

6- برای اضافه نمودن آیتم (Item) جدید (مانند یک فرم وب دیگر) به پروژه بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه Add New Item را انتخاب کنید. سپس آیتم مورد نظر را انتخاب کنید. شکل ۵-۱ را ببینید.



شکل ۵-۱. افزودن آیتم جدید به پروژه

7- توجه داشته باشید که می توانید کد مربوط به صفحه code_behind (.aspx.VB) را در همان صفحه فرم (.aspx) نیز قرار دهید.

برای این کار تیک مربوط به گزینه Place code In seprate file را بردارید. در تمامی مثالهای کتاب این گزینه فعال است و Code – behind ها در فایل های جداگانه قرار می گیرد.

نکته: هنگامیکه می خواهید (توسط کلید F5) پروژه تان را Debug کنید پروژه شما به یک فایل Web. Config نیاز دارد. بطور پیش فرض چنانچه هنگام Debug پروژه، فایل web.config موجود نباشد از شما می خواهد آنرا اضافه نماید. در VS 2005 (بر خلاف VS.Net 2003) فایل Web. Config به طور اتوماتیک به پروژه تان اضافه نمی شود. برای افزودن آن گزینه add new Item را انتخاب و بر روی گزینه web configuration file از visual studio Installed templates کلیک کنید.

1-1-2 پاورقی بخش...

... اصلاح کد تولید شده توسط Visual Designer

چنانچه به code – behind فرم بروید؛ مشاهده می کنید که بلوک مربوط به کد User Interface توسط VS نمایش داده نمی شود، برخلاف ASP.Net 1.x شما تنها یک Partial class مشاهده می کنید.

Partial Class Default_aspx

End Class

تمام کدنویسی شما درون این کلاس صورت می پذیرد. بر خلاف ASP.Net 1.x که شامل کدهای تولید شده توسط Visual Designer بود، ASP.Net 2.0 این بخش را نمایش نمی دهد. در ضمن در زمان ترجمه Visual Designer (compile) بطور اتوماتیک Partial class مورد نیاز User Interface را تولید و آنرا با code behind شما ادغام می کند.

نکاتی راجع به Partial class ها :

یکی از قابلیت های ASP.Net 2.0 مخصوص برنامه نویسان VB 2005 و C# 2.0، Partial class ها هستند. بطور خلاصه Partial class به معنی تجزیه کلاس تعریف شده به چندین فایل فیزیکی است. منطقاً Partial class ها از دوگانگی در مترجم (Compiler) جلوگیری می کنند. در زمان ترجمه، مترجم خیلی ساده انواع مختلف Partial class ها را گروه بندی کرده و با آنها همانند یک موجودیت واحد رفتار می کند. (کدها توسط VS Designer تولید می شوند) یکی از بزرگترین مزیت Partial class ها جداسازی کدهای User Interface (که توسط VS Designer تولید می شوند) از کدهای کاربر است.

با استفاده از Partial class ها کدهای مربوط به User Interface که کاربر یازی به آنها ندارد از دید او مخفی می شوند، همچنین Partial class ها با تقسیم کدها به فایل های جداگانه امکان Debugging ساده تر را فراهم آورده اند.

این قابلیت همچنین به اعضای یک تیم بزرگ توسعه دهنده وب این امکان را می دهد تا بر روی اجزاء مختلف پروژه در قالب فایل های فیزیکی مجزا کار کنند.

... انتخاب فضا برای ذخیره برنامه های وب

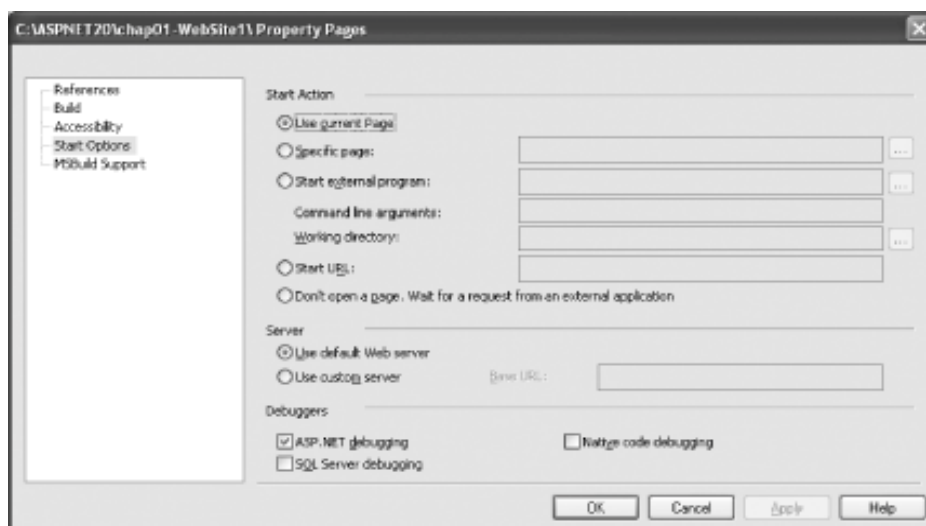
VS 2005 چهار فضا برای یک برنامه وب در نظر گرفته است. چنانچه شما در حال طراحی یک برنامه وب ساده هستید (یا اینکه قصد آشنایی با قابلیت های جدید Net . ASP را دارید) ساده ترین و سریع ترین راه استفاده از متد File system است. این متد همچنین برای کسانی که وب سرور (مانند IIS) در اختیار ندارند یا اینکه از نسخه Home Edition ویندوز XP استفاده می کنند بسیار مفید است. اگر بر روی سیستمتان وب سرور نصب کرده اید و می خواهید از آن بعنوان میزبان (Host) برنامه وبتان بهره بگیرید، از متد Local IIS استفاده کنید. چنانچه به وب سرور بصورت Remote دسترسی دارید می توانید از متد FTP استفاده کنید.

چنانچه میزبان شما دسترسی به FTP را برایتان فراهم آورده است از این دسترسی استفاده کنید. شما می توانید از طریق HTTP وب تان را بر روی سرور میزبان قرار دهید؛ برای این کار می بایستی سرور مورد نظرتان توسط Front page Server Extensions پیکربندی شده باشد.

بهبود Debugging در ASP. Net 2.0

در ASP.Net 1.x شما بایستی صفحه ای را بعنوان صفحه شروع (Start page) پروژتان معرفی کنید. به طوری که هنگام Debug برنامه پس از فشردن کلید F5، این صفحه بارگذاری (Load) می شود. اگر چه در ASP. Net 2.0 همچنان می توانید یک صفحه خاص را بعنوان صفحه شروع معرفی کنید، اما بطور عادی صفحه اصلی شما همان صفحه ای است که توسط شما به حالت انتخاب در آورده است. (این انتخاب می تواند به علت ویرایش آن صفحه و یا انتخاب از طریق Solution Explorer باشد) با این قابلیت دیگر مجبور نیستید برای Debug هر صفحه آنرا بعنوان صفحه شروع معرفی کنید.

این حالت از طریق پنجره Property pages و گزینه Start options قابل تنظیم است. (بر روی نام پروژه در Solution Explorer راست کلیک کنید و سپس گزینه Property pages را انتخاب نمایید). همانگونه که شکل ۶-۱ نشان می دهد.



شکل ۶-۱ پنجره Project property pages

۳-۱-۱. منابع اطلاعاتی :

جهت کسب اطلاع از آخرین تغییرات در مورد VS 2005 به آدرس زیر مراجعه کنید.

<http://lab.msdn.microsoft.com/vs2005>

۲-۱. استفاده از چند زبانی

بر خلاف Asp.Net 1.x که می بایستی حتماً از یک زبان در سرتاسر برنامه استفاده کنید. Asp. Net 2.0 به شما این امکان را می دهد در طول پروژه زبان برنامه نویسی را از یک صفحه به صفحه دیگر تغییر دهید.

یادداشت : شما مجبور نیستید تنها از یک زبان در برنامه های وبتان استفاده کنید.

نکته: اگر چه پشتیبانی از چند زبان قابلیت مفیدی به شمار می آید اما برنامه نویسان باید از این قابلیت در حد اعتدال و بر اساس ضرورت استفاده نمایند. استفاده از چند زبان در یک پروژه پشتیبانی را دشوار می سازد. خصوصاً اگر پشتیبانی پروژه بر عهده فردی ناآشنا به زبان های مورد در پروژه استفاده باشد.

شکل ۷.۱ پروژه ای را به سه صفحه نشان می دهد که هر صفحه با یک زبان کدنویسی شده است :

VB. Net (Default. aspx), C# (Default 2.aspx), VJ # (Default 3.aspx)

نکته: VJ# از صفحات code_behind پشتیبانی نمی کند، به همین دلیل در solution Explorer صفحه ای برای default3.aspx وجود ندارد.



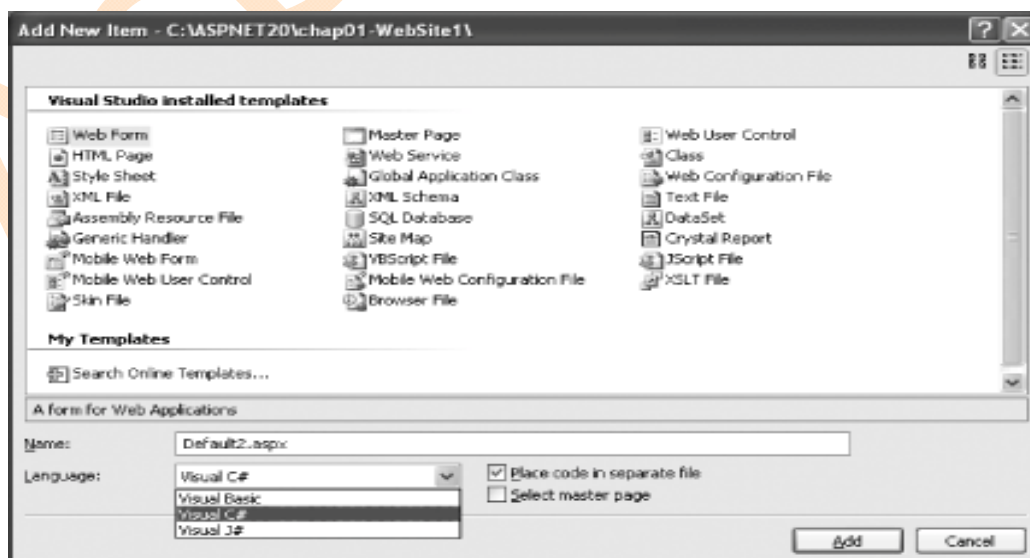
شکل ۷-۱. یک پروژه با صفحات چند زبانه

۱-۲-۱- نحوه انجام کار :

برای اینکه ببینید چگونه می توان چند زبان را درون یک برنامه وب Asp.Net 2.0 ادغام کرد برنامه ای می نویسیم که از دو زبان استفاده می کند. C#, VB. Net.

1- پروژه بکار رفته در مثال قبلی را باز کنید، یک فرم وب جدید به پروژه اضافه کنید بر روی نام پروژه در Solution Explorer راست کلیک کنید، سپس گزینه Add new Item... را انتخاب و از Installed templates گزینه web form را انتخاب کنید.

2- در پنجره Add new Item می توانید زبان دلخواهتان را انتخاب کنید. در شکل ۸-۱ سه زبانی که ASP. Net 2.0 پشتیبانی می کند را مشاهده می کنید: VB, C# , VJ#. نام پیش فرض Default2.aspx را تغییر ندهید و تنها زبان آنرا C# انتخاب کنید، سپس دکمه Add را کلیک کنید.



شکل ۸-۱ انتخاب زبان

3- به code view صفحه Default.aspx رفته و کد زیر را بنویسید :

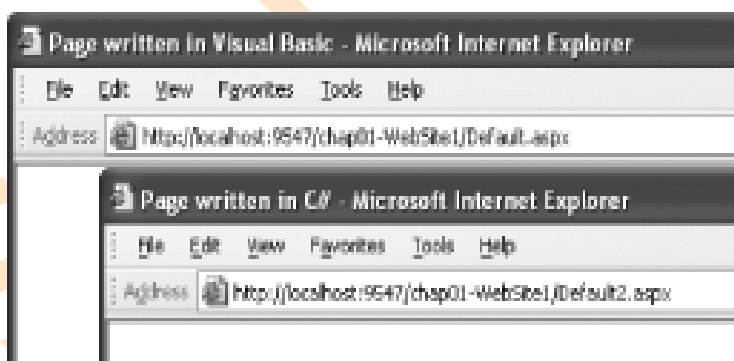
```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load
    Page.Title = "Page written in Visual Basic"
End Sub
```

4- در code view صفحه Default2.aspx نیز کد زیر را وارد کنید:

```
void Page_Load(object sender, EventArgs e)
{
    Page.Title = "Page written in C#";
}
```

5- برای اجرای برنامه، Default.aspx را در Solution Explorer انتخاب نموده و کلید F5 را فشار دهید. به عنوان صفحه در IE (Internet Explorer) توجه کنید.

حال آدرس (URL) را به Default 2.aspx تغییر دهید. شکل ۹-۱ نتیجه اجرای دو صفحه را نشان می دهد.



شکل ۹-۱ نمایش دو صفحه نوشته شده با دو زبان متفاوت ؛ در یک برنامه

۲-۲-۱. پاورقی بخش ...

... استفاده از سایر زبان های پشتیبانی شده توسط .NET. از جمله ++C, python, perl.

VS 2005 تنها زبان های VB 2005, C#, #VJ را برای برنامه های وب پشتیبانی می کند و از سایر زبان ها در انواع دیگر برنامه ها (غیر وب) می توانید استفاده کنید.

۱-۲-۳. منابع اطلاعاتی :

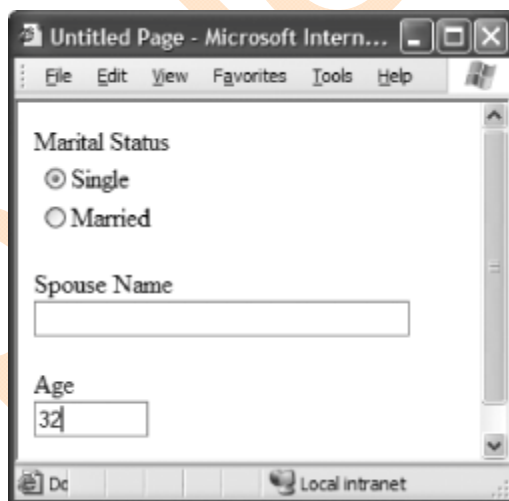
اگر می خواهید درباره ادغام زبان ها در یک پروژه NET. بیشتر بدانید به آدرس زیر مراجعه کنید:

<http://weblogs.asp.net/dreilly/archive/2003/05/17/7164.aspx>

۱-۳-۳. فوکوس (Focus) کردن کنترل ها

در نسخه های قبلی ASP.Net انتساب حالت فوکوس به کنترل ها نیازمند نوشتن اسکریپت های سمت مشتری (Client side scripts) بود (مانند Java script) در ASP.Net 2.0 این عملیات بسیار ساده شده است و شما می توانید با استفاده از متد Focus مربوط به کنترل ، آن کنترل را به حالت فوکوس در آورید. شکل ۱-۱۰ را نگاه کنید.

چنانچه کاربر کلید رادیویی مجرد (single) را انتخاب کند انتظار دارید تکست باکس (text box) مربوط به سن (Age) فعال شود. اما اگر کاربر دکمه متأهل (Married) را انتخاب کرد بایستی تکست باکس مربوط به نام همسر (Spouse name) فعال شود.



شکل ۱-۱۰. اعمال فوکوس بر روی کنترل ها

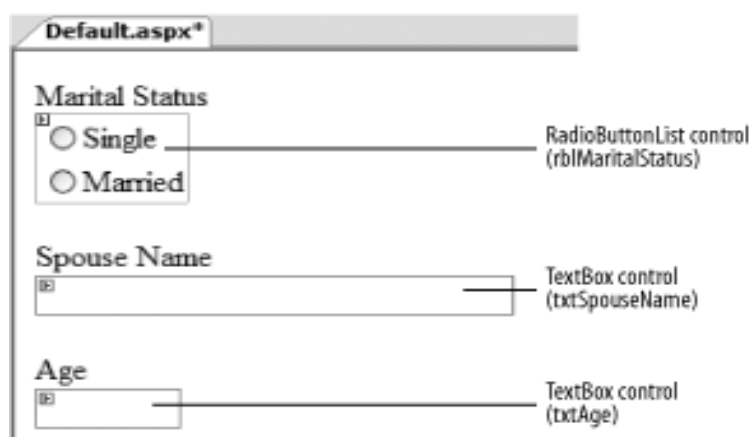
یادداشت: هم اکنون بدون استفاده از جاوا اسکریپت (Java script) و با استفاده از متد جدید Focus قادرید کنترل ها را به حالت فوکوس در آورید.

۱-۳-۱. مراحل انجام کار :

برای آشنایی با متد جدید FOCUS مراحل زیر را جهت طراحی فرم شکل ۱-۱۰ انجام دهید.

1- در VS 2005، پروژه جدیدی با نام و آدرس `C:\ASPNET20\chap01-ControlFocus` ایجاد کنید.

2- یک کنترل `Radio Button` و دو کنترل `Text Box` همانند شکل ۱-۱۱ به فرم اضافه کند.



شکل ۱-۱۱. طراحی فرم

3- دو آیتم `Singel,Married` را به کنترل `RadioButtonList` اضافه کنید. برای این کار به `Source View` رفته و خط های درشت (**Bold**) کد زیر را به آن اضافه کنید.

```
<asp:RadioButtonList ID="rblMaritalStatus"
    runat="server" AutoPostBack="True">
    <asp:ListItem>Single</asp:ListItem>
    <asp:ListItem>Married</asp:ListItem>
</asp:RadioButtonList><br />
```

4- برای اینکه مطمئن شوید هنگامیکه کاربر دکمه `Married` را انتخاب می کند تکست با کس `Spouse Name` فعال می شود و هنگام انتخاب دکمه `Singel`، گزینه `Age` به حالت فعال در می آید بایستی خاصیت `AutoPostBack` کنترل `RadioButtonList (rblMaritalStatus)` را `True` کنید.

برای این کار در پنجره `Propertes` کنترل `RedioButtonList` گزینه `Enable Auto Post Back` را `True` کنید.

5- جهت تکمیل کار بر روی کنترل `RadioButtonList` کلیک مضاعف کنید و در `Code-Behind` کد زیر را وارد کنید.

```
Protected Sub rblMaritalStatus_SelectedIndexChanged( _
    ByVal sender As Object, _
    ByVal e As System.EventArgs) _
```

```
Handles rblMaritalStatus.SelectedIndexChanged
If rblMaritalStatus.SelectedValue = "Single" Then
    txtAge.Focus( )
Else
    txtSpouseName.Focus( )
End If
End Sub
```

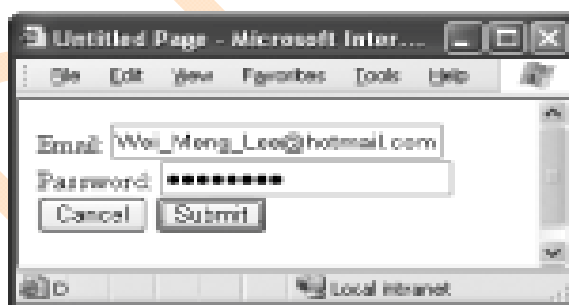
نکته: میتوانید از طریق متد `SetFocus()` از کلاس `Page` یک کنترل را به حالت فوکوس در آورید . بصورت زیر :

```
Page.SetFocus(controlName)
```

۱-۳-۲. پاورقی بخش ...

... قرار دادن دکمه پیش فرض بر روی فرم

در Asp.Net 2.0 می توانید یک دکمه را بعنوان دکمه پیش فرض صفحه معرفی کنید . بعنوان مثال شکل ۱-۱۲ یک صفحه با دو کنترل `TextBox` و دو کنترل `Button` را نشان می دهد . شما می توانید دکمه `Submit` را به عنوان دکمه پیش فرض تعریف نمائید . بطوریکه هنگامیکه کاربر کلید `Enter` را فشرده بطور اتوماتیک دکمه `Submit` کلیک شود.

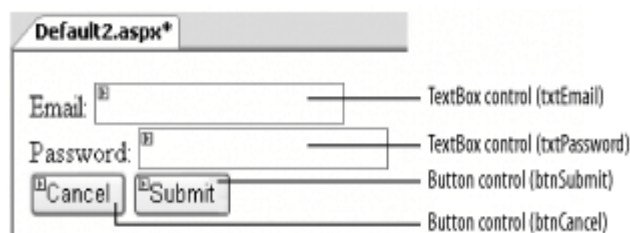


شکل ۱-۱۲ قرار دادن دکمه `Submit` بعنوان دکمه پیش فرض فرم

جهت انجام کار مراحل زیر را انجام دهید :

یادداشت : اکنون در Asp.Net 2.0 می توانید یک دکمه را به عنوان دکمه پیش فرض تعریف کنید بطوریکه هنگام فشردن کلید `Enter` توسط کاربر دکمه بطور اتوماتیک کلیک شود .

۱- همانند شکل ۱-۱۳ فرمی طراحی کنید.

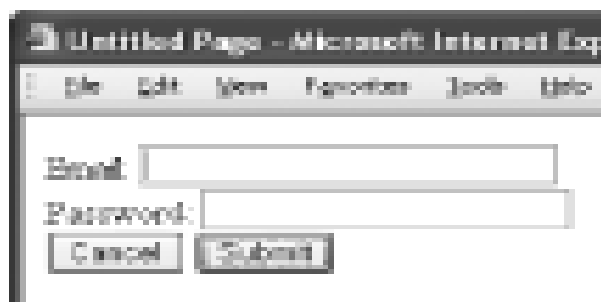


شکل ۱-۱۳. فرم وب به همراه دو TextBox و دو Button

2- به Source View فرم رفته و خطوط درشت قطعه کد زیر را به آن اضافه کنید. خاصیت default Focus کنترل مربوطه را هنگام بارگذاری (load) صفحه فوکوس می کند. خاصیت default button دکمه مورد نظر را هنگام فشردن کلید Enter فعال می کند.

```
<form id="form1" runat="server"
    defaultbutton="btnSubmit"
    defaultfocus="txtEmail"
>
<div>
    Email:
    <asp:TextBox ID="txtEmail" runat="server"
Width="174px"></asp:TextBox>
    <br />
    Password:
    <asp:TextBox ID="txtPassword" runat="server" Width="153px"
        TextMode="Password"></asp:TextBox>
    <br />
    <asp:Button ID="btnCancel" runat="server" Text="Cancel" />
    <asp:Button ID="btnSubmit" runat="server" Text="Submit" /><br />
    <br />
</div>
</form>
```

3- کلید F5 را فشار دهید تا فرم بارگذاری شود؛ دقت کنید که فوکوس بر روی کنترل Textbox مربوط به Email عمل می کند (شکل ۱-۱۴ را ببینید) بعنوان دکمه پیش فرض تعیین شده است.



شکل ۱-۱۴. Textbox مربوط به Email فعال شده و Submit دکمه پیش فرض صفحه است

۱-۳-۳ منابع اطلاعاتی :

همانطور که احتمالاً متوجه شده اید متد Focus در Asp.Net 2.0 سمت سرور (server Side) عمل می کند . ازاینرو هر زمان بخواهید یک کنترل را به حالت فوکوس در آورید به یکPostBack نیاز دارید چنانچه می خواهید کنترل ها را بدون Post Back و سمت مشتری (client Side) فوکوس نمائید به آدرسهای زیر مراجعه کنید :

- <http://www.ondotnet.com/pub/a/dotnet/2003/09/15/aspnet.html>
- <http://ryanfarley.com/blog/archive/2004/12/21/1325.aspx>

۱-۴-۲ تعریف اعتبار سنج های گروهی در یک صفحه

در Asp.Net 1.x اعتبار سنجی تمامی کنترل ها در یک صفحه با هم صورت می پذیرد . بعنوان مثال فرض کنید یک دکمه Search و یک دکمه GetTitle دارید (همانند شکل ۱-۱۵). چنانچه از کنترل اعتبار سنج Required Filed Validator برای تکست باکس ISBN استفاده کنید . سپس دکمه Search را کلیک کنید . تا زمانیکه تکست باکس ISBN خالی است هیچ نتیجه ای در بر نخواهد داشت و تا زمانیکه یکی از کنترل ها از نظر اعتبار سنج (Validator) مشکل داشته باشد تمامی صفحه نامعتبر محسوب می شود.



شکل ۱-۱۵ . چند فرم در یک صفحه

یادداشت: بوسیله گروه بندی کنترل ها توسط اعتبار سنجی گروهی ، می توانید مجموعه ای از کنترلها را بصورت مجزا کنترل نمائید .

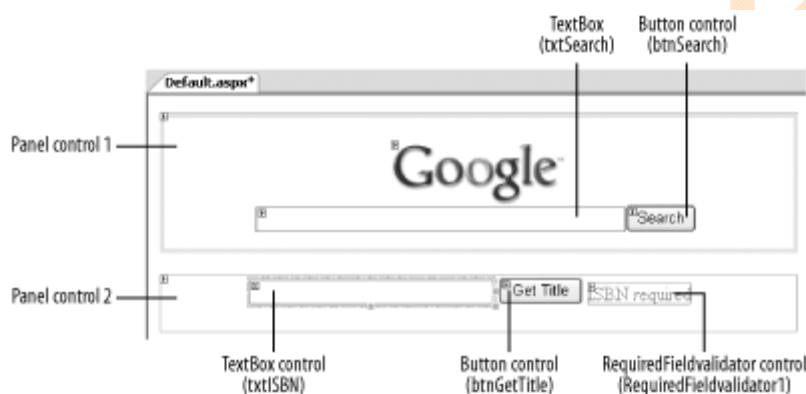
در Asp.Net 2.0 اعتبار سنجی گروهی این امکان را فراهم آورده است کنترل های موجود در یک صفحه بصورت مستقل و مجزا از یکدیگر توسط اعتبار سنج ها کنترل شوند .

۱-۴-۱ مراحل انجام کار :

جهت آشنایی با این قابلیت جدید یک فرم با چندین کنترل طراحی می کنیم ؛ سپس یک اعتبار سنج به یکی از آنها اضافه می کنیم و هر کنترل را به گروهی مجزا انتساب می دهیم . ابتدا فرم نشان داده شده در شکل ۱-۱۵ را طراحی می کنیم :

1- VS 2005 را باز کنید و یک پروژه جدید با نام و آدرس *C:\ASPNET20\chap01-Validation* ایجاد کنید .

2- همانند شکل ۱-۱۶ کنترل های Panel و TextBox و Button را به فرم اضافه کنید .



شکل ۱-۱۶ فرم طراحی شده

3- در پنل بالایی (Panel Control 1) دکمه Search را بگونه ای تنظیم می کنیم که رشته جستجو را به سرور ارسال و در بازگشت کاربر را به سایت گوگل (Google) هدایت (Redirect) نماید. برای انجام این کار بر روی دکمه Search (btn Search) کلیک مضاعف کنید و کد زیر را به آن اضافه کنید:

```
Sub btnSearch_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles btnSearch.Click
    Dim queryStr As String = HttpUtility.UrlEncode(txtSearch.Text)
    Response.Redirect("http://www.google.com/search?q=" & queryStr)
End Sub
```

- 4- کنترل TextBox درون پانل پایین (Panel Control 2) شماره ISBN را می گیرد و عملی را انجام می دهد .
(مثلاً بازایی اطلاعات مربوط به کتاب از سایت Amazon.com)
- یک کنترل Required Filed Valdator به تکست باکس ISBN انتساب دهید. (برای این کار می توانید کد زیر را در Source View اضافه کنید)

```
<asp:RequiredFieldValidator
    ID="RequiredFieldValidator1" Runat="server"
    ErrorMessage="RequiredFieldValidator"
    ControlToValidate="txtISBN"
    SetFocusOnError="True">
    ISBN required</asp:RequiredFieldValidator>
```

- 5- برای تقسیم کنترل ها به دو گروه مجزا به Source View رفته و از خاصیت ValidationGroup استفاده کنید .
(خطوط ضخیم کد زیر)

...

```
<asp:TextBox ID="txtSearch" Runat="server"
    Width="320px" Height="22px">
</asp:TextBox>

<asp:Button ID="btnSearch" Runat="server"
    Text="Search" ValidationGroup="SearchGroup" />
```

...

```
<asp:TextBox ID="txtISBN" Runat="server"
    Width="212px" Height="22px">
</asp:TextBox>

<asp:Button ID="btnGetTitle" Runat="server"
    Text="Get Title"
ValidationGroup="BooksGroup" />
```

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator1"
    Runat="server" ErrorMessage="RequiredFieldValidator"
    ControlToValidate="txtISBN"
ValidationGroup="BooksGroup">ISBN required
</asp:RequiredFieldValidator>
```

...

- 6- هم اکنون دکمه Search به Search Group تعلق دارد در حالیکه دکمه Get Title و Required Filed به Valdator به گروه Book Group .
- 7- کلید F5 را فشار دهید تا نتیجه کار را مشاهده کنید . حالا شما می توانید دکمه های Search و Get Titele را مستقل از یکدیگر کلیک کنید .

نکته : کنترل های کد فاقد خاصیت ValidetinGroup باشند ، به گروه پیش فرض تعلق خواهند گرفت و همانند Asp.net 1.x عمل خواهند کرد .

۱-۴-۲ . پاورقی بخش ...

... فوکوس کردن یک فیلد هنگام خطا

تمامی کنترل های اعتبار سنج در Asp.Net 2.0 از خاصیت SetFocusOnError پشتیبانی می کنند .

تنها کافیت خاصیت SetFocusOnError اعتبار سنج را True کنید بدین ترتیب مرورگر (Browser) هنگام بروز خطا کنترل را فعال می کند .

در مثال قبل چنانچه خاصیت SetFocusOnError کنترل Valdator Required Filed ؛ True باشد :

```
<asp:RequiredFieldValidator
    ID="RequiredFieldValidator1" Runat="server"
    ErrorMessage="RequiredFieldValidator"
    ControlToValidate="txtISBN"
    ValidationGroup="BooksGroup"
    SetFocusOnError="True">
```

مادامیکه تکست باکس ISBN خالی باشد و دکمه Get Titele را کلیک کنید ، تکست باکس ISBN فعال خواهد بود .

۱-۴-۳ . منابع اطلاعاتی :

چنانچه علاقه مند هستید ValidationGroup های موجود در Asp.Net 2.0 را در Asp.Net 1.x پیاده سازی کنید ، لینک زیر می تواند به شما کمک کند .

<http://weblogs.asp.net/skoganti/archive/2004/12/05/275457.aspx>

۵-۱. قراردادن کدهای سمت مشتری (Client Script) درون صفحه

موارد بسیاری وجود دارد که نیازمند استفاده از کدهای سمت مشتری (Client Script) Javascript جهت انجام برخی اعمال هستید. بعنوان مثال با نكرداری الکترونیکی را در نظر بگیرید. می خواهید در صورت غیر فعال بودن کاربر پس از مدت زمان معینی توسط پنجره ای از او سوال کنید که مایل به ادامه فعالیت هست یا خیر؟ نمایش این پنجره برای کاربر توسط کد نویسی سمت مشتری امکان پذیر است.

یادداشت: هم اکنون قادرید به همان راحتی کد نویسی سمت سرور، کدهای سمت مشتری را درون صفحاتتان قرار دهید. در Asp.Net 2.0 می توانید از طریق خاصیت ClientScript از کلاس Page، کد نویسی سمت مشتری انجام دهید.

۱-۵-۱. مراحل انجام کار:

جهت نمایش نحوه قرار دادن کدهای سمت مشتری درون یک برنامه وب Asp.Net 2.0؛ پروژه ای ایجاد می کنیم که با استفاده از پنجره Javascript زمان فعلی سیستم را هنگام بارگذاری صفحه نمایش دهد.

- 1- در VS 2005 پروژه جدیدی با نام و آدرس `C:\ASPNET20\chap01-ClientScript` ایجاد کنید.
- 2- بر روی فرم پیش فرض دوبار کلیک کنید تا به Code-behind فرم بروید.
- 3- درون رویداد (Event) Form-Load با استفاده از متد `RegisterClientScriptBlock()` کد Javascript مورد نظرتان را وارد کنید. قطعه کد زیر این عمل را انجام می دهد:

```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load
    '--inserting client-side script
    Dim script As String = _
        "alert('Time on the server is " & Now & "');"
    Page.ClientScript.RegisterClientScriptBlock(_
        Me.GetType, "MyKey", script, True)
End Sub
```

شکل ۱-۱۷ خروجی برنامه در هنگام بارگذاری صفحه را نشان می دهد.

نکته : متد `RegisterClientScriptBlock()` را در `Asp.Net 1.x` می توانید در کلاس `Page` پیدا کنید . در `Asp.Net 2.0` فضای نام `(Name Space)` این متد تغییر یافته است و از طریق کلاس `ClientScriptManager` قابل دسترسی است . شما می توانید از طریق `Page.ClientScript` به کلاس `ClientScriptManager` دست یابید .



شکل ۱-۱۷ اجرای اسکریپت سمت مشتری

پارامترهای `(Paramter)` متد `RegisterClientScriptBlock()` به شرح زیر می باشد :

Type

نحوه فراخوانی صفحه را مشخص می کند .

Key

کلیدی جهت شناسایی اسکریپت .

Script

کد مربوط به اسکریپت ارسال شده به سمت مشتری .

AddScript

مشخص می کند که آیا اسکریپت در میان بلوک `< Script>` محصور باشد یا خیر . در صورت `True` بودن اسکریپت تولید شده در سمت مشتری به شکل زیر خواهد بود :

```
<script type="text/javascript">
<!--
alert ('Time on the server is 7/18/2007 12:07:23 PM');//
-->
</script>
```

چنانچه پارامتر `AddScript` را `False` قرار دهید اسکریپت در سمت مشتری اجرا نخواهد شد و تنها در صفحه وب نمایش داده میشود . (هیچ کدی برای سمت مشتری تولید نخواهد شد)

نکته: دقت داشته باشید در Asp.Net 1.1 پارامتر Key می تواند رشته خالی باشد اما در Asp.Net 2.0 رشته خالی خطای زمان اجرا (Runtime Error) را به همراه خواهد داشت.

۱-۵-۲ پاورقی بخش ...

... استفاده از یک فایل حاوی اسکریپت

ممکن است بخواهید بجای قرار دادن اسکریپت سمت مشتری درون صفحات برنامه آنها را درون فایل های جداگانه قرار دهید. بعنوان مثال فرض کنید فایلی با نام hello.js در اختیار دارید که حاوی کدهای جاوااسکریپت است. (بدون تگ (<script> Tag)

```
alert("Hello world, from JavaScript");
```

شما می توانید این فایل را از طریق متد (RegisterScriptInclude) به صفحه خود اضافه کنید:

```
'--including script files
Dim scriptURL As String = "./hello.js"
Page.ClientScript.RegisterClientScriptInclude( _
    Me.GetType, "MyKey", scriptURL)
```

خروجی تولید شده بصورت زیر خواهد بود:

```
<script src="./hello.js" type="text/javascript"></script>
```

۱-۵-۳ منابع اطلاعاتی :

در MSDN می توانید مقاله جامعی در خصوص نحوه اضافه نمودن اسکریپت قسمت مشتری به صفحات Asp.Net پیدا کنید :

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspp/html/aspnet-injectclientsidesc.asp>

۱-۶ انتقال اطلاعات بین صفحات

یادداشت : Asp.Net 2.0 ارسال مقادیر (Values) بین صفحات را بسیار آسان نموده است.

اکثر برنامه نویسان Asp.Net با قابلیت PostBack کنترلرها آشنا هستند، اما در بسیاری از موارد PostBack به صفحه ارسال شده عمل مورد نظر شما نیست و ممکن است بخواهید اطلاعاتتان را به صفحه دیگری ارسال کنید.

بعنوان مثال فرض کنید میخواهید در برنامه تان صفحه ای را به نظرسنجی از کاربران اختصاص دهید. مطمئناً بجای طرح ۵۰ سوال در یک صفحه ترجیح میدهید سوالات را در چند صفحه به کاربر ارائه دهید، مثلاً در هر صفحه ۱۰ سوال. ممکن است در این صفحه پاسخ برخی سوالات به سوالات صفحه یا صفحات دیگر وابسته باشند. در این حالت نیاز پیدا می کنید برخی مقادیر (مثلاً پاسخ سوالات) را بین صفحات جابجا کنید. در این وضعیت به موازات ارسال اطلاعات بایستی در صفحه دیگر دریافت اطلاعات داشته باشید. (Cross Post)

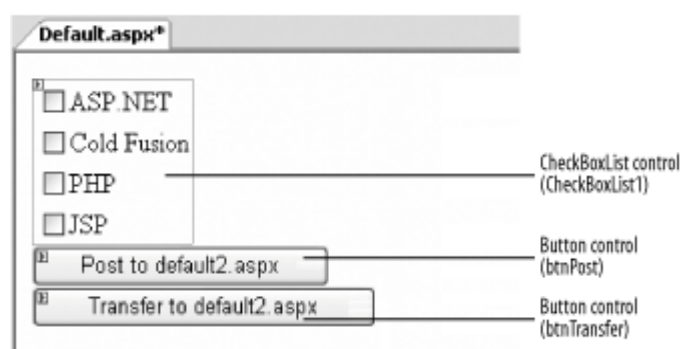
یک راه حل ساده در Asp.Net 1.x جهت انتقال اطلاعات به صفحه دیگر استفاده از **Server.Transfer** است. اما Asp.Net 2.0 از طریق **Cross Page Posting** انتقال اطلاعات را بصورت آسان تر و بهینه تر در اختیار شما قرار داده است. هم اکنون کنترلرها به کمک خاصیت **PostBackURL** انتقال اطلاعات میان صفحات را به آسانی انجام میدهند.

۱.۶.۱. مراحل انجام کار :

برای اینکه با نحوه عملکرد **Cross Page Posting** در Asp.Net 2.0 آشنا شوید پروژه جدیدی با دو صفحه طراحی می کنیم. یکی از صفحات به شما اجازه میدهد از یک **CheckBoxList** گزینه ای را انتخاب و به صفحه بعد ارسال کنید. بعلاوه چگونگی بازیابی مقادیر ارسالی در مقصد را خواهید دید. همچنین در این مثال با تفاوت های میان **Cross-Page Posting** و **Server.Transfer** متد آشنا خواهید شد.

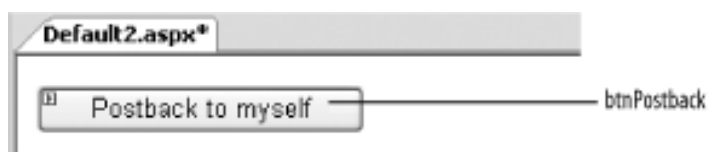
1- در VS 2005 پروژه جدیدی با نام و آدرس **C:\ASPNET20\chap01-CrossPagePosting** ایجاد کنید.

2- کنترل های نشان داده شده در شکل ۱-۱۸ را به فرم پیش فرض اضافه کنید. **checkboxlist1** شامل چندین **Check Box** است که چندین گزینه را در اختیار کاربر قرار می دهند. هنگامی که دکمه "Post to Default2.aspx" کلیک می شود با استفاده از **cross-page posting** مقادیر به صفحه **default2.aspx** ارسال می شوند. دکمه "Transfer to Default2.aspx" از متد **Server.Transfer** جهت ارسال اطلاعات به صفحه **default2.aspx** استفاده می کند.



شکل ۱-۱۸ . طراحی صفحه default.aspx

3- فرم وب جدیدی شامل یک دکمه به پروژه اضافه کنید . (شکل ۱-۱۹ را ببینید)



شکل ۱-۱۹. طراحی صفحه default2.aspx

به Source view صفحه default2.aspx رفته و خاصیت PostBackUrl را به **btnPost** اضافه کنید . به این ترتیب این صفحه به صفحه default2.aspx ؛ cross-page post خواهد داشت .

```
<asp:Button ID="btnPost"
    PostBackUrl="~/Default2.aspx"
    runat="server"
    Text="Post to Default2.aspx" />
```

4- کد زیر را به **btnTransfer** اضافه کنید :

```
Protected Sub btnTransfer_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles btnTransfer.Click
    Server.Transfer("Default2.aspx")
End Sub
```

به منظور دسترسی default2.aspx به مقادیر کنترل CheckBoxlist1 موجود در صفحه default.aspx کد زیر را به Code-behind صفحه default.aspx اضافه کنید : (خطوط درشت)

```
Partial Class Default_aspx
    Inherits System.Web.UI.Page

    Public ReadOnly Property ServerSide( ) As CheckBoxList
        Get
            Return CheckBoxList1
        End Get
    End Property
```

```
Protected Sub btnTransfer_Click(ByVal sender As Object, _
```

```

        ByVal e As System.EventArgs) _
        Handles btnTransfer.Click
    Server.Transfer("Default2.aspx")
End Sub
End Class

```

نکته: به منظور دسترسی سایر صفحات به مقادیر موجود در صفحه اول نیازمند استفاده از خواص عمومی هستید این کار انقیاد زودرس (early binding) و سهولت کد نویسی را نیز به همراه دارد.

7- در default2.aspx بایستی مشخص کنید که مقادیری از طریق default.aspx به این صفحه ارسال خواهند شد. این کار را از طریق رهنمود Previouspagetype(Directive) انجام دهید. برای این کار به Source view رفته و کد زیر را اضافه کنید:

```

<%@ Page Language="VB" AutoEventWireup="false"
    CodeFile="Default2.aspx.vb" Inherits="Default2_aspx" %>
<%@ PreviousPageType VirtualPath="~/Default.aspx" %>

```

8- هنگامیکه مقادیر از default.aspx به default2.aspx ارسال می شوند تمامی اطلاعات مربوط به Default2.aspx در خاصیتی بنام Previouspage نگهداری می شوند. هنگام بارگذاری صفحه default2.aspx، ابتدا بایستی مشخص کنید که بارگذاری این صفحه توسط default.aspx بوده یا از طریق خودش برای اینکار باید چک کنید که آیا Previouspage مرجعی دارد یا خیر:

```

Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
    Me.Load
    If PreviousPage IsNot Nothing Then
        ...
    End If
End Sub

```

9- سپس از طریق خاصیت جدید IsCrossPagePostBack می توانید چک کنید که ارسال بصورت Cross-Page بوده یا Server.Transfer. چنانچه مقادیر از طریق Cross - page از default.aspx به default2.aspx ارسال شده باشند این خاصیت True و در حالت Server.Transfer؛ False خواهد بود.

```

If PreviousPage IsNot Nothing Then
    '--checks the type of posting
    If PreviousPage.IsCrossPagePostBack Then
        Response.Write("This is a cross-post")
    Else
        Response.Write("This is a Server.Transfer")
    End If
End If

```

End If

یادداشت : می توانید از خاصیت *IsCrossPagePostBack* برای تشخیص Cross posting استفاده کنید.

10- در نهایت گزینه های انتخاب شده CheckBoxList را نمایش دهید.

```
Response.Write("<br/>You have selected :")
Dim i As Integer
For i = 0 To PreviousPage.ServerSide.Items.Count - 1
    If PreviousPage.ServerSide.Items(i).Selected Then
        Response.Write( _
            PreviousPage.ServerSide.Items(i).ToString & " ")
    End If
Next
```

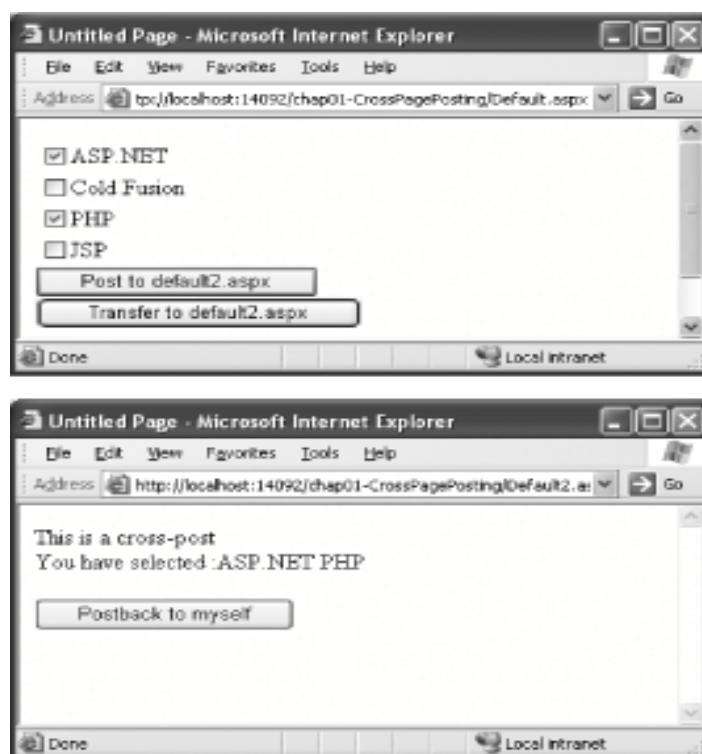
نکته: خاصیت ServerSide به خاصیت عمومی تعریف شده در صفحه Default.aspx اشاره می کند.

کد مربوط به رویداد page_load بصورت کامل در زیر آمده است :

```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load
    If PreviousPage IsNot Nothing Then
        '--checks the type of posting
        If PreviousPage.IsCrossPagePostBack Then
            Response.Write("This is a cross-post")
        Else
            Response.Write("This is a Server.Transfer")
        End If

        Response.Write("<br/>You have selected :")
        Dim i As Integer
        For i = 0 To PreviousPage.ServerSide.Items.Count - 1
            If PreviousPage.ServerSide.Items(i).Selected Then
                Response.Write( _
                    PreviousPage.ServerSide.Items(i).ToString & " ")
            End If
        Next
    End If
End Sub
```

شکل ۲۰-۱ نتیجه cross-page posting را نشان می دهد.



شکل ۲۰-۱ ارسال از Default.aspx به Default2.aspx به روش cross – posting

شکل ۲۱-۱ ارسال اطلاعات به default2.aspx را به روش Server.Transfer نشان می دهد. به URL هر دو صفحه دقت کنید. (در Server.Transfer هر دو یکسان هستند)



شکل ۱-۲۱. ارسال از Default. Aspx به Default2.aspx با استفاده از Server.Transform

چنانچه دکمه " Postback to myself " در Default. aspx را کلیک کنید. متوجه می شوید که اطلاعات ارسال شده از Defalt.aspx از بین می روند. علت هم این است که عمل post back شیئی را که خاصیت pervous page به آن رجوع می کرد، از بین می برد.

نکته: تفاوت اصلی میان Server. Transfer و cross – page post این است که در حالت Server.Transfer آدرس URL در صفحه جدید تغییر نمی کند.

۱-۶-۲ پاورقی بخش ...

... ارسال اطلاعات از چند صفحه به یک صفحه

در این حالت دیگر قادر به استفاده از انقیاد زودرس که توسط خاصیت PreviousPage فراهم شده بود نیستند. چرا که رهنمود PreviousPageType نوع صفحه را از قبل مشخص می کند:

```
<%@ PreviousPageType VirtualPath="~/default.aspx" %>
```

بنابراین، چنانچه ارسال های متفاوتی به یک صفحه مشترک داشته باشیم، استفاده از این راهنما مفید نخواهد بود زیرا صفحات ممکن است کنترلها و انواع مختلفی داشته باشند. و در واقع صفحه مشخصی در این حالت وجود ندارد. بنابراین بهترین راه استفاده از انقیاد دیررس (late – binding) و متد Findcontrol است.

مثال ۱-۱ نشان می دهد که چگونه می توان با استفاده از خاصیت AppRelativeVirtualPath از خاصیت PerviousPage آدرس آخرین صفحه ای که مقادیری را ارسال نموده پیدا کرد. در ادامه برنامه از طریق متد FindControl () کنترل های موجود در صفحه اصلی پیدا می کند.

یادداشت : در صورتیکه ارسال از چندین صفحه به یک صفحه صورت می گیرد می توانید از متد Find control () جهت پیدا کردن مقادیر موجود در صفحه قبل استفاده کنید.

```
If PreviousPage.AppRelativeVirtualPath = "~/Default.aspx" Then
```

```
Dim serverSide As CheckBoxList
```

```
serverSide = CType(PreviousPage.FindControl("checkboxlist1"), _
```

```
System.Web.UI.WebControls.CheckBoxList)
```

```

If serverSide IsNot Nothing Then
    Dim i As Integer
    For i = 0 To serverSide.Items.Count - 1
        If serverSide.Items(i).Selected Then
            Response.Write(serverSide.Items(i).ToString & " ")
        End If
    Next
End If

ElseIf PreviousPage.AppRelativeVirtualPath = "~/Default3.aspx" Then
    Dim userName As TextBox
    userName = CType(PreviousPage.FindControl("txtName"), _
        System.Web.UI.WebControls.TextBox)

    If userName IsNot Nothing Then
        Response.Write(userName.Text)
    End If
End If

```

مثال ۱-۱ جستجوی کنترل‌های موجود در آخرین صفحه ارسال کننده

نکته : چنانچه از cross – post بین دو برنامه (پروژه) متفاوت استفاده کنید، خاصیت، Pervious page ؛ Nothing بر می گرداند. (در C#، Null بر می گرداند)

۱-۶-۳. منابع اطلاعاتی :

جهت کسب اطلاعات بیشتر از روش های هدایت کاربران به صفحات دیگر در MSDN عبارت " Redirecting Users to Another Page " را جستجو کنید.

۱-۷. نمایش گروهی کنترل ها از طریق انتخاب

یکی از اعمال مشترک در برنامه های وب گردآوری اطلاعات است. بعنوان مثال ممکن است بخواهید صفحه ای برای ثبت نام کاربران طراحی کنید.

در این صفحه احتمالاً می خواهید اطلاعات گوناگونی را جمع آوری کنید از قبیل: نام کاربر، تاریخ تولد، و احتمالاً سئوالات امنیتی. یک راه حل خوب این است که سئوالات را در چندین صفحه قرار دهید تا کاربر نیازی به پیمایش یک صفحه بزرگ نداشته باشد.

البته برنامه نویسان ASP. Net 1.x ترجیح می دهند از کنترل Panel استفاده کنند، و با انتخاب هر پنل و مخفی کردن پنل های دیگر در هر پنل مجموعه ای از سئوالات را به نمایش در آورد.

یادداشت: با استفاده از کنترل های جدید Wizard, Multiview کنترل ها را از طریق انتخاب نمایش داده و مخفی کنید.

در Asp.Net 2.0 کنترل Multi view زحمت ساخت چندین صفحه را از میان برداشته است. این کنترل می تواند از چندین کنترل view (کنترل جدید Asp.Net 2.0) تشکیل شود که هر کدام از این view ها چندین کنترل دیگر در برداشته باشند. سپس از طریق برنامه نویسی هر کدام را نمایش دهید.

۱-۷-۱. مراحل انجام کار:

برای اینکه با نحوه کار کنترل Multiview آشنا شوید، برنامه ای با یک کنترل Multiview که از سه کنترل view تشکیل شده طراحی می کنیم.

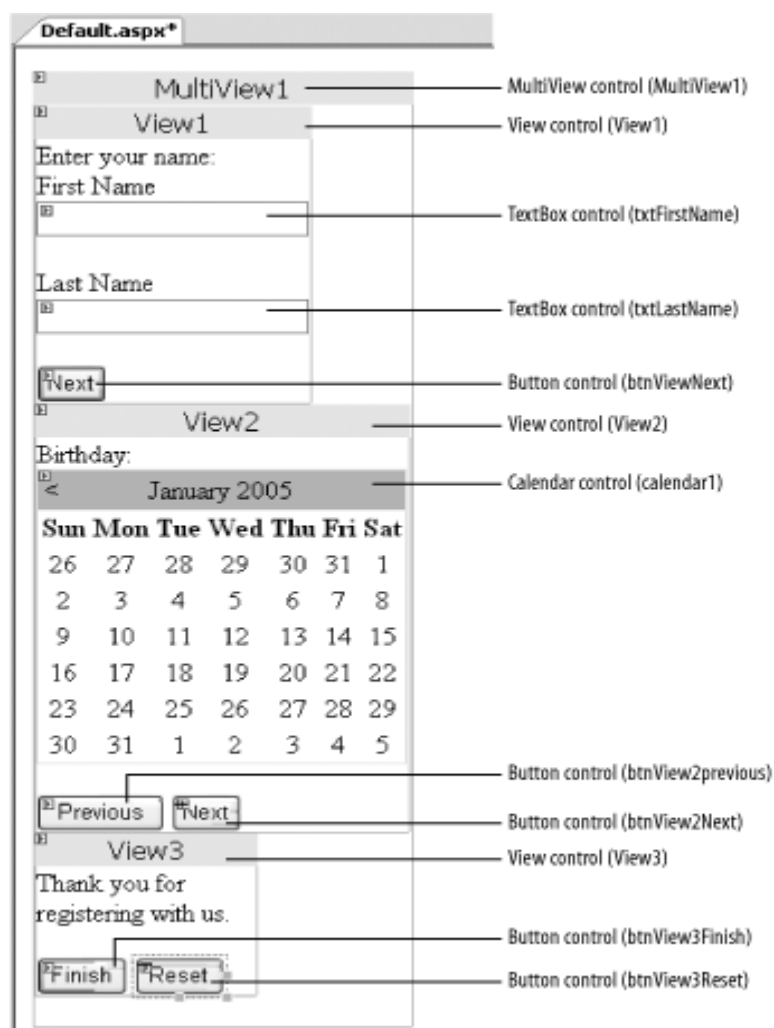
هر کدام از view ها همانند یک فرم وب عادی عمل خواهند کرد. بگونه ای این view ها را به هم ارتباط می دهیم که کاربر بتواند از هر کدام به طور جداگانه استفاده کند.

1- پروژه جدیدی با نام و آدرس C:\ASPNET20\chap01-MultiView در VS 2005 ایجاد کنید.

2- بر روی کنترل Multi view درون Tool Box کلیک کرده و آنرا به فرمتان اضافه کنید.

3- از طریق Tool Box سه کنترل View به Multi view اضافه کنید.

4- همانند شکل ۱-۲۲ چندین کنترل به هر کدام از view ها اضافه کنید.



شکل ۱-۲۲ طراحی فرم با استفاده از کنترل ها

5- بر روی فرم کلیک مضاعف کنید و به code_behind صفحه وارد شوید. از کد مثال ۱-۲ جهت تعریف عملکرد تمامی Button های صفحه استفاده کنید.

```
Protected Sub btnAllButtons_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles btnView1Next.Click, _
    btnView2Next.Click, btnView2Previous.Click, _
    btnView3Finish.Click, btnView3Reset.Click
    Select Case CType(sender, Button).Text
        Case "Next"
            MultiView1.ActiveViewIndex += 1
        Case "Previous"
            MultiView1.ActiveViewIndex -= 1
        Case "Finish"
```

```

Response.Write("You have registered as " & _
    txtFirstName.Text & _
    txtLastName.Text & "<br/>")
Response.Write("Birthday " & _
    Calendar1.SelectedDate)
btnView3Finish.Enabled = False
btnView3Reset.Enabled = False
Case "Reset"
    MultiView1.ActiveViewIndex = 0
End Select
End Sub

```

مثال: ۱-۲. قطعه کد مربوط به رویداد Click

6- خاصیت ActiveViewIndex کنترل Multi view جهت نمایش View ها بکار می رود. چنانچه این حالت برابر با صفر باشد در هنگام بار گذاری صفحه اولین View موجود در Multi view به نمایش در می آید.

7- کلید F5 را جهت اجرای برنامه فشار دهید. شکل ۱-۲۳ مراحل کار برنامه را نشان می دهد.



شکل ۱-۲۳. استفاده از کنترل Multi view

دقت کنید، به ازاء هر تغییر در کنترل View یک "Post Back" به سرور رخ خواهد داد.

۱-۷-۲. پاورقی بخش ...

... روشی بهتر جهت جمع آوری اطلاعات کاربر

اگرچه در این بخش با استفاده از کنترل MultiView توانستیم یک صفحه بزرگ را به چندین کوچکتر تقسیم کنیم؛ اما عیب MultiView این است که خودتان باید کاربر را جهت جابجایی بین View های مختلف هدایت کنید. (مثلا از طریق دکمه های Next و Previous) به همین جهت برای اعمالی چون جمع آوری اطلاعات کاربران استفاده از کنترل Wizard میتواند بهتر باشد.

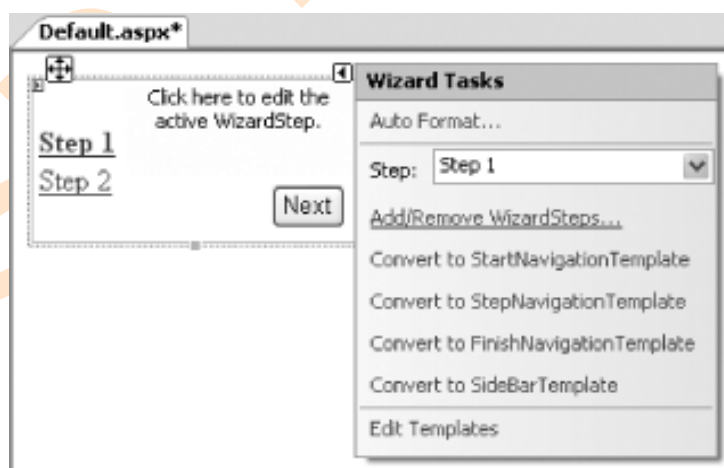
کنترل Wizard را در Tool Box می توانید پیدا کنید.

1- VS 2005 را باز کنید و پروژه جدیدی با نام و آدرس `C:\ASPNET20\chap01-Wizard` ایجاد کنید.

2- کنترل wizard را به فرمتان اضافه کنید.

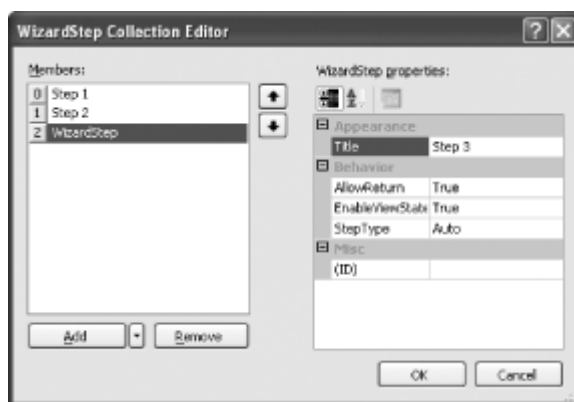
4- جهت اضافه کردن مرحله جدید در پنجره Wizard Tasks بر روی لینک Add/Remove

WizardSteps... کلیک کنید. (شکل ۱-۲۴) (بطور پیش فرض دو مرحله برایتان ساخته می شود)



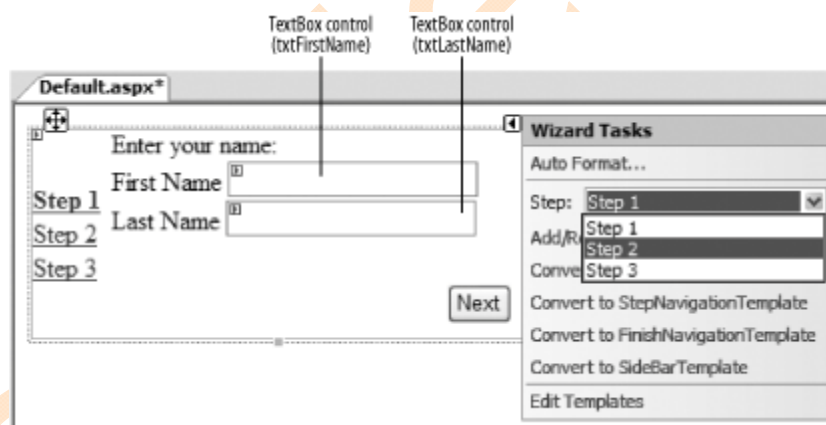
شکل ۱-۲۴. استفاده از کنترل Wizard

5- در پنجره WizardStep Collection دکمه Add را کلیک کنید. سپس درون تکست باکس Title عبارت "step 3" را تایپ کنید (شکل ۱-۲۵) و ok کنید.



شکل ۱-۲۵. اضافه کردن Step جدید

5- به Step 1 دو کنترل TextBox اضافه کنید. (شکل ۱-۲۶) برای رفتن به مرحله بعدی از منوی Wizard Tasks گزینه Step 2 را انتخاب کنید.



شکل ۱-۲۶. طراحی Step 1

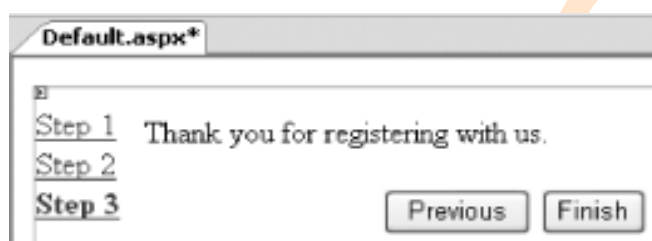
یادداشت : کنترل Wizaed قابلیت سفارشی سازی (customizable) بالایی دارد. از طریق پنجره Properties می توانید به امکانات مختلف این کنترل دست یابید.

6- Step2 همانند شکل ۱-۲۷ طراحی کنید.



شکل ۱-۲۷ طراحی Step2

7- در step3 متن نوشته شده در عکس ۱-۲۸ را بنویسید.

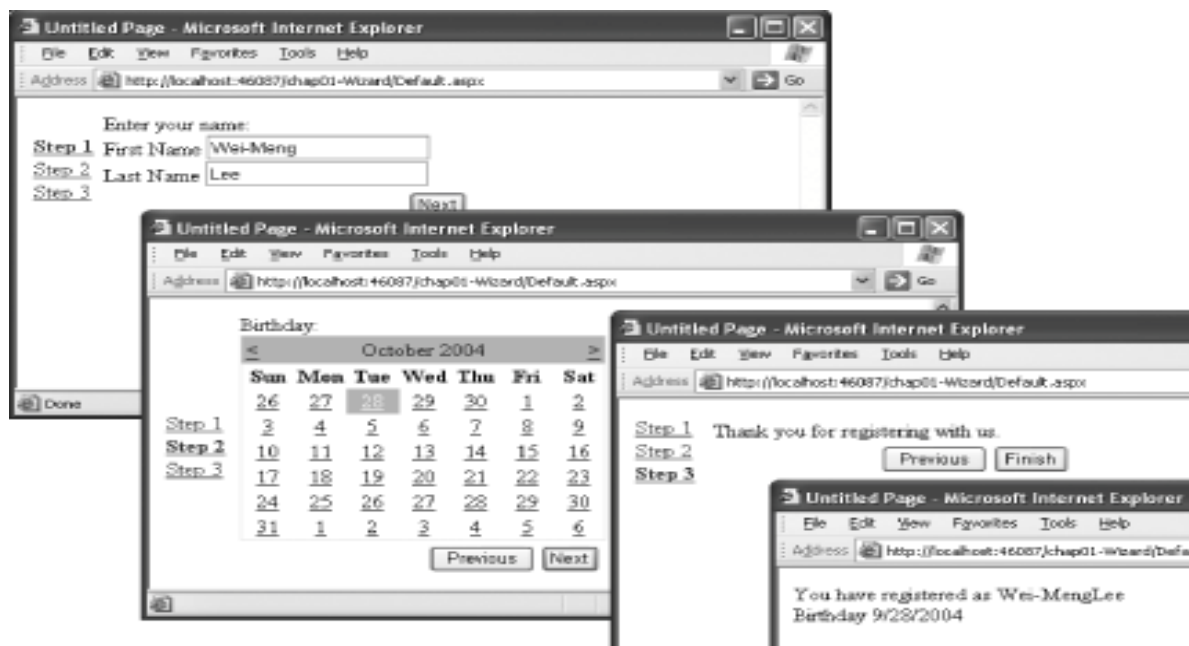


شکل ۱-۲۸ طراحی Step3

8- بر روی کنترل wizard دوبار کلیک کنید و کد زیر را جهت تعریف دکمه Finish در آخرین مرحله (Step3) به آن اضافه کنید :

```
Protected Sub Wizard1_FinishButtonClick(ByVal sender As Object, _
    ByVal e As System.Web.UI.WebControls.WizardNavigationEventArgs) _
    Handles Wizard1.FinishButtonClick
    Response.Write("You have registered as " & _
        txtFirstName.Text & _
        txtLastName.Text & "<br/>")
    Response.Write("Birthday " & _
        Calendar1.SelectedDate)
    Wizard1.Visible = False
End Sub
```

9- کلید F5 را فشار دهید تا خروجی برنامه را مشاهده کنید. (شکل ۱-۲۹)



شکل ۱-۲۹. خروجی برنامه

نکته: در این مثال هنگامی که کاربر از دکمه های Next , Previous استفاده می کند، هیچگونه عمل Postback صورت نمی پذیرد. اما اگر کاربرد در Step 2 بر روی تقویم (Calendar) کلیک کند یک Postback رخ می دهد.

۱-۷-۳. منابع اطلاعاتی :

در VS 2005 Help اطلاعات جامع تری در مورد قابلیت های کنترل های Wizard , Multi View می توانید پیدا کنید.

۱-۸. ارسال (Upload) فایل به سرور از طریق سایت

Asp. Net 2.0 از طریق کنترل File Upload امکان ارسال فایل بر روی وب سرور را برای کاربران فراهم آورده است. بعنوان مثال دانش آموزان یک مدرسه نیاز دارند پروژه ها و تکالیفشان را جهت بررسی و دریافت نمره به سرور مدرسه ارسال کنند.

کنترل File Upload را می توانید در Toolbox پیدا کنید.

یادداشت: ارسال فایل به سایت از طریق کنترل FileUpload بسیار آسان شده است.

۱-۸-۱. مراحل انجام کار :

برنامه نویسان Visual Studio کنترل File Upload را از طریق قرار دادن یک کنترل Text box و یک Button در برنامه نمایش می دهند. برای ارسال فایل شما نیازمند تعریف یک رویداد (همانند کلیک یک دکمه) هستید. (دکمه Submit شکل ۱-۳۰ را ببینید) در مثالی که ارائه خواهد شد شما برنامه ای طراحی می کنید که به کاربر اجازه می دهد فایلهایش را در شاخه (Directory) خاصی بر روی سرور ذخیره کند. در این برنامه شما همچنین اندازه فایل ارسال شده کاربر را چک می کنید تا کاربر فایلهای بزرگ تر از حد مجاز را ارسال نکند.

1- VS 2005 را باز کنید و پروژه ای با نام و آدرس `C:\ASPNET20\chap01-FileUpload` ایجاد کنید.

2- از Tool Box یک کنترل File Upload به فرمتان اضافه کنید.

3- یک کنترل Button به فرمتان اضافه کنید و گزینه Text آنرا به Submit و نامش را به btnSubmit تغییر دهید.

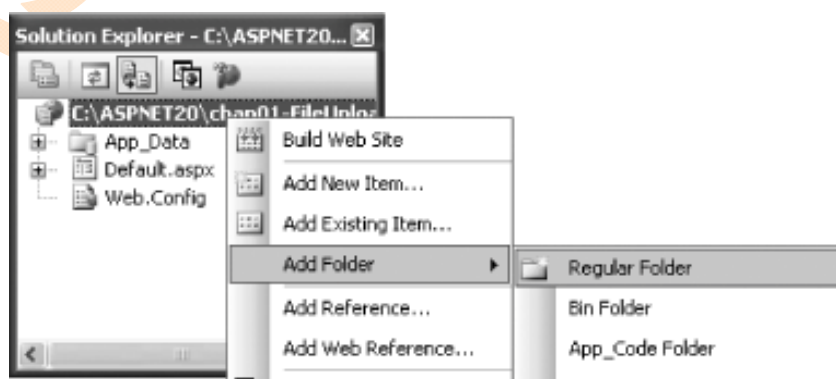
4- فرم شما بایستی همانند شکل ۱-۳۰ شده باشد.



شکل ۱-۳۰. کنترل File Upload بر روی فرم

یادداشت : به خاطر داشته باشید یک دکمه Submit جهت کار با کنترل FileUpload به صفحه اضافه کنید.

5- بر روی نام پروژه در Soultion Explorer راست کلیک کنید. و گزینه Add Folder → Regular Folder را انتخاب کنید (شکل ۱-۳۱ را ببینید). نام شاخه (folder) را "Uplaods" قرار دهید. از این شاخه جهت ذخیره فایل های ارسالی کاربران استفاده خواهیم کرد.



شکل ۱-۳۱. افزودن شاخه جدید به برنامه

6- بر روی دکمه submit کلیک مضاعف کنید و کد زیر را به آن اضافه کنید:

```
Sub btnSubmit_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles btnSubmit.Click

    ' get the application path
    Dim savePath As String = Request.PhysicalApplicationPath
    ' uploads to a special upload folder
    savePath += "uploads\"
    If FileUpload1.HasFile Then ' verify if there is file to upload
        savePath += FileUpload1.FileName
        ' existing file will be overwritten
        FileUpload1.SaveAs(savePath)
        Response.Write("File uploaded successfully!")
    Else
        Response.Write("No file to upload")
    End If
End Sub
```

نکته: از متد () `FileUpload1.SaveAs` جهت ذخیره فایل در یک شاخه مشخص استفاده می کنیم. چنانچه فایلی با همین نام قبلاً ارسال شده باشد این متد فایل جدید را جایگزین قبلی می کند. از اینرو ضروریست حتماً قبل از ذخیره فایل بر روی سرور برخی موارد را چک کنید.

تعریف Event Handlers در ASP. Net 2.0:

ASP. Net 2.0 شیوه جدیدی برای تعریف Event Handlers فراهم آورده است که در Asp. Net 1.x وجود ندارد. برای استفاده از این تکنیک جدید ابتداء خاصیتی به فرمتان اضافه می کنید که رویداد (Event) مورد نظرتان را مشخص می کند، سپس کد مربوط به رویداد را در `code_behind` صفحه وارد می کنید. بعنوان مثال فرض کنید می خواهیم رویداد click دکمه `btnSubmit` را به این شیوه پیاده سازی کنیم؛ ابتداء در source view فرم کد زیر را وارد کنید:

```
<asp:Button ID="btnSubmit"
    runat="server" Text="Submit"
```

```
OnClick="Submit_Click" />
```

حال کد مربوط به رویداد click را در code_behind صفحه وارد کنید. همانند زیر:

```
Protected Sub Submit_Click(ByVal sender As Object, _  
    ByVal e As System.EventArgs)  
    ' code to handle the Submit button click  
End Sub
```

همانطور که مشاهده می کنید، با این تکنیک کلمه کلیدی (key word) مربوط به handles حذف شد.

با استفاده از خاصیت PhysicalApplicationPath از شیء Request (object) می توانید مسیری را که برنامه در آن قرار دارد پیدا کنید.

خاصیت HasFile از کنترل FileUpload مشخص می کند که آیا کاربر فایلی را برای ارسال انتخاب کرده است یا خیر. فایل انتخاب شده در خاصیت FileName کنترل FileUpload ذخیره می شود.

۱-۸-۲. پاورقی بخش ...

... محدود کردن اندازه فایل های ارسالی

به دلایل امنیتی بهتر است اندازه فایل های ارسالی کاربران را محدود کنید. با استفاده از خاصیت ContentLength می توانید اندازه فایل ارسالی کاربر را چک کنید. به مثال زیر توجه کنید:

یادداشت: حتماً اندازه فایل های ارسالی کاربران را چک کنید، عدم انجام این کار امکان ارسال فایل های بزرگ را به کاربر می دهد که نتیجه آن کند شدن سرور است.

```
' ensure size if below 3MB
```

```
If FileUpload1.PostedFile.ContentLength <= 3145728 Then
```

```
    savePath += FileUpload1.FileName
```

```
    ' existing file will be overwritten
```

```
    FileUpload1.SaveAs(savePath)
```

```
    Response.Write("File uploaded successfully!")
```

```
Else
```

```
    Response.Write("File size exceeds 3MB.")
```

```
End If
```

۱-۸-۳. منابع اطلاعاتی :

چنانچه می خواهید در مورد ارسال فایل (FileUpload) در ASP.Net 1.x اطلاعات بیشتری بدست آورید مقاله زیر را مطالعه کنید.

<http://www.ondotnet.com/pub/a/dotnet/2002/04/01/asp.html>

۱-۹. ساخت Image Map

وب سایت های مدرن از Image Map ها جهت هدایت کاربران به بخشهای مختلف سایت و یا انجام اعمال مختلف استفاده می کنند. بعنوان مثال فرض کنید نقشه ای از جهان را در اختیار دارید که می خواهید با کلیک کاربر بر روی هر قسمت از نقشه اطلاعات مربوط به یک کشور را برای کاربر نمایش دهید. Asp.Net 2.0 از طریق کنترل جدید Image Map این امکان را فراهم آورده است .

این کنترل را مانند کنترل های دیگر می توانید در Toolbox پیدا کنید .

یادداشت : از کنترل جدید Image Map برای نمایش یک Image Map در سایتان استفاده کنید .

۱-۹-۱. مراحل انجام کار :

برای اینکه با شیوه کار Image Map آشنا شوید با استفاده از این کنترل یک کامپیوتر جیبی (PocketPc) طراحی می کنیم شما با مشخص کردن ۳ قسمت مختلف بر روی عکس PocketPc به کاربر این امکان را می دهید که با کلیک بر روی هر قسمت اطلاعات دیگری مشاهده نماید.

- 1- در VS2005 پروژه جدیدی با نام و آدرس `C:\ASPNET20\chap01-ImageMap` ایجاد کنید .
- 2- از طریق Toolbox یک کنترل Image Map به فرمتان اضافه کنید .
- 3- بر روی نام پروژه Soultion Explorer راست کلیک کرده و شاخه ای با نام Images جهت ذخیره عکسهای پروژه به برنامه اضافه کنید . عکس `ellaxim.jpg` (شکل ۱-۳۲) را در این شاخه ذخیره کنید :

`C:\ASPNET20\chap01-ImageMap\Images\`



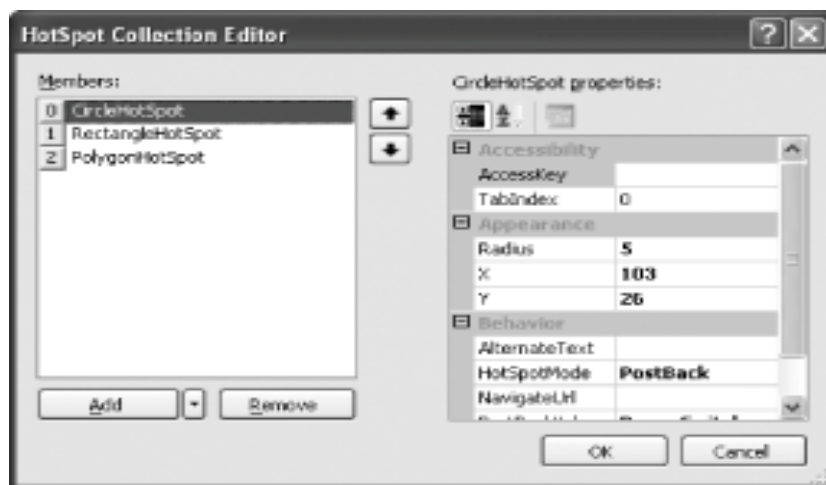
شکل ۱-۳۲ نمایش یک PoocketPc از طریق کنترل Image Map

۴- برای اینکه عکس را در کنترل Image Map نمایش دهید خاصیت ImageURL کنترل Image Map را برابر *Images/dellaxim.jpg* قرار دهید اکنون Image Map ؛ PocketPc نشان داده شده در شکل ۱-۳۲ را نشان می دهد .

عملکرد Image Map بر اساس یک یا چند Hotspot است که هر کدام عملی را انجام می دهند . بعنوان مثال ممکن است یکی کاربر را به صفحه دیگر هدایت کند و دیگری بر روی عکس خود کنترل ایجاد تغییر کند . برای تعریف HotSpot ها خاصیت Hotspots کنترل Image Map را از پنجره Properties آن انتخاب کنید . سپس بر روی دکمه Collections کلیک کنید تا پنجره HotsPot Collections Editor باز شود . (شکل ۱-۳۲)

کنترل ImageMap سه نوع Hotspots را پشتیبانی می کند :

دایره ای (*CircleHotSpot*) ، چهار ضلعی (*RectangleHotSpot*) و چند ضلعی (*PolygonHotSpot*) از طریق منوی کنار دکمه Add می توانید Hotspot مورد نظرتان را اضافه کنید . هنگامی که یک Hotspot را اضافه می کنید پنجره Properties مربوط به آن نمایش داده می شود . از طریق این پنجره می توانید اندازه و محل قرار گیری Hot spot را مشخص کنید . شکل ۱-۳۳ پنجره Properties مربوطه به یک Circle HotsPot را نشان می دهد که دارای شعاع ۵ می باشد و در مختصات ۱۰۳ و ۲۶ از عکس قرار گرفته است .



شکل ۱-۳۳ اضافه کردن Hotspot به Image Map

چنانچه می خواهید Hot spot های مشخص شده را بصورت دستی ایجاد کنید . به Source view رفته و آنها را همانند کد زیر ایجاد کنید :

```
<asp:ImageMap ID="ImageMap1" runat="server"
    ImageUrl="Images/dellaxim.jpg">
    <asp:CircleHotSpot HotSpotMode="PostBack"
        X="103" Y="26" PostBackValue="PowerSwitch"
        Radius="5" />
    <asp:RectangleHotSpot Top="48" Bottom="242"
        Left="30" NavigateUrl="./screen.aspx"
        Right="177" />
    <asp:PolygonHotSpot
        Coordinates="92,273,112,273,123,287,112,303,92,303,83,287"
        PostBackValue="Navigation" HotSpotMode="PostBack" />
</asp:ImageMap>
```

نکته : می توانید از Microsoft Paint جهت پیدا کردن مختصات قسمتهای مختلف عکس استفاده کنید . برای این کار کافیت عکس را در paint باز کنید و موس (mouse) را بر روی نقطه ای از عکس که می خواهید برایش Hotspot تعریف کنید قرار دهید سپس مختصات کرسر (Cursor) موس را از انتهای پنجره بخوانید .

چنانچه مراحل بالا را بصورت کامل انجام داده باشید سه Hotspot همانند شکل ۱-۳۴ ایجاد کرده اید .



شکل ۱-۳۴. Hotspot های تعریف شده در Image Map

یادداشت: کنترل Image Map از سه نوع Hotspot پشتیبانی می کند. دایره ای، چهار ضلعی و چند ضلعی.

شکل ۱-۳۴ یک Image Map را به همراه سه Hotspot نشان می دهد: یک CircleHotSpot یک RectangleHotSpot و یک PolygonHotSpot.

CircleHotSpot کلید روشن و خاموش شدن PocketPc را شبیه سازی کرده است RectangleHotSpot صفحه PocketPc را نشان می دهد و PolygonHotSpot کلیدهای کنترلی PocketPc را شبیه سازی نموده است.

هر Hot pot چهار حالت رفتاری دارد که در خاصیت HotSpotMode مشخص می شوند:

Not Set

چنانچه HotSpotMode با گزینه "Not Set" تنظیم شده باشد: Hotspot بر اساس HotSpotMode تنظیم شده برای Image Map رفتار می کند.

Navigate

با انتخاب این گزینه، هنگام فشردن Hotspot صفحه دیگری بارگذاری می شود.

PostBack

یک PostBack به سرور رخ می دهد. این گزینه هنگامیکه نیازمند انجام عملی هستید مؤثر است. (به عنوان مثال افزایش یک متغیر)

Inactive

Hot spot را غیر فعال می کند.

نکته: خاصیت HotSpotMode کنترل Image Map رفتار پیش فرض تمامی Hot spot ها را مشخص می کند.

هر Hotspot از طریق خاصیت HotSpotMode خودش می تواند این تنظیم را بازنویسی (Override) کند.

جهت نمایش چگونگی ارسال Post back به سرور توسط یک Hotspot خاصیت HotSpotMode از CircleHotSpot را Postback قرار دهید و خاصیت Postback Value را Power Switch بگذارید. حال برای دریافت مقدار ارسالی به سرور رویداد زیر را برای کنترل Image Map بنویسید.

```
Protected Sub ImageMap1_Click(ByVal sender As Object, _  
    ByVal e As System.Web.UI.WebControls.ImageMapEventArgs) _  
    Handles ImageMap1.Click  
    Dim str As String = "You have clicked " & e.PostBackValue  
    Response.Write(str)  
End Sub
```

5- فرم جدیدی به پروژه تان اضافه کنید (بر روی نام پروژه در Solution Explorer راست کلیک و گزینه های Add New Item... پس Web Form را انتخاب کنید) و نام آنرا Screen.aspx قرار دهید. این صفحه هنگام کلیک Hotstop نمایش داده می شود.

6- خاصیت Hotspot mode از RectangleHotSpot را Navigate و خاصیت NavigateURL آنرا Screen.aspx قرار دهید.

7- کلید F5 را فشار دهید تا برنامه اجرا شود. به حالت موس در هنگام قرار گرفتن بر روی Hotspot های مختلف دقت کنید.

۱-۹-۲ پاورقی بخش ...

... افزودن Tool Tip به Hotspot ها

با استفاده از خاصیت AlternateText می توانید به کنترل های مختلف یک Image map، Tooltip اضافه کنید. افزودن Tooltip به Hotspot ها قابلیت بکارگیری برنامه ها را بالا می برد و کاربر جزئیات بیشتری در مورد یک Hotspot بدست می آورد. شما می توانید در Source view متن Tooltip مورد نظر تان را اضافه کنید:

```
<asp:CircleHotSpot  
    HotSpotMode="PostBack"  
    X="103" Y="26" Radius="5"
```

```
PostBackValue="PowerSwitch"
AlternateText="Power Switch" />
```

همچنین می توانید از طریق برنامه نویسی این کار را انجام دهید :

```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load
    ImageMap1.HotSpots(0).AlternateText = "Power Switch"
    ImageMap1.HotSpots(1).AlternateText = "Screen"
    ImageMap1.HotSpots(2).AlternateText = "Navigation"
End Sub
```

زمانیکه کاربر موس را بر روی Hotspot می برد متن Tooltip نمایش داده می شود. (شکل ۱-۳۵)



۱-۳۵. نمایش Tooltip

۱-۹-۳. منابع اطلاعاتی :

علاوه بر Imagemap می توانید با استفاده از ImageButton یک Imagemap برای سایتتان طراحی کنید. جهت آشنایی با چگونگی انجام این کار به آدرس زیر مراجعه کنید.

http://samples.gotdotnet.com/quickstart/aspplus/samples/webforms/ctrlref/webctrl/imagebutton/doc_imagebut.aspx

فصل دوم

Master Pages ها

و

Site Navigations

یکی از مهمترین امکاناتی که توسط Asp.Net در VS 2005 فراهم آمده است، پشتیبانی از ارث بری صفحات است. همانند ارث بری فرم های ویندوز (windows form) در Asp.Net 2.0 می توانید یک Master page شامل تعدادی عنصر و کنترل؛ داشته باشید بطوریکه سایر صفحات از این صفحه استفاده نمایند. Master page ها به شما کمک می کنند تا فرمهای وب با ظاهری یکسان طراحی کنید و نهایتاً وب سایتی با ظاهر همگون و منظم داشته باشید.

نکته: پیشنهاد می شود Master page ها را جایگزین Web User control ها ننمایند.

قابلیت جدید Master page در Asp.Net 2.0 محدودیت های استفاده از User control را از میان برداشته است. در Asp.Net 2.0 ابتداء یک Master page طراحی می کنید سپس هر صفحه از سایتتان را با ارث بری از Master page طراحی شده، می سازید.

یادداشت: با استفاده از Master page ها قادرید سایت های با ظاهر همسان و منظم داشته باشید.

۱-۲. طراحی یک Master Page

برای آشنایی با نحوه کار Master page ها وب سایتی را به کمک Master page طراحی می کنیم. در این بخش شما Master page ای به همراه چند کنترل ایجاد خواهید کرد.

1- در VS 2005، پروژه جدیدی با نام و آدرس `C:\ASPNET20\chap02-MasterPages` ایجاد کنید.

2- شاخه ای با نام Images به پروژه تان اضافه کنید. (بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های `Add Folder → regular Folder` را انتخاب نمایید).

تصاویر نمایش داده شده در شکل ۱-۲ را در شاخه ساخته شده کپی کنید. از این عکسها برای ساخت Master page استفاده می کنیم.

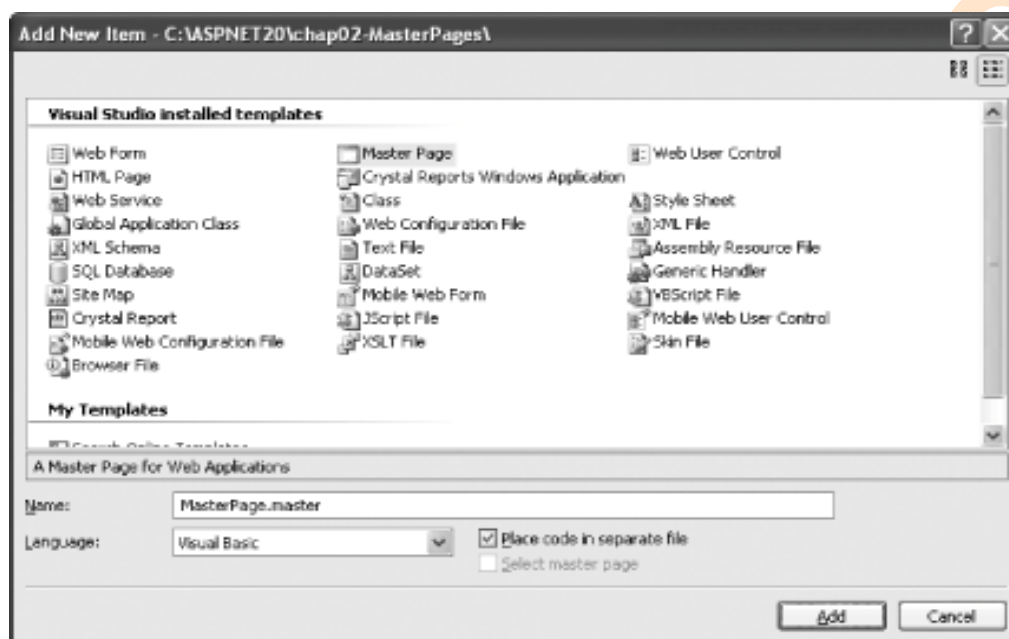
نکته: عکسها را می توانید از آدرس <http://www.oreilly.com/catalog/aspnetadn/> دانلود کنید.



شکل ۲-۱. عکسهای موجود در شاخه Images

3- اکنون یک Master Page به پروژه اضافه کنید. برای این کار بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه Add New Item... را انتخاب کنید.

4- در پنجره Add New Item گزینه Master page را با نام پیش فرض *MasterPage.master* انتخاب کنید، همانند شکل ۲-۲. در انتها بر روی دکمه Add کلیک کنید.



شکل ۲-۲. افزودن یک Master page به پروژه

5- در Solution Explorer بر روی Master page.Master کلیک مضاعف کنید. صفحه ای که مشاهده می کنید تنها حاوی یک کنترل ContentPlaceholder می باشد.

یادداشت: کنترل ContentPlaceholder محل نگهداری فرم هایی است که از Master page ارث می برند.

6- بر روی Master page دو کنترل Image، یک Panel و چهار Button link قرار دهید. همانند شکل ۲-۳. خواص هر کدام از کنترلها را به صورت زیر تنظیم کنید:

- کنترل های Image:

نام اولین کنترل Image را ImageLogo و خاصیت ImageURL آنرا *Images/oreilly_header.gif* قرار دهید. دومین کنترل Image را با نام imgTitle و آدرس آنرا *Images/oranet_logo.jpg* تنظیم کنید.

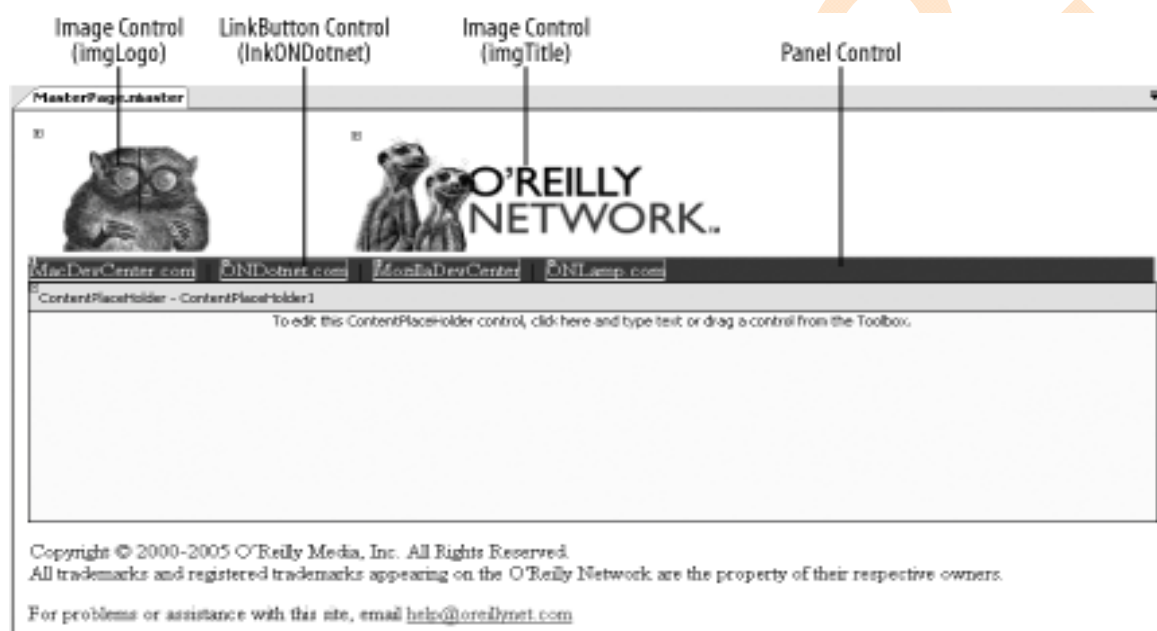
- کنترل Panel:

با استفاده از موس Panel را سرتاسر صفحه بکشید و خاصیت Backcolor آنرا به Maroon تغییر دهید.

کنترل‌های linkButton:

چهار کنترل Link Button را به درون Panel انتقال دهید و همانند شکل ۲-۴ آنها را نامگذاری کنید. (از طریق خاصیت Text آنها) خاصیت PostbackUrl از link Button دوم (InkONDotnet) را [ONDotnet.aspx](#) قرار دهید.

همانند شکل ۲-۳ متنی را در انتهای صفحه پس از کنترل ContentPlaceHolder تایپ کنید. برای این کار در انتهای ContentPlaceHolder کلید Enter را فشار دهید سپس شروع به تایپ کنید.



شکل ۲-۳. طراحی Master Page

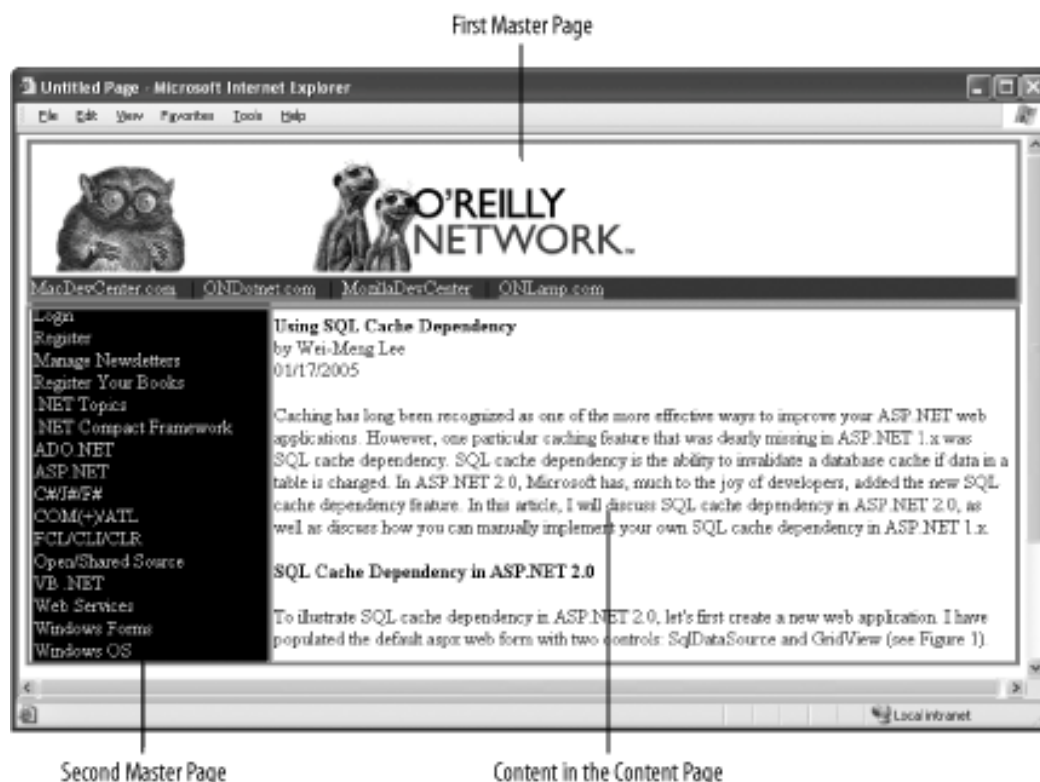
هم اکنون Master page شما ساخته شده است، در بخش بعد نحوه استفاده از آنرا خواهید آموخت.

۲-۱-۱. پاروقی بخش ...

... Master page های تو در تو

قرار دادن یک Master page درون Master page دیگر می تواند هنگامی که می خواهید قسمت مشخصی از ویجت را تغییر دهید، مفید واقع شود. بعنوان مثال فرض کنید بخواهیم در برخی صفحات لینکهای هدایت کننده جداگانه ای نمایش دهیم.

شکل ۲-۴ نحوه انجام این کار را از طریق صفحه ای با دو Master page که دومی درون اولی قرار گرفته است، نشان می دهد.



شکل ۲-۴. Master page تو در تو

برای اینکه ببینید چگونه یک Master page درون دیگری قرار می گیرد مراحل زیر را دنبال کنید:

- 1- یک Master page جدید به پروژه (C:\ASPNET20\chap02-MasterPages) اضافه کنید. و نام آنرا Masterpage2.Master قرار دهید.
 - 2- Master page دوم حاوی یک جدول (table) با یک سطر و دو ستون می باشد. ستون اول شامل یک کنترل content (Content1) است که کنترل ContentPlaceholder از Master page اول را نگهداری می کند (ContentPlaceholder1) و حاوی عبارات (جهت سهولت از متن به جای لینک استفاده شده است) Register, login و... است.
 - ستون دوم شامل کنترل ContentPlaceholder (ContentPlaceholder2) مربوط به فرم است.
- در source view مربوط به Master page قطعه کد مثال ۲-۱ را اضافه کنید.

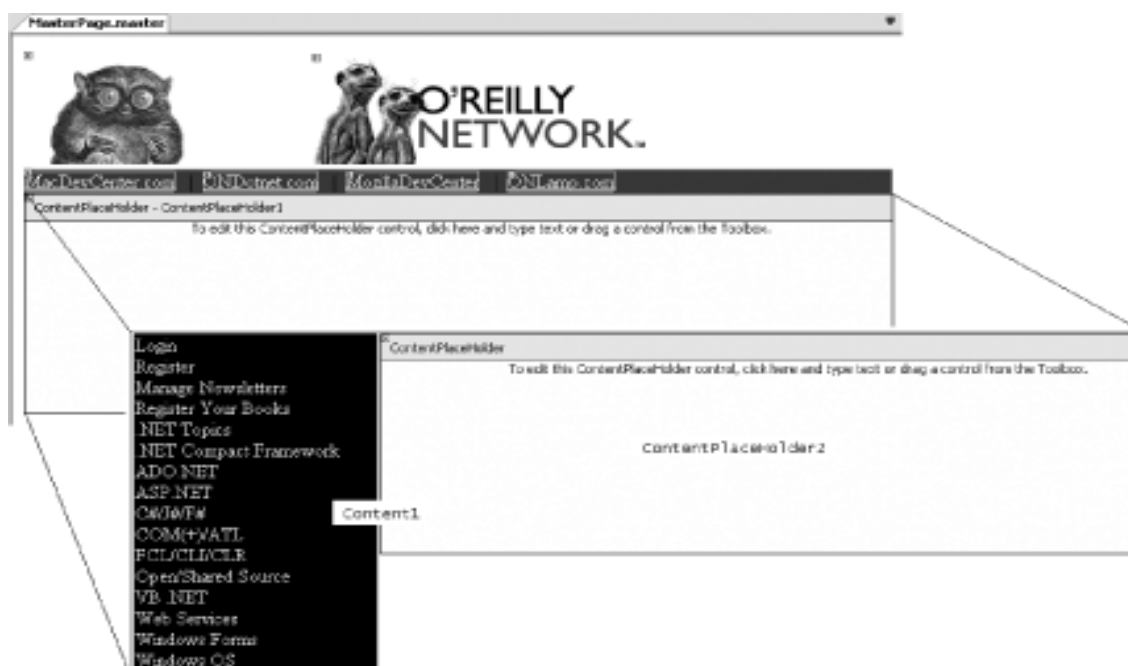
نکته: VS 2005 ساخت و ویرایش Master page های تو در تو را در حالت Design پشتیبانی نمی کند. این کار را بایستی در حالت Source view انجام دهید.


```
<%@ Master Language="VB" MasterPageFile="~/MasterPage.master"
    CodeFile="MasterPage2.master.vb" AutoEventWireup="false"
    Inherits="MasterPage2_master" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
    Runat="Server">
<table>
    <tr>
        <td width=200 bgcolor=black>
            <font color=white>
                Login<br />Register<br />Manage Newsletters<br />
                Register Your Books<br />.NET Topics<br />
                .NET Compact Framework<br />ADO.NET<br />
                ASP.NET<br />C#/J#/F#<br />COM(+)/ATL<br />
                FCL/CLI/CLR<br />Open/Shared Source<br />
                VB .NET<br />Web Services<br />Windows Forms<br />
                Windows OS<br />
            </font>
        </td>
        <td>
            <asp:contentplaceholder id="ContentPlaceHolder2" runat="server">
                <asp:contentplaceholder>
            </td>
        </tr>
    </table>
</asp:Content>
```

مثال ۲-۱. ساخت Master Page های تو در تو

۳. شکل ۲-۵ ارتباط بین دو Master page را نشان می دهد.



شکل ۲-۵ ارتباط میان دو Master page

در بخش بعد به بررسی نحوه استفاده از Master page ساخته شده در این بخش خواهیم پرداخت.

۲-۱-۲. منابع اطلاعاتی :

برای آشنایی با ساخت Master page توسط Visual Web Developer 2005 Express به آدرس زیر مراجعه کنید:

<http://beta.asp.net/GuidedTour/s15.aspx>

۲-۲. بکارگیری Master page بعنوان قالب (Template)

یادداشت : با استفاده از Master page ها و ارث بری صفحات از آنها می توانید وب سایت های با صفحات همگون طراحی کنید.

تنها کفایت یک مرتبه یک Master page طراحی کنید و سپس از آن به عنوان قالب صفحات استفاده کنید.

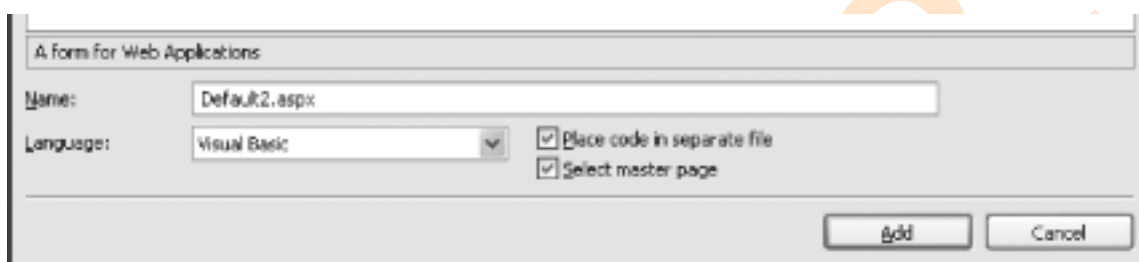
۲-۲-۱. مراحل انجام کار:

یادداشت : Content page همان فرمی است که از Master page استفاده می کند.

در این بخش چندین Content page به پروژه اضافه می کنیم و Master page ساخته شده در بخش قبل را بعنوان قالب برای آنها تعریف می کنیم.

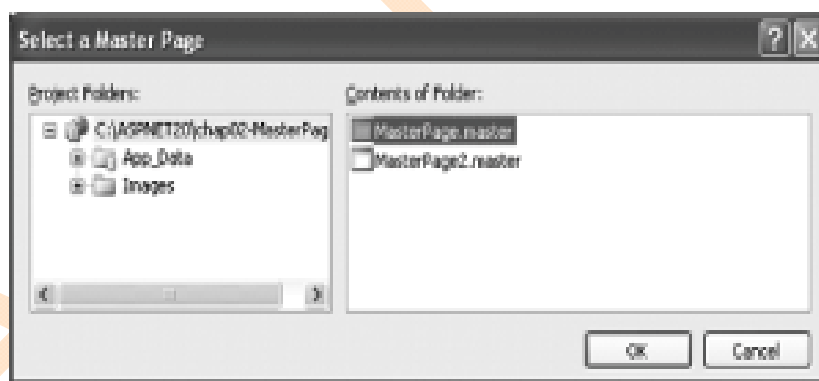
سپس چندین کنترل به Content page اضافه خواهیم کرد و نحوه ترکیب Master page و Content page ها را در زمان اجرا مشاهده خواهید نمود.

1- ابتدا صفحه جدیدی به پروژه بخش قبل اضافه کنید. برای این کار بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه Add New Item... سپس web form را انتخاب کنید. نام صفحه را Default2.aspx قرار دهید. چک باکس Select Master page انتهای کار را انتخاب کنید. (شکل ۶-۲)



شکل ۶-۲. ساخت Content page و انتخاب Master page برای آن

2- در پنجره دیگری از شما خواسته می شود تا Master page مورد نظران را انتخاب کنید. در این قسمت MasterPage.master را انتخاب کنید. (شکل ۷-۲ را ببینید)



شکل ۷-۲. انتخاب Master Page

یادداشت: یک Content page تنها می تواند یک Master page داشته باشد.

3- دکمه ok را کلیک کنید. Visual Studio صفحه Default2.aspx را بصورت بخشی از Master page به شما نشان می دهد. (شکل ۸-۲) همانطور که مشاهده می کنید محتویات Master page در صفحه Default2.aspx نمایش داده می شوند اما در این صفحه قابل ویرایش نیستند. همچنین توجه داشته باشید که صفحه توسط یک کنترل Content ایجاد شده است.



شکل ۲-۸. ساخت یک Content page

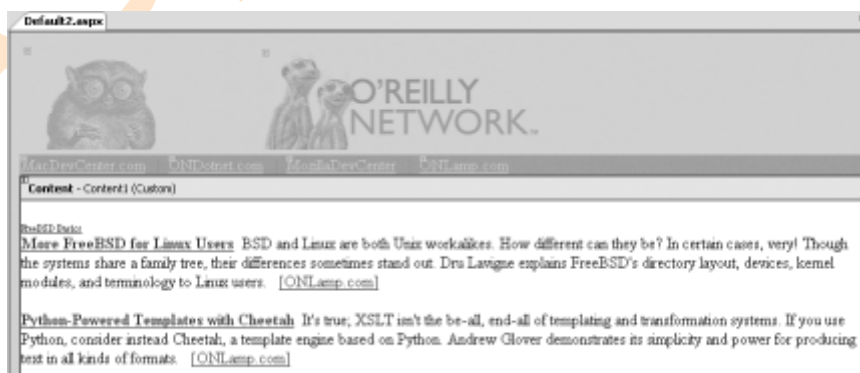
4- می توانید صفحه Default2.aspx را از طریق افزودن کنترل های مورد نظر تان به Content به شکل دلخواه خود در آورید. فراموش نکنید که Master page در این صفحه قابل ویرایش نیست.

یادداشت: Content control مکانیست که Content page شما در آنجا قرار می گیرد.

نکته: برای ویرایش Master page می توانید بر روی هر قسمت از صفحه که مربوط به Master page است راست کلیک کنید و گزینه Edit Master را انتخاب نمایید و یا اینکه در Solution Explorer بر روی Master page دوبار کلیک کنید. هر دو عمل باعث نمایش Master page می شوند.

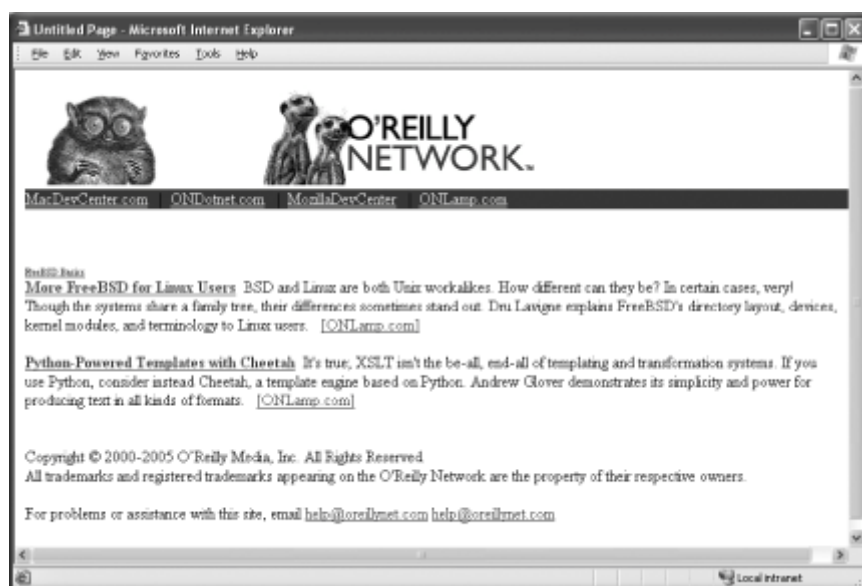
5- برای افزودن عناصر به کنترل Content تنها کافیست کنترل های مورد نظر را از Tool Box انتخاب و درون Content رها کنید. (Drag & Drop)

همانند شکل ۲-۹ چند خط متن درون Content تایپ کنید.



۲-۹ افزودن به Contentpage

6- کلید F5 را فشار دهید تا نتیجه کار را مشاهده کنید. صفحه شما بایستی همانند شکل ۲-۱۰ باشد.



شکل ۲-۱۰ نمایش Content page در Internet Explorer

نکته: به خاطر دارید که Master page دارای یک کنترل Content place Holder است که می توانید عناصر Content page را به آن اضافه کنید. اما دقت کنید اگر چه Content place Holder محتویات Content page در زمان بارگذاری صفحه نمایش می دهد اما Content page در زمان اجرا کنترل ها و عناصر موجود در Content خودش را جایگزین Content place Holder می کند. (Override)

۲-۲-۲. تشریح نحوه انجام کار:

در این برنامه یک فرم وب (Default2.aspx) ایجاد کردیم که از یک Master page (master . Master page) ارث می برد.

نکته جالب در مورد Master page این است که شما می توانید قالب یک صفحه را یکبار طراحی می کنید و سایر فرم ها از آن استفاده کنند و چنانچه تغییری در Master page ایجاد شود این تغییرات بصورت خودکار در تمامی صفحاتی که از آن ارث می برند نمایان می شوند.

برای اینکه صفحه ای را با Master page های تودرتو مشاهده کنید، فرم جدیدی به پروژهتان اضافه کنید. (نام آنرا SqlArticle.aspx قرار دهید) و صفحه MasterPage2.master را بعنوان Master page آن معرفی کنید.

در Source view صفحه SqlArticle.aspx کدهای Hmll و متون نمایش داده شده در شکل ۲-۱۱ را اضافه کنید.

```

SqlArticle.aspx
Client Objects & Events (No Events)
1 <%@ Page Language="VB" MasterPageFile="~/MasterPage2.master"
2   AutoEventWireup="false" CodeFile="SqlArticle.aspx.vb"
3   Inherits="Default4.aspx" title="Untitled Page" %>
4 <asp:Content ID="Content1"
5   ContentPlaceHolderID="ContentPlaceHolder2" Runat="Server">
6   <b>Using SQL Cache Dependency</b><br />
7   by Wei-Meng Lee<br />
8   01/17/2005<br />
9   <p>Caching has long been recognized as one of the more effective
10  ways to improve your ASP.NET web applications. However, one
11  particular caching feature that was dearly missing in ASP.NET 1.x
12  was SQL cache dependency. SQL cache dependency is the ability to
13  invalidate a database cache if data in a table is changed. In
14  ASP.NET 2.0, Microsoft has, much to the joy of developers, added
15  the new SQL cache dependency feature. In this article, I will
16  discuss SQL cache dependency in ASP.NET 2.0, as well as discuss
17  how you can manually implement your own SQL cache dependency in
18  ASP.NET 1.x. </p>
19
20  <b>SQL Cache Dependency in ASP.NET 2.0</b>
21  <p>To illustrate SQL cache dependency in ASP.NET 2.0, let's first
22  create a new web application. I have populated the default.aspx
23  web form with two controls: SqlDataSource and GridView (see Figure
24  1). </p>
25  </asp:Content>

```

شکل ۲-۱۱. ویرایش صفحه ای که از Master page های تودرتو استفاده می کند

هنگام بارگذاری صفحه پس از فشردن کلید F5 مشاهده می کنید که Master page ها با یکدیگر ترکیب شده اند. (شکل ۲-۱۲ را ببینید)



شکل ۲-۱۲. بارگذاری صفحه شامل Master page های تودرتو

... تبدیل Web Page به Content Page

اگر به Source یک فرم وب معمولی (که از Master page استفاده نمی کند) نگاه کنید متوجه یکسری کدهای استاندارد HTML می شوید. که انتظار دارید آنها را در صفحات Asp. Net پیدا کنید مانند عنصر <Form>:

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Default.aspx.vb"
Inherits="Default_aspx" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/
DTD/xhtml11.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Untitled Page</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>

    </div>
  </form>
</body>
</html>
```

یک Content page شامل عناصری که صفحات Asp.Net را ایجاد می کنند (مانند <html>, <body>, <form>) نیست؛ در عوض از یک عنصر <asp: content> تشکیل شده که کنترل Content و خصوصیات آنرا نمایش می دهد.

عنصر <asp: content> تمامی کنترل هایی که درون کنترل Content قرار می دهید نگهداری می کند.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
AutoEventWireup="false" CodeFile="Default3.aspx.vb"
Inherits="Default3_aspx" title="Untitled Page"
%>
<asp:Content ID="Content1"
ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
...
...
</asp:Content>
```


برای تبدیل Content page web form از خاصیت Master page file در رهنمود (Divective) صفحه استفاده کنید، سپس تمامی عناصر HTML صفحه را حذف کنید. کد زیر این کار را انجام می دهد:

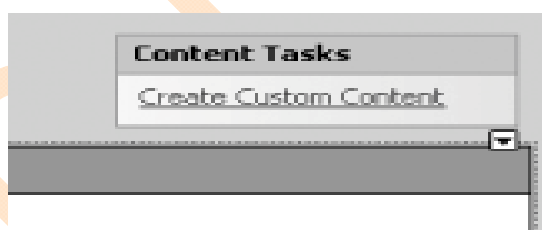
```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
AutoEventWireup="false" CodeFile="Default.aspx.vb"
Inherits="Default_aspx" %>
```

این کد صفحه Default.aspx را به یک Content page تبدیل می کند. از کد زیر نیز برای نمایش کنترل Content استفاده کنید:

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
AutoEventWireup="false" CodeFile="Default.aspx.vb"
Inherits="Default_aspx" %>
```

```
<asp:Content ID="Content1"
ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
</asp:Content>
```

همچنین می توانید این کار را با کلیک بر روی لینک Create Custom Content از Context Tasks menu در گوشه سمت راست Content control انجام دهید. (شکل ۲-۱۳)



شکل ۲-۱۳. ساخت کنترل Content جدید

نکته: اگر Content control را در کد برنامه مشخص نکنید. همچنان قادر به دیدن آن در Design view خواهید بود، ولی قادر به افزودن هیچ کنترلی به آن نیستید، مگر اینکه بر روی Create Custom Content در Context Tasks menu کلیک کنید.

... استفاده از یک Master page برای تمام سایت

برای اینکه تمامی صفحات سایت از یک Master page مشخص استفاده کنند کافست عنصر <pages> را در فایل Web. Config بصورت زیر اضافه کنید:


```
<system.web>
```

```
<pages masterPageFile="MasterPage.master" />
```

```
...
```

با این تعریف اگر در رهنمود صفحه ای هیچ Master page ای تعریف نشده باشد، آن صفحه از فایل Master page ارث خواهد برد.

چنانچه می خواهید صفحه از Master page دیگری ارث ببرد کافست نام Master page مورد نظر را در رهنمود صفحه وارد کنید تا جایگزین Master page شود.

... شناسایی مرورگر (browser) و تغییر Master page بر اساس نوع آن

چنانچه کاربران برنامه شما از مرورگرهای مختلفی استفاده کنند، می توانید برای هر مرورگر Master page خاصی در نظر بگیرید که برای آن مرورگر بهینه باشد. در این حالت بایستی در زمان اجرا به صورت پویا (Dynamic) بین Master page ها جابجا شوید. برای این کار کد زیر را به رویداد PreInt از Content page اضافه کنید.

در قطعه کد فوق فرض بر این است که سه Master page متفاوت دارید: *Master page* (برای IE) *MasterPage_FireFox.master* (برای fire fox) و *MasterPage_Others.master* (برای سایر مرورگرها).

صفحه نوع مرورگر درخواست کننده اش را شناسایی می کند و سپس Master page مربوطه به صفحه اعمال می شود.

بجای جابجایی Master page ها از طریق برنامه نویسی می توانید از طریق تعریف نیز این کار را انجام دهید:

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

```
MasterPageFile="~/MasterPage_Others.master"
```

```
ie: MasterPageFile="~/MasterPage.master"
```

```
mozilla: MasterPageFile="~/MasterPage_FireFox.master"
```

```
...
```

۲-۲-۴. منابع اطلاعاتی :

Scott Guthrie (مدیر فروش ASP. Net در میکروسافت) در وبلاگش مطالبی در مورد Master page ها بر اساس انواع مرورگرها نوشته است. که می تواند راهنمای خوبی برایتان باشد.

<http://blogs.msdn.com/scottgu/archive/2004/11/20/267362.aspx>

۲-۳. ویرایش Master page در زمان اجرا

هنگام بارگذاری؛ فرم وبی که از یک Master page استفاده می کند در زمان اجرا محتویات Master page را به همراه محتویات خودش نمایش می دهد.

اما زمانی وجود دارد که می خواهید بخشی از Master page را تغییر دهید.

بعنوان مثال در سایت O'Reilly (<http://www.oreillynet.com>) صفحاتی که به زیر دومین (Subdomain) OnDotNet تعلق دارند، Logo متفاوتی نسبت به صفحات اصلی سایت دارند. این مطالب را در سایت هایی همچون (Perl.com , onlamp.com) نیز می توانید ببینید.

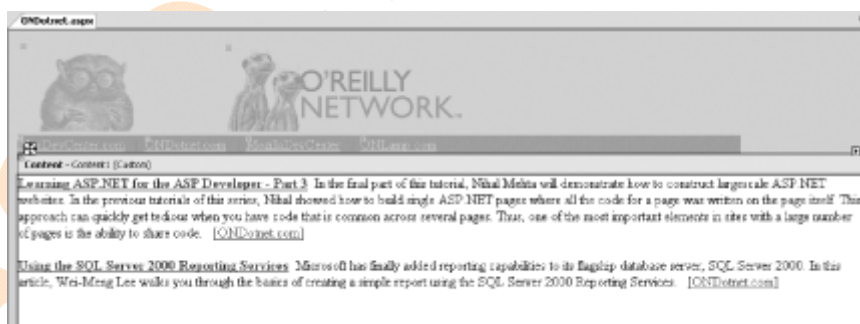
۲-۳-۱. مراحل انجام کار:

برای آشنایی با شیوه اصلاح Master page به صورتی که گفته شد صفحه ای به پروژه قبلی (بخش ۲-۲) اضافه می کنیم که از صفحه Master page استفاده می کند و از طریق آن logo موجود در Master page را تغییر خواهیم داد.

1- پروژه بخش قبلی را باز کنید و فرم جدیدی به آن اضافه کنید (بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه Add New Item... و سپس WebForm را انتخاب نمایید) چک باکس "select Master page" را تیک بزنید و نام صفحه را OnDontnet.aspx قرار دهید. سپس دکمه add را کلیک کنید.

2- فایل Masterpage.Master را بعنوان Master page صفحه انتخاب کنید.

3- صفحه OnDotNet.aspx را با متن پر کنید. (شکل ۲-۱۴ را ببینید)



شکل ۲-۱۴. صفحه OnDotNet.aspx

4- به Source view صفحه رفته و رهنمود Master page را پس از رهنمود page تایپ کنید. این رهنمود به شما امکان دسترسی به Master page را می دهد به طوری که از طریق برنامه نویسی بتوانید محتوای آنرا تغییر دهید. (همانند خواص متد های عمومی ((Public))

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
AutoEventWireup="false" CodeFile="ONDotnet.aspx.vb"
```

```
Inherits="ONDotnet_aspx" title="Untitled Page" %>
<%@ MasterType VirtualPath="~/MasterPage.master" %>
...
```

5- در code_behind از Master page (*MasterPage.master.vb*) متد `setLinkColor()` را اضافه کنید. از طریق این متد می توانید با برنامه نویسی رنگ خاصی را برای linkButton های Content page مشخص کنید.

```
Imports System.Drawing
Partial Class MasterPage_master
    Inherits System.Web.UI.MasterPage

    Public Sub setLinkColor(ByVal ctrl As String, _
        ByVal InkColor As Color)
        Select Case ctrl
            Case "MacDevCenter"
                InkMacDevCenter.ForeColor = InkColor
            Case "ONDotnet"
                InkONDotnet.ForeColor = InkColor
            Case "MozillaDevCenter"
                InkMozillaDevCenter.ForeColor = InkColor
            Case "ONLamp"
                InkONLamp.ForeColor = InkColor
        End Select
    End Sub
End Class
```

یادداشت: برای اینکه Content page بتواند متد `setLinkColor()` را فراخوانی کند بایستی این متد را به صورت عمومی تعریف کنید.

6- در code_behind صفحه `ONDotnet.Aspx.Vb` (`ONDotnet.Aspx.Vb`) رویداد `load (Event)` page را به صورت زیر وارد کنید:

```
Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load
    '---change the link color
    Master.setLinkColor("ONDotnet", Drawing.Color.Yellow)

    '---change the title
    Master.Page.Header.Title = _
        "O'Reilly and Associates Conferences"
```

```

'---change the image
Dim masterImage As Image
masterImage = CType(Master.FindControl("imgTitle"), _
    Image)
If Not (masterImage Is Nothing) Then
    masterImage.ImageUrl = "Images/ondotnet_logo.gif"
End If
End Sub

```

6- جهت ارجاع به Master page از خاصیت master استفاده کردیم. در کد بالا ابتدا رنگ linkButton مربوط به ONDotnet.com را با استفاده از متد (setLinkColor) تغییر دادیم. سپس عنوان صفحه را از طریق خاصیت Master.Page.Header.Title عوض نموده و در نهایت با استفاده از متد find control لوگوی (logo) موجود در Master page را تغییر می دادیم.

نکته: خاصیت master یک خاصیت ویژه است جهت دسترسی فرمهای وب به Master page.

8- دکمه F5 را فشار دهید. ابتدا صفحه Default. Aspx را لود کنید. سپس بر روی لینک ONDotnet.com کلیک کنید. شکل ۲-۱۵ تفاوت میان صفحات را نشان می دهد.



شکل ۲-۱۵ نمایش صفحات Default.aspx , OnDotNet.aspx

۲-۳-۲. تشریح نحوه انجام کار:

دو روش برای ویرایش محتویات Master page در زمان اجرا وجود دارد:

- فراخوانی متدهای عمومی و تنظیم خواص عمومی Master page.

- استفاده از متد (Find control) (از خاصیت master) جهت پیدا نمودن کنترل ها در صفحه و سپس فراخوانی متدها و جایگزین نمودن خواص کنترلها.

اخطار: خاصیت Master تنها در صفحاتی معتبر است که از طریق MasterPageFile در رهنمود صفحه به Master page ارجاع داشته باشند. چنانچه از خاصیت master در صفحه بدون ارجاع استفاده کنید، نتیجه دسترسی Null خواهد بود.

تا جائیکه ممکن است از روش اول یعنی استفاده از خواص و متدهای عمومی Master page استفاده کنید. این روش، روش بسیار مفیدتری برای تغییر کنترل های Master page است زیرا از انقیاد زودرس (early binding) استفاده می کند.

یادداشت: هر جا که امکان دارد از انقیاد زودرس استفاده کنید این تکنیک برنامه هایتان را ساده تر و کم خطا تر (Error – prone) می کند.

در روش دوم از طریق متد (findcontrol) ابتداء مکان کنترلی که باید اصلاح شود را پیدا می کنیم. سپس نام و نوع آنرا تغییر می دهیم.

مانند قطعه کد زیر که یک کنترل Image در Master page را با یک فایل جدید جایگزین می کند :

```
Dim masterImage As Image
masterImage = CType(Master.FindControl("imgTitle"), _
    Image)
If Not (masterImage Is Nothing) Then
    masterImage.ImageUrl = "Images/ondotnet_logo.gif"
End If
```

این روش از انقیاد دیررس برای دسترسی به کنترل های Master page استفاده می کند، چرا که نیازمند تبدیل کنترلها به نوع دلخواه در زمان اجرا هستید.

یادداشت: متد (FindControl) از انقیاد دیررس استفاده می کند در حالیکه استفاده از خواص عمومی مزیت انقیاد زودرس را به همراه دارد.

۲-۳-۳. پاورقی بخش ...

... دسترسی به کنترلهایی که هنگام بارگذاری Master page بصورت پویا ایجاد می شوند

فرض کنید یک Master page اخبار فوری (Breaking News) را از طریق ساخت label هایی به صورت پویا نمایش دهد. تعداد کنترل های تولید شده بستگی به اهمیت و نوع خبر دارد.

اگر چه بنا به دلایلی که گفته شد استفاده از متدها و خواص عمومی بهتر است اما مستلزم این است که در زمان طراحی بدانید به چه خواص و متدی نیاز دارید. به همین دلیل در مثال مطرح شده مجبورید به روش دوم متوسل شوید، چرا که نیازمند استفاده از متد Findcontrol هستید.

۲-۳-۴. منابع اطلاعاتی :

برای آشنایی با انقیدهای زودرس و دیررس و مزایا و معایب هر کدام به آدرس زیر مراجعه کنید:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbcn7/html/vaconearlylatebinding.asp>

۲-۴. ساخت Site Map برای سایت

یادداشت: با ساخت Site map برای سایت هیچ وقت کاربر در سایت سردرگم نخواهد شد.

اگر سایت شما بیش از یک صفحه داشته باشد. نیازمند روش برای هدایت کاربر بین صفحات می باشید. در Asp.Net 2.0 چهار کنترل جدید جهت بهبود هدایت کاربران در نظر گرفته شده است.

SiteMapDataSource

کنترلی جهت نمایش site map .sitemap فایل Xml ای است که ساختار کلی سایت و ارتباط میان صفحات سایت را توصیف می کند. کنترل SitemapDataSource هنگام بارگذاری صفحه نمایش داده نمی شود بلکه تأمین کننده اطلاعات سایت برای کنترل sitemappath به شمار می آید.

SiteMapPath

کنترلی که مسیر صفحه جاری را نمایش می دهد.

Menu

کنترلی جهت نمایش اطلاعات سایت بصورت منو. کنترل منو انقید داده (Data binding) به site map را پشتیبانی می کند.

TreeView

آخرین کنترل، کنترل Tree view است که اطلاعات را بصورت سلسله مراتبی نمایش می دهد. این کنترل نیز همانند menu انقید به site map را پشتیبانی می کند.

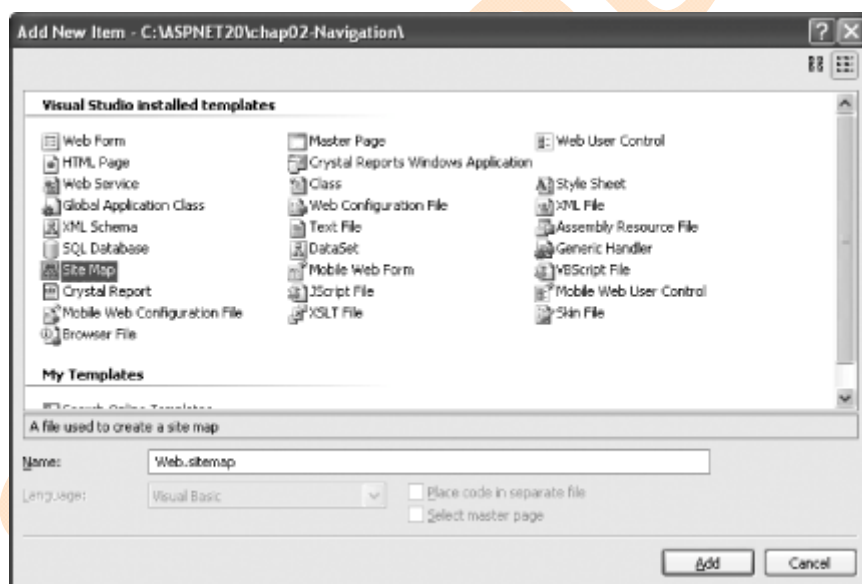
در بخش های باقیمانده فصل با نحوه استفاده از این کنترل ها آشنا خواهید شد اما اجازه دهید ابتداء نگاه دقیق تری به کنترل های Site map path , SitemapDatasource داشته باشیم.

۲-۴-۱. مراحل انجام کار:

در این بخش خواهید دید که چگونه می توان با استفاده از کنترل SiteMapPath به همراه فایل *Web.sitemap* مسیرهای هدایت کاربر را در صفحات سایت فراهم آورد. ابتدا پروژه ای طراحی می کنیم که از یک Master page استفاده کند. هنگامی که از Master page به عنوان قالب سایر صفحات استفاده کنیم، موقعیت مناسبی جهت استفاده از sitemap و سایر کنترل های هدایت کاربر فراهم می آید. چرا که اطلاعات سایت در تمامی صفحاتی که کاربر مشاهده می کند به نمایش در می آیند. شما یک site map جهت توصیف ارتباط میان صفحات سایت ایجاد می کنید سپس یک کنترل Site map path به Master page اضافه می کنید بطوریکه کاربر بصورت سلسله مراتبی به صفحات بعدی و قبلی اش دسترسی خواهد داشت.

1- VS 2005 را باز کنید و پروژه جدیدی با نام و آدرس *C:\ASPNET20\Chap02-Navigation* ایجاد کنید.

2- از طریق Solution Explorer یک Site map به پروژه اضافه کنید. برای این کار بر روی نام پروژه در Solution Explorer راست کلیک و گزینه های sitemap → Add New Item را انتخاب کنید. (شکل ۲-۱۶ را ببینید)



شکل ۲-۱۶. اضافه کردن Sitemap به پروژه

یادداشت: Sitemap یک سند XML است که نقشه وب سایت را توصیف می کند.

3- فایل *Web.sitemap* را باز کنید و کد مثال ۲-۲ را به آن اضافه کنید.

نکته: دقت داشته باشید که فایل *Web. Sitemap* بایستی بر اساس قوانین XML نوشته شود.


```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap>
  <siteMapNode url="main.aspx"
    title="Windows Mobile Home"
    description=" Windows Mobile Home " roles="">
    <siteMapNode url="whatiswinmobile.aspx"
      title="What is Windows Mobile"
      description=" What is Windows Mobile " roles="">
      <siteMapNode url="windowsmobileoverview.aspx"
        title=" Windows Mobile Overview"
        description=" Windows Mobile Overview "
        roles="" />
      <siteMapNode url="comparedevices.aspx"
        title="Compare Devices"
        description=" Compare Devices " roles="" />
      <siteMapNode url="software.aspx"
        title="Software Features Tour"
        description=" Software Features Tour "
        roles="" />
      <siteMapNode url="faq.aspx"
        title="Frequently Asked Questions"
        description=" Frequently Asked Questions "
        roles="" />
    </siteMapNode>

    <siteMapNode url="devices.aspx" title="Devices"
      description=" Devices " roles="" />
    <siteMapNode url="downloads.aspx" title="Downloads"
      description=" Downloads " roles="" />
    <siteMapNode url="howto.aspx"
      title="Help and How To"
      description=" Help and How To " roles="" />
    <siteMapNode url="communities.aspx"
      title="Communities"
      description=" Communities " roles="" />
    <siteMapNode url="events.aspx"
      title="News and Events"
      description=" News and Events" roles="" />
    <siteMapNode url="Administrator/Admin.aspx"
      title="Administrator"
      description=" Administrator" roles="" />
  </siteMapNode>
</siteMap>
```


مثال ۲-۲ فایل web. Sitemap

فایل web.Sitemap آرایش منطقی سایت را به صورت خیلی آسان نمایش می دهد. فایلها می توانند از نظر فیزیکی هر جایی واقع شده باشند. دقت داشته باشید که فایل قالب مشخص و درستی دارد و عناصر تودرتو برای نمایش گروهی صفحات بکار رفته اند.

4-هر عنصر <siteMapNode> چهار خاصیت دارد:

title

عنوان مربوط به هر گره (Node) از کنترل های SiteMapPath , TreeView یا Menu را مشخص می کند.

description

متن Tooltip مربوط به هر گره جهت نمایش هنگام قرار گرفتن موس بر روی گره.

roles

نقش یا نقش های مربوط به گره.

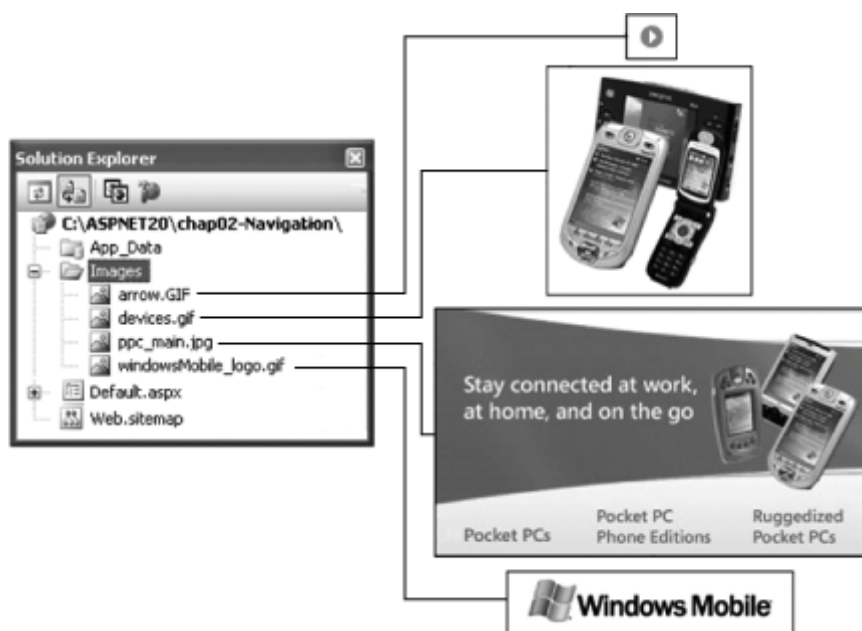
url

آدرس (URL) صفحه مربوط به گره، که هنگام کلیک بر روی گره کاربر را به آن صفحه هدایت می کند.

نکته: دقت کنید که مقدار URL در هر عنصر <sitemapnode> باید مقداری یکتا باشد.

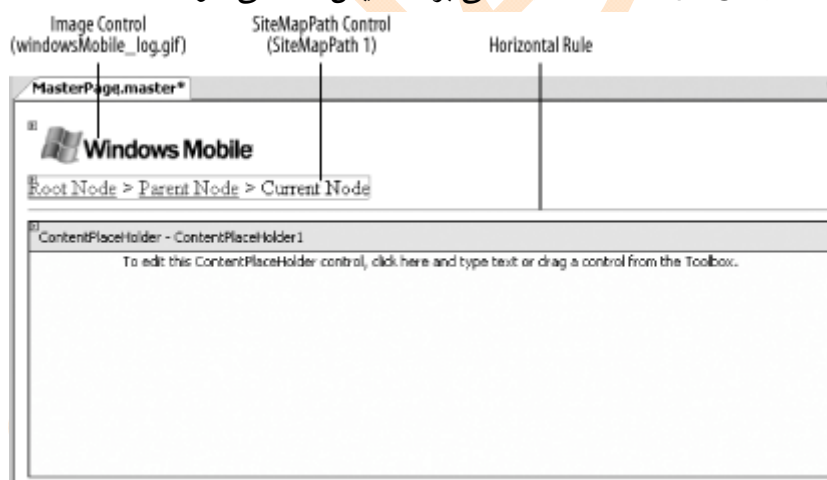
5- شاخه جدیدی به نام Images به پروژه اضافه کنید. (بر روی نام پروژه در Solution Explorer راست کلیک و گزینه های Add folder → regular folder را انتخاب کنید) عکس های شکل ۲-۱۷ را در شاخه C:\ASPNET20\chap02-Navigation\Images\ کپی کنید.

از این عکس ها در طراحی Master page استفاده خواهیم کرد.



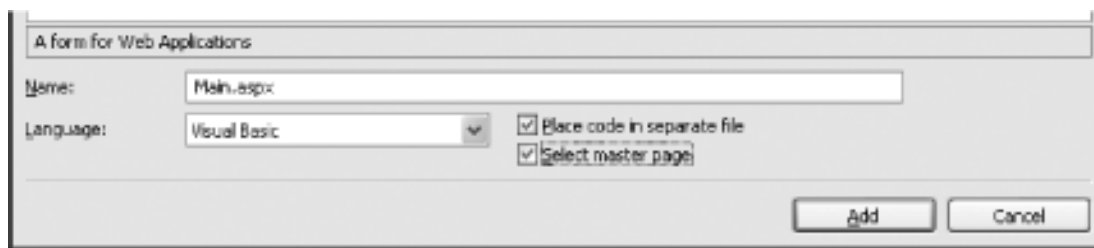
شکل ۲-۱۷. عکس های شاخه Images

6- یک Master page جدید به پروژه اضافه کنید. کنترل های نمایش داده شده در شکل ۲-۱۸ را به Master page اضافه کنید. Master page بهترین مکان جهت قراردادن کنترل site map path است چرا که تمامی کنترل های این صفحه در تمامی صفحاتی که از Master page ارث می برند نمایش داده می شوند.



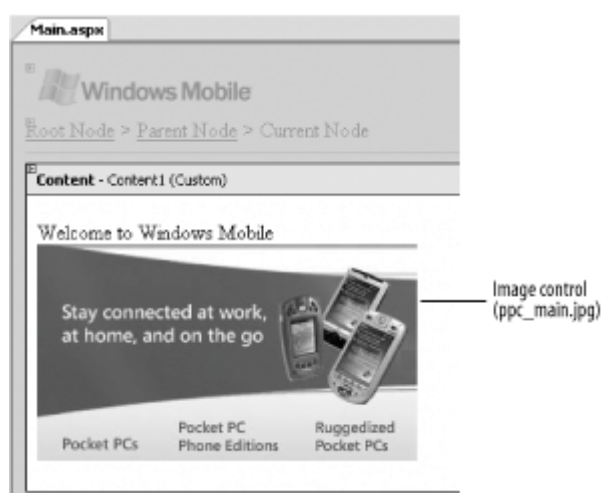
شکل ۲-۱۸. طراحی Master page

7- فرم جدیدی با نام main.aspx به پروژه اضافه کنید، گزینه "select Master page" را برای این صفحه فعال و Master page ساخته شده در مرحله قبل را برای صفحه main.aspx انتخاب کنید. (شکل ۲-۱۹ را ببینید).



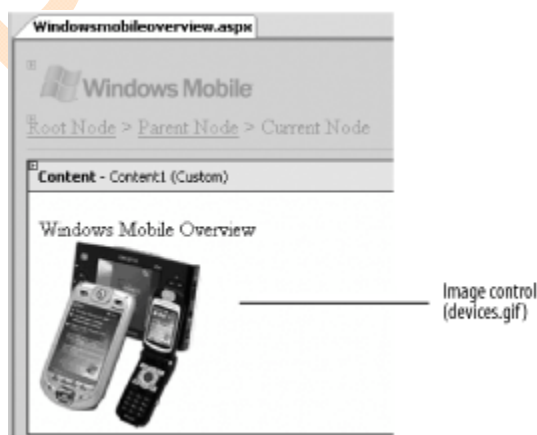
شکل ۲-۱۹. انتخاب Master page برای فرم

8- کنترل های موجود در شکل ۲-۲۰ را به صفحه main.aspx اضافه کنید.



شکل ۲-۲۰. صفحه main.aspx

9. فرم جدیدی به نام *Windowsmobileoverview.aspx* به پروژه اضافه کنید کنترل هایی شکل ۲-۲۱ را به این صفحه اضافه کنید.



شکل ۲-۲۱ صفحه Windowsmobileoverview.aspx

10- برای تست برنامه کلید F5 را فشار دهید. خروجی را در شکل ۲-۲۲ می توانید مشاهده کنید



شکل ۲-۲۲. تست کنترل Sitmappath

۲-۴-۲ تشریح نحوه عملکرد برنامه :

کنترل Sitmappath بصورت پیش فرض محتویات فایل Web.sitemap را خوانده و بصورت سلسله مراتبی در صفحه در حال نمایش نشان می دهد .

نکته : توجه داشته باشید مسیرهای کد کنترل Sitmappath نمایش می دهد بر مبنای اطلاعاتی است که در فایل Web.sitemap نوشته شده است . و این لزوماً به این معنی نیست که آنچه نمایش داده می شود با ساختار فیزیکی سایت مطابقت دارد. لذا چنانچه اطلاعات موجود در فایل Web.sitemap با ساختار اصلی سایت متفاوت باشند ، اطلاعات نمایش داده شده توسط Sitmappath نیز نادرست خواهند بود.

دقت کنید که در Asp.Net 2.0 تنها یک تامین کننده (Provider) برای sitemap می توانید داشته باشید:

XmlSiteMapProvide

XmlSiteMapProvider اطلاعات مربوط به sitemap را از سند XML از پیش تعریف شده می خواند ، که این سند بصورت پیش فرض فایل Web.sitemap می باشد.

برای تغییر فایل پیش فرض Sitemap بایستی از web.Config استفاده کنید. مثال ۲-۳ نحوه حذف XmlSiteMapProvider پیش فرض و اضافه نمودن فایل جدیدی به نام My.sitemap را نشان می دهد.

```
<configuration>
  <system.web>
    <siteMap>
```

```

<providers>
  <remove name="AspNetXmlSiteMapProvider" />
  <add name="AspNetXmlSiteMapProvider"
    type="System.Web.XmlSiteMapProvider,
      System.Web, Version=1.2.3400.0,
      Culture=neutral,
      PublicKeyToken=b03f5f7f11d50a3a"
    siteMapFile="My.sitemap" />
</providers>
</siteMap>
</system.web>
</configuration>

```

مثال ۲-۳ تغییر Sitemap پیش فرض در web.config

نکته : چنانچه میخواهید خودتان یک SiteMapProvider تعریف کنید ، خاصیت SiteMapProvider کنترل Sitemappath را به نامی که انتخاب کرده اید تغییر دهید.

یادداشت : در فصل پنجم بطور مفصل راجع به انواع تامین کننده ها (Providers) بحث خواهیم کرد.

۲-۴-۳ پاورقی بخش :

... سفارشی کردن کنترل Sitemappath

کنترل Sitemappath بصورت پیش فرض آدرسها را بصورت رشته ای مشتمل بر عنوان نوشته شده در عنصر <siteMapNode> نمایش می دهد که گره ها توسط علامت ">" از یکدیگر مجزا شده اند.

شما می توانید قالب پیش فرض نمایش گره ها توسط Sitemappath را در Source View صفحه ای که Sitemappath در آن قرار دارد تغییر دهید. (در مثال ما این صفحه همان Master Page است)

در مثال ۲-۴ علامت ">" که برای جدا نمودن گره ها از یکدیگر بکار می رفت ، با یک کنترل Image عوض می شود.)

عکس مورد نظر در شاخه Images و با نام arrow.gif ذخیره شده است)بعلاوه این کد مسیر جاری را با استفاده از پس

زمینه سبز برجسته و مشخص می کند.

```
<asp:SiteMapPath ID="SiteMapPath2"
    Font-Name="Garamond"
    Font-Size="12pt" RunAt="server">
    <CurrentNodeStyle Height="24px"
        BackColor="LightGreen"
        Font-Bold="true" />
    <NodeStyle Height="24px" />
    <PathSeparatorTemplate>
        <ItemTemplate>
            <asp:Image ID="Image2"
                ImageUrl="~/Images/arrow.gif"
                RunAt="server" />
        </ItemTemplate>
    </PathSeparatorTemplate>
</asp:SiteMapPath>
```

مثال ۲-۴ سفارشی نمودن کنترل Sitemappath

شکل ۲-۲۳ کنترل Sitemappath را بس از تغییر نشان می دهد .



شکل ۲-۲۳ Sitemappath سفارش شده

کنترل Sitemappath از طریق برنامه نویسی و از طریق خواص زیر نیز قابل تغییر است.

PathDirection

ترتیب مسیر نمایش را مشخص می کند. پیش فرض آن **RootToCurrent** می باشد که از بالاترین مسیر تا مسیر جاری را نمایش می دهد ، برای نمایش ترتیب بصورت معکوس این خاصیت را به **CurrentToRoot** تغییر دهید.

PathSeparator

علامت جدا کننده را مشخص می کند گزینه پیش فرض ">" می باشد.

RenderCurrentNodeAsLink

نمایش مسیر جاری بصور ابرمتن (Hyperlink) گزینه پیش فرض False می باشد.

... دسترسی به اطلاعات Sitemap از طریق برنامه نویسی

چنانچه از کنترل Sitemappath استفاده نمی کنید می توانید با استفاده از API مربوط به Sitemap (که در فضای نام System.Web.SiteMap (NameSpace قرار دارد) و از طریق برنامه نویسی به اطلاعات sitemap دسترسی داشته باشید مثال ۲-۵ نحوه دسترسی به این روش را نشان می دهد (کد در code-behind مربوط به master page نوشته شده است)

Partial Class MasterPage_master

```
Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load
```

```
Dim myCurrentNode As SiteMapNode
```

```
'---Gets the current node
myCurrentNode = SiteMap.CurrentNode
```

```
'---displays the navigational path---
Dim path As String = myCurrentNode.Title
While myCurrentNode.ParentNode IsNot Nothing
    myCurrentNode = myCurrentNode.ParentNode
    path = " > " & path
    path = myCurrentNode.Title & path
End While
```

```
Response.Write(path)
End Sub
```

```
End Class
```

مثال ۲-۵ نمایش اطلاعات سایت از طریق برنامه نویسی

علاوه بر خاصیت CurrentNode، خواص دیگری نیز برای دسترسی به اطلاعات وجود دارد:

- SiteMap.CurrentNode.ParentNode
- SiteMap.CurrentNode.PreviousSibling
- SiteMap.CurrentNode.RootNode
- SiteMap.CurrentNode.NextSibling

... مخفی کردن گره ها از کاربران فاقد هویت

اگر فایل *Web.sitemap* را با دقت بررسی کنید متوجه خاصیت *role* می شوید که تا اینجا هیچ استفاده ای از آن نکردیم و همواره یک رشته خالی بود. بطور پیش فرض کنترل *SiteMapDataSource* اطلاعات تمامی گره ها را در *Web.sitemap* قابل مشاهده (*visible*) در نظر می گیرد. اما مواقعی پیش می آید که نیاز دارید دسترسی به برخی گره ها را برای کاربران محدود کنید بعنوان مثال کاربران عادی حق دسترسی به صفحه مربوط به *admin* را ندارند پس نبایستی آدرس این صفحه برای این کاربران نمایش داده شود پس *SiteMapDataSource* بایستی امکان مخفی نمودن گره ها را فراهم کند که این قابلیت را *security trimming* می نامند. *XmlSiteMapProvider* قابلیت *security trimming* را بطور پیش فرض غیر فعال کرده است. برای فعال نمودن این خاصیت مراحل زیر را انجام دهید:

1- خاصیت *securityTrimmingEnabled* را *true* کنید. برای این کار *AspNetXmlSiteMapProvider* را در *Web.config* حذف و بصورت زیر اضافه کنید:

```
<system.web>
  <siteMap>
    <providers>
      <remove name="AspNetXmlSiteMapProvider" />
      <add name="AspNetXmlSiteMapProvider"
        type="System.Web.XmlSiteMapProvider,
        System.Web, Version=1.2.3400.0,
        Culture=neutral,
        PublicKeyToken=b03f5f7f11d50a3a"
        securityTrimmingEnabled="true"
        siteMapFile="Web.sitemap" />
    </providers>
  </siteMap>
```

2- در این مرحله گره ای را که می خواهید *security trimming* آن فعال باشد مشخص کنید. از فایل *web.sitemap* ساخته شده در ابتدا بخش استفاده کنید و خاصیت *Roles* مربوط به آخرین گره را *admin* قرار دهید. در این حالت *SiteMapPath* گره *Administrator* را تنها برای کاربران با نقش *admin* نمایش می دهد. (گره *Administrator* به فایل *Admin.aspx* موجود در شاخه *Administrator* اشاره می کند)

```
...
<siteMapNode url="Administrator/Admin.aspx"
  title="Administrator"
  description=" Administrator" roles="admin" />
</siteMapNode>
```


3-در مرحله آخر بایستی شاخه *Administrator* را طوری پیکر بندی (configure) کنید که همه کاربران به آن دسترسی نداشته باشند و تنها کاربران با نقش admin بتوانند آن را درخواست کنند. این کار را از طریق افزودن یک فایل web.Config به شاخه *Administrator* انجام دهید . کد مربوط به فایل web.Config در مثال ۲-۶ آورده شده است .

```
<configuration xmlns=
"http://schemas.microsoft.com/.NetConfiguration/v2.0">
<system.web>
<compilation debug="false"/>
<!-- deny all users -->
<authorization>
<deny users="*" />
</authorization>
</system.web>

<!-- allow only Admin role users to see Admin.aspx -->
<location path="Admin.aspx">
<system.web>
<authorization>
<allow roles="Admin" />
</authorization>
</system.web>
</location>
</configuration>
```

مثال ۲-۶. web.Config مخصوص شاخه administrator

پس از اعمال این تغییرات تنها کاربران با نقش admin قادر به دیدن گره Administrator خواهند بود.

یادداشت: برای کسب اطلاعات بیشتر راجع به نقشها و عضویتها به فصل پنجم مراجعه کنید.

۲-۴-۴. منابع اطلاعاتی:

برای کسب اطلاعات راجع به دسترسی ها در Asp.Net به آدرس زیر مراجعه کنید.

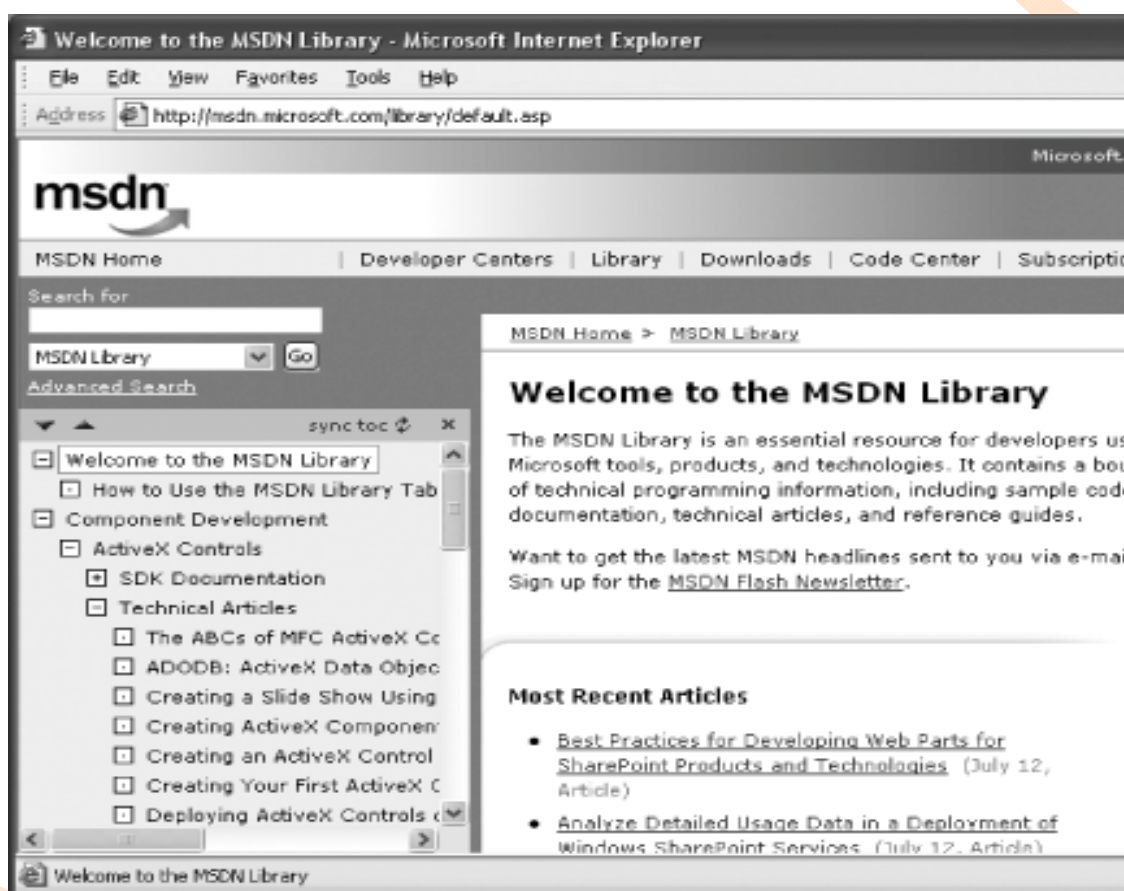
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconaspnetauthorization.asp>

چنانچه می خواهید در مورد Site Map Providers مطالب بیشتری بدانید در MSDN عبارت "ASP.NET Site Navigation Providers" را جستجو کنید.

اگر می خواهید اطلاعات را در فایل های غیر از اسناد XML (مانند پایگاه داده oracle ذخیره کنید) در MSDN عبارت "Implementing a Site Map Provider" را جستجو کنید.

۲-۵. نمایش سلسله مراتبی اطلاعات با استفاده از کنترل Tree view

اگر تا کنون به کتابخانه MSDN سر زده باشید حتما متوجه منوی درخت مانند موجود در سمت چپ صفحه اصلی شده اید . (عکس ۲-۲۴ را ببینید)



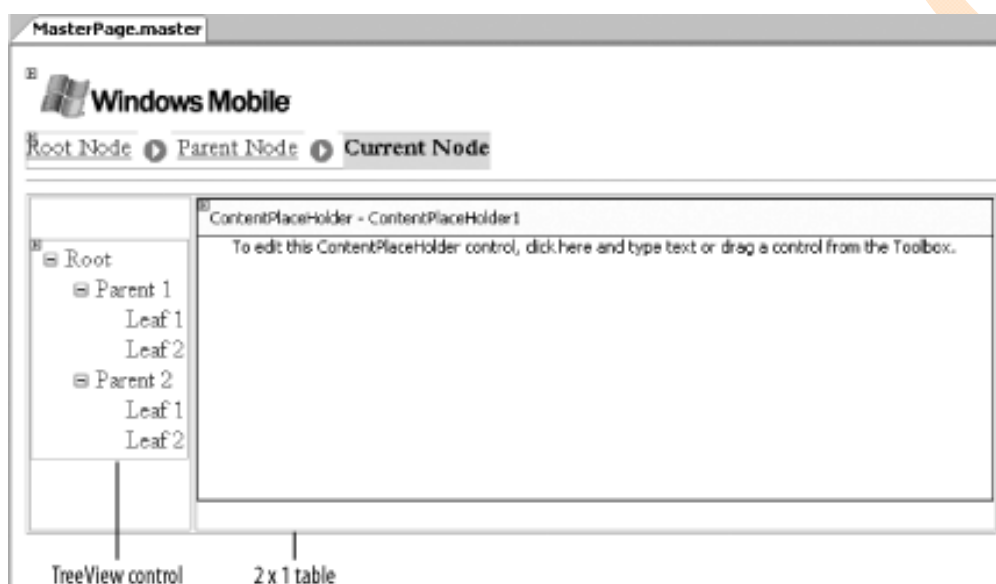
شکل ۲-۲۴. راهنمای درختی شکل MSDN

این اطلاعات با استفاده از کنترل Tree view نمایش داده شده اند که می توان از آن بعنوان جایگزین Sitemapprovider استفاده کرد. TreeView اطلاعات سلسله مراتبی را به خوبی نمایش می دهد و برخلاف Sitemapprovider تمامی سایت را نمایش می دهد .

۲-۵-۱. مراحل انجام کار :

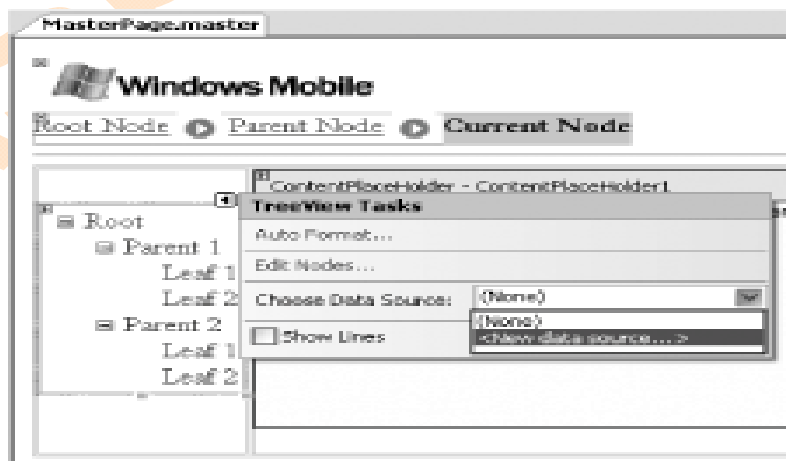
از پروژه بخش قبل برای افزودن یک TreeView به سایت استفاده کنید این کنترل را به Master page اضافه خواهیم کرد و از طریق انقیاد آن به Sitemap امکان دسترسی به تمامی صفحات سایت را فراهم می کنیم.

1- به فایل *MasterPage.master* یک جدول با دو ستون و یک سطر اضافه کنید (Layout → Insert Table) درون ستون سمت چپ یک کنترل TreeView و در ستون سمت راست ContentPlaceHolder را قرار دهید. (شکل ۲-۲۵ را ببینید)



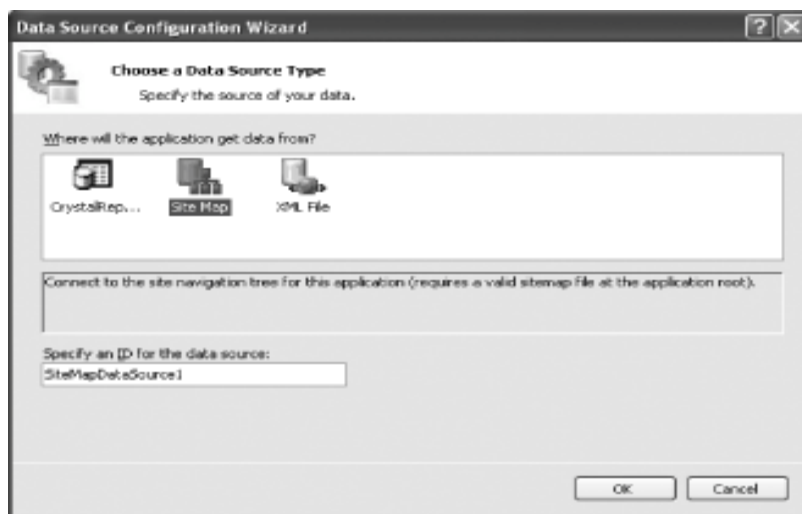
شکل ۲-۲۵. ویرایش Master page

2- از منوی Treeview Tasks کنترل TreeView گزینه <New data source...> را انتخاب کنید. (شکل ۲-۲۶)



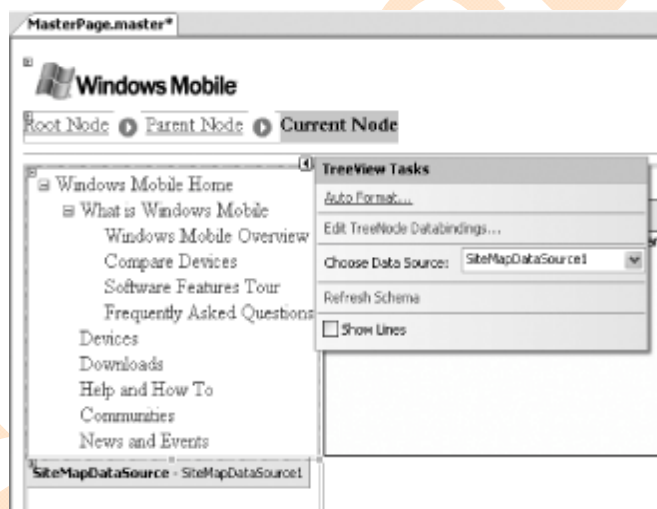
شکل ۲-۲۶. انتخاب Data Source

3- در پنجره Data Source Configuration Wizard، گزینه Site Map را انتخاب و ok کنید.
(شکل ۲-۲۷)



شکل ۲-۲۷. انتخاب Site Map بعنوان Data Source

4- شمای MSDN theme را (از طریق لینک "Auto Format..." در منوی TreeView) برای
TreeView انتخاب و خاصیت Back color آنرا با Gray (کد #EOEOEO) قرار دهید. (شکل ۲-۲۸)



شکل ۲-۲۸. انتخاب تم برای Treeview

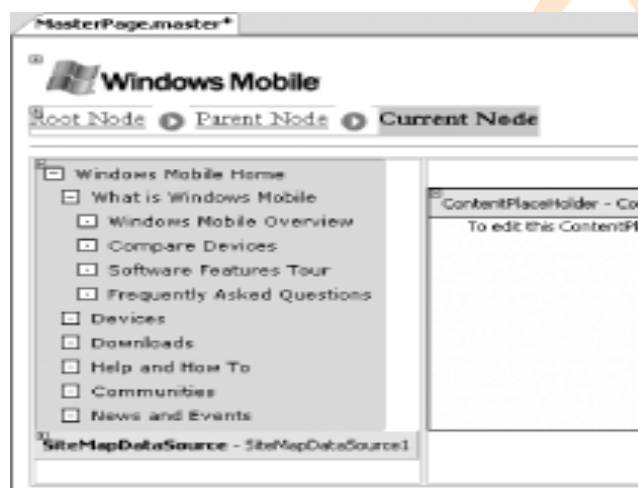
کنترل SiteMapDataSource یک Sitemap را نمایش می دهد. همانطور که در بخش ۲-۴ شرح داده شد
Sitemap ساختار منطقی سایت و ارتباطات میان صفحات را توصیف می کند.

نکته: VS 2005 از قالبهای متنوعی برای نمایش TreeView پشتیبانی می کند. در شکل ۲-۲۹ چند نمونه را می توانید
بینید.



شکل ۲-۲۹. چند نمونه از قالبهای طراحی شده برای TreeView

5- masterpage طراحی شده در شکل ۲-۳۰ نمایش داده شده است .



شکل ۲-۳۰. کنترل Tree view درون master page

6- دکمه F5 را فشار دهید تا نتیجه کار را مشاهده کنید. اکنون از طریق آیتمهای (Items) کنترل tree view به تمامی صفحات سایت دسترسی دارید. (شکل ۲-۳۱ را ببینید)

نکته : چنانچه از فریم (Frames) در صفحه تان استفاده می کنید می توانید از خاصیت target کنترل tree view جهت نمایش نام فریم برای نمایش صفحه هنگام کلیک آیتم استفاده کنید.



شکل ۲-۳۱. تست tree view

۲-۵-۲. پاورقی پخش ...

...استفاده از فایل Xml دلخواه بجای sitemap

صرف نظر از استفاده از فایل sitemap می توانید فایل Xml دیگری را به tree view انقیاد (bind) دهید. بعنوان مثال فرض کنید می خواهید با استفاده از tree view کتابی را پیاده سازی کنید که کاربر بتواند کتاب را پیمایش کند.

برای پیکربندی tree view جهت کار با فایل xml مراحل زیر را انجام دهید.

- 1- یک سند xml به پروژه اضافه کنید برای این کار بر روی نام پروژه در Solution Explorer راست کلیک کنید ، گزینه Add New Item... سپس XML File را انتخاب کنید نام فایل را Book.xml قرار دهید.
- 2- کد مثال ۲-۷ را در فایل Book.xml وارد کنید :

```
<Book>
  <Content Title=".NET Compact Framework Pocket Guide"
    URL="booktop.aspx">
    <Content Title="Part I" URL="PartI.aspx">
      <Content Title="The .NET Framework and Mobile Devices"
        URL="PartIMain.aspx">
        <Content Title="The .NET CLR" URL="PartI1.aspx" />
        <Content Title="Windows Powered Mobile Devices"
          URL="PartI2.aspx" />
        <Content Title="Tool and Language Support"
          URL="PartI3.aspx" />
        <Content Title="Hello, Pocket PC" URL="PartI4.aspx" />
      </Content>
    </Content>
  </Book>
```

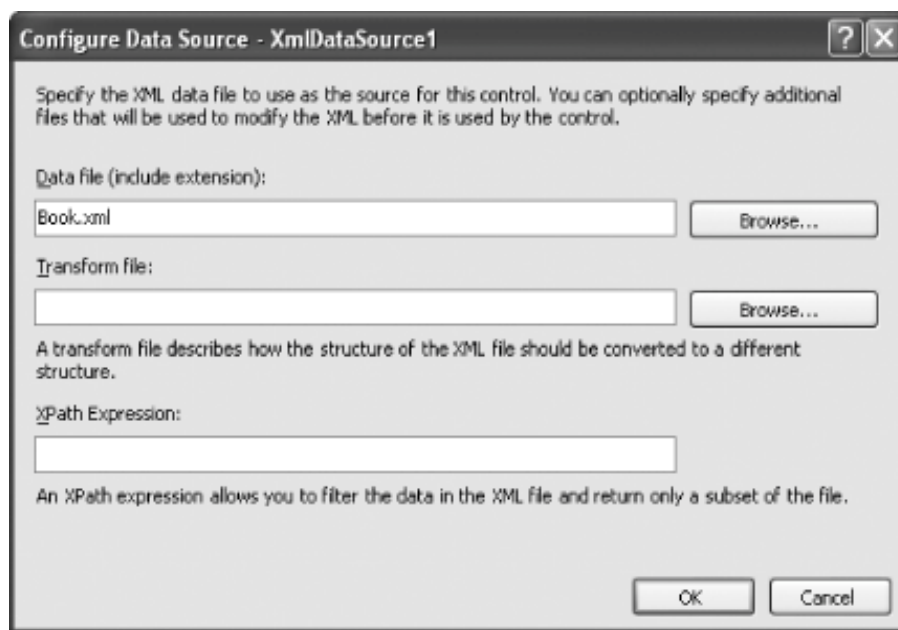
```

    <Content Title="Debugging the Application"
      URL="PartI5.aspx" />
    <Content Title="Troubleshooting" URL="PartI6.aspx" />
  </Content>
</Content>
<Content Title="Part II" URL="PartIIMain.aspx">
  <Content Title="User Interface Design" URL="PartII.aspx">
    <Content Title="Using the Windows Forms Controls"
      URL="PartII1.aspx" />
    <Content Title="Design Considerations for Smartphone Applications"
      URL="PartII2.aspx" />
  </Content>
</Content>
<Content Title="Part III" URL="PartIIIMain.aspx">
  <Content Title="Projects" URL="PartIII.aspx">
    <Content Title="Project A: Currency Converter"
      URL="PartIIIA.aspx" />
    <Content Title="Project B: Book Ordering System"
      URL="PartIIIB.aspx" />
    <Content Title="Project C: Bluetooth Chat"
      URL="PartIIIC.aspx" />
  </Content>
</Content>
<Content Title="Part IV" URL="PartIVMain.aspx">
  <Content Title="Deploying .NET Compact Framework Applications"
    URL="PartIV.aspx">
    <Content Title="Packaging CAB Files" URL="PartIV1.aspx" />
    <Content Title="Redistributing the .NET Compact Framework"
      URL="PartIV2.aspx" />
    <Content Title="Deploying Smartphone Applications"
      URL="PartIV3.aspx" />
  </Content>
</Content>
</Content>
</Book>

```

مثال ۲-۷. Book.xml

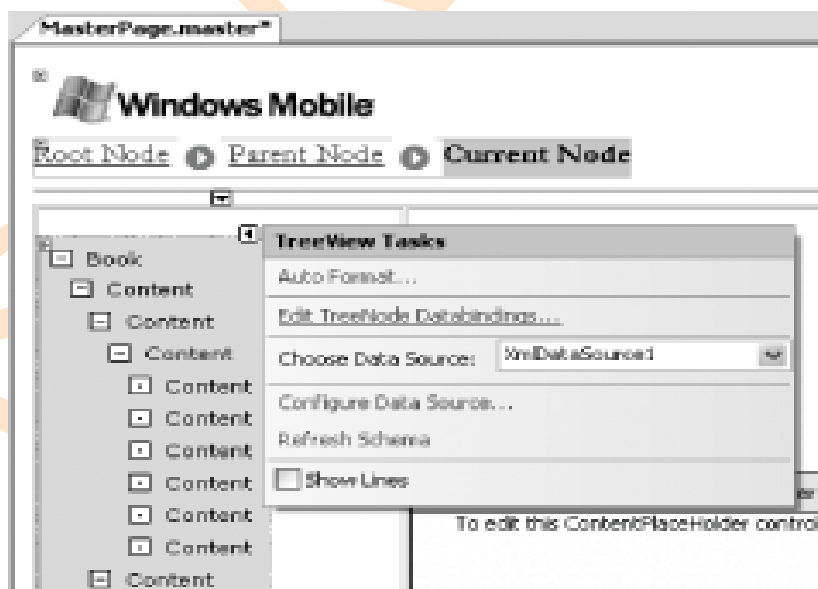
3- گزینه Configure Date Source را از منوی Treeview Tasks انتخاب و در بخش Date file فایل Book.xml را وارد کنید (شکل ۲-۳۲)



شکل ۲-۳۲. انتخاب منبع برای XmlDataSource

یادداشت: دقت کنید می توانید با استفاده از گزینه Transform file از یک فایل XSLT برای تبدیل سند xml به قالب دیگر و از گزینه XPath Expression برای فیلتر نمودن فایل xml استفاده کنید.

4- جهت ویرایش گره های tree view بر روی لینک Edit TreeNode Databindings... در Tree view tasks کلیک کنید. (شکل ۲-۳۳)



شکل ۲-۳۳. ویرایش گره ها

4- در کادر Available DataBindings (شکل ۲-۳۴) آیتم Content را انتخاب و دکمه Add را کلیک کنید. برخی خواص مهم مربوط به Content به شرح ذیل می باشند:

DataMember

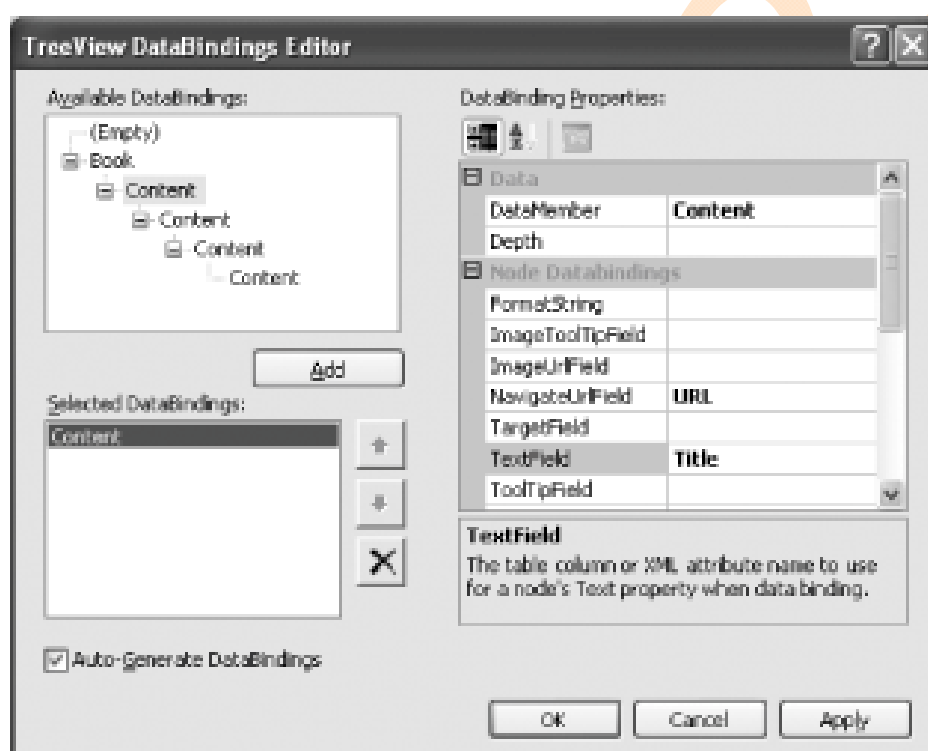
عنصری از فایل xml که برای انقیاد به هر گره از Tree view بکار می رود. این خاصیت را Content قرار دهید.

NavigateUrlField

آدرس URL مربوط به صفحه ای که هنگام کلیک بر روی گره tree view، کاربر به آن هدایت می شود. این خاصیت را URL قرار دهید.

TextField

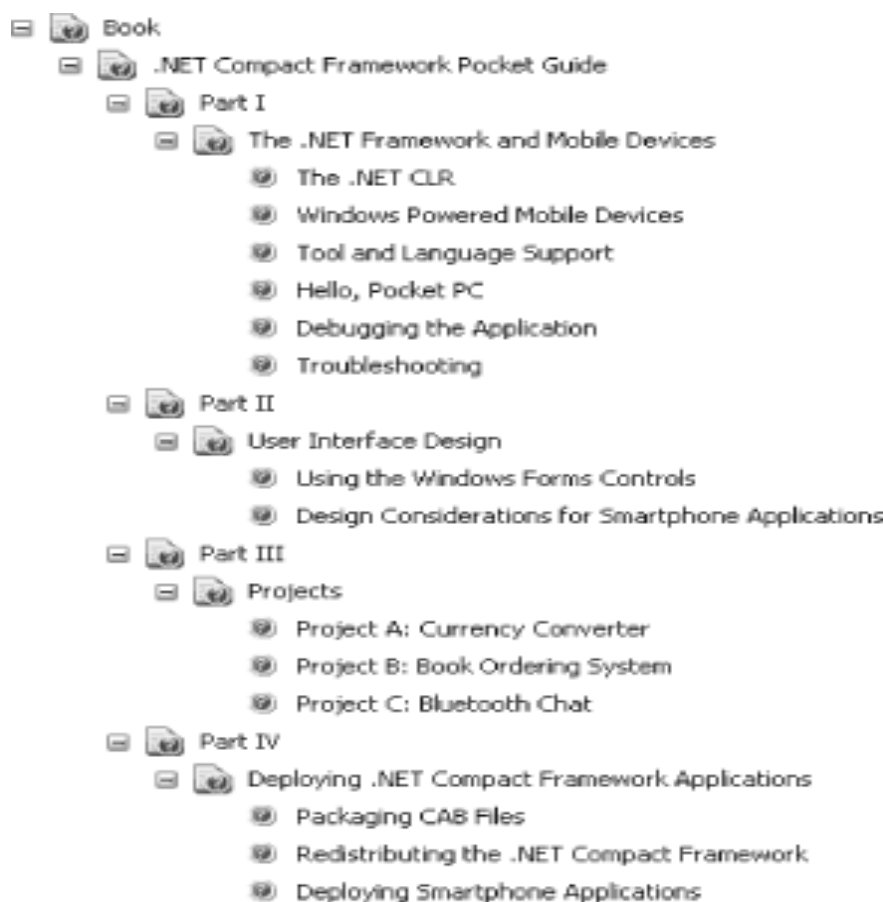
از این خاصیت جهت نمایش هر گره از tree view استفاده می شود، این خاصیت را Title قرار دهید.



شکل ۲-۳۴. استفاده از DataBindings Editor

6- Ok را کلیک کنید.

7- کلید F5 را جهت اجرای برنامه فشار دهید. tree view ایجاد شده در شکل ۲-۳۵ نمایش داده شده است.



شکل ۲-۳۵ نمایش tree view با استفاده از فایل xml

۲-۵-۳. منابع اطلاعاتی :

برای کسب اطلاعات بیشتر در مورد کنترل tree view در Asp.Net به آدرسهای زیر مراجعه کنید.

- <http://www.obout.com/obout/treeview/treeview.asp>
- <http://www.asp.net/ControlGallery/default.aspx?Category=38&tabindex=2>
- <http://www.telerik.com/Default.aspx?PageId=1828>

۲-۶. طراحی Treeview از طریق برنامه نویسی :

یادداشت : بجای انقیاد tree view به sitemap یا یک فایل xml به صورت ایستا می توان در زمان اجرا انقیاد داده را انجام داد.

در ادامه بخش خواهید دید که چگونه می توان همانند زمان طراحی یک tree view را به Sitemap یا سند xml متصل کرد. با استفاده از این روش می توانید در هر لحظه یک گره ایجاد کنید بعنوان مثال ممکن است بخواهید شاخه

خاصی از درایوهای هاردتان را از طریق tree view نمایش دهید و یا سایتی دارید که مرتباً در حال تغییر است در این حالت ساخت tree view به صورت پویا بسیار موثرتر خواهد بود.

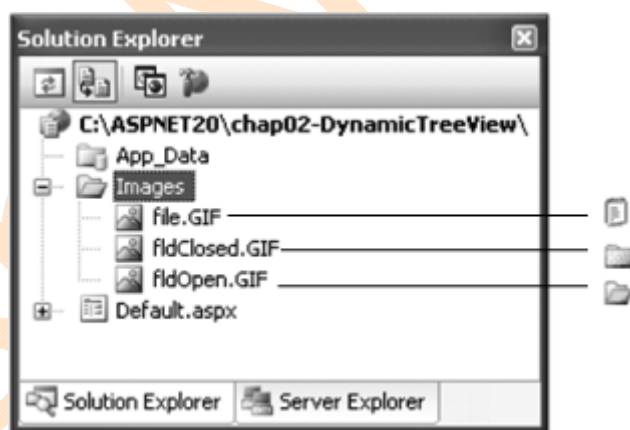
در این بخش با استفاده از tree view آدرس پروژه جاری و زیر شاخه های آنرا نمایش خواهیم داد.

۲-۶-۱. مراحل انجام کار :

در این بخش می آموزید چگونه یک tree view را به همراه آیتمهایش بصورت پویا به نمایش در آورید.

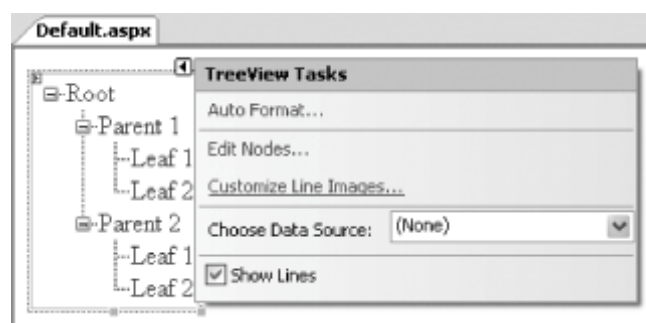
- 1- در VS 2005 پروژه جدیدی با نام و آدرس `C:\ASPNET20\chap02-DynamicTreeView` ایجاد کنید.
- 2- یک tree view به فرم جاری اضافه کنید.
- 3- در Solution Explorer بر روی نام پروژه راست کلیک کنید و گزینه های `Add New Folder → Regular` Folder را انتخاب کنید نام شاخه ایجاد شده را `Images` قرار دهید ، از این شاخه جهت ذخیره عکسها استفاده می کنیم.
- 4- عکس های شکل ۲-۳۶ را به `C:\ASPNET20\chap02-DynamicTreeView\Images` اضافه کنید.

نکته : عکس را می توانید از آدرس <http://www.oreilly.com/catalog/aspnetadn/> دانلود کنید.



شکل ۲-۳۶. عکسهای شاخه Images

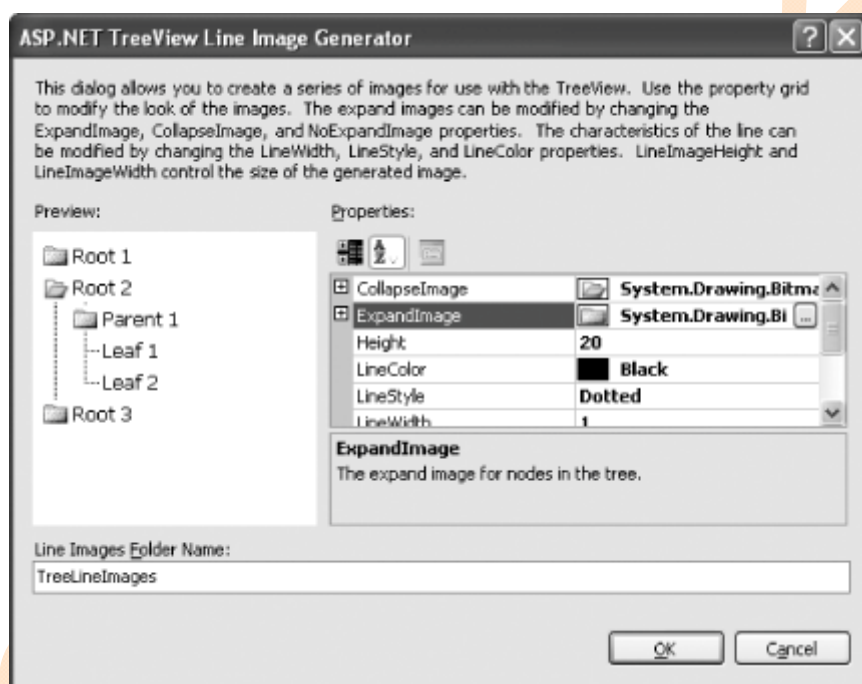
- 5- در پنجره treeview tasks ، چک باکس Show Lines را تیک بزنید و بر روی لینک Customize Line Images... کلیک کنید (شکل ۲-۳۷ را ببینید)



شکل ۲-۳۷. پیکر بندی tree view

6- در پنجره ASP.NET TreeView Line Image Generator خواص زیر را اضافه کنید. (شکل ۲-۳۸)

- CollapseImage: *fldOpen.gif*
- ExpandImage: *fldClosed.gif*



شکل ۲-۳۸. تنظیم خواص tree view

7- دکمه ok را کلیک کنید. هنگامیکه پرسیده شد شاخه *TreeLineImages* ساخته شود yes را انتخاب کنید. *TreeLineImages* شامل مجموعه ای از عکسها است که برای نمایش tree view استفاده می شوند. هنگامیکه هر گره tree view باز (Expand) می شود عکس *fldOpen.gif* نمایش داده می شود و در زمان بسته شدن عکس *fldClosed.gif* به نمایش در می آید.

8- بر روی فرم کلیک مضاعف کنید و به Code-behind صفحه بروید.

9- کد مربوط به زیر برنامه () *getSubDirectories* (مثال ۲-۸) بصورت بازگشتی تمامی زیر شاخه های شاخه جاری را باز می گرداند و در گره های tree view نمایش می دهد.

```

Public Sub getSubDirectories(ByVal path As String, _
                             ByVal node As TreeNode)
    '---get sub directories under the current one
    Dim dirs As String( ) = Directory.GetDirectories(path)

    '---no subdirectories
    If dirs.Length = 0 Then
        Exit Sub
    Else
        '---for each subdirectory, add to TreeView and
        ' recursively call itself again
        Dim dir As String
        For Each dir In dirs
            '---Add the relative path to the TreeView control
            Dim newNode As New TreeNode( _
                dir.Substring(dir.LastIndexOf("\") + 1))
            newNode.ToolTip = dir
            node.ChildNodes.Add(newNode)
            '---Find its subdirectories
            getSubDirectories(dir, newNode)
            '---Find files under itself
            getFiles(dir, newNode)
            '---close the current node
            newNode.CollapseAll( )
        Next
    End If
End Sub

```

مثال ۸-۲ زیرنامه () getSubDirectories

10- زیر نامه () getFiles (مثال ۹-۲) تمامی فایل‌های موجود در شاخه جاری را در گره های tree view لیست می کند.

```

Public Sub getFiles(ByVal path As String, ByVal node As TreeNode)
    Dim files As String( ) = Directory.GetFiles(path)
    Dim file As String

    '---no files in directory and no subdirectory
    If files.Length = 0 And node.ChildNodes.Count = 0 Then
        Dim newNode As New TreeNode("Directory is empty")
        node.ChildNodes.Add(newNode)
    End If
    Exit Sub

```

End If

For Each file In files

'---Add the file to the TreeView control

Dim newNode As New TreeNode(file.Substring(path.Length + 1))

newNode.ToolTip = file

newNode.ImageUrl = "Images\file.gif"

node.ChildNodes.Add(newNode)

Next

End Sub

مثال ۲-۹. زیر برنامه () getFiles

نکته: می توانید از طریق خاصیت ImageUrl برای هر گره عکسی انتخاب کنید.

10- نهایتا در رویداد page-load صفحه کد زیر را وارد کنید تا هنگام بارگذاری صفحه اطلاعات tree view به نمایش در آیند.

Protected Sub Page_Load(ByVal sender As Object, _

ByVal e As System.EventArgs) _

Handles Me.Load

If Not IsPostBack Then

TreeView1.Nodes.Add(New _

TreeNode(Request.PhysicalApplicationPath))

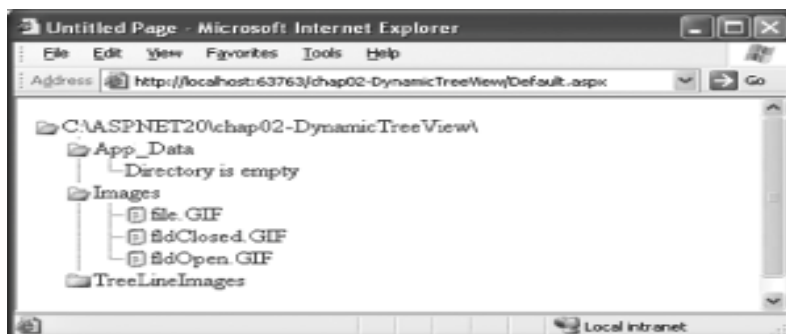
getSubDirectories(Request.PhysicalApplicationPath, _

TreeView1.Nodes(0))

End If

End Sub

11- برای مشاهده خروجی کلید F5 را فشار دهید. بایستی شکل ۲-۳۹ را مشاهده کنید.



شکل ۲-۳۹. نمایش tree view ایجاد شده بصورت پویا

...ویرایش TreeView از طریق برنامه نویسی

با استفاده از متدهای ExpandAll() و CollapseAll() میتوانید تمامی گره های TreeView را باز کنید یا ببندید.

TreeView1.CollapseAll()

TreeView1.ExpandAll()

...ویرایش جداگانه آیتمهای TreeView

چندین متد رفتاری برای هر کدام از آیتمهای TreeView وجود دارد که از طریق Source View در مثال ۱۰-۲ نمایش داده شده اند.

```
<asp:TreeView ID="TreeView2" runat="server">
  <Nodes>
    <asp:TreeNode SelectAction="Expand"
      Value="Windows" Text="Windows">
      <asp:TreeNode SelectAction="SelectExpand"
        Value="Windows XP" Text="Windows XP">
        <asp:TreeNode ShowCheckBox="True"
          Value="Home Edition" Checked="True"
          NavigateUrl="xphome.aspx"
          Text="Home Edition">
        </asp:TreeNode>
        <asp:TreeNode ShowCheckBox="True"
          Value="Professional"
          NavigateUrl="xpProfessional.aspx"
          Text="Professional">
        </asp:TreeNode>
      </asp:TreeNode>
    <asp:TreeNode Value="Windows 2003 Server"
      NavigateUrl="win2003.aspx"
      Text="Windows Server 2003">
    </asp:TreeNode>
```

```
</asp:TreeNode>
</Nodes>
</asp:TreeView>
```

مثال ۲-۱۰. تنظیم خواص آیتمهای TreeView

هنگام بارگزاری صفحه در IE (Internet Explorer) خروجی مشابه شکل ۲-۴۰ خواهد بود.

```
[-] Windows
    [-] Windows XP
        [x] Home Edition
        [-] Professional
    Windows Server 2003
```

شکل ۲-۴۰. کنترل TreeView ویرایش شده

بر اساس مثال ۲-۱۰ آیتمهای TreeView بصورت زیر عمل می کنند:

- کلیک بر روی گزینه "Windows" باعث تمامی زیر مجموعه ها میشود. بدون اینکه PostBack به سرور رخ دهد. (SelectAction="Expand")
- کلیک گزینه Windows XP باعث نمایش آیتمهای زیر مجموعه آن میشود و یک PostBack به سرور رخ میدهد. (SelectAction="SelectExpand") اطلاعات آیتم انتخاب شده را میتوانید در رویداد Page_Load و از طریق خاصیت SelectedNode بدست آورید.
- آیتمهای Home Edition و Professional دارای چک باکس هستند. (ShowCheckBox="True") که Home Edition بطور پیش فرض تیک خورده است. (Checked="True") کلی ک بر روی Home Edition باعث لود شدن صفحه XPHome.aspx میشود. (NavigateUrl="xpHome.aspx") و کلیک بر روی Professional صفحه xpProfessional.aspx را لود میکند. (NavigateUrl="xpProfessional.aspx")
- گزینه Windows server 2003 نیز صفحه win2003.aspx را لود می کند.

نکته: گزینه پیش فرض خاصیت SelectonAction گزینه select می باشد برای نمایش صفحه در یک صفحه جدید از خاصیت target استفاده کنید.

۲-۶-۳. منابع اطلاعاتی:

برای آشنایی بیشتر با خواص و متدهای کنترل tree view به MSDN مراجعه کنید.

۲-۷. ساخت منو با استفاده از کنترل Menu

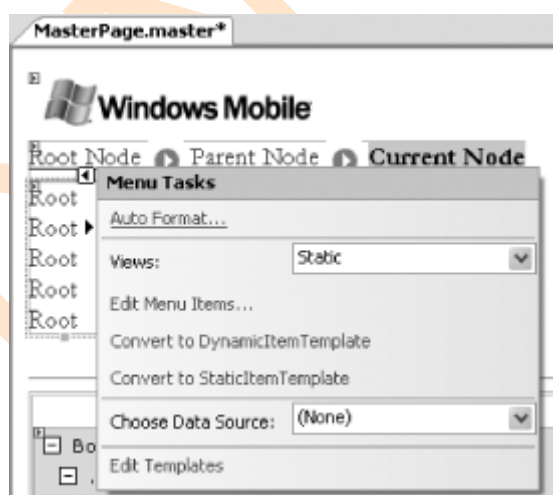
یادداشت: هم اکنون با استفاده از Asp.Net 2.0 می توانید در سمت مشتری منو ایجاد کنید.

بجای استفاده از کنترل treeview جهت نمایش اطلاعات سایت، می توانید از کنترل menu برای تولید خودکار یک منوی drop-down در سمت مشتری استفاده کنید. در بسیاری موارد منوها برای نمایش اطلاعات سایت بسیار مفیدتر هستند. بعنوان مثال کارخانه نیسان موتور (<http://www.nissanusa.com>) با استفاده از منو به کاربرانش امکان انتخاب و نمایش خودروی موردنظرشان را می دهد.

۲-۷-۱. مراحل انجام کار:

در این بخش یک کنترل Menu به Master Page بخش ۲-۵ اضافه می کنیم و از طریق آن کاربر را به قسمتهای مختلف سایت هدایت می کنیم.

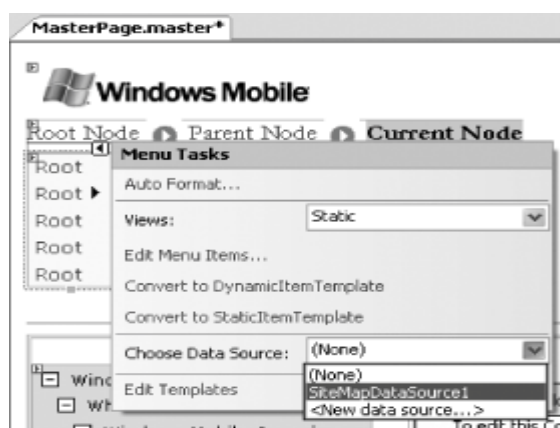
1- پروژه ای که در بخش ۲-۵ ایجاد کردیم باز کنید، یک کنترل menu (این کنترل را در Toolbox می توانید پیدا کنید) به master page اضافه کنید و شمای "Colorful" را به آن اعمال کنید. (از طریق کلیک Auto Format... در منوی menu tasks) شکل ۲-۴۱ را ببینید.



شکل ۲-۴۱ ویرایش کنترل menu

2- در پنجره menu tasks گزینه **SiteMapDataSource1** را به عنوان Data source انتخاب کنید (شکل ۲-۴۲)

SiteMapDataSource1 را قبلاً هنگام پیکربندی tree view اضافه کردیم.



شکل ۲-۴۲. اضافه کردن منو به Master page

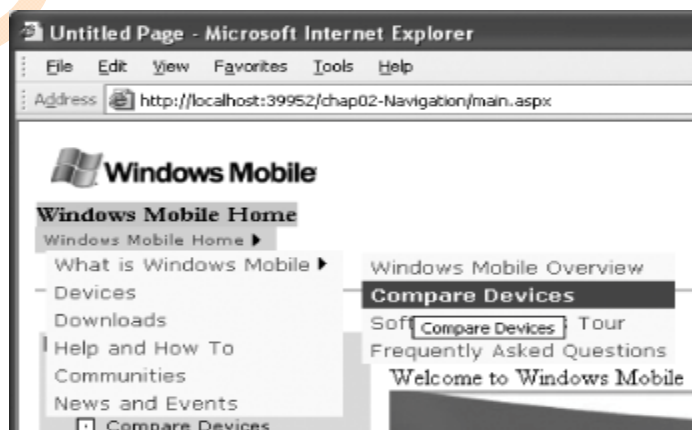
3- کلید F5 را فشار دهید. همانطور که در شکل ۲-۴۳ می بینید کنترل menu به SiteMapDataSource1 متصل شد و صفحات به صورت منو در اختیار کاربر قرار گرفتند.



شکل ۲-۴۳. تست کنترل menu

5- کنترل menu دو جهت (orientation) نمایش اطلاعات را پشتیبانی می کند: افقی (horizontal) و عمودی (vertical) با استفاده از خاصیت orientation می توانید حالت نمایش منو را تغییر دهید. (شکل ۲-۴۴ را ببینید)

نکته: حالت پیش فرض نمایش اطلاعات در کنترل menu، عمودی (vertical) می باشد.



شکل ۲-۴۴. تغییر حالت نمایش در menu

... ساخت آیتمهای منو از طریق تعریف (declaratively)

بجای انقیاد کنترل menu به فایل xml یا sitemap ، می توانید آیتمهای منو را بصورت دستی ایجاد کنید بعلاوه از طریق Menu Item Editor (از منوی menu tasks و گزینه Edit Menu Items.. می توانید آنرا پیدا کنید) می توانید آیتمهای منو را بصورت دستی وارد کنید.

برای طراحی آیتمهای منو از طریق کد ، یک کنترل menu به صفحه اضافه کنید و سپس در source View آیتمهای مورد نظر را به منو اضافه کنید. مثال ۱۱-۲ ساختار منو را با استفاده از کد ایجاد می کند.

```
<asp:Menu ID="Menu1" Target="newWindow" runat="server">
  <Items>
    <asp:MenuItem Value="Windows" Text="Windows">
      <asp:MenuItem Value="Windows XP" Text="Windows XP">
        <asp:MenuItem Value="Home Edition" Text="Home Edition"
          navigateurl="xphome.aspx">
        </asp:MenuItem>
        <asp:MenuItem Value="Professional" Text="Professional"
          navigateurl="xpprof.aspx">
        </asp:MenuItem>
      </asp:MenuItem>
    <asp:MenuItem Value="Windows 2003 Server"
      Text="Windows 2003 Server"
      navigateurl="win2003.aspx">
    </asp:MenuItem>
  </asp:MenuItem>
</Items>
</asp:Menu>
```

مثال ۱۱-۲. افزودن آیتمهای منو از طریق کد نویسی

نکته : چنانچه می خواهید آیتم انتخابی در صفحه جدید باز شود از خاصیت target استفاده کنید. دقت کنید این خاصیت هم برای عنصر <menu> و هم برای <menu Items> کاربرد دارد. چنانچه target در سطح menu تعیین شده باشد ، تمامی زیر منوها با target مشخص شده تنظیم می شوند مگر اینکه برای هر آیتم منو بصورت مجزا target تعیین کرده باشید. اگر خاصیت target را مشخص نکرده باشید (یا اینکه با رشته خالی تنظیم شده باشد) ؛ target صفحه جاری در نظر گرفته می شود. به یاد داشته باشید باز کردن صفحه جدید باعث سردرگمی کاربر می شود مگر اینکه آیتم مورد نظر صریحا به یک صفحه یا سایت خارجی اشاره داشته باشد.

۲-۷-۳ منابع اطلاعاتی :

جهت کسب اطلاعات بیشتر راجع به ساخت منوها از طریق اکسریپت های سمت مشتری می توانید از آدرسهای زیر استفاده کنید.

- <http://dhtml-menu.com/menu-demos/demo117.html>
- <http://wsabstract.com/javatutors/crossmenu.shtml>
- <http://archive.devx.com/dhtml/articles/td041801/td041801-1.asp>

Asp.Net 2.0

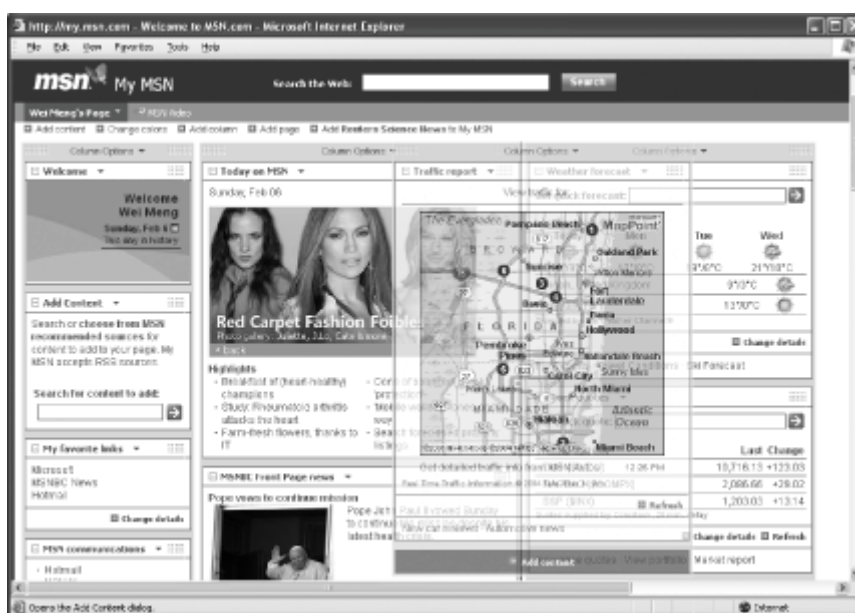
فصل سوم

وب پارت ها

(Web Parts)

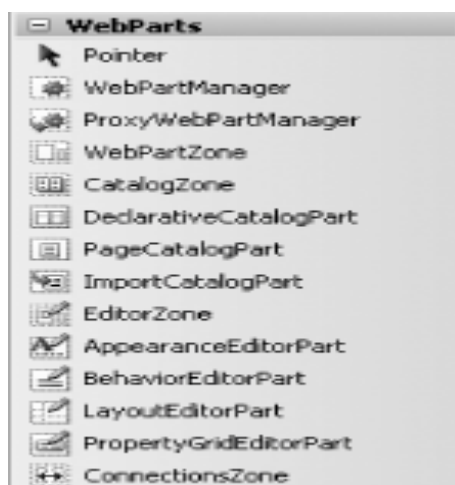
امروزه وب سایتها حاوی اطلاعات زیادی هستند ، بطوریکه یک طراحی ضعیف باعث سردرگمی کاربر در سایت می شود.

به منظور هدایت و راهنمایی بهتر کاربران ، پرتال ها (مانند msn) اطلاعاتشان را به بخش های مجزایی تقسیم می کنند که هر بخش به طور جداگانه سفارشی (costumsize) شده است. اطلاعات به بخشهای جداگانه ای تقسیم می شوند و کاربران می توانند آنها را بر اساس قالب مورد علاقه خود مرتب نمایند. علاوه بر شخصی سازی (personalization) کاربران می توانند بخشهایی که به آنها نیاز ندارند و یا مورد علاقه آنها نیست را مخفی نمایند. علاوه بر این کاربران می توانند تنظیماتشان را ذخیره نمایند و در رجوع مجدد به سایت از تنظیمات خودشان استفاده نمایند. در ASP.NET 2.0 می توانید با کنترلهای جدید Web Parts پرتالهایی با چنین امکانات طراحی و مدیریت کنید.



یادداشت : استفاده از webparts ها در صفحه کاربر را قادر می سازد بخشهای مختلف صفحه را به صورت دلخواه خود چیدمان کند.

ASP.NET 2.0 مجموعه ای از کنترلهای آماده و از پیش طراحی شده را در اختیار شما قرار می دهد که از آنها جهت ساخت پرتالها می توانید استفاده کنید این کنترلهای جدید را می توانید در بخش webparts از toolbox پیدا کنید.



اجازه دهید کار را از پایه آغاز کنیم . در این بخش یاد می گیرید چگونه برای برنامه وب خود web part طراحی کنید.

بکارگیری web part ها

بمنظور استفاده از web part ها در یک صفحه پرتال کاربر بایستی تائید هویت شده باشد. (authenticated) بطور پیش فرض Asp.Net از مدیریت windows جهت تائید هویت کاربران در وب استفاده می کند ، از اینرو تمامی کاربران برای استفاده از سیستم شما بایستی ابتدا تائید هویت شوند. در هر حال این حالت هنگامی که می خواهید پروژه تان از طریق اینترنت در اختیار کاربران قرار دهید روش مناسبی نیست ، بدین منظور شما نیازمند استفاده از تائید هویت مبنی بر فرمها (Forms authentication) هستید. برای کسب اطلاعات بیشتر راجع به Forms authentication به فصل ۵ مراجعه کنید. چنانچه کاربر تائید هویت نشده سعی در تغییر یک صفحه حاوی webpart داشته باشد خطای حین اجرا (runtime error) رخ می دهد.

۱-۳. آشنایی با وب پارتها

برای درک نحوه کارکرد webpart ها ، در این بخش یک پرتال ساده شامل سه webpart طراحی خواهیم کرد.

دو روش برای افزودن webpart به صفحه وجود دارد :

- استفاده از کنترلها (table, textbox, calendar, ...)
- استفاده از Web User control

در هر صورت ، ابتدا می بایست راهی برای افزودن این کنترلها به صفحه پیدا کنید. این کار از طریق افزودن یک **WebPartManagerControl** و یک یا بیش از یک کنترل **WebPartZone** امکان پذیر است.

WebPartManager

مدیریت تمامی webpart های موجود در صفحه را بر عهده دارد.

WebPartZone

فراهم کننده یک لایه سراسری برای کنترلهای webpart است بطوریکه user Interface اصلی صفحه را مشخص می کند. این کنترل نقش لنگرگاه (Anchor) را برای کنترلهای webpart ایفا می کند.

برای درک بهتر مطالب فوق یک webpart از یک کنترل استاندارد طراحی می کنیم .

- 1- ابتدا به یک صفحه Home page برای webpart ها نیاز داریم. Vs 2005 را باز کنید پروژه جدیدی ایجاد کنید و نام آنرا `C:\ASPNET20\chap03-Webparts` قرار دهید.
- 2- سپس یک کنترل webpart manager از Toolbox (برگه WebParts) به صفحه اضافه کنید. کنترل webpart manager تمامی webpart هایی را که بر روی صفحه قرار می دهید مدیریت خواهد کرد. این کنترل اولین کنترلی است که باید به صفحه اضافه شود.
- 3- برای اینکه مکان قرار گیری webpart ها در صفحه را مشخص کنید یک جدول با ۱ سطر و ۳ ستون به فرم اضافه کنید (Layout → Table)، سپس در هر یک از خانه های جدول یک کنترل webpart zone (از Toolbox برگه WebParts) قرار دهید. (شکل ۱-۳)

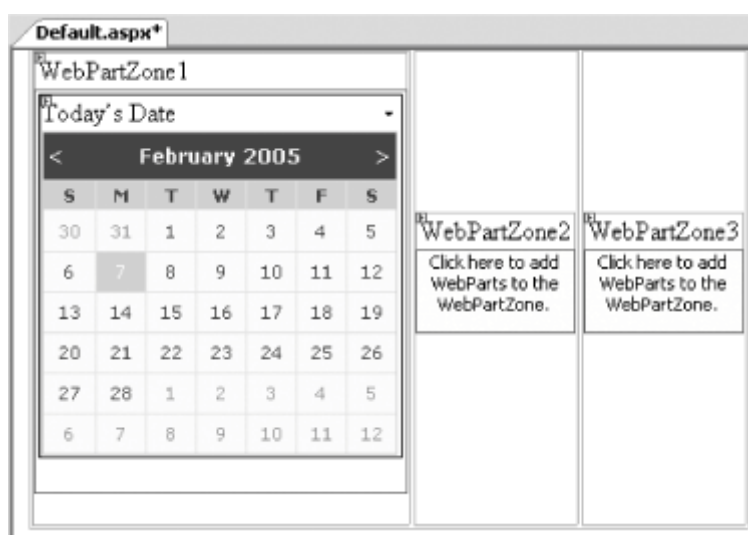


شکل ۱-۳ طراحی فرم

- 4- Web Part zone ناحیه ای است که webpart به آن تعلق دارد همچنین طرح و ظاهر پیش فرض هر webpart را مشخص می کند.
- 5- اکنون زمان آن است که یک کنترل وب سرور به صفحه اضافه کنید. یک کنترل calendar به webpart Zone اضافه کنید. از طریق لینک Auto Format... در Task menu مربوط به calendar تم Colorful 1 را به آن اعمال کنید.
- هنگامیکه یک کنترل به webpart Zone اضافه می شود، webpart ها آنرا با کنترل خاصی به نام GenericWebPart پوشش می دهند بطوریکه کنترل قابلیت های مورد نیاز جهت تبدیل شدن به یک webpart واجد شرایط را به دست می آورد.
- 6- به Source view صفحه (default.aspx) رفته و خاصیت <title> را با مقدار "Today's Date" به کنترل Calendar اضافه کنید:

```
<asp:Calendar ID="Calendar1"
    title="Today's Date"
    runat="server" BorderWidth="1px"
    Font-Names="Verdana"
    ...
```

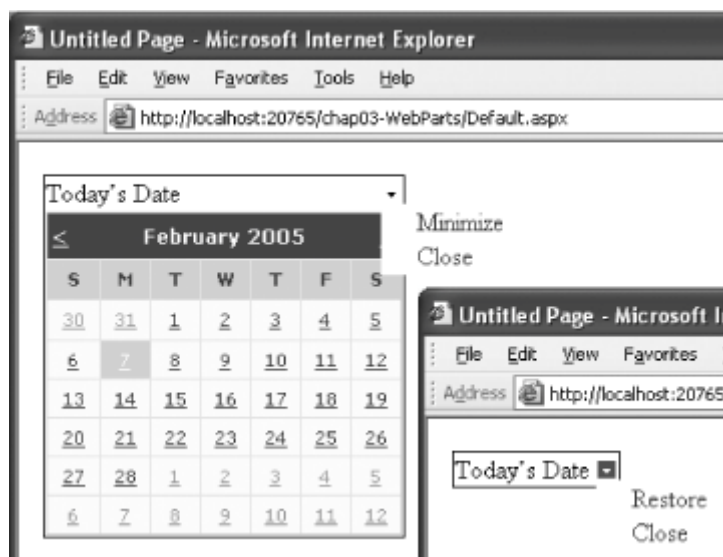

- 7- دقت داشته باشید که کنترل Calendar از خاصیت <title> پشتیبانی نمی کند. همانطور که پیش تر گفته شد هنگامیکه یک کنترل به webpart Zone اضافه می شود بوسیله یک GenericWebPart پوشش داده می شود ، که این کنترل از <title> پشتیبانی می کند.مقداری که در <title> قرار می گیرد هدر (header) کنترل GenericWebPart به شمار می آید.
- 8- هم اکنون فرم شما شبیه شکل ۲-۳ شده است .



شکل ۲-۳. کنترل calendar بعنوان یک webpart

- نکته : توجه داشته باشید ، چنانچه بیش از یک کنترل را به webpart Zone اضافه کنید هر کنترل بعنوان webpart جدا گانه رفتار خواهد کرد و هر کدام دکمه های minimize و close مربوط به خودشان را خواهند داشت . به منظور ترکیب چند کنترل درون یک webpart می بایست از web user Control استفاده کنید.
- همچنین می توانید گروهی از کنترلها را درون یک placeholder یا یک panel قرار دهید و سپس آنها را درون webpart Zone قرار دهید .

- کلید F5 را فشار دهید . توجه کنید آیکن (Icon) پیکان (Arrow) در قسمت بالا و راست کنترل Calendar مشاهده می شود چنانچه بر روی آن کلیک کنید لینک های Close و minimize نمایش داده می شوند. گزینه minimize کنترل را در کمینه (minimize) می کند.برای باز گذراندن کنترل به حالت اول از گزینه Restore استفاده کنید.(شکل ۳-۳)گزینه close جهت بستن webpart بکار می رود.



شکل ۳-۳. کنترل calendar

نکته : چنانچه Close را کلیک کنید ، دیگر قادر به نمایش webpart نخواهید بود. برای باز کردن webpart بسته شده به بخش ۳-۴ همین فصل مراجعه کنید.

9- اکنون مشاهده می کنید که چگونه یک کنترل معمولی را به یک webpart تبدیل کردیم. اجرای برنامه را متوقف کنید و به VS باز گردید.

حال می خواهیم به بررسی شیوه دوم افزودن webpart به صفحه یعنی ساخت user control و افزودن آن به webpart zone بپردازیم.

کنترلی که طراحی می کنیم ، امکان جستجو در گوگل را برای کاربر فراهم می کند.

نکته : web user control کنترل مرکبی است که می تواند شامل یک یا چند کنترل معمولی باشد. استفاده از web user Control امکان دسته بندی چند کنترل درون یک کنترل را فراهم می کند.

1- ابتدا باید کنترل جدیدی به پروژه اضافه کنیم یک Web User control به پروژه اضافه کنید ، بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های Add New Item... سپس Web User control را انتخاب نمایید، نام آنرا Google.ascx قرار دهید.

2- از آنجا که می خواهیم لوگوی گوگل را برای مشخص کردن webpart نمایش دهیم ، شاخه ای جهت ذخیره عکس ایجاد می کنیم ، شاخه جدید را به پروژه اضافه کنید و نام آنرا Images قرار دهید. (بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های Add Folder → Regular Folder را انتخاب کنید. عکس Google.gif را در شاخه C:\ASPNET\chap03-Webparts\Images ذخیره کنید (شکل ۳-۴ را ببینید)

نکته: عکس را می توانید از آدرس <http://www.oreilly.com/catalog/aspnetadn> دانلود کنید.



شکل ۳-۴. عکس موجود در شاخه Images

3- حالا کنترل اضافه شده را کامل می کنیم ابتدا یک جدول با ۲ سطر و ۲ ستون به User control اضافه کنید (Layout → Table) همانند شکل ۳-۵ یک کنترل Image ، یک Textbox و یک Button به سه خانه جدول اضافه کنید. فایل *Google.gif* را به کنترل Image انتساب دهید و text مربوط به Button را Search قرار دهید.



شکل ۳-۵. طراحی user Control توسط کنترل‌های مختلف

4- بر روی دکمه Search کلیک مضاعف نموده و کد زیر را بنویسید ، این کد رشته جستجو را از textbox دریافت و به google.com ارسال می کند.

```
Protected Sub btnSearch_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles btnSearch.Click
    Response.Write(Page.IsValid)
    Dim queryStr As String = HttpUtility.UrlEncode(txtSearch.Text)
    Response.Redirect("http://www.google.com/search?q=" & queryStr)
End Sub
```

6- فایل *Google.ascx* را از Solution Explorer انتخاب و درون *WebPartZone2* در صفحه *Default.aspx* رها کنید (شکل ۳-۶ را ببینید)



شکل ۳-۶. انتقال user control به webpart Zone

7- دقت کنید کنترل ساخته شده فاقد عنوان *title* می باشد برای افزودن عنوان به آن به *source view* رفته و خاصیت *title* را به *User Control* اضافه کنید.

8- دکمه F5 را فشار دهید تا خروجی کار را مشاهده کنید خروجی برنامه شکل ۳-۷ خواهد بود.

استفاده از Iwebpart Interface

برای استفاده جامع از *User Control* بعنوان یک *web part*، نیازمند استفاده از واسطه *Iwebpart* و خواص آن هستید. استفاده از این واسطه به *User Control* امکان دسترسی به خواص ویژه ای از *web part* از جمله عنوان (*title*) کنترل *webpart*، یک آیکن برای *webpart* و ... را می دهد.

Partial Class *Google_ascx*

Inherits *System.Web.UI.UserControl*

Implements *IWebPart*

چنانچه در *web User Control* از *Iwebpart Interface* استفاده کرده باشید خواصی مانند *title* و *TitleUrl* در پنجره *properties* از *web User Control* ظاهر می شوند. در حالیکه در حالت عادی این خواص وجود ندارند. برای کسب اطلاعات بیشتر راجع به *Iwebpart Interface*، در MSDN عبارت *Iwebpart members* را جستجو کنید.



شکل ۳-۷. نمایش webpart در صفحه

۳-۱-۱. پاورقی بخش ...

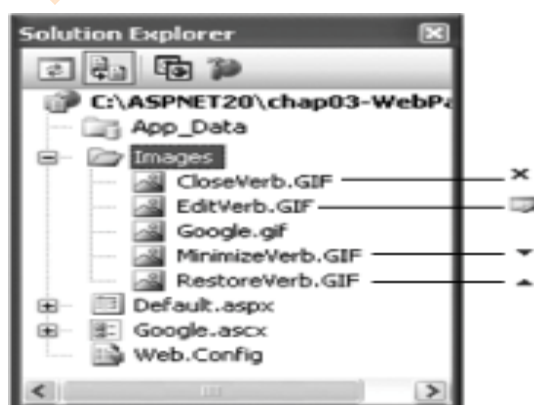
... سفارشی نمودن (Customize) ظاهر webpart ها

به طور پیش فرض ظاهر کنترل‌های webpart توسط لایه پیش فرض webpartzone طراحی می شوند همچنین گزینه های webpart مانند Close, minimize و ... بطور پیش فرض بصورت لینک نمایش داده می شوند . شما می توانید ظاهر یک webpart را بر اساس سلیقه خود تغییر دهید و خواصی را با خواص پیش فرض تعریف شده در webpart zone جایگزین نمائید. در ادامه به بررسی نحوه انجام این کار می پردازیم.

یادداشت : به یاد داشته باشید تنظیمات webpart zone بر روی webpart و محتویات آن اعمال می شود و برای تغییر ظاهر webpart مجبورید تنظیمات webpart zone را تغییر دهید.

1- ابتدا به چند عکس به عنوان آیکن (Icon) نیاز داریم . عکسهای نشان داده شده در شکل ۳-۸ را به شاخه Images پروژه اضافه کنید.

نکته : عکسها را می توانید از آدرس <http://www.oreilly.com/catalog/aspnetadn/> دانلود کنید.

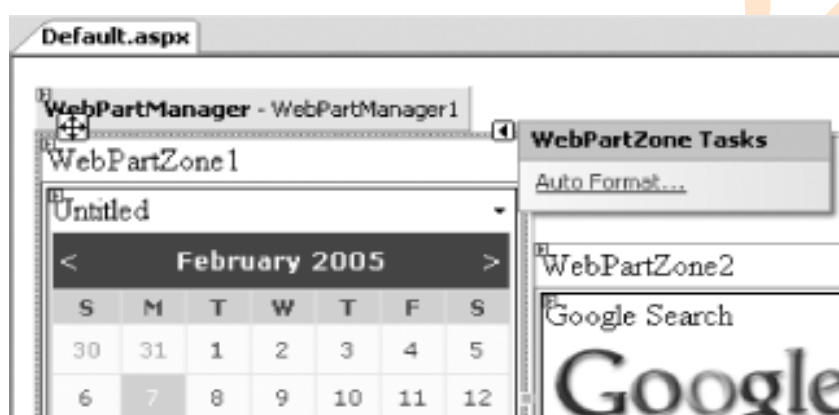


شکل ۳-۸. عکسهای اضافه شده به شاخه Images

2- حالا می خواهیم هر کدام از عکسها را به یکی از گزینه های webpart اختصاص دهیم ، webpart zone را انتخاب و پنجره properties آنرا باز کنید. در پنجره properties خواص زیر را پیدا کنید و خاصیت Imageurl هر کدام را مانند زیر تنظیم کنید :

- CloseVerb.ImageUrl="Images/CloseVerb.gif"
- EditVerb.ImageUrl="Images/EditVerb.gif"
- MinimizeVerb.ImageUrl="Images/MinimizeVerb.gif"
- RestoreVerb.ImageUrl="Images/RestoreVerb.gif"

3- بر روی لینک Auto format... در WebPartZone Tasks menu کلیک کنید و شمای Professional را به آن اعمال کنید. (شکل ۳-۹ را ببینید) همچنین تم Color به WebPartZone2 اعمال کنید.



شکل ۳-۹. اعمال Professional بر روی WebPartZone

یادداشت : تنظیمات اعمال شده بر یک webpart zone بر روی تمامی webpart های موجود در آن webpart zone اعمال می شود. چنانچه یک webpart از یک webpart zone به webpart zone دیگری انتقال یابد ، تنظیمات webpart zone دوم را بخود می گیرد.

4- کلید F5 را فشار دهید. همانگونه که مشاهده می کنید کنترل Calendar در webpart zone اول بجای لینک از آیکن جهت نمایش گزینه های minimize و Close استفاده کرده است (شکل ۳-۱۰ را ببینید)، در حالیکه Web User control موجود در WebPartZone دوم هیچ آیکنی ندارد. در بخش بعدی به بررسی چیدمان webpart ها توسط کاربر می پردازیم.



شکل ۳-۱۰. نمایش دو webpart در دو webpart zone

۳-۲. منابع اطلاعاتی :

یکی از عمومی ترین سئوالات راجع به webpart ها در Aspert 2.0 سازگاری آنها را SharePoint Portal Server است . نسخه فعلی SharePoint Portal Server هنوز با webpart های Asp.Net 2.0 سازگار نیست . اما مطمئناً " نسخه بعدی آن سازگار خواهد بود. برای آشنایی با نحوه استفاده برای webpart برای SharePoint Portal Server 2003 در دات نت مقاله زیر را مطالعه کنید.

<http://www.devx.com/dotnet/Article/17518>

۳-۲. ساخت webpart با قابلیت شخصی سازی (Personalizable)

یادداشت : در این بخش با نحوه ساخت webpart های شخصی و چگونگی صادر کردن (Export) آن ها به فایلها آشنا می شوید.

در بخش قبل آموختید که چگونه با استفاده از یک کنترل وب سرور و یا یک web user's control یک webpart طراحی کنید در این بخش شیوه دیگری از ساخت webpart ها را بررسی می کنیم : استفاده از کنترل سفارشی شده.

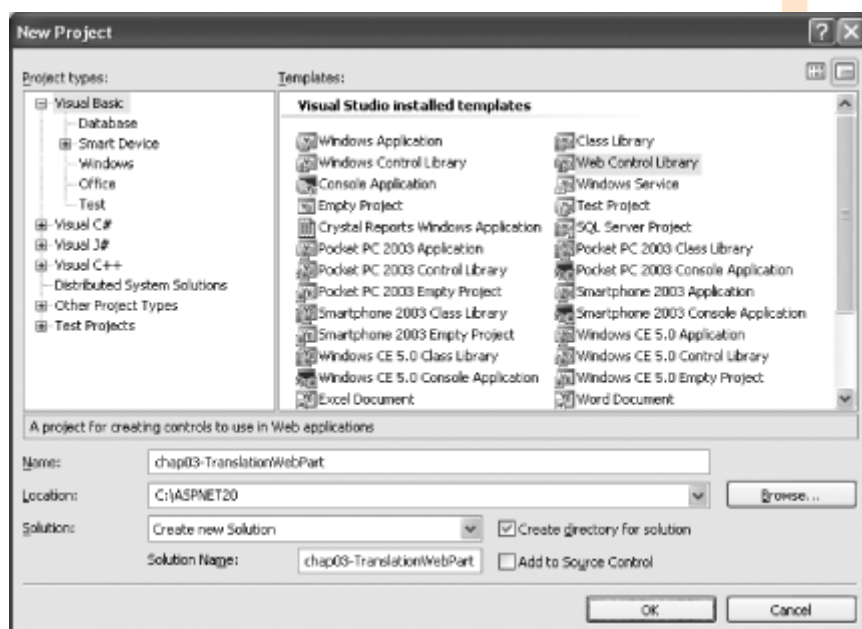
از طریق یک کنترل سفارشی شده انعطاف پذیری بیشتری برای طراحی UI (User Interface) خواهید داشت. علاوه بر این با استفاده از ارث بری از کلاس webpart قادر به انجام اعمالی هستید که بطور عادی در دسترس نیستند، مانند صدور یک webpart به فایل همچنین یاد می گیرید webpart های شخصی قادر به نگهداری اطلاعات کاربران طراحی کنید.

نکته : webpart های شخصی سازی شده بر روی web user control هم قابل اعمال هستند.

۳-۲-۱. مراحل انجام کار :

در این بخش ، می خواهیم یک کتابخانه web Control سفارشی به برنامه ای که در بخش قبل ایجاد کردیم ، اضافه کنیم. این کتابخانه امکان ساخت یک web Control سفارشی را فراهم می کند ، در ادامه خواهید دید چگونه می توان این web Control را به یک Web Part تبدیل کرد. این کنترل سفارشی شده یک web service جهت ترجمه کلمات انگلیسی به فرانسه فراهم می کند. کنترل همچنین آخرین کلمه (ترجمه شده) را نگهداری می کند بطوریکه هنگام بازدید مجدد صفحه توسط کاربر ، کلمه ترجمه شده نمایش داده می شود.

- 1- ابتدا بایستی کتابخانه توضیح داده شده را به پروژه اضافه کنید. VS 2005 را باز کنید، پروژه جدیدی به پروژه بخش قبل (*C:\ASPNET20\chap03-Webparts*) اضافه کنید. (File → Add → New (Project) نام آنرا *C:\ASPNET20\chap03-TranslationWebPart* قرار دهید. (شکل ۳-۱۱)



شکل ۳-۱۱. افزودن پروژه جدید به پروژه فعلی

- 2- یک web Refrence جهت استفاده از Web service مخصوص ترجمه از آدرس <http://www.webservices.com/TranslateService.asmx?WSDL> به پروژه دوم اضافه کنید. (بر روی *chap03-TranslationWebPart* در Solution Explorer راست کلیک کنید و گزینه Add Web Reference... را انتخاب نمایید)
این مرحله مترجم را بعنوان یک web Service فراهم می کند.

- 3- برای حفظ انسجام پروژه فایل سورس پروژه را از نام پیش فرض *WebCustomControl1.vb* به *TranslationWebPart.vb* تغییر نام دهید.

4- اکنون پنجره Solution Explorer پروژه مشابه شکل ۳-۱۲ شده است.



شکل ۳-۱۲. Solution Explorer شامل دو پروژه

5- برای اینکه کنترل طراحی شده قابلیت اجرا داشته باشد قطعه کد مثال ۳-۱ را درون فایل *TranslationWebPart.vb* وارد کنید:

```
Imports System.Web.UI.WebControls.WebParts
```

```
Public Class TranslationWebPart
    Inherits WebPart
```

```
    Private pStrText As String = Nothing
    Private txtStringToTranslate As TextBox
```

```
    Private lblTranslatedString As Label
```

```
    Public Sub New( )
        Me.AllowClose = True
    End Sub
```

```
<Personalizable( ), WebBrowsable( )> _
    Public Property strToTranslate( ) As String
        Get
            Return pStrText
        End Get
        Set(ByVal value As String)
            pStrText = value
        End Set
    End Property
```

```

Protected Overrides Sub CreateChildControls( )
    Controls.Clear( )

    '---display a textbox
    txtStringToTranslate = New TextBox( )
    txtStringToTranslate.Text = Me.strToTranslate
    Me.Controls.Add(txtStringToTranslate)

    '---display a button
    Dim btnTranslate As New Button( )
    btnTranslate.Text = "Translate"
    AddHandler btnTranslate.Click, AddressOf Me.btnTranslate_Click
    Me.Controls.Add(btnTranslate)

    '---display a label
    lblTranslatedString = New Label( )
    lblTranslatedString.BackColor = _
        System.Drawing.Color.Yellow
    Me.Controls.Add(lblTranslatedString)
    ChildControlsCreated = True
End Sub

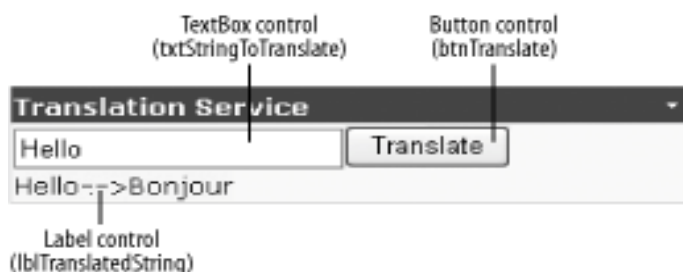
Private Sub btnTranslate_Click(ByVal sender As Object, _
    ByVal e As EventArgs)
    '---display the translated sentence
    If txtStringToTranslate.Text <> String.Empty Then
        Me.strToTranslate = txtStringToTranslate.Text
        txtStringToTranslate.Text = String.Empty

        '---access the web service
        Dim ws As New com.webservices.www.TranslateService
        lblTranslatedString.Text = "<br/>" & Me.strToTranslate & "-->" & _
            ws.Translate(com.webservices.www.Language.EnglishTOFrench, _
                Me.strToTranslate)
    End If
End Sub
End Class

```

مثال ۳-۱. فایل *TranslationWebPart.vb*

6- web part ساخته شده در شکل ۳-۱۳ نشان داده شده است این کنترل امکان ورود یک رشته انگلیسی و ترجمه آن به فرانسه را برای کاربران فراهم می کند. متن ورودی پس از ترجمه از طریق یک lable نمایش داده می شود.

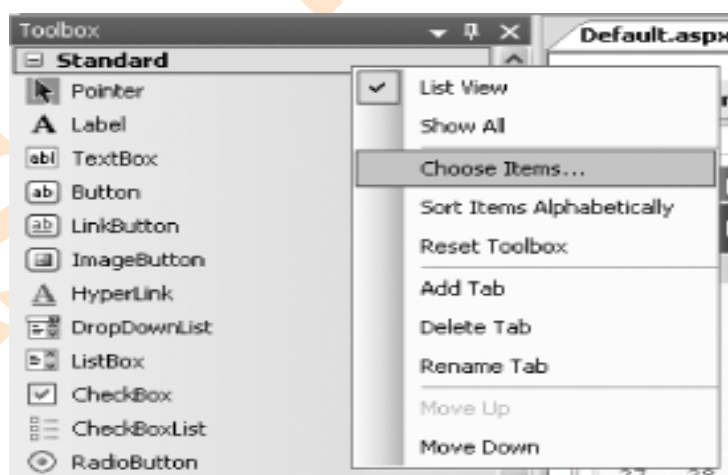


شکل ۳-۱۳. کنترل ساخته شده

7- اکنون شما یک کنترل سفارشی ایجاد کرده اید اما برای استفاده از آن در پروژه اصلی (-chap03 WebParts) بایستی DLL آنرا ایجاد کنید. برای این کار بر روی TranslationWebPart در Solution Explorer راست کلیک و گزینه Build را انتخاب کنید.

8- حالا به صفحه Default.aspx پروژه اصلی باز گردید.

9- برای استفاده از کنترل ساخته شده در این صفحه، آنرا به Toolbox اضافه می کنیم. برای این منظور بر روی بخش Standard از toolbox راست کلیک کنید و گزینه Choose Items... را انتخاب نمایید. (شکل ۳-۱۴ را ببینید)



شکل ۳-۱۴. افزودن آیتم جدید به toolbox

10- در پنجره باز شده دکمه Browse... را کلیک کنید و فایل chap03-TranslationWebPart.dll را انتخاب کنید. (فایل در آدرس C:\ASPNET20\chap03-TranslationWebPart\bin قرار دارد) گزینه OK و باز هم OK را انتخاب کنید.

11- اکنون می توانید کنترل را در toolbox مشاهده کنید. (شکل ۳-۱۵)



شکل ۳-۱۵. کنترل TranslationWebPart اضافه شده به toolbox

12- کنترل TranslationWebPart موجود در toolbox را به WebPartZone2 در صفحه Default.aspx اضافه کنید. (شکل ۳-۱۶)

13- همانطور که می بینید web part شما فاقد عنوان است برای افزودن عنوان به Source View صحنه رفته و همانند زیر خاصیت Title را به WebPartZone2 اضافه کنید.

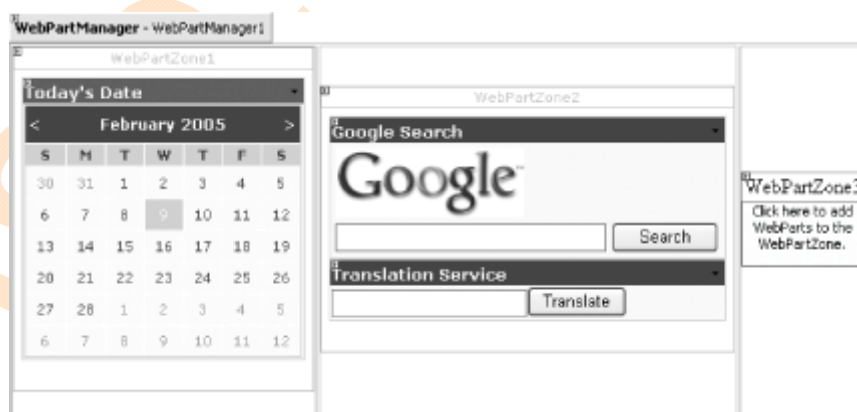
<ZoneTemplate>

<uc1:Google ID="Google1" title="Google Search" runat="server" />

<cc1:TranslationWebPart runat="server" ID="TranslationWebPart1" title="Translation Service" />

</ZoneTemplate>

14- به Design View باز گردید و نتیجه کار را مشاهده کنید.



شکل ۳-۱۶. افزودن webpart سفارشی به WebPartZone2

15- کلید F5 را فشار دهید . هم اکنون قادر به استفاده از کنترل طراحی شده هستید. بعنوان مثال کلمه Hello را تایپ کنید و نتیجه را مشاهده نمایید. (شکل ۳-۱۷) کنترل ایجاد شده همانند سایر web part ها می تواند کمینه (minimize) یا بسته (close) شود.



شکل ۳-۱۷. استفاده از web part

۳-۲-۲. تشریح عملکرد برنامه :

آنچه در این بخش انجام دادیم طراحی web part سفارشی بود. چنانچه به خط تعریف `strToTranslate` دقت کنید ، متوجه دو خاصیت قبل از آن می شوید :

```
<Personalizable( ), WebBrowsable( ) > _
Public Property strToTranslate( ) As String
...
```

خاصیت `Personalizable` امکان نگهداری اطلاعات درون web part را جهت خصوصی سازی web part برای هر کاربر فراهم می کند. دو محدوده برای خاصیت `personalizable` می توانید انتخاب کنید : `User` (حالت پیش فرض) و `Shared`. در این مثال از حالت پیش فرض استفاده کردیم ، از اینرو هر زمان که کاربر متنی را جهت ترجمه وارد می کند این متن ذخیره می شود. به عبارت دیگر webpart آخرین متن وارد شده هر کاربر را برایش ذخیره می کند.

یادداشت : خصوصی سازی در web part ها برای کنترل‌های معمولی (Server Controls) کنترل‌های سفارشی (Costume Controls) و User Controls ها استفاده می شود.

حالت دوم ، (Shard) اطلاعات را ذخیره می کند ولی این اطلاعات را در اختیار تمامی کاربران قرار می دهد. بمنظور استفاده از این حالت از تعریف زیر استفاده کنید:

```
<Personalizable(PersonalizationScope.Shared), WebBrowsable( ) > _
Public Property strToTranslate( ) As String
...
```

با تعریف فوق تمامی کاربران قادر به دیدن متن نوشته شده توسط آخرین کاربر خواهند بود. راجع به خاصیت `<WebBrowsable>` در بخش ۳-۵ از همین فصل بحث خواهیم کرد.

بررسی خصوصی سازی کاربر

از آنجا که از تائید هویت پیش فرض windows در پروژه استفاده کردیم درک نحوه خصوصی سازی کنترلها برای کاربران مختلف کمی مشکل بود. برای درک بهتر مطلب مراحل زیر را انجام دهید : از طریق ASP.NET Web Site Administration Tool (ASP.NET Configuration) → (WebSite) دو کاربر به وب سایتان اضافه کنید. برای آشنایی با نحوه اضافه کردن کاربر به سایت به فصل ۵ مراجعه کنید.

یک صفحه با نام *Login.aspx* شامل یک کنترل login به پروژه اضافه کنید.

فایل web. config را بصورت زیر اصلاح کنید :

```
<authentication mode="Forms" />
```

```
<authorization>
```

```
<deny users="?" />
```

```
</authorization>
```

Default.aspx را انتخاب کنید و دکمه F5 را فشار دهید. از آنجا که تائید هویت نشده اید به صفحه login.aspx هدایت می شوید با حساب کاربر اولی که ایجاد کردید به سایت وارد شوید درون Textbox متنی را وارد و Translate را کلیک کنید.

مجددا صفحه login.aspx را لود کنید و با حساب کاربری دوم وارد شوید ، مانند قبل کلمه ای را وارد و دکمه translate را کلیک کنید.

برای بار دوم به صفحه login.aspx بروید و با حساب کاربری اول وارد شوید ، کلمه ای را که بار اول تایپ کردید مشاهده خواهید کرد.

۳-۲-۳. پاورقی بخش ...

... صادر کردن یک web part

شما می توانید web part را به یک فایل صادر (Export) کنید و آن فایل را (مثلا از طریق پست الکترونیکی) در اختیار سایر کاربران قرار دهید.

برای انجام این کار اعمال زیر را انجام دهید :

1- به Source view صفحه Default.aspx رفته، خاصیت Export mode را همانند کد زیر به Translation Web Part اضافه کنید:

```
<cc1:TranslationWebPart title="Translation Service"
    ExportMode=All
    runat="server" ID="TranslationWebPart1" />
```

نکته: فقط هنگامیکه کنترل سفارشی از کلاس web part ارث بری داشته باشد می توانید آنرا Export نمایید.

2- عنصر < web parts > را به web.Config اضافه کنید:

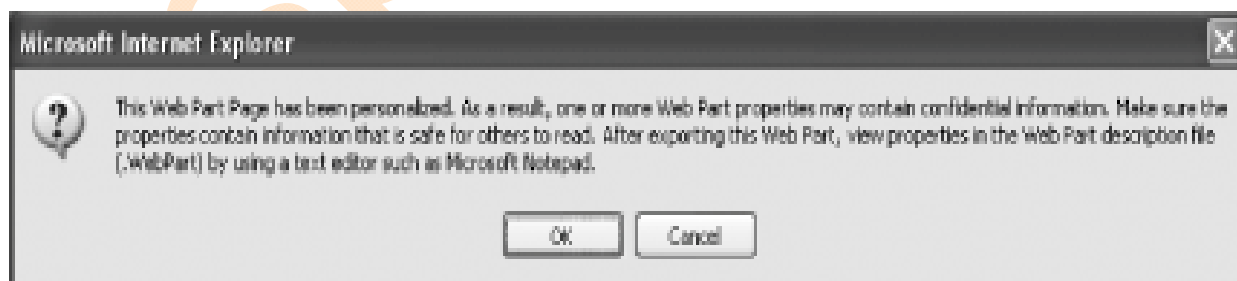
```
<system.web>
    <webParts enableExport="true" />
    ...
```

3- کلید F5 را فشار دهید و پیکان قسمت بالای web part را کلیک کنید، حالا گزینه Export را انتخاب نمایید. (شکل ۱۸-۳)



شکل ۱۸-۳. صدور یک web part

4- هنگام اجرای برنامه یک اخطار (warning) مبنی بر احتمال وجود اطلاعات محرمانه در web part نمایش داده می شود (شکل ۱۹-۳)



شکل ۱۹-۳. اخطار قبل از ذخیره web part

کنترل صدور اطلاعات :

خاصیت ExportMode یکی از مقادیر : All, none, NonSensitiveData را می پذیرد. حالت پیش فرض none است. حالت NonSensitiveData تنها اجازه صدور داده های که به صورت sensitive (از طریق پارامتر isSensitive در خاصیت <Personalizable>) مشخص شده اند را می دهد.

برای اینکه داده های درون web part را sensitive تعریف کنید ، پارامتر دوم خاصیت <Personalizable> را true کنید:

```
<Personalizable(PersonalizationScope.User, True) , _
    WebBrowsable( ) > _
Public Property strToTranslate( ) As String
...
```

5- ok کنید و در پنجره باز شده دکمه save را جهت ذخیره web part درون فایل کلیک کنید.

6- نام پیش فرض (*TranslationService.WebPart*) را تغییر ندهید و فایل را ذخیره کنید.

7- فایل *TranslationService.WebPart* را از طریق notepad باز کنید ، آنچه که می بینید یک فایل Xml است محتویات آنرا در مثال ۲-۳ میتوانید ببینید.

```
<webParts>
<webPart>
<metaData>
<type name="chap03_TranslationWebPart.TranslationWebPart" />
<importErrorMessage>Cannot import this Web Part.</importErrorMessage>
</metaData>
<data>
<properties>
<property name="AllowClose">True</property>
<property name="Width" />
<property name="strToTranslate">Hello</property>
<property name="AllowMinimize">True</property>
<property name="AllowConnect">True</property>
<property name="ChromeType">Default</property>
<property name="TitleIconImageUrl" />
<property name="Description" />
<property name="Hidden">False</property>
<property name="TitleUrl" />
<property name="AllowEdit">True</property>
<property name="Height" />
```



```

<property name="HelpUrl" />
<property name="Title">Translation Service</property>
<property name="CatalogIconImageUrl" />
<property name="Direction">NotSet</property>
<property name="ChromeState">Normal</property>
<property name="AllowZoneChange">True</property>
<property name="AllowHide">True</property>
<property name="HelpMode">Navigate</property>
<property name="ExportMode">All</property>
</properties>
</data>
</webPart>
</webParts>

```

مثال ۳-۲. محتویات فایل *TranslationService.WebPart*

دقت کنید؛ کلمه Hello در اینجا نیز مشاهده می شود.

۳-۲-۴. منابع اطلاعاتی :

برای کسب اطلاعات بیشتر راجع به ساخت کنترل‌های سفارشی می توانید به آدرسهای زیر مراجعه کنید:

- http://www.ondotnet.com/pub/a/dotnet/excerpt/progaspdotnet_14/index2.html
- <http://samples.gotdotnet.com/quickstart/aspplus/doc/webctrlauthoring.aspx>

۳-۳. جابجایی web part ها توسط کاربران

یادداشت: شما می توانید به کاربران سایتتان امکان چیدمان دلخواه web part ها را بدهید.

تا اینجا راجع به ساخت و پیکربندی web part ها بحث کردیم اما هنوز راجع به مهمترین قابلیت آنها یعنی امکان جابجایی از مکانی به مکان دیگر صحبت نکردیم.

۳-۳-۱. مراحل انجام کار :

در این بخش خواهید دید که چگونه کاربران web part ها را بطور مستقیم درون مرورگر (browser) جابجا می کنند. برای اینکار از یک جفت دکمه رادیویی جهت فعال و غیر فعال نمودن این قابلیت استفاده می کنیم.

سپس مشاهده نحوه رفتار web part هنگام تغییر مکان آن خواهید بود.

1- از پروژه ای که در بخش قبل ایجاد کردیم استفاده کنید (C:\ASPNET20\chap03-Webparts) و یک کنترل RadioButtonList با ۲ گزینه به فرم اضافه کنید. یکی به نام Browse Display Mode و دیگری Design Display Mode همچنین چک باکس Auto post Back این کنترل را در RadioButtonlist Tasks بزنید. (شکل ۳-۲۰ را ببینید)

```
<asp:RadioButtonList ID="rbIMode" runat="server" AutoPostBack="True">
    <asp:ListItem>Browse Display Mode</asp:ListItem>
    <asp:ListItem>Design Display Mode</asp:ListItem>
</asp:RadioButtonList>
```



شکل ۳-۲۰. افزودن یک کنترل Radio Button list

2- بر روی Radio Button list کلیک مضاعف کنید و در Code_behind صفحه کد زیر را وارد کنید کنترل WebPartManager حالت نمایش تمامی web part های موجود در صفحه را کنترل می کند. بطور پیش فرض این کنترل حالت نمایش تمامی web part را browse Display قرار می دهد (BrowseDisplayMode) برای اینکه کاربر بتواند web part ها را جابجا کند بایستی این حالت به Design Display تغییر یابد. (DesignDisplayMode).

```
Protected Sub rbIMode_SelectedIndexChanged( _
    ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles rbIMode.SelectedIndexChanged
    Select Case rbIMode.SelectedIndex
        Case 0 : WebPartManager1.DisplayMode = _
            WebPartManager.BrowseDisplayMode
```

Case 1 : WebPartManager1.DisplayMode = _
WebPartManager.DesignDisplayMode

End Select

End Sub

نکته : برای اجرای این مثال بایستی از کاربر تائید هویت شده استفاده کنید. برای راحتی کار حالت شناسایی را در فایل Web.Config گزینه Windows قرار دهید. بصورت زیر :

<authentication mode="Windows" />

3- برای تست برنامه کلید F5 را فشار دهید . هنگامیکه صفحه لود شد گزینه Design Display Mode را انتخاب کنید. همانطور که مشاهده می کنید کادر زمینه هر سه کنترل WebPartZone به نمایش در می آیند . اکنون می توانید هر کدام از web part ها را از طریق نوار عنوانشان جابجا کنید و به هر جای صفحه که می خواهید انتقال دهید. (شکل ۳-۲۱ را ببینید)



شکل ۳-۲۱. انتقال web part از یک ناحیه به ناحیه دیگر

4- می توانید متن پیش فرض header هر WebPartZone را از طریق خاصیت HeaderText تغییر دهید.

5- چنانچه Google Search Web Part را به داخل اولین Web Part zone انتقال دهید حاصل شکل ۳-۲۲ خواهد بود، دقت کنید ؛ اکنون Google Search از تنظیمات WebPartZone1 استفاده می کند. همچنین هر کدام از لینکهای Minimize و close دارای یک آیکن هستند.



شکل ۳-۲۲. تغییر ظاهر Google Search

6- هنگامیکه جابجایی web part به اتمام رسید . گزینه Browse Display mode را انتخاب کنید تا صفحه به حالت Browse باز گردد.

۳-۲-۲. تشریح نحوه عملکرد برنامه :

عملی که انجام دادید تنها پیکر بندی یک web part بود بگونه ای که بتواند از یک Web Part zone به Web Part zone دیگری انتقال یابد. همچنین شاهد تغییر رفتار web part هنگام ورود به Web Part zone جدید بودید. تمامی این اعمال توسط کنترل WebPartManager مدیریت می شوند. هنگامیکه گزینه Display Mode را انتخاب می کنید با حالت نمایش WebPartManager با WebPartManager.DesignDisplayMode تنظیم می شود:

Case 1 : `WebPartManager1.DisplayMode = _
WebPartManager.DesignDisplayMode`

نهایتا جهت ثابت نمودن web part ها در مکان جدید نشان کافیست به حالت browse باز گردید:

Case 0 : `WebPartManager1.DisplayMode = _
WebPartManager.BrowseDisplayMode`

حال چنانچه مرورگر (IE) را ببندید و کلید F5 را مجددا فشار دهید متوجه می شوید web part ها در مکان جدید باقی می مانند. web part ها قابلیت نگهداری آخرین مکان استقرار خود را دارند. اما چگونه ؟ !

در واقع اطلاعات مربوط به موقعیت هر web part درون جدولی (Table) به نام aspNet_PersonalizationPerUser ذخیره می شود ، این جدول بخشی از پایگاه داده ASPNETDB.MDF (SQL Express 2005) است که در شاخه App_Data پروژه قرار دارد. جدول ۱-۳ رکوردهای (Records) این جدول را نشان می دهد.

یادداشت : هنگامیکه برنامه ای را بر روی سیستمتان اجرا می کنید ، نام کاربری شما (user name) به عنوان ID در جدول aspNet_Users نگهداری می شود ، این جدول در پایگاه داده ASPNETDB.MDF قرار دارد.

Field	Value
Id	{94CACA48-36D3-4F64-A6AB-BD5352FF4522}
PathId	{BF32CB1F-DEB7-4384-B3DB-5D9B45F59FF9}
UserId	{F9D14F92-540B-4A64-A4B2-75DB80857E78}
PageSettings	<Binary>
LastUpdatedDate	2/7/2007 8:48:58 PM

جدول ۱-۳ محتویات یک رکورد از جدول aspNet_PersonalizationPerUser

برای بازگرداندن web part به مکان اصلی خود کافیت سطر (متعلق به کاربر) مربوطه را پاک کنید.

۳-۳-۳. پاورقی بخش ...

... باز گرداندن web part بسته (close) شده

حین اجرای برنامه های این بخش ممکن است بر روی دکمه close یک web part کلیک کنید، اگر این کار را انجام دهید در باز گرداندن آن دچار مشکل می شوید. اما نگران نباشید web part برای همیشه ناپدید نشده است تنها کافیت روش باز گرداندن آنرا بدانید.

دو راه برای باز گرداندن web part بسته شده وجود دارد :

- پاک کردن سطر مربوطه از جدول aspNet_PersonalizationPerUser که پیشتر ارائه شد.
- استفاده از کنترل PageCatalogPart که در بخش ۳-۴ به آن می پردازیم .

همچنین می توانید لینک close یک web part را غیر فعال کنید ، برای این کار کافیت مشخصه Visible از خاصیت CloseVerb را false کنید.

۳-۳-۴. منابع اطلاعاتی :

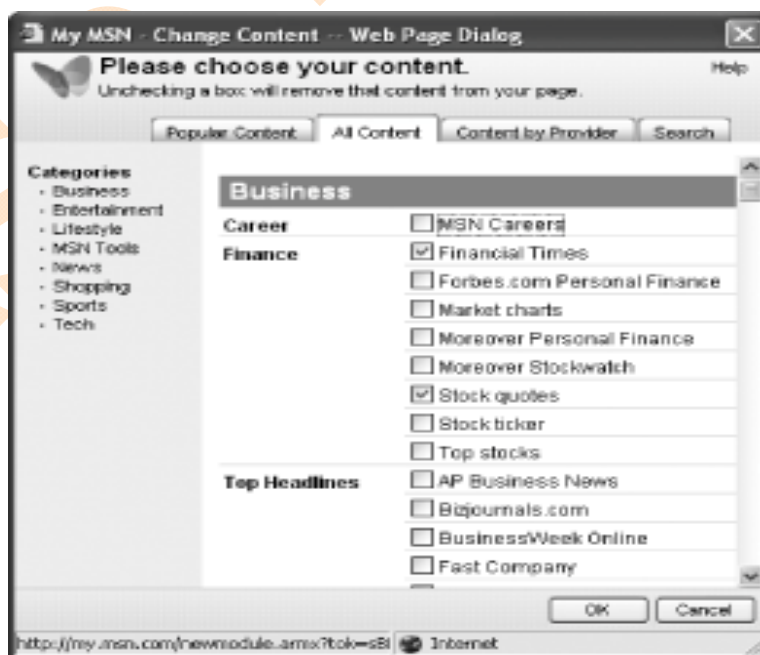
جهت کسب اطلاعات بیشتر راجع به نحوه نوشتن سرویس دهنده های (providers) سفارشی مقالات زیر را مطالعه کنید:

- <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspnet/html/asp02182004.asp>
- <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspnet/html/asp04212004.asp?frame=true>

۴-۳. افزودن Web Part در زمان اجرا توسط کاربر

یادداشت : شما می توانید لیستی از Web Part ها برای انتخاب در اختیار کاربر قرار دهید .

علاوه بر نمایش Web Part ها در زمان بارگذاری صفحه، می توانید لیستی از سایر Web Part ها در اختیار کاربر قرار دهید بطوریکه بتواند آنها را انتخاب و به صفحه اضافه کند . بعنوان مثال در سایت MSN لیستی در اختیار شما قرار می گیرد که می توانید از آن، موضوع مورد نظرتان را انتخاب و به صفحه تان اضافه کنید . (شکل ۳-۳۳ را ببینید)



شکل ۳-۳۳ . افزودن اطلاعات موضوعی در <http://my.msn.com>

برای فراهم نمودن امکان انتخاب Web Part توسط کاربر از کترهای DeclarativeCatalogPart، PageCatalogPart و ImportCatalogPart استفاده می کنیم .

۳-۴-۱. مراحل انجام کار :

در این بخش می آموزید با استفاده از کنترل CatalogZone مجموعه ای از Web Part های موجود را برای انتخاب توسط کاربر لیست کنید .

علاوه بر CatalogZone سه کنترل DeclarativeCatalogZone و PageCatalogZone و ImportCatalogZoneso در انجام این کار به شما کمک می کنند .

قبل از هر چیز بطور خلاصه با این کنترلها آشنا می شویم :

CatalogZone

محل نگهداری کنترلهای CatalogPart (DeclarativeCatalogPart, PageCatalogPart, ImportCatalogPart) می باشد . از این کنترل برای ایجاد مجموعه ای از Web Part ها استفاده می شود که کاربر می تواند آنها را انتخاب کند .

DeclarativeCatalogPart

این کنترل امکان افزودن مجموعه ای از Web Part ها به صفحه را فراهم می آورد .

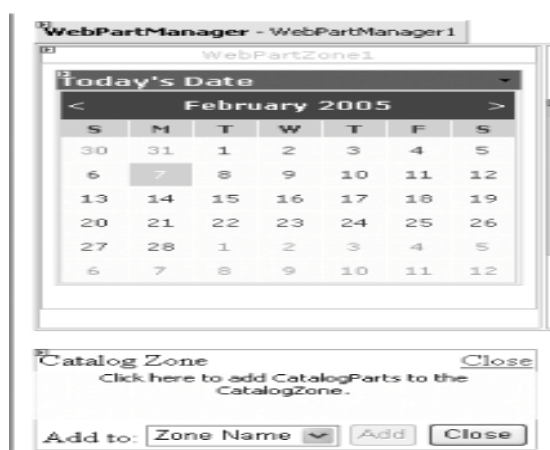
PageCatalogPart

مجموعه Web Part های بسته (Close) شده توسط کاربر را نگهداری می کند و به کاربر امکان بازیابی (Restore) آنها را می دهد .

ImportCatalogPart

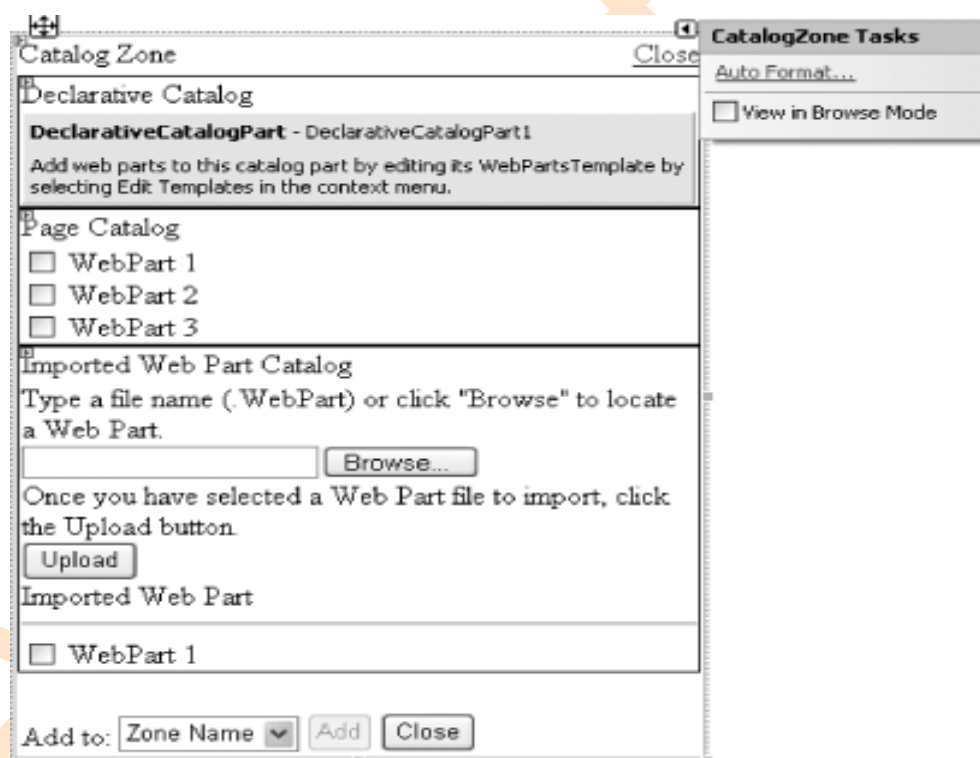
امکان وارد کردن یک Web Part به همراه تنظیمات از قبل تعریف شده به صفحه را می دهد . در ادامه بخش از پروژه ایجاد شده ابتدای فصل (C:\ASPNET20\chap03-Webparts) استفاده می کنیم .

1- یک کنترل CatalogZon از بخش WebParts در Tool Box به فرم اضافه کنید . (شکل ۳-۲۴) کنترل CatalogZon کنترل اصلی جهت میزبانی کنترلهای Catalog Part صفحه است .



شکل ۳-۲۴. افزودن کنترل CatalogZone به فرم

2- در این قسمت کنترل‌هایی که به کاربر امکان انتخاب بخش‌های مختلف لیست را می‌دهند، اضافه می‌کنید. یک کنترل DeclarativeCatalogPart، یک PageCatalogPart، و یک ImportCatalogPart به CatalogZone اضافه کنید. CatalogZone مشابه شکل ۳-۲۵ خواهد شد.



شکل ۳-۲۵. کنترل CatalogZone

3- کنترل DeclarativeCatalogPart به شما امکان افزودن مجموعه‌ای از Web Part ها به صفحه را می‌دهد. سپس کاربر می‌تواند Web Part های مورد نظرش را انتخاب و آنها را به صفحه اضافه کند. کنترل PageCatalogPart مجموعه Web Part های بسته شده توسط کاربر را نگهداری می‌کند که کاربر در صورت

تمایل بتواند آنها را باز گرداند . کنترل ImportCatalogPart به کاربر امکان می دهد Web Part ذخیره شده را درون صفحه لود کند .

4- از طریق لینک Auto Format... از CatalogZone Tasks شمای Classic را به CatalogZone اعمال کنید.

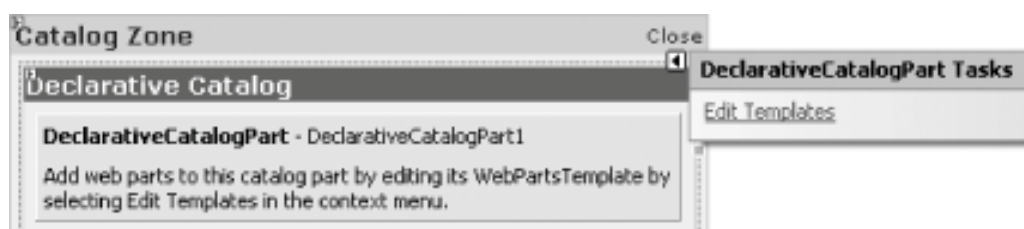
5- به کنترل Radio Batton list انتهای صفحه همانند کد زیر یک گزینه اضافه کنید :

```
<asp:RadioButtonList ID="rbIMode" runat="server" AutoPostBack="True">
    <asp:ListItem>Browse Display Mode</asp:ListItem>
    <asp:ListItem>Design Display Mode</asp:ListItem>
    <asp:ListItem>Catalog Display Mode</asp:ListItem>
</asp:RadioButtonList>
```

6- در رویداد SelectedIndexChanged از کنترل Radio Batton list کد زیر را اضافه کنید. از این کد برای نمایش CatalogZone هنگام انتخاب Catalog Display Mode استفاده می شود .

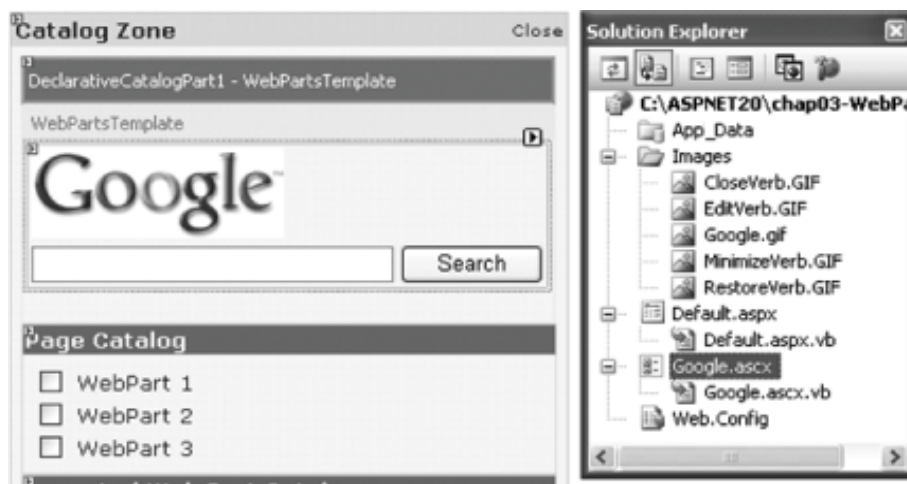
```
Protected Sub rbIMode_SelectedIndexChanged( _
    ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles rbIMode.SelectedIndexChanged
    Select Case rbIMode.SelectedIndex
        Case 0 : WebPartManager1.DisplayMode = _
            WebPartManager.BrowseDisplayMode
        Case 1 : WebPartManager1.DisplayMode = _
            WebPartManager.DesignDisplayMode
        Case 2 : WebPartManager1.DisplayMode = _
            WebPartManager.CatalogDisplayMode
    End Select
End Sub
```

7- در DeclarativeCatalogPart Tasks بر روی Edit Templates کلیک کنید . (شکل ۳-۲۶) وب پارت google Search را به این کنترل اضافه می کنیم تا کاربران امکان افزودن آن به صفحاتشان را داشته باشند .



شکل ۳-۲۶ ویرایش کنترل DeclarativeCatalogPart

8- فایل *Google.ascx* را از Solution Explorer به DeclarativeCatalogPart اضافه کنید. (شکل ۳-۲۷)
 ۲۷) با این کار Google Search Web Part از طریق DeclarativeCatalogPart برای انتخاب در اختیار کاربران قرار می گیرد.



شکل ۳-۲۷. افزودن Google Search به DeclarativeCatalogPart

9- بر روی Google Search درون DeclarativeCatalogZone راست کلیک کنید و گزینه Properties را انتخاب نمایید. خاصیت Title آنرا به Google Search تغییر دهید. این نام زمان اجرا در کنترل DeclarativeCatalogPart نمایش داده می شود.

10- کلید F5 را فشار دهید و گزینه Catalog Display Mode را انتخاب کنید. CatalogZone را مشاهده خواهید کرد.

نکته: نام پیش فرض کنترل DeclarativeCatalogPart در اجرا Declarative Catalog است، این نام برای کاربران چندان واضح نیست، برای تغییر آن بر روی کنترل DeclarativeCatalogPart در Visual Studio راست کلیک کنید، Properties را انتخاب و سپس خاصیت Title آنرا با عبارتی مثل "Available Web Parts" تنظیم نمایید.

11- محتویات DeclarativeCatalogPart همیشه نمایش داده می شوند. (شکل ۳-۲۸) برای افزودن Google Search؛ چک باکس google Search انتهای Catalog Zone را انتخاب سپس وب پارتی را که می خواهید google Search به آن اضافه شود در قسمت To: add انتخاب کنید: (در این مثال WebPartZone2) سپس دکمه add را کلیک کنید.



شکل ۳-۲۸. کنترل Catalog Zone در عمل

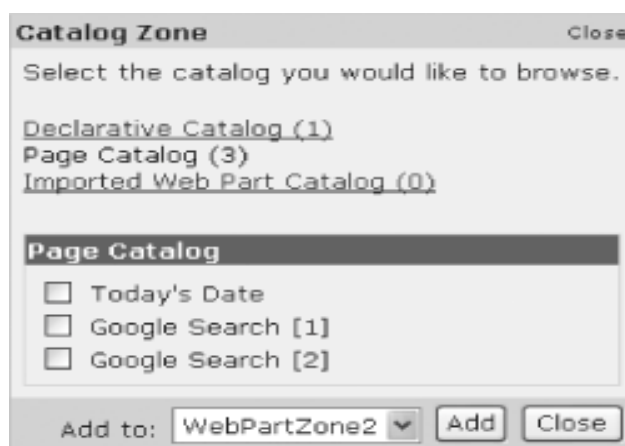
12- شکل ۳-۲۹ نتیجه افزون google Search به WebPartZone2 را نشان میدهد :



شکل ۳-۲۹. افزودن Google Search به WebPartZone2

13- با استفاده از لینک Close گوشه Web Part ها تمامی Web Part های موجود در صفحه را ببندید . در ادامه خواهید دید که چگونه می توان آنها را به صفحه باز گرداند .

14- مجدداً دکمه Catalog Display Mode را انتخاب کنید . این بار لینک Page Catalog را در Catalog Zone کلیک کنید (شکل ۳-۳۰) همانطور مشاهده می کنید Page Catalog سه Web Part بسته شده را نشان می دهد .

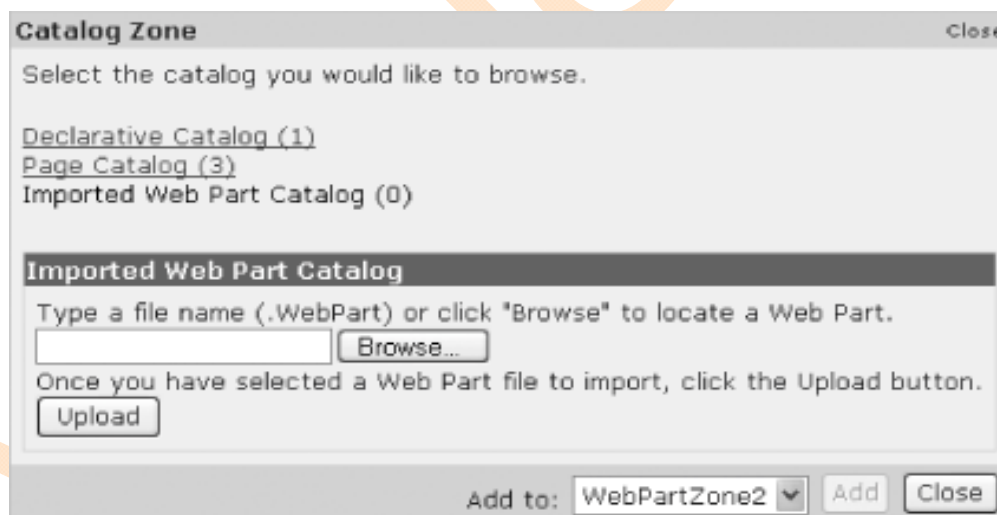


شکل ۳-۳۰. نمایش page catalog

15- برای بازگشت مجدد Web Part های بسته شده به صفحه ، Web Part مورد نظر را علامت بزنیید و پس از انتخاب Web Part Zone دکمه add را کلیک کنید .

16- در نهایت بر روی لینک Imported Web Part Catalog در Catalog Zone کلیک کنید . با استفاده دکمه Browse می توانید Web Part ذخیره شده را جهت لود شدن به صفحه انتخاب کنید .

17- WebPartZone مورد نظر را انتخاب و دکمه Add را کلیک کنید. (شکل ۳-۳۱)



شکل ۳-۳۱. Import کردن Web Part ذخیره شده

۳-۴-۲. پاورقی بخش...

... صدور (Export) وب پارت

ممکن است Web Part ای ایجاد کرده باشد و بخواهد آنرا با سایر کاربران به اشتراک گذارد (Sharing) در این حالت بایستی امکان صادر کردن Web Part را برای کاربر فراهم آورید .

برای اینکه بتوانید یک Web Part را صادر کنید ، بایستی یک کنترل سفارشی ایجاد کنید که از کلاس Web Part ارث بری داشته باشد ، برای انجام کار به بخش ۳-۲ همین فصل که قبلاً آنرا مطالعه کردید مراجعه کنید .

۳-۴-۳. منابع اطلاعاتی :

چنانچه می خواهید در مورد کنترل های Catalog Part اطلاعات بیشتری کسب کنید در MSDN موضوع "CatalogPart Class" را جستجو کنید .

۳-۵. ویرایش Web Part ها توسط کاربر در زمان اجرا

یادداشت : شما می توانید به کاربران سایت امکان سفارشی نمودن Web Part ها را بدهید .

علاوه بر جابجایی می توانید امکاناتی فراهم کنید که کاربران بتوانند ظاهر Web Part ها را نیز تغییر دهند . در این بخش می آموزید که چگونه از کنترل های AppearanceEditorPart, EditorPart (BehaviorEditorPart, LayoutEditorPart, PropertyGridEditorPart) برای این منظور استفاده کنید .

ابتداء به توصیف اجمالی این کنترل ها می پردازیم :

AppearanceEditorPart

این کنترل یک ویرایشگر در اختیار کاربر قرار می دهد که از طریق آن می تواند برخی خصوصیات Web Part (مثل عنوان (Title) ، عرض (With) و طول (Height)) را تغییر دهد.

BehaviorEditorPart

این کنترل نیز امکاناتی (از جمله نمایش دکمه های Edit , Close , Minimize) را برای کاربر فراهم می کند .

LayoutEditorPart

با استفاده از این کنترل کاربر می تواند برخی لایه های واسط کاربری (UI) را ویرایش کند .

PropertyGridEditorPart

این کنترل امکان ویرایش خصوصیات سفارشی Web Part را به کاربر می دهد .

۱-۵-۳. مراحل انجام کار :

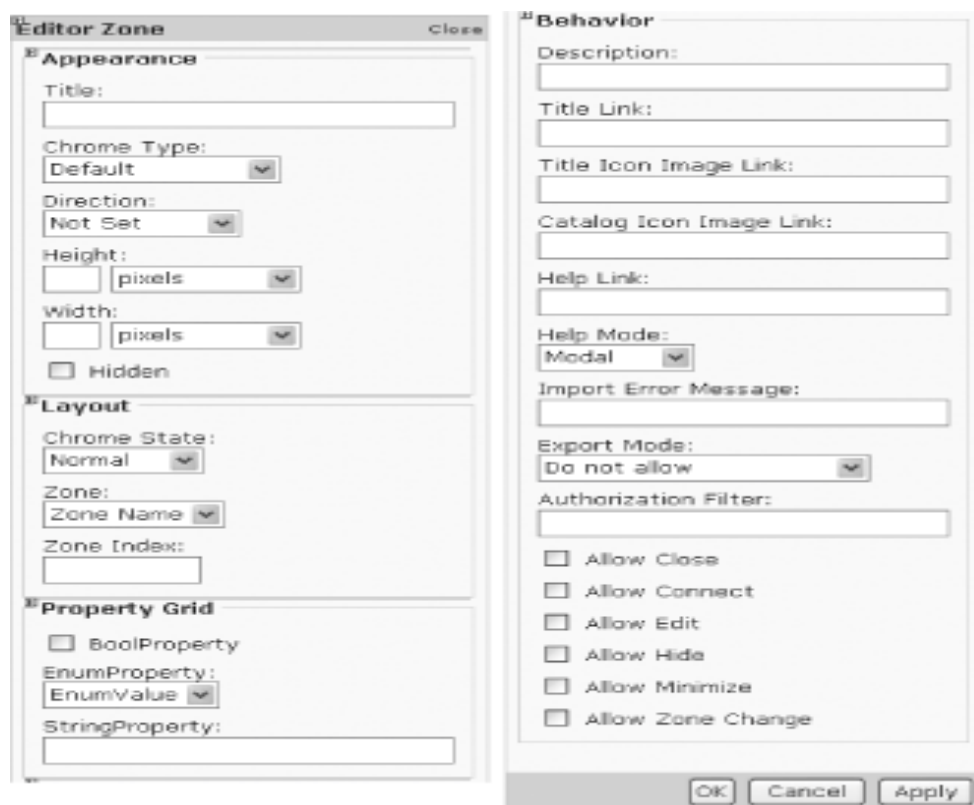
در این بخش خواهید دید که چگونه کاربر می تواند ظاهر و رفتار Web Part ها را تغییر دهد . ابتداء یک Editor Zone شامل چهار کنترل : appearanceEditorPart , BehaviorEditorPart , PropertyGridEditorPart , LayoutEditorPart به صفحه اضافه می کنیم . این کنترلها به کاربر اجازه ویرایش Web Part ها را می دهند ، بعنوان مثال کاربران می توانند عنوان یک Web Part را تغییر دهند ، یا اینکه از طریق انتخاب برخی دکمه های Web Part را فعال یا غیر فعال نمایند .

1- پروژه بخش قبل (C:\ASPNET20\chap03-Webparts) را باز کنید . یک کنترل EditorZone به فرم اضافه کنید (شکل ۳-۳۲) کنترل Editor Zone کنترل اصلی است ؛ در واقع محل نگهداری (میزبان) کنترلهای EditorPart در صفحه است .



شکل ۳-۳۲ . افزودن کنترل EditorZone به فرم

2- برای ویرایش Web Part به کنترل‌های Editor Part نیاز دارید. کنترل‌های LayoutEditorPart، BehaviorEditorPart، appearanceEditorPart، PropertyGridEditorPart، را به Editor Zone اضافه کنید. شمای Colorful را به EditorZone اعمال کنید (از طریق لینک Auto Format.... در منوی Editor Zone Tasks) هم اکنون Editor Zone همانند شکل ۳-۳۳ شده است.



شکل ۳-۳۳. کنترل EditorZone (به منظور نمایش بهتر عکسها آنها را جدا کردیم)

3- به کنترل Radio Button List انتهای فرم یک گزینه اضافه کنید. با انتخاب آن کاربر می تواند از قابلیت ویرایش Web Part استفاده کند:

```
<asp:RadioButtonList ID="rbIMode" runat="server" AutoPostBack="True">
  <asp:ListItem>Browse Display Mode</asp:ListItem>
  <asp:ListItem>Design Display Mode</asp:ListItem>
  <asp:ListItem>Catalog Display Mode</asp:ListItem>
  <asp:ListItem>Edit Display Mode</asp:ListItem>
</asp:RadioButtonList>
```

4- در رویداد SelectedIndexChanged کنترل Redio button list کد زیر را اضافه کنید:

```
Protected Sub rbIMode_SelectedIndexChanged( _
    ByVal sender As Object, _
    ByVal e As System.EventArgs) _
```

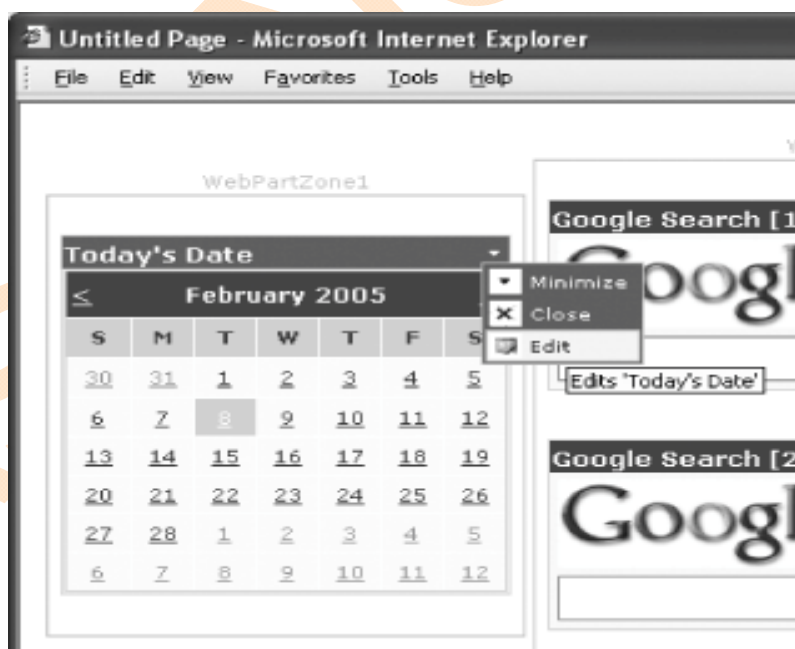
```

Handles rbIMode.SelectedIndexChanged
Select Case rbIMode.SelectedIndex
    Case 0 : WebPartManager1.DisplayMode = _
              WebPartManager.BrowseDisplayMode
    Case 1 : WebPartManager1.DisplayMode = _
              WebPartManager.DesignDisplayMode
    Case 2 : WebPartManager1.DisplayMode = _
              WebPartManager.CatalogDisplayMode
    Case 3 : WebPartManager1.DisplayMode = _
              WebPartManager.EditDisplayMode
End Select
End Sub
    
```

با این کد هنگامی که کاربر گزینه Edit Display Mode را انتخاب می کند؛ کنترل EditorZone نمایش داده میشود.

5- کلید F5 را فشار دهید و گزینه Edit Display Mode را انتخاب کنید. دقت کنید که تغییر خاصی به صفحه اعمال نمی شود.

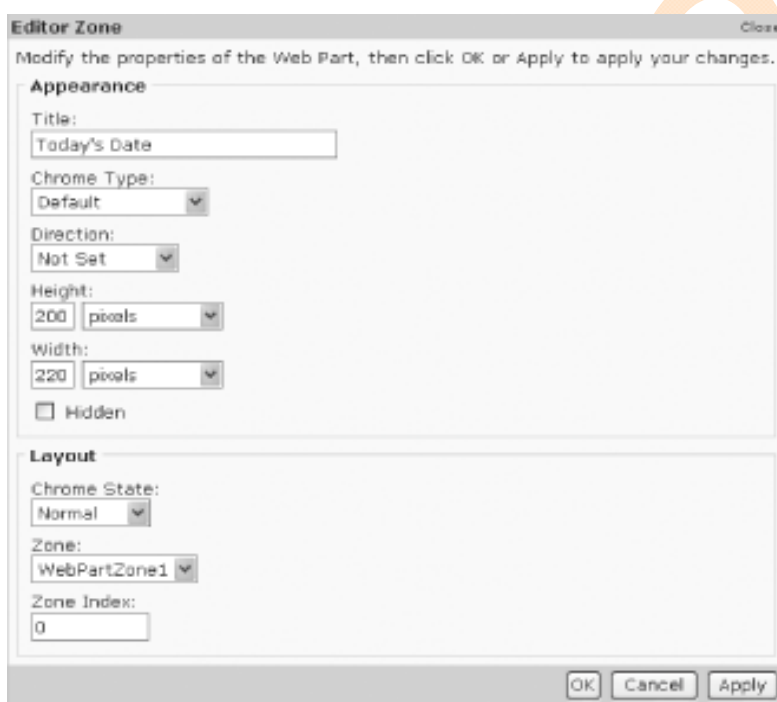
6- گوشه بالای وب پارت Calendar را کلیک کنید. لینک جدیدی با نام Edit نمایش داده میشود. (شکل ۳-۳۴) بر روی آن کلیک کنید.



شکل ۳-۳۴. ویرایش یک Web Part

7- در این مرحله کنترل EditorZone نمایش داده می شود. شما می توانید از طریق کنترل های AppearanceEditorPart و LayoutEditorPart خواص و ظاهر Web Part مورد نظر را تغییر دهید. (شکل ۳-۳۵). در نهایت دکمه OK را کلیک کنید.

نکته: دقت کنید که کنترل PropertyGridEditorPart برای این Web Part نمایش داده نمی شود، در ادامه (بخش ۲.۵.۳) راجع به PropertyGridEditorPart بیشتر توضیح خواهیم داد.



شکل ۳-۳۵. ویرایش ظاهر Web Part

۲.۵.۳ پاورقی بخش...

... نمایش کنترل PropertyGridEditorPart

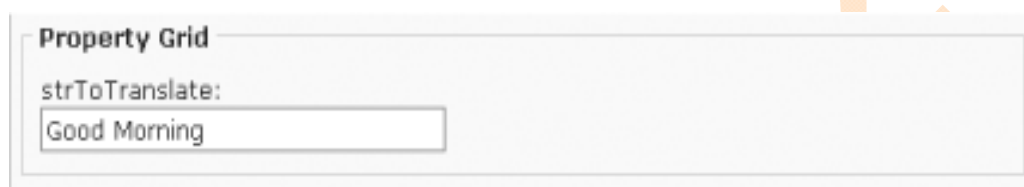
همانطور که مشاهده شد در حالت Edit صفحه؛ کنترل PropertyGridEditorPart نمایش داده نشد. پس کاربرد این کنترل چیست؟

برای نمایش کنترل PropertyGridEditorPart، یک Web Part باید دارای خواص عمومی باشد تا قابل نمایش (از طریق خاصیت WebBrowsable) و شخصی سازی (از طریق خاصیت Personalizable) باشد. Web User

Control ها این خواص را پیشانی نمی کنند. از این رو تنها کنترل هایی که در کلاس Web Part ارث بری دارند می توانند از کنترل PropertyGridEditorPart استفاده کنند.

اگر بر روی گزینه Edit از Translation Service Web Part کلیک کنید، کنترل PropertyGridEditorPart را مشاهده می کنید (شکل ۳-۳۶) علت هم این است که Translation Service دارای خواص عمومی و قابل نمایش است:

```
<Personalizable( ), WebBrowsable( ) > _
Public Property strToTranslate( ) As String
...
```



شکل ۳-۳۶. نمایش کنترل PropertyGridEditorPart

شما می توانید مقداری را برای خاصیت عمومی strToTranslate وارد کنید، هنگام بارگذاری مجدد Translation Service این مقدار به نمایش در می آید.

۳.۵. ۳. منابع اطلاعاتی :

برای کسب اطلاعات بیشتر راجع به کنترلهای EditorPart در MSDN عبارت، "EditorPart Class" را جستجو کنید.

۳.۶. برقراری ارتباط میان Web Part ها

یادداشت: در این بخش می آموزید چگونه میان Web Part های مختلف ارتباط برقرار کنید.

۳.۶. ۱. مراحل انجام کار:

برای مشاهده چگونگی برقراری ارتباط میان Web Part ها، ابتدا یک Web User Control جدید شامل یک کنترل Calendar طراحی می کنیم، هنگامیکه کاربر تاریخی را در Calendar انتخاب کرد، تاریخ انتخاب شده به Google web user control فرستاده می شود تا از طریق Text box موجود در آن نمایش داده شود.

1- پروژه بخش قبل (*C:\ASPNET20\chap03-Webparts*) را باز کنید و یک کلاس (class) جدید به آن اضافه کنید (بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های add new Item.... و سپس class را انتخاب کنید) نام آن را *ISelectedDate.vb* قرار دهید.

2- کد مربوط به کلاس *ISelectedDate.vb* در زیر آمده است. این کلاس هم توسط ارسال کننده (*CalendarUC.ascx*) و هم توسط دریافت کننده (*Google.ascx*) استفاده می شود. در واقع این کلاس رابط میان تولید کننده (provider) و مصرف کننده (Consumer) است.

Imports Microsoft.VisualBasic

Public Interface ISelectedDate

ReadOnly Property SelectedDate() As Date

End Interface

3- یک Web user Control جدید به صفحه اضافه کنید. برای این کاربر روی نام پروژه (*C:\ASPNET20\chap03-Webparts*) در Solution Explorer راست کلیک کنید و گزینه های Add New Item... و Web User Control را انتخاب نمایید.

یادداشت: ارتباط بین web part ها بر مبنای یک مدل ارتباطی میان تولید کننده و مصرف کننده استوار است. برای اینکه مصرف کننده اطلاعات را به درستی از فرستنده دریافت کند، تولید کننده بایستی از طریق قراردادی (کلاس *ISelectedDate.vb* در این مثال) نوع داده ارسالی را مشخص کند. در این صورت مصرف کننده می تواند از داده های ارسالی استفاده نماید.

4- نام Web User control اضافه شده را *CalendarUC.ascx* قرار دهید.

5- یک کنترل Calendar به *CalendarUC.ascx* اضافه کنید و شمای Simple را به آن اعمال کنید. (از طریق لینک Auto Format... در Calendar Tasks) اکنون *CalendarUC.ascx* مانند شکل ۳-۳۷ شده است.



شکل ۳-۳۷ . CalendarUC.ascx

6- بر روی *CalendarUC.ascx* کلیک مضاعف کنید و کد زیر را در آن بنویسید. این کد web user control را به عنوان سرویس دهنده پیکر بندی می کند :

```
Partial Class CalendarUC_ascx
    Inherits System.Web.UI.UserControl
    Implements ISelectedDate

    Public ReadOnly Property SelectedDate( ) As Date _
        Implements ISelectedDate.SelectedDate
        Get
            Return Calendar1.SelectedDate.Date
        End Get
    End Property

    <ConnectionProvider("SelectedDate", "SelectedDate")> _
    Public Function GetSelectedDate( ) As ISelectedDate
        Return Me
    End Function
End Class
```

7- پارامتر اول ConnectionProvider نامی را برای نقطه اتصال سرویس دهنده در نظر می گیرد. و پارامتر دوم یک ID یکتا برای آن است.

8- اساساً از طریق ConnectionProvider به رابط ISelectedDate دسترسی دارید و از طریق آن میتوانید نقطه اتصال را ایجاد کنید. همچنین از خاصیت SelectedDate جهت بازگرداندن تاریخ انتخاب شده در Calendar استفاده می کنید.

9- قطعه کد زیر را به کنترل google.ascx اضافه کنید :

```
Partial Class Google_ascx
    Inherits System.Web.UI.UserControl

    Protected Sub btnSearch_Click(ByVal sender As Object, _
        ByVal e As System.EventArgs) _
        Handles btnSearch.Click
        Response.Write(Page.IsValid)
        Dim queryStr As String = HttpUtility.UrlEncode(txtSearch.Text)
        Response.Redirect("http://www.google.com/search?q=" & queryStr)
    End Sub

    Private _selectedDate As ISelectedDate
```

```
<ConnectionConsumer("SelectedDate", "SelectedDate")> _
Sub setSearchText(ByVal SearchText As ISelectedDate)
    Me._selectedDate = SearchText
End Sub
```

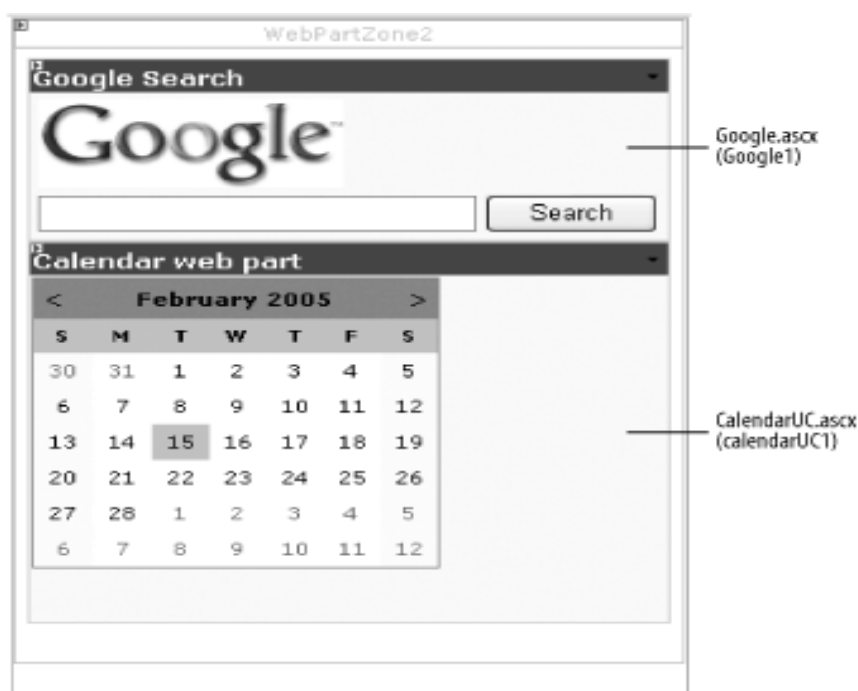
```
Protected Sub Page_PreRender(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.PreRender
    If _selectedDate IsNot Nothing Then
        txtSearch.Text +=
            _selectedDate.SelectedDate.ToShortDateString
    End If
End Sub
End Class
```

10- همانند ConnectionProvider پارامتر اول ConnectionConsumer یک نام و پارامتر دوم یک ID واحد برای اتصال ایجاد می کند.

11- ConnectionConsumer یک نقطه اتصال تعریف می کند که امکان، دریافت اطلاعات از واسط ISelectedDate را فراهم می کند. داده های دریافت شده در رویداد PreRender صفحه google.ascx به Text Box اضافه می شود.

12- کنترل CalendarUC.ascx را به WebPartZone2 صفحه default.aspx اضافه کنید. کنترلها را همانند شکل ۳-۳۸ نامگذاری کنید (google web part در بخش قبل اضافه شد) به کنترل Calendar web user خاصیت Title نیز اضافه کنید:

```
<ZoneTemplate>
    <uc1: Google title="Google Search" runat="server"
        ID="Google1" />
    <uc3: CalendarUC title="Calendar Web Part" runat="server"
        ID="CalendarUC1" />
</ZoneTemplate>
```



شکل ۳-۳۸. افزودن web Part های سرویس دهنده و سرویس گیرنده به صفحه

- 13- جهت ایجاد ارتباط (Connection) میان سرویس دهنده (Provider) و مصرف کننده (Consumer) کد زیر را به صفحه Default.aspx پس از WebPartManager اضافه کنید:

```
<asp:WebPartManager ID="WebPartManager1" runat="server">
  <StaticConnections>
    <asp:WebPartConnection ID="Connection"
      ProviderID="CalendarUC1"
      ProviderConnectionPointID="SelectedDate"
      ConsumerID="Google1"
      ConsumerConnectionPointID="SelectedDate" />
  </StaticConnections>
</asp:WebPartManager>
```

- 14- کلید F5 را فشار دهید. چنانچه تاریخی را در Calendar انتخاب کنید، آن تاریخ در Text Box مربوط به google web part نشان داده میشود. (شکل ۳-۳۹)



شکل ۳-۳۹. ارتباط میان دو web part

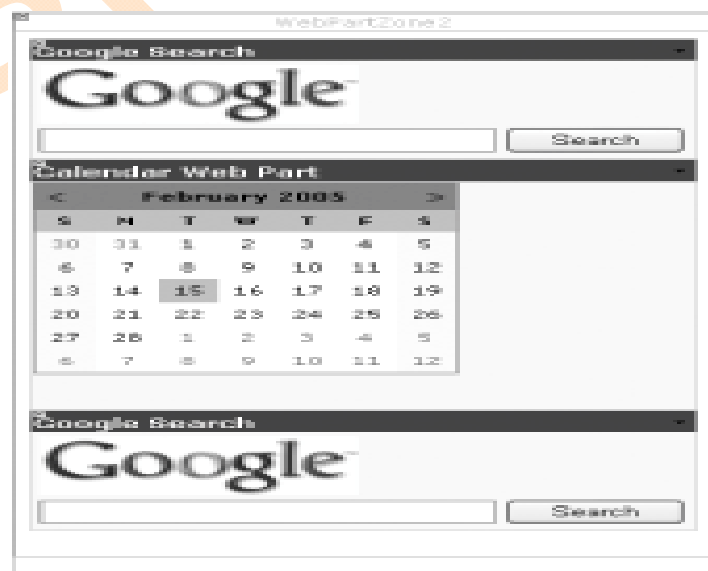
۳-۶-۲. پاروقی بخش...

... برقراری ارتباط میان Web Part ها در زمان اجرای برنامه

تا اینجا شیوه برقراری ارتباط میان دو web part را به صورت ایستا (Static) آموختید. اما فرض کنید در زمان اجرا web part جدیدی به صفحه اضافه کنید، حال چگونه web part جدید را با سایر web part ها ارتباط می دهید؟ در این حالت نیازمند برقراری ارتباط به صورت پویا (Dynamic) هستید؛ در واقع این شما نیستید که ارتباط را ایجاد می کند بلکه کاربر آن را انجام می دهد.

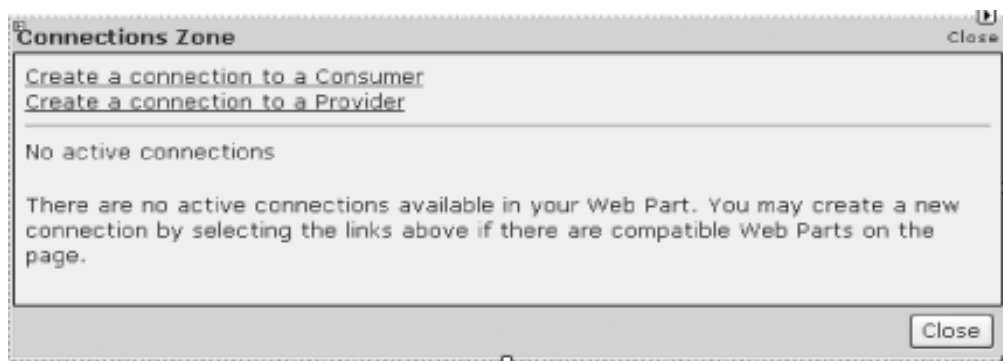
در ادامه خواهید دید که چگونه از طریق کنترل ConnectionsZone می توان دو web part را به یکدیگر متصل کرد.

1- Google.ascx را به WebPartZone2 اضافه کنید. اکنون صفحه شامل دو کنترل google sench web part و یک calendar web part است. (شکل ۳-۴۰)



شکل ۳-۴۰. web part های موجود در WebPartZone2

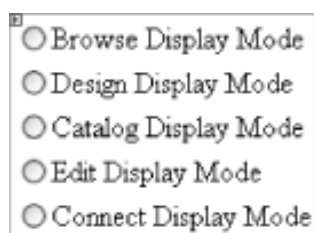
2- یک کنترل ConnectionsZone (این کنترل را در بخش web part از Toolbox می توانید پیدا کنید) به صفحه Default.aspx اضافه کنید. شمای Classic را به این کنترل اعمال کنید. (از طریق لینک Auto format... در ConnectionsZone Tasks) (شکل ۳-۴۱)



شکل ۳-۴۱. اعمال شمای Classic به کنترل ConnectionsZone

3- یک گزینه جدید با نام Connect Display mode به Radiolist Button اضافه کنید (به همان صورت که در بخشهای قبل انجام دادیم) و کد زیر را به رویداد SelectedIndexchanged آن اضافه کنید. (شکل ۳-۴۲) این کد باعث نمایش ConnectionsZone هنگام انتخاب گزینه Connect Display mode می شود:

```
Protected Sub rbIMode_SelectedIndexChanged( _
    ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles rbIMode.SelectedIndexChanged
    Select Case rbIMode.SelectedIndex
        Case 0 : WebPartManager1.DisplayMode = _
            WebPartManager.BrowseDisplayMode
        Case 1 : WebPartManager1.DisplayMode = _
            WebPartManager.DesignDisplayMode
        Case 2 : WebPartManager1.DisplayMode = _
            WebPartManager.CatalogDisplayMode
        Case 3 : WebPartManager1.DisplayMode = _
            WebPartManager.EditDisplayMode
        Case 4 : WebPartManager1.DisplayMode = _
            WebPartManager.ConnectDisplayMode
    End Select
End Sub
```

شکل ۳-۴۲. افزودن گزینه ConnectDisplayMode به کنترل Radio Button list

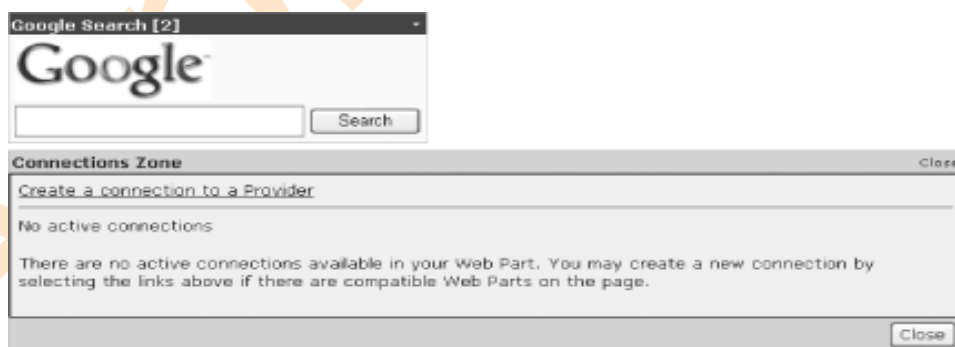
4- دکمه F5 را فشار دهید و Connect Display mode را انتخاب نمایید. همانطور که مشاهده می کنید اتفاق خاصی رخ نمی دهد.

5- گوشه بالای Google Search [2] را کلیک کنید. همانطور که می بینید Connect به آن اضافه شده است. (شکل ۳-۴۳) بر روی این لینک کلیک کنید.



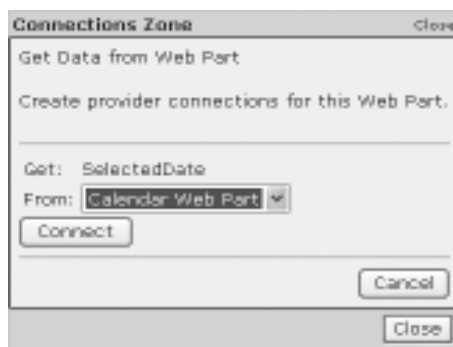
شکل ۳-۴۳. پیکر بندی اتصال به web Part

6- اکنون پنجره Connections Zone را مشاهده می کنید (شکل ۳-۴۴) بر روی لینک "Create a connection to a Provider" کلیک کنید تا Google search را به web part دیگری متصل کنید.



شکل ۳-۴۴ ایجاد اتصال برای یک web part

7- در این قسمت می توانید web part سرویس دهنده را مشخص کنید. در drop-down listbox و جلوی عبارت From: گزینه "Calendar web part" را انتخاب و دکمه Connect را کلیک کنید. (شکل ۳-۴۵)



شکل ۳-۴۵. انتخاب سرویس دهنده

8- چنانچه دقت کنید می بینید که هر دو google sereh web part به Calendar web part متصل شده اند.

9- برای قطع ارتباط دکمه Disconnect را در ConnectionZone کلیک کنید.

۳.۶.۳. منابع اطلاعاتی :

برای آشنایی بیشتر با نحوه عملکرد اتصالات web part ها مقاله زیر را مطالعه کنید :

<http://www.nikhilk.net/CrossPageConnections.aspx>

فصل چهارم

دسترسی به داده ها

(Data Access)

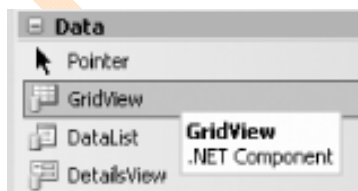
یکی از معمول ترین و مهمترین اعمالی که در برنامه های وب بکار میرود دسترسی به داده هاست . وجود کنترل های متنوع دسترسی به داده در VS 2005 نشان دهنده اهمیت همین موضوع است . آشکارترین این کنترل های ، کنترل جدید GridView است که نسخه بهبود یافته کنترل محبوب Datagrid است . (اگرچه از Datagrid همچنان در Asp.Net 2.0 پیشانی می شود) بعلاوه Asp.Net 2.0 کنترل های Data source جدیدی را به همراه آورده است که امکان استفاده از انواع مختلف داده را برایتان فراهم آورده اند .

۱-۴- نمایش اطلاعات درون جدول

چنانچه برنامه نویس Asp.Net باشید حتماً از کنترل Datagrid استفاده کرده اید Datagrid کنترل قدرتمند و انعطاف پذیری جهت نمایش اطلاعات از منابع اطلاعاتی (Data source) همچون پایگاه داده (Data Base) می باشد . با وجود این برای انجام برخی اعمال از جمله ویرایش سطرهای Datagrid و یا یک مرتب سازی ساده ستونی نیازمند نوشتن کد هستید تا به یک Datagrid سفارشی دست یابید .

به همین دلیل کنترل GridView جایگزین Datagrid شده است . به جای نوشتن چندین خط کد برای انجام اعمال ساده و معمولی در Datagrid کنترل GridView همین اعمال را با کمترین کد نویسی انجام می دهد .

کنترل GridView را در قسمت Data از Toolbox می توانید پیدا کنید . (شکل ۱-۴)



شکل ۱-۴. کنترل GridView

یادداشت : با استفاده از کنترل جدید GridView می توانید نسبت به Datagrid اعمال بیشتری را با کمترین وقت و کد نویسی انجام دهید .

۱-۴-۱-۴- مراحل انجام کار :

برای آشنایی بیشتر با کنترل GridView صفحه ای شامل یک drop-down listbox و یک GridView طراحی می کنیم . از پایگاه داده pubs (این مثال در SQL Server 2000 وجود دارد) به همراه SQL Server 2005 Express استفاده می کنیم تا لیستی از نواحی محل زندگی مؤلفین را نمایش دهیم . هنگامی که کاربر ناحیه را انتخاب کرد ، نام تمامی مؤلفینی که در آن ناحیه زندگی می کنند نمایش داده می شود .

یادداشت : به کمک GridView می توانید رکوردهایتان را بر اساس نام فیلدها مرتب کنید و یا اینکه رکوردها را ویرایش و حذف نمایید بدون اینکه مجبور به نوشتن کد باشید .

نکته : SQL Server 2005 Express پایگاههای pubs و Northwind را به همراه ندارد .

اما می توانید اسکریپت‌های (scripts) نصب آنها را از آدرس <http://www.microsoft.com/downloads/search.aspx?displaylang=en> دانلود کنید . پس از

نصب اسکریپت‌ها بر روی سیستم به خط فرمان (Command prompt) VS 2005

(Start → Programs → Microsoft Visual Studio 2005 → Visual Studio Tools → Visual Studio 2005 Command Prompt)

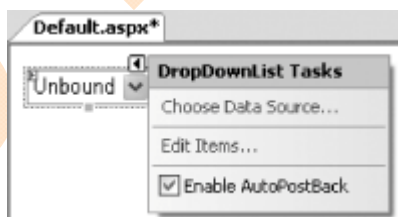
رفته و دستور زیر را برای نصب pubs و Northwind وارد کنید: (با فرض اینکه اسکریپت‌های شما در آدرس C:\ ذخیره شده اند)

C:\>sqlcmd -S .\SQLEXPRESS -i instpubs.sql C:\>sqlcmd -S .\SQLEXPRESS -i instnwnd.sql

1. VS 2005 را باز کنید و پروژه جدیدی با نام C:\ASPNET20\Chap04-GridView ایجاد کنید .

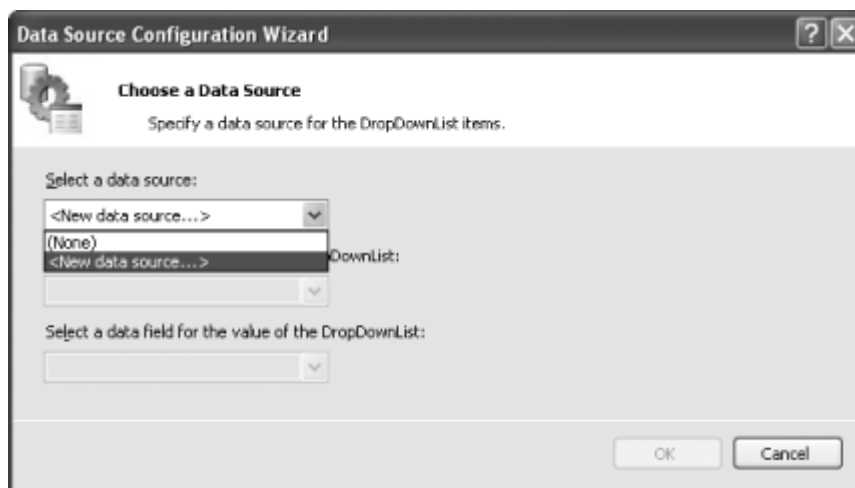
2. یک کنترل Dropdown list از Tool Box به فرم پیش فرض اضافه کنید .

3. در منوی Dropdown list Tasks گزینه Choose Data Source ... را انتخاب کنید . (شکل ۴-۲)



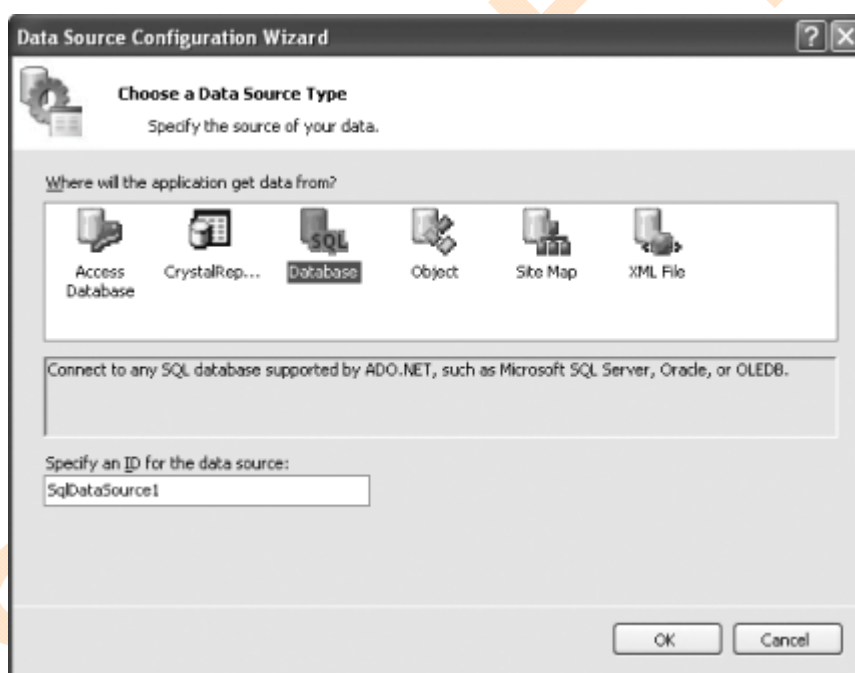
شکل ۴-۲. پیکر بندی کنترل DropDownList

4. در پنجره Data Source Configuration Wizard و در قسمت "Select a data source" گزینه <New Data source> را جهت ایجاد ارتباط جدید انتخاب کنید . (شکل ۴-۳)



شکل ۴-۳. ساخت منبع اطلاعاتی جدید

5. در بخش "Where will the application get data from" گزینه Data Base را انتخاب و OK کنید (شکل ۴-۴) نام پیش فرض **SqlDataSource1** را به عنوان ID منبع اطلاعاتی انتخاب کنید.



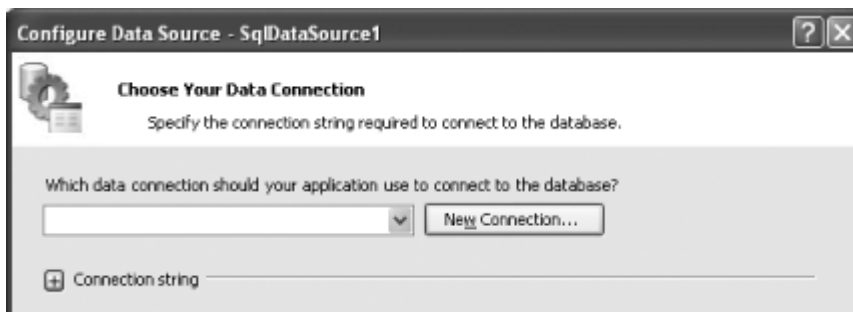
شکل ۴-۴. انتخاب نوع منبع اطلاعاتی

نکته: منبع اطلاعاتی Data Base جهت ارتباط با هر نوع پایگاه داده SQL (از جمله Oracle) به کار می رود و تنها مختص SQL Server نیست.

با استفاده از کنترل **SqlDataSource** بدون نوشتن کد با یک منبع اطلاعاتی از نوع SQL ارتباط برقرار کردید. زمانیکه از **Asp.net 1.x** استفاده می کردید، مجبور بودید از طریق کدهای استاندارد و با استفاده از **ADO.Net** با منبع اطلاعاتی ارتباط برقرار کنید.

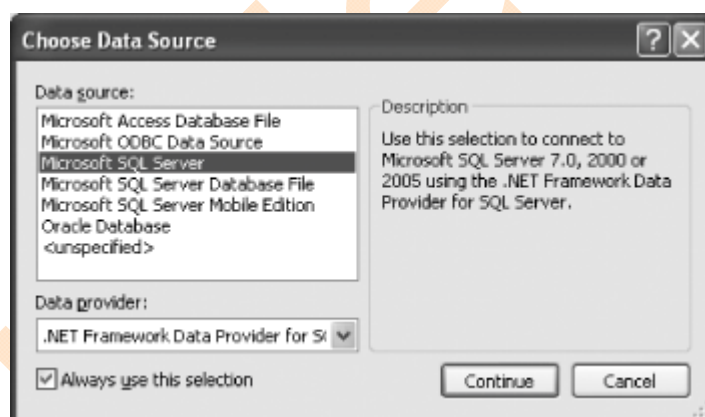
در Asp.Net 2.0 کنترل‌های دسترسی به منابع اطلاعاتی تمامی نیازمندیهای منطقی جهت دسترسی به اطلاعات را در خود نگهداری می‌کنند. (راجع به این موضوع در همین فصل بحث خواهیم کرد)

6. دکمه New Connection ... را کلیک کنید یک ارتباط جدید تعریف نماییم. (شکل ۴-۵)



شکل ۴-۵ ساخت یک ارتباط جدید

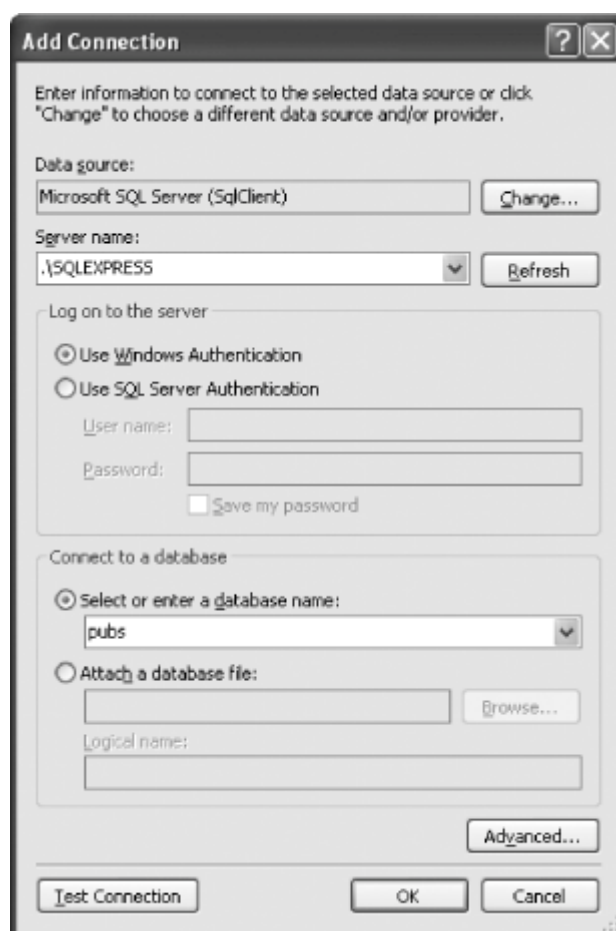
7. در پنجره "choose Data Source" از شما خواسته می‌شود تا نوع منبع اطلاعاتی که می‌خواهید به آن وصل شوید را مشخص کنید. در این قسمت گزینه Microsoft SQL server را انتخاب و دکمه Continue را کلیک کنید. (شکل ۴-۶)



شکل ۴-۶ تعریف نوع ارتباط

8. در این قسمت بایستی اطلاعات مربوط به پایگاه داده ای که می‌خواهید استفاده کنید مشخص نمایید. (شکل ۴-۷ را ببینید) در این مثال از پایگاه داده pubs استفاده خواهیم کرد.

بر روی دکمه Test Connection کلیک کنید تا مطمئن شوید ارتباط به درستی برقرار شده است. سپس ok کنید.



شکل ۴-۷. مشخص کردن اطلاعات مربوط به پایگاه داده و سرور

نکته: چنانچه از SQL Server 2000 استفاده می کنید ، در مرحله ۸ در قسمت **Server name** نام سرور مربوط به SQL Server 2000 را وارد کنید .

چنانچه موقع نصب SQL Server گزینه **SQL server Authentication** را انتخاب کرده اید و به آن **username** و **password** داده اید در قسمت **log on to the server** گزینه **SQL server use Authentication** را انتخاب و **user name** و **password** وارد شده موقع نصب را وارد کنید . در غیر این صورت مانند مثال فوق گزینه **Use Windows Authentication** را انتخاب کنید .

۹. در این قسمت می توانید **Data Connection** انتخاب شده را مشاهده کنید .

(**winxp2\sqlexpress.pubs.dbo**) دکمه **Next** را کلیک کنید . (شکل ۴-۸)



شکل ۴-۸ ارتباط ساخته شده

10. در ادامه از شما سؤال می شود آیا مایلید ارتباط (connection) ساخته شده در فایل پیکربندی پروژه (web.config) ذخیره شود یا خیر؟

انتخاب این گزینه به شما امکان می دهد تنظیمات سرور و پایگاه داده را بدون تغییر کد اصلاح کنید. به همین دلیل پیشنهاد می شود همیشه از این گزینه استفاده کنید. برای ذخیره رشته اتصال در فایل web.config مقدار پیش فرض **pubsConnectionString** را انتخاب و مطمئن شوید چک باکس گزینه "Yes, save this connection as:" تیک خورده باشد. سپس دکمه Next را کلیک کنید. (شکل ۴-۹)

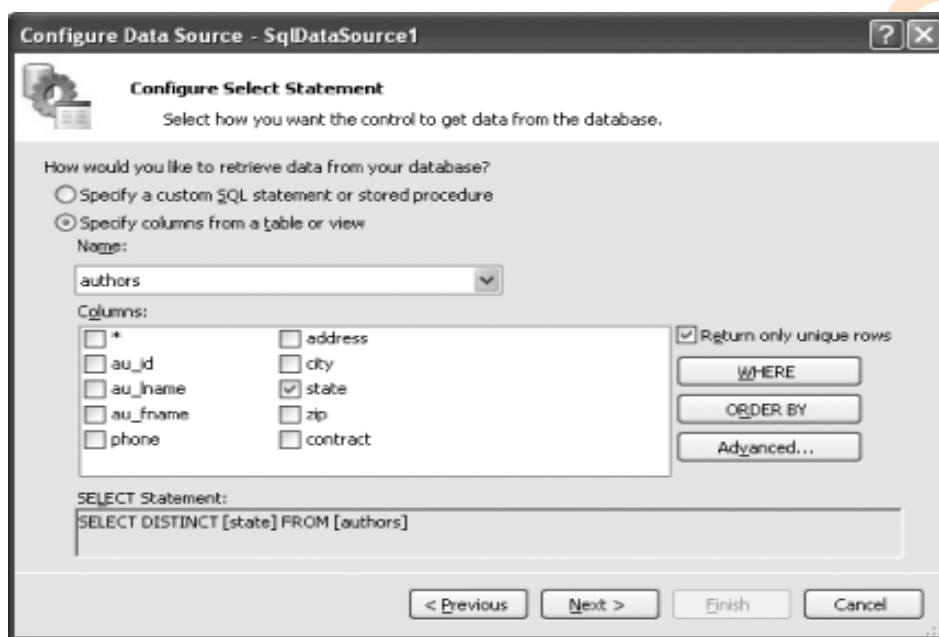
نکته: از طریق ذخیره رشته اتصال (connection string) در فایل web. Config هر گونه تغییر در ارتباط پایگاه داده به آسانی اصلاح می شود.



شکل ۴-۹. ذخیره رشته اتصال در فایل web. Config

11. اکنون می توانید فیلدهای مورد نیازتان را انتخاب نمایید . برای این کار می توانید از یک دستور SQL (و یا یک زیربرنامه ذخیره شده) استفاده کنید و یا اینکه فیلدهای مورد نظرتان را تیک بزنید .

برای این مثال گزینه "Specify columns from a table or view" و جدول Authors را انتخاب کنید . همچنین چک باکس "Return only unique rows" را تیک بزنید . نهایتاً دکمه Next را کلیک کنید . (شکل ۴-۱۰ را ببینید)



شکل ۴-۱۰. پیکربندی جمله انتخاب از پایگاه داده

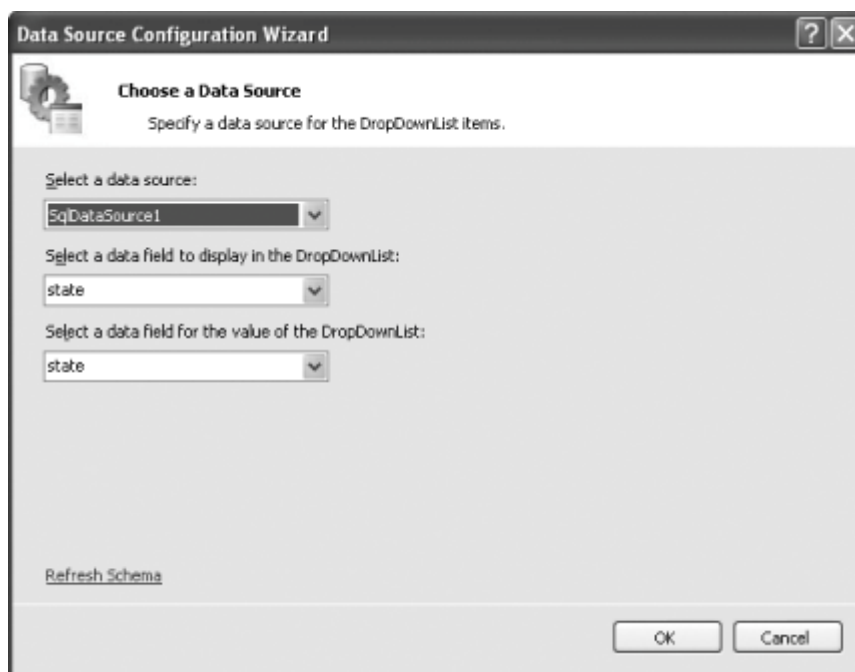
12. در این قسمت می توانید رشته (Query) ایجاد شده را آزمایش کنید . دکمه Test Query را کلیک کنید . نتیجه بازگردانده شده را مشاهده کنید . همانطور که در شکل ۴-۱۱ مشاهده می کنید لیستی از نواحی نمایش داده می شود .



شکل ۴-۱۱. تست رشته تقاضا

13. در نهایت **SqlDataSource1** ساخته شده را به عنوان منبع اطلاعاتی کنترل Drop-down list انتخاب کنید .

همچنین فیلد **state** را جهت نمایش و مقدار (Value) مورد استفاده Drop down list وارد و ok کنید . (شکل ۴-۱۲)



شکل ۴-۱۲- مشخص کردن فیلد مخصوص نمایش و انقیاد (binding)

14. هم اکنون صفحه شما شبیه شکل ۴-۱۳ شده است و یک کنترل SQL Data source به صفحه اضافه شده است .



شکل ۴-۱۳- کنترل SQL Data source

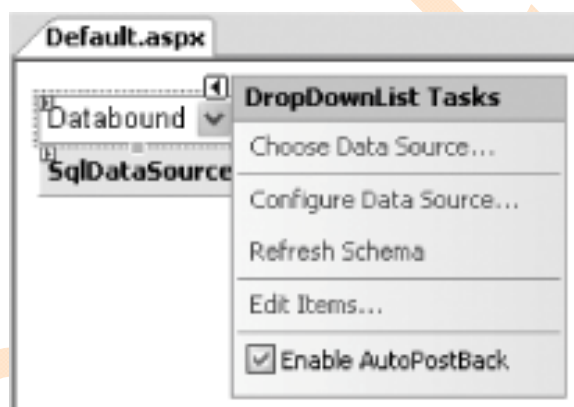
15. کلید F5 را فشار دهید . همان طور که مشاهده می کنید کنترل Dropdown list لیستی از نواحی محل زندگی مؤلفین را ارائه می دهد . (شکل ۴-۱۴)



شکل ۴-۱۴. نمایش نواحی در کنترل Dropdown list

در ادامه بخش یک کنترل GridView به فرم وب اضافه خواهیم کرد تا هنگام انتخاب ناحیه از Dropdown list، تمامی مؤلفینی که در آن ناحیه زندگی می کنند در GridView نمایش داده شوند.

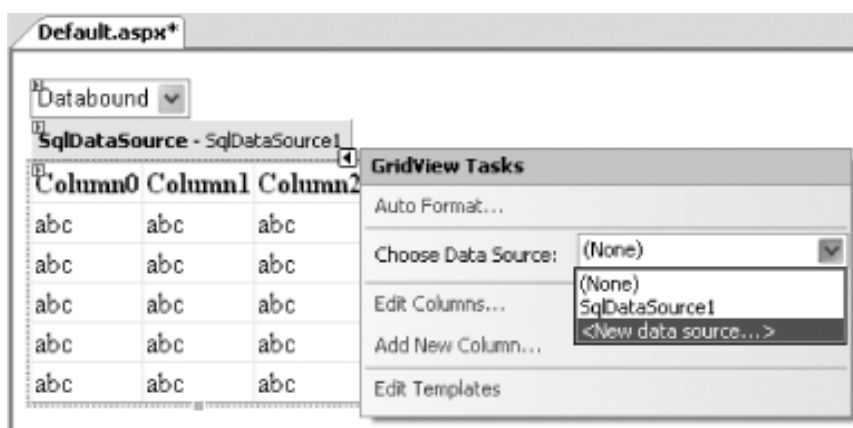
1. ابتدا چک باکس Enable Auto postback کنترل Dropdown list را از طریق منوی Dropdown list Tasks تیک بزنید. (شکل ۴-۱۵) با این کار هنگام انتخاب هر ناحیه در Dropdown list کنترل GridView رکورد مربوطه را نمایش دهد.



شکل ۴-۱۵. تنظیم ارسال postback به سرور هنگام تغییر آیتم های Dropdown list

2. یک کنترل GridView به فرم اضافه کنید. (این کنترل در قسمت Data از Toolbox قرار دارد)

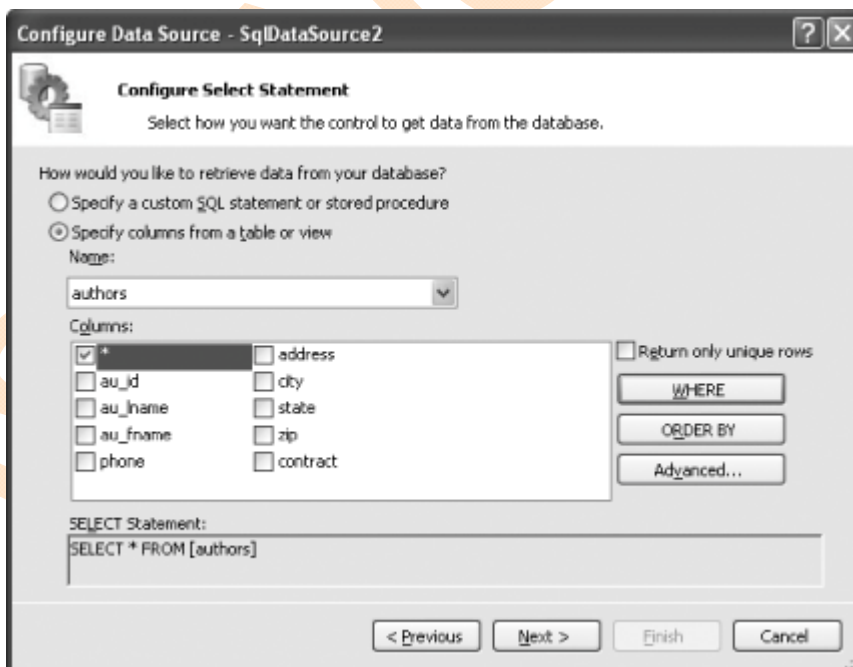
3. از GridView Tasks گزینه '<New Data Source ...>' را انتخاب کنید. (شکل ۴-۱۶) در این قسمت می خواهیم یک کنترل SQL Data source جدید را به گونه ای پیکربندی کنیم که بر اساس آیتم انتخاب شده در Dropdown list، اطلاعات مربوطه از جدول pubs بازایی (retrieve) شوند.



شکل ۴-۱۶- پیکربندی کنترل Grid view

4. در ادامه نوع منبع اطلاعاتی از شما پرسیده می شود. (Database را انتخاب کنید) Data base connection را همانند آنچه برای Dropdown list انجام دادیم تنظیم کنید .

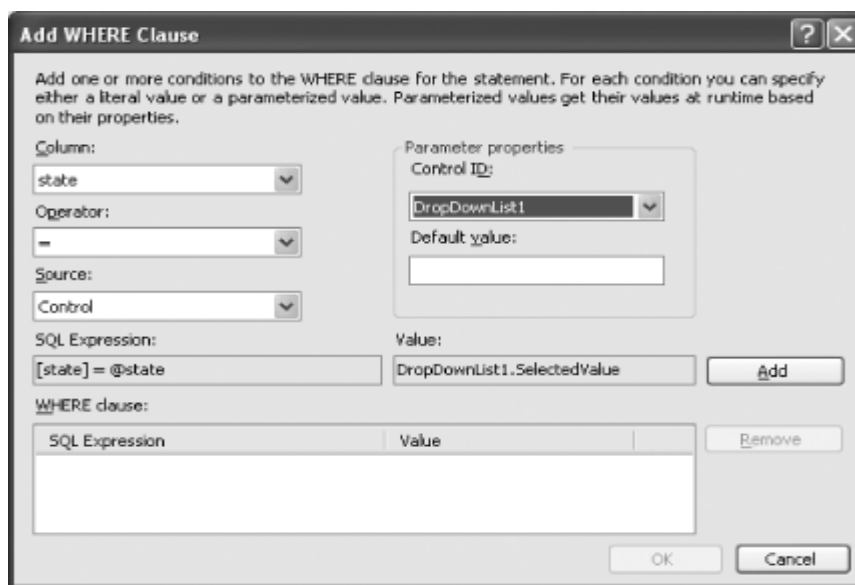
5. بقیه مراحل را همانند dropdown list پیکربندی کنید تا به پنجره شکل ۴-۱۷ برسید. در این پنجره گزینه "Specify columns from a table or view" را انتخاب کنید . چک باکس علامت (*) را تیک بزنید . این گزینه باعث نمایش تمامی فیلدهای جدول Authors می شود. برای نمایش مؤلفین هر ناحیه به یک دستور SQL خاص نیاز دارید بنابراین بر روی دکمه Where کلیک کنید .



شکل ۴-۱۷- تعریف دستور SQL

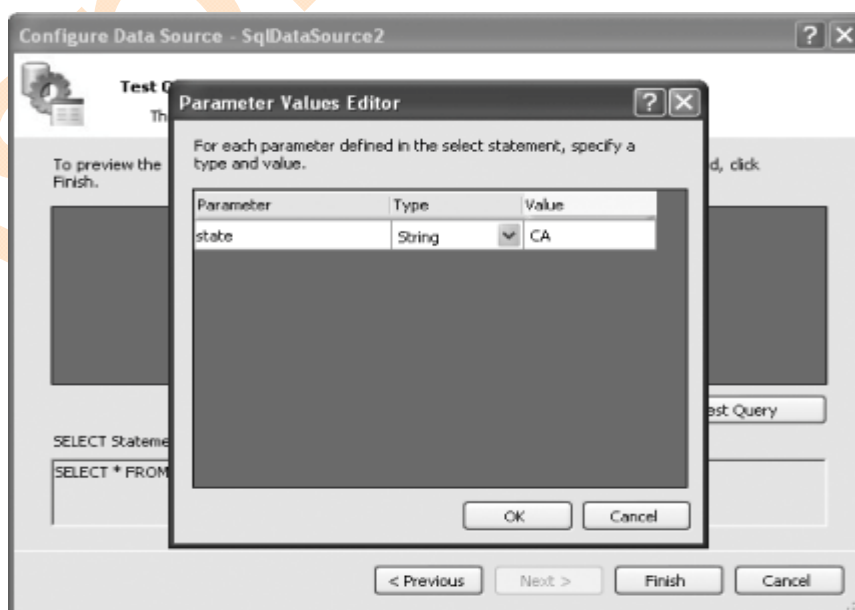
6. در پنجره Add WHERE Clause گزینه های شکل ۴-۱۸ را انتخاب کنید . در واقع مشخص می کنید که فیلد State با گزینه انتخاب شده در Dropdown list برابری کند . دکمه Add را کلیک کنید . همان طور که مشاهده می کنید دستور SQL زیر در کادر "WHERE clause" نمایش داده می شود :

`SELECT * FROM [authors] WHERE ([state] = @state)`



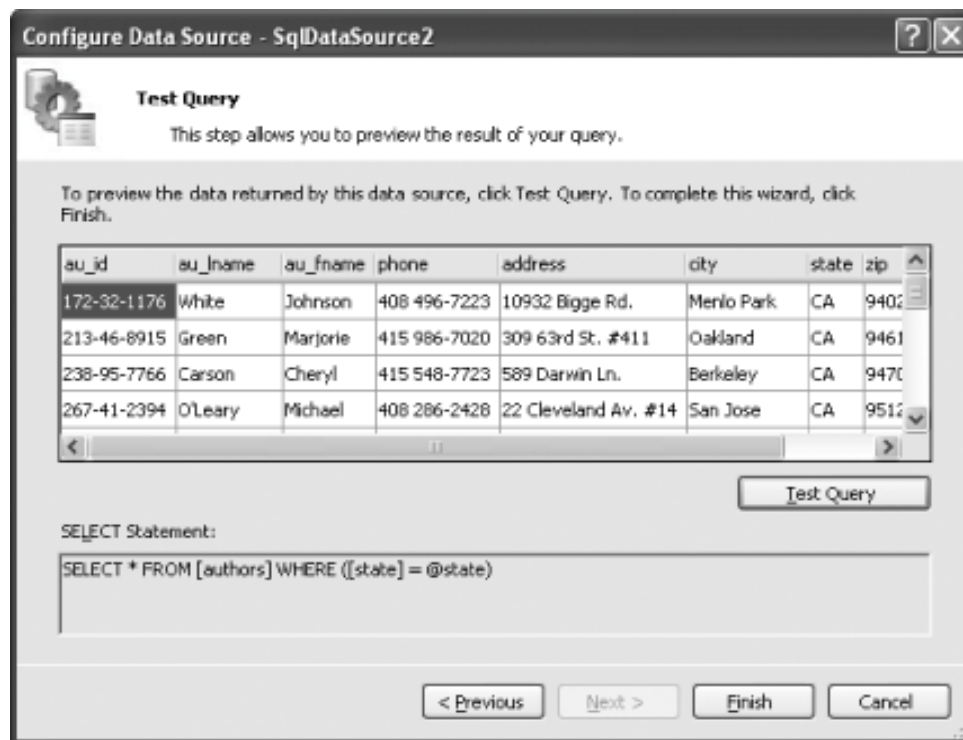
شکل ۴-۱۸- تعریف پارامترهای دستور SQL

7. دکمه OK را در پنجره Add WHERE Clause کلیک کنید . اکنون Test Query را کلیک کنید . در پنجره Parametr Values Editor از شما خواسته می شود مقداری را برای فیلد State مشخص کنید . مقدار CA را وارد کنید تا لیست مؤلفین ساکن کالیفرنیا برایتان نمایش داده شود . (شکل ۴-۱۹)



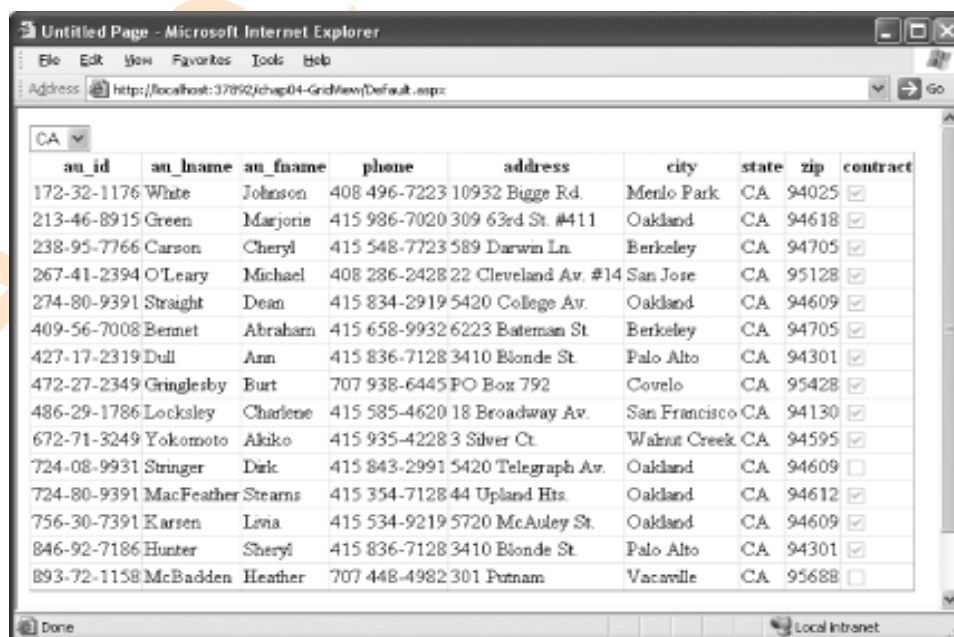
شکل ۴-۱۹- تست رشته تقاضا (query)

8. دکمه Finish را جهت تکمیل پیکربندی کلیک کنید . (شکل ۴-۲۰) . کنترل SqlDataSource دیگری (SqlDataSource2) ایجاد کردیم .



شکل ۴-۲۰- بازایی لیست مؤلفین ساکن کالیفرنیا از پایگاه داده

9. کلید F5 را فشار دهید . هنگامی که ناحیه ای را از طریق Dropdown list انتخاب می کنید. GridView مؤلفین ساکن آن ناحیه را نمایش می دهد.(شکل ۴-۲۱)



شکل ۴-۲۱- کنترل GridView

... پیگر بندی Dropdown list از طریق تعریف

یکی از قابلیت‌های مفید Asp.Net 2.0 این است که تمامی اعمالی را که از طریق Wizardها انجام می دهید می توانید از طریق تعریف در فایل web.config نیز انجام دهید. به تدریج متوجه خواهید شد که پیگر بندی از طریق تعریف چقدر مؤثر و کاراست. مثلاً به جای ساخت رشته اتصال به پایگاه داده از طریق wizard می توانید با استفاده از عنصر < connection string به سادگی آن را به فایل web.config اضافه کنید. به عنوان مثال رشته اتصال pubsConnectionString جهت اتصال به sql server 2005 Express بر روی کامپیوتر محلی (Local Computer) و دسترسی به پایگاه داده pubs به کار می رود :

```
<?xml version="1.0"?>
<configuration
  xmlns="http://schemas.microsoft.com/.NetConfiguration/
v2.0">
  <appSettings/>
  <connectionStrings>
    <add name="pubsConnectionString" connectionString=
      "Data Source=.\SQLEXPRESS; Initial
      Catalog=pubs; Integrated
      Security=True" providerName=
      "System.Data.SqlClient"/>
  </connectionStrings>
  <system.web>
  ...
  ...
  </system.web>
</configuration>
```

نکته: در Asp.Net 1.x نیز امکان ذخیره رشته اتصال در فایل web.config وجود دارد. با این روش تنها یک بار رشته اتصال را در web.config تعریف می کنید و در تمامی صفحاتی که نیازمند اتصال به پایگاه داده هستید، رشته اتصال را از web.config فراخوانی می کنید. مزیت دیگر این روش این است که با تغییر رشته اتصال تنها در web.config آن را اصلاح می کنید نه در تک تک صفحات.

برای ایجاد ارتباط با پایگاه داده pubs می توانید از کنترل SQL Data source استفاده کنید. شما می توانید این کنترل را طوری تنظیم نمایید که از رشته اتصال تعریف شده در web.config استفاده نماید. برای این کار کد زیر را در view فرم اضافه کنید :


```
<asp:SqlDataSource
    ID="SqlDataSource1" runat="server"
    SelectCommand="SELECT DISTINCT [state] FROM [authors]"
    ConnectionString=
        "<%"$ ConnectionStrings:pubsConnectionString %">">
</asp:SqlDataSource>
```

To display the records in the pubs database, bind a DropDownList control to the SqlDataSource control through the DataSourceID attribute:

```
<asp:DropDownList ID="DropDownList1" runat="server"
    DataSourceID="SqlDataSource1"
    DataTextField="state"
    DataValueField="state">
</asp:DropDownList
```

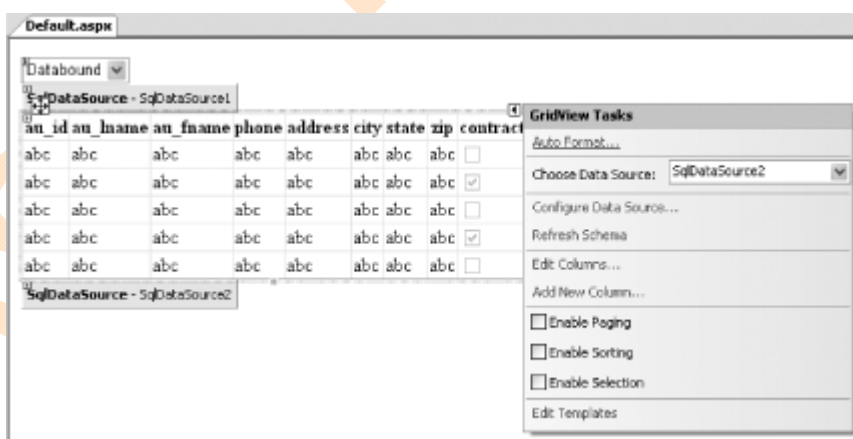
همچنین می توانید از طریق برنامه نویسی SqlDataSource و DropDownList را مقید سازید. (binding) به صورت زیر:

```
DropDownList1.DataSource = SqlDataSource1
```

```
DropDownList1.DataBind( )
```

... تغییر ظاهر Gridview

برای ایجاد تغییر ظاهر Gridview و زیباسازی آن می توانید از شماهای ارائه شده برای آن استفاده کنید. برای اعمال شما بر روی Gridview لینک ... Auto format را از منوی Gridview Tasks کلیک کنید. (شکل ۴-۲۲)



شکل ۴-۲۲- استفاده از قابلیت Auto format کنترل Gridview

اشکال ۴-۲۳ و ۴-۲۴ حاصل اعمال شماهای professional و Sand & sky پروری Gridview هستند.

au_id	au_lname	au_fname	phone	address	city	state	zip	contract
172-32-1176	White	Johnson	408 496-7223	10932 Bigge Rd.	Menlo Park	CA	94025	<input checked="" type="checkbox"/>
213-46-8915	Green	Marjorie	415 986-7020	309 63rd St. #411	Oakland	CA	94618	<input checked="" type="checkbox"/>
238-95-7766	Carson	Cheryl	415 548-7723	589 Darwin Ln.	Berkeley	CA	94705	<input checked="" type="checkbox"/>
267-41-2394	O'Leary	Michael	408 286-2428	22 Cleveland Av. #14	San Jose	CA	95128	<input checked="" type="checkbox"/>
274-80-9391	Straight	Dean	415 834-2919	5420 College Av.	Oakland	CA	94609	<input checked="" type="checkbox"/>
409-56-7008	Bennet	Abraham	415 658-9932	6223 Bateman St.	Berkeley	CA	94705	<input checked="" type="checkbox"/>
427-17-2319	Dull	Ann	415 836-7128	3410 Blonde St.	Palo Alto	CA	94301	<input checked="" type="checkbox"/>
472-27-2349	Gringlesby	Burt	707 938-6445	PO Box 792	Covelo	CA	95428	<input checked="" type="checkbox"/>
486-29-1786	Locksley	Charlene	415 585-4620	18 Broadway Av.	San Francisco	CA	94130	<input checked="" type="checkbox"/>
672-71-3249	Yokomoto	Aluko	415 935-4228	3 Silver Ct.	Walnut Creek	CA	94595	<input checked="" type="checkbox"/>
724-08-9931	Stringer	Dirk	415 843-2991	5420 Telegraph Av.	Oakland	CA	94609	<input type="checkbox"/>
724-80-9391	MacFeather	Stearns	415 354-7128	44 Upland Hts.	Oakland	CA	94612	<input checked="" type="checkbox"/>
756-30-7391	Karsen	Livia	415 534-9219	5720 McAuley St.	Oakland	CA	94609	<input checked="" type="checkbox"/>
846-92-7186	Hunter	Sheryl	415 836-7128	3410 Blonde St.	Palo Alto	CA	94301	<input checked="" type="checkbox"/>
893-72-1158	McBadden	Heather	707 448-4982	301 Putnam	Vacaville	CA	95688	<input type="checkbox"/>

شکل ۴-۲۳- اعمال شمای professional بر روی GridView

au_id	au_lname	au_fname	phone	address	city	state	zip	contract
172-32-1176	White	Johnson	408 496-7223	10932 Bigge Rd.	Menlo Park	CA	94025	<input checked="" type="checkbox"/>
213-46-8915	Green	Marjorie	415 986-7020	309 63rd St. #411	Oakland	CA	94618	<input checked="" type="checkbox"/>
238-95-7766	Carson	Cheryl	415 548-7723	589 Darwin Ln.	Berkeley	CA	94705	<input checked="" type="checkbox"/>
267-41-2394	O'Leary	Michael	408 286-2428	22 Cleveland Av. #14	San Jose	CA	95128	<input checked="" type="checkbox"/>
274-80-9391	Straight	Dean	415 834-2919	5420 College Av.	Oakland	CA	94609	<input checked="" type="checkbox"/>
409-56-7008	Bennet	Abraham	415 658-9932	6223 Bateman St.	Berkeley	CA	94705	<input checked="" type="checkbox"/>
427-17-2319	Dull	Ann	415 836-7128	3410 Blonde St.	Palo Alto	CA	94301	<input checked="" type="checkbox"/>
472-27-2349	Gringlesby	Burt	707 938-6445	PO Box 792	Covelo	CA	95428	<input checked="" type="checkbox"/>
486-29-1786	Locksley	Charlene	415 585-4620	18 Broadway Av.	San Francisco	CA	94130	<input checked="" type="checkbox"/>
672-71-3249	Yokomoto	Aluko	415 935-4228	3 Silver Ct.	Walnut Creek	CA	94595	<input checked="" type="checkbox"/>
724-08-9931	Stringer	Dirk	415 843-2991	5420 Telegraph Av.	Oakland	CA	94609	<input type="checkbox"/>
724-80-9391	MacFeather	Stearns	415 354-7128	44 Upland Hts.	Oakland	CA	94612	<input checked="" type="checkbox"/>
756-30-7391	Karsen	Livia	415 534-9219	5720 McAuley St.	Oakland	CA	94609	<input checked="" type="checkbox"/>
846-92-7186	Hunter	Sheryl	415 836-7128	3410 Blonde St.	Palo Alto	CA	94301	<input checked="" type="checkbox"/>
893-72-1158	McBadden	Heather	707 448-4982	301 Putnam	Vacaville	CA	95688	<input type="checkbox"/>

شکل ۴-۲۴- اعمال شمای Sand & sky بر روی GridView

۴-۱-۳- منابع اطلاعاتی :

در این بخش با قابلیتها ، خصوصیات و نحوه کار با کنترل GridView آشنا شدید برای آشنایی بیشتر با خواص و متدهای این کنترل پیشرفته در MSDN موضوع "GridView Class" را جستجو کنید .

۴-۲- مرتب سازی و نمایش رکوردها در چند صفحه

یادداشت : چنانچه می خواهید اطلاعات GridView را در چند صفحه نمایش دهید ، ادامه بخش را مطالعه کنید .

بطور پیش فرض کنترل GridView تمامی رکوردهای موجود در DataSet خودش را در یک صفحه نمایش می دهد . و چنانچه رکوردهای زیادی برای نمایش داشته باشید ، این شیوه نمایش GridView چندین زیبا به نظر نمی رسد .

به عنوان مثال ممکن است بخواهید مشخصات تمام کارمندان یک شرکت را نمایش دهید ، چنانچه این نامها لیست بلند و بالایی را تشکیل دهند ، بهتر است آنها را در چند صفحه نمایش دهید . همچنین ممکن است بخواهید اسامی لیست شده را بر

اساس حروف الفبا مرتب کنید و یا اینکه به صفحه خاصی از صفحات Gridview بروید . شما می توانید تمامی این اعمال را با انتخاب چند Check Box انجام دهید . بدون اینکه نیازمند نوشتن کد باشید . (بر خلاف آنچه در Asp.Net 1.x وجود داشت)

۴-۲-۱- مراحل انجام کار :

برای مشاهده چگونگی پیکر بندی کنترل Gridview جهت نمایش رکوردها در چند صفحه و مرتب سازی آنها از Gridview بخش قبل استفاده می کنیم .

1. پروژه ایجاد شده در بخش قبل را باز کنید .

2. از منوی Gridview Tasks چک باکسهای زیر را انتخاب کنید . (شکل ۴-۲۵)

– Enable Paging

امکان نمایش داده ها در چند صفحه را فراهم می کند .

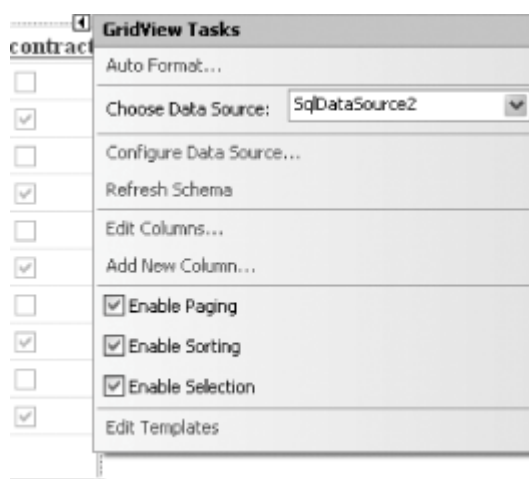
– Enable Sorting

امکان مرتب سازی داده ها بر اساس فیلدها را مقدور می سازد .

– Enable Selection

امکان انتخاب سطرها را فراهم می کند .

نکته : گزینه Enable Selection هنگامی مفید است که می خواهید سطری از Gridview را انتخاب نمائید . به عنوان مثال ممکن است بخواهید با انتخاب یک سطر از Gridview اطلاعات جزئی تری راجع به آن سطر در Gridview دیگری نمایش دهید . هنگامیکه گزینه Enable Selection فعال باشد ، با انتخاب یک سطر از Gridview رویداد Selected Index Changed کنترل Gridview اجرا (Run) می شود .



شکل ۴-۲۵- انتخاب گزینه های paging، sorting و selection برای کنترل Gridview

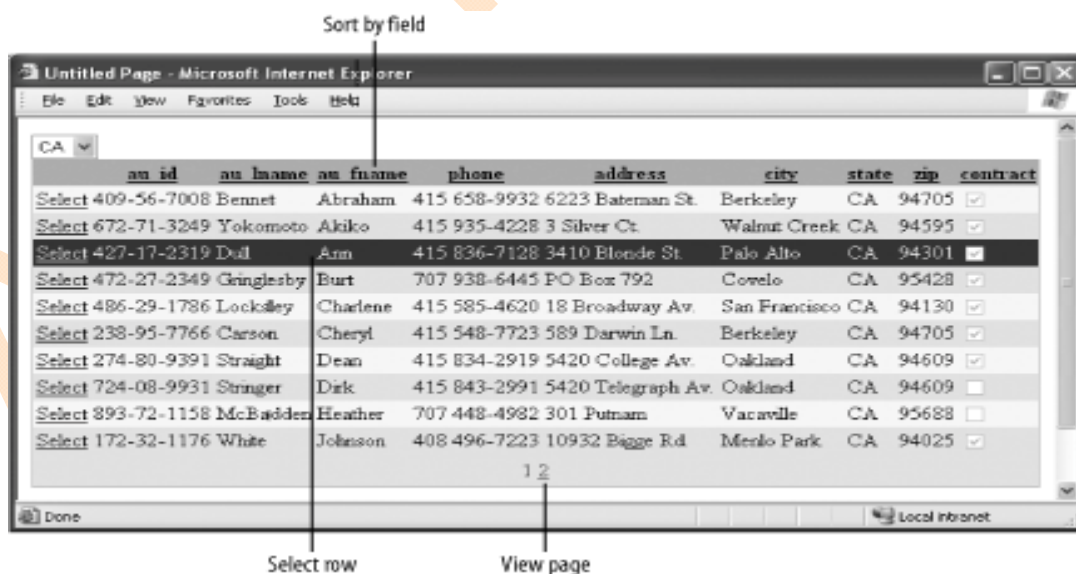
3. هم اکنون کنترل Gridview مشابه شکل ۴-۲۶ شده است .

	au_id	au_lname	au_fname	phone	address	city	state	zip	contract
Select	abc	abc	abc	abc	abc	abc	abc	abc	<input type="checkbox"/>
Select	abc	abc	abc	abc	abc	abc	abc	abc	<input checked="" type="checkbox"/>
Select	abc	abc	abc	abc	abc	abc	abc	abc	<input type="checkbox"/>
Select	abc	abc	abc	abc	abc	abc	abc	abc	<input checked="" type="checkbox"/>
Select	abc	abc	abc	abc	abc	abc	abc	abc	<input type="checkbox"/>
Select	abc	abc	abc	abc	abc	abc	abc	abc	<input checked="" type="checkbox"/>
Select	abc	abc	abc	abc	abc	abc	abc	abc	<input type="checkbox"/>
Select	abc	abc	abc	abc	abc	abc	abc	abc	<input checked="" type="checkbox"/>
Select	abc	abc	abc	abc	abc	abc	abc	abc	<input type="checkbox"/>
Select	abc	abc	abc	abc	abc	abc	abc	abc	<input checked="" type="checkbox"/>

شکل ۴-۲۶- کنترل Gridview پس از اعمال برخی تنظیمات

4. شمای Sand & sky را (از طریق لینک Auto Format... در پنجره Gridview Tasks) به Gridview اعمال کنید تا ظاهر زیبا تری پیدا کند .

5. کلید F5 را فشار دهید . (شکل ۴-۲۷ را ببینید) هم اکنون می توانید (با کلیک بر روی نام فیلد در Gridview) سطرها را بر اساس فیلدها مرتب کنید . یک سطر خاص را انتخاب نمایید (با کلیک بر روی لینک Select) همچنین داده ها در چند صفحه (در این مثال ۲ صفحه) نمایش داده شده اند .



شکل ۴-۲۷- کنترل Gridview با قابلیت مرتب سازی، انتخاب و نمایش چند صفحه ای اطلاعات

نکته : کنترل Gridview می تواند اعمال سمت مشتری (Client – Side) مرتب سازی و صفحه بندی را بدون Refresh صفحه انجام دهد . برای این کار کافیت خاصیت EnableSortingAndPagingCallbacks این کنترل را True کنید . در این حالت از Client Callback manager استفاده می کند . برای کسب اطلاعات بیشتر در مورد Client Callback manager به فصل ۶ مراجعه کنید .

۴-۲-۲- پاورقی بخش ...

... اصلاح UI (User Interface) مربوط به صفحه بندی (paging)

واسط کاربری مربوط به قابلیت صفحه بندی لینکهایی را نمایش می دهد که کاربر از طریق آنها از یک صفحه به صفحه دیگر می رود . چنانچه می خواهید از صفحه بندی در Gridview استفاده کنید می توانید UI مربوط به آن را بر اساس سلیقه خود ویرایش کنید .

خاصیت Mode (از خاصیت PagerSettings) کنترل Gridview امکان نمایش صفحات با قابلیت های متفاوت را فراهم می کند . خاصیت Mode می تواند یکی از مقادیر زیر را داشته باشد :

NextPrevious

نمادهای قبلی (Previous) و بعدی (Next) را نمایش می دهد مانند : < , > و ...

Numeric

صفحات را شماره گذاری می کند مانند : 1 2 3 4 , ...

NextPreviousFirstLast

نمادهای قبلی (Previous) و بعدی (Next) ، آخرین (Last) و اولین (First) را نمایش می دهد . مانند > > < <

NumericFirstLast

شماره صفحات به همراه نمادهای آخرین (Last) و اولین (First) صفحه را نمایش می دهد . مانند > > 456 < <

همچنین برخی خواص pagerstting عبارتند از :

- FirstPageText
- NextPageText
- PageButtonCount
- Position
- PreviousPageText

شکل ۴-۲۸ دو نمونه از صفحات پیکر بندی شده را نشان می دهد .

au_id	au_fname	au_lname	phone	address	city	state	zip	contract
Select 238-95-7766	Carson	Cheryl	415 548-7723	589 Darwin Ln.	Berkeley CA	94705		<input checked="" type="checkbox"/>
Select 267-41-2394	O'Leary	Michael	408 286-2428	22 Cleveland Av. #14	San Jose CA	95128		<input checked="" type="checkbox"/>

au_id	au_fname	au_lname	phone	address	city	state	zip	contract
Select 172-32-1176	White	Johnson	408 496-7223	10932 Bigge Rd.	Menlo Park CA	94025		<input checked="" type="checkbox"/>
Select 213-46-8915	Green	Marjorie	415 986-7020	309 63rd St. #411	Oakland CA	94618		<input checked="" type="checkbox"/>

شکل ۴-۲۸ نمونه ای از تنظیمات اعمال شده بر روی صفحه بندی (paging) در GridView

علاوه بر آنچه گفته شد می توانید از عکس به جای نمایش لینکهای صفحات استفاده کنید . خواص زیر برای این منظور به کار می روند. (شما کافیسست آدرس عکس مورد نظر را در هر خاصیت قرار دهید)

- FirstPageImageUrl
- LastPageImageUrl
- PreviousPageImageUrl
- NextPageImageUrl

۴-۲-۳- منابع اطلاعاتی :

اگر می خواهید راجع به نحوه عملکرد مرتب سازی و صفحه بندی کنترل قدیمی grid Data اطلاعاتی کسب کنید به سایت <http://datagridgirl.com> مراجعه کنید . این سایت همه چیز در مورد Data grid را پوشش می دهد .

۴-۳- ویرایش و حذف رکوردها (Edit & Delete)

یادداشت : می توانید به آسانی به کاربران اجازه ویرایش و حذف رکوردها را بدهید .

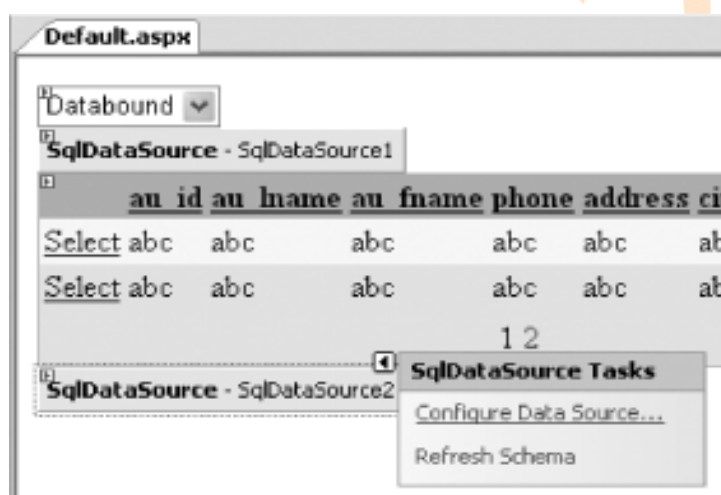
علاوه بر نمایش رکوردها می توانید به کاربران امکان ویرایش و حذف آنها را نیز بدهید . برخلاف کنترل قدیمی Datagrid کنترل GridView ویرایش و حذف را بسیار آسان نموده است . و با استفاده از wizard های پیکر بندی تمامی اعمال GridView را بدون نوشتن کد انجام می دهید .

۴-۳-۱- مراحل انجام کار :

در بخش قبل شاهد انقیاد Gridview به کنترل SQL Data Source بودید . همچنین دیدید که چگونه بدون نوشتن کد رکوردها مرتب و در چند صفحه نمایش داده شدند . در این بخش یک گام جلوتر می رویم و به نحوه پیکربندی Gridview جهت ویرایش و حذف رکوردها می پردازیم .

1- پروژه بخش قبل را باز کنید . در این قسمت کنترل جدیدی با نام SQL Data Source2 جهت ویرایش و حذف رکوردها پیکربندی می کنیم .

2- در پنجره SqlDataSource Tasks از کنترل **SqlDataSource2** بر روی **Configure Data Source...** کلیک کنید . (شکل ۴-۲۹)

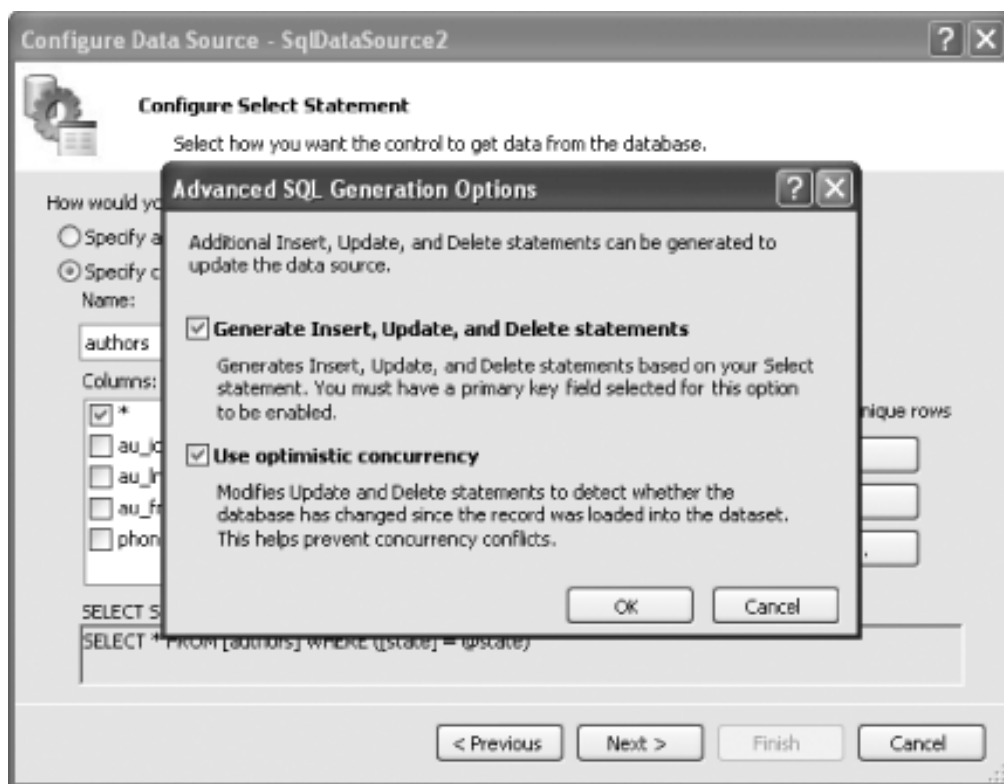


شکل ۴-۲۹- پیکربندی کنترل SqlDataSource2

3- در پنجره بعدی دکمه Next را کلیک کنید و سپس در پنجره **Configure Select Statement** دکمه Advance را کلیک کنید .

... AsAdvanced را انتخاب کنید .

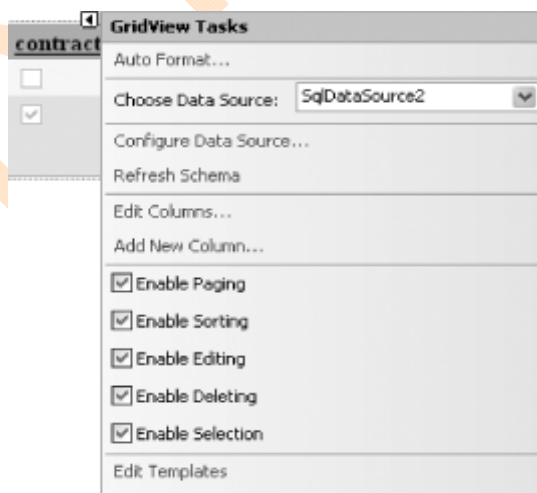
در پنجره ای که نشان داده می شود گزینه "Generate Insert, Update, and Delete statements" را انتخاب کنید. (شکل ۴-۳۰) به منظور جلوگیری از برخوردهای همزمان (concurrency conflicts) (به پاورقی بخش مراجعه کنید) گزینه "Use optimistic concurrency" را نیز تیک بزنید و سپس ok کنید .



شکل ۴-۳۰- تعریف جمله مناسب جهت اعمال تغییرات در جدول

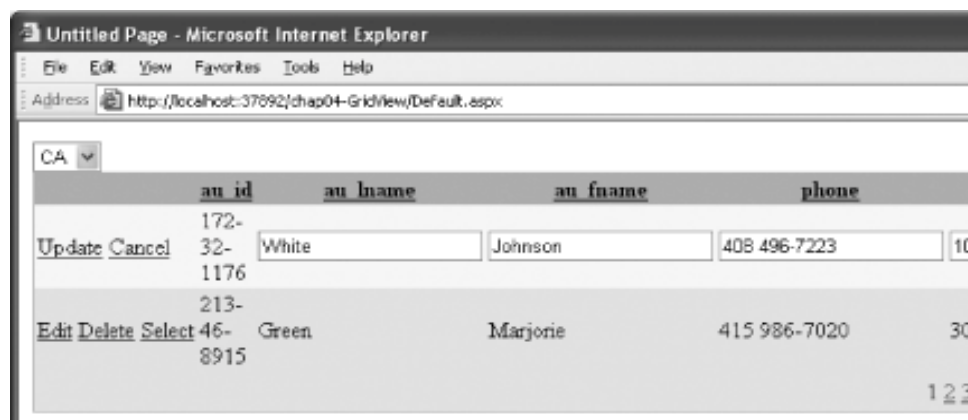
4- به پنجره Configure Select Statement باز گردید و گزینه Next و نهایتاً Finish را کلیک کنید .

5- برای اینکه GridView امکان ویرایش و حذف رکوردها را داشته باشد در پنجره GridView Tasks گزینه های Enable Editing و Enable Deleting را انتخاب نمایید . (شکل ۴-۳۱)



شکل ۴-۳۱- انتخاب گزینه های Enable Editing و Enable Deleting

6- کلید F5 را جهت اجرای برنامه فشار دهید . اکنون از طریق لینکهای Edit و Update میتوانید رکوردها را ویرایش یا حذف کنید . (شکل ۴-۳۲)



شکل ۴-۳۲. ویرایش و حذف رکوردها

۴-۳-۲ پاورقی بخش ...

... جلوگیری از برخوردهای همزمان (Concurrency Conflicts)

شکل ۴-۳۰ را به یاد آورید. در این شکل گزینه "Use optimistic concurrency" را تیک زدیم. با انتخاب این گزینه کنترل SqlDataSource هر گونه تغییر در جدول را قبل از انجام Update شناسایی (Detect) می کند.

حالتی را در نظر بگیرید که دو کاربر به طور همزمان در حال ویرایش چند رکورد هستند هنگامیکه رکوردی توسط یکی از کاربران ویرایش شود، کاربر دیگر مشاهده می کند که مقادیر اصلی دچار تغییر شده اند و هنگامی که سعی در ویرایش رکورد دارد، موفق به انجام این کار نمی شود.

چنانچه به Source View صفحه بروید، متوجه می شوید که کنترل SqlDataSource مقادیر جدیدی را به همراه دارد. در واقع دستورات Insert Command، Delete Command و Update Command ویرایش، درج (Insert) و حذف (Delete) رکوردها در پایگاه داده را کنترل می کنند.

قالب (syntax) این دستورات در مثال ۴-۱ آورده شده است.

```
DeleteCommand="DELETE FROM [authors]
WHERE [au_id] = @original_au_id
AND [au_lname] = @original_au_lname
AND [au_fname] = @original_au_fname
AND [phone] = @original_phone
AND [address] = @original_address
AND [city] = @original_city
AND [state] = @original_state
AND [zip] = @original_zip
AND [contract] = @original_contract"
```

```
InsertCommand="INSERT INTO [authors] ([au_id], [au_lname],
    [au_fname], [phone], [address], [city],
    [state], [zip], [contract]) VALUES (@au_id,
    @au_lname, @au_fname, @phone, @address,
    @city, @state, @zip, @contract)"
```

```
UpdateCommand="UPDATE [authors] SET [au_lname] = @au_lname,
    [au_fname] = @au_fname, [phone] = @phone,
    [address] = @address, [city] = @city,
    [state] = @state, [zip] = @zip,
    [contract] = @contract
WHERE
    [au_id] = @original_au_id
    AND [au_lname] = @original_au_lname
    AND [au_fname] = @original_au_fname
    AND [phone] = @original_phone
    AND [address] = @original_address
    AND [city] = @original_city
    AND [state] = @original_state
    AND [zip] = @original_zip
    AND [contract] = @original_contract"
```

```
ConflictDetection="CompareAllValues"
```

مثال ۴-۱ قالب دستورات Insert Cammand ، Delete Cammand و Update Cammand

ساختار دستورات SQL به گونه ای است که قادر به تشخیص اعمال همزمان هستند . به عنوان مثال دستور UpdateCammand را در نظر بگیرید . این دستور زمانی موفق عمل می کند که مقادیر با فیلدهای اصلی یکسان باشند .

(این حالت از طریق پارامتر @original_fieldname چک می شود)

برای استفاده از استراتژی "First – One Wins" می بایست خاصیت ConflictDetection را CompareAllValues قرار دهید .

یادداشت : استراتژی ای که باعث می شود اولین کاربر موفق به انجام عمل Update شود را "First – One Wins" می گویند .

عنصر <updateParameters> مقادیر اولیه هر فیلد را نگهداری می کند :

<UpdateParameters>

<asp:Parameter Type="String" Name="au_lname" />

<asp:Parameter Type="String" Name="au_fname" />

```
<asp:Parameter Type="String" Name="phone" />
<asp:Parameter Type="String" Name="address" />
<asp:Parameter Type="String" Name="city" />
<asp:Parameter Type="String" Name="state" />
<asp:Parameter Type="String" Name="zip" />
<asp:Parameter Type="Boolean" Name="contract" />
<asp:Parameter Type="String" Name="original_au_id" />
<asp:Parameter Type="String" Name="original_au_lname" />
<asp:Parameter Type="String" Name="original_au_fname" />
<asp:Parameter Type="String" Name="original_phone" />
<asp:Parameter Type="String" Name="original_address" />
<asp:Parameter Type="String" Name="original_city" />
<asp:Parameter Type="String" Name="original_state" />
<asp:Parameter Type="String" Name="original_zip" />
<asp:Parameter Type="Boolean" Name="original_contract" />
</UpdateParameters>
```

دقت کنید ؛ مقدار اولیه هر پارامتر با رشته "original_" شروع شده است . این مقدار را می توانید با استفاده از خاصیت OldValuesParameterFormatString از کنترل SqlDataReader تغییر دهید .

... کنترل خطا حین به روزرسانی (updating) رکوردها

چنانچه هنگام به روزرسانی رکوردها در GridView خطایی رخ دهد ، این خطا در رویداد RowUpdated نگهداری می شود ، همانند زیر :

```
Protected Sub GridView1_RowUpdated(
    ByVal sender As Object, _
    ByVal e As System.Web.UI.WebControls. _
        GridViewUpdatedEventArgs) _
    Handles GridView1.RowUpdated
    If e.Exception IsNot Nothing Then
        Response.Write("Error in updating record.")
        e.ExceptionHandled = True
    End If
End Sub
```

۴-۳-۳- منابع اطلاعاتی :

برای کسب اطلاعات بیشتر در مورد همزمانیها مقاله زیر را مطالعه کنید .

<http://www.15seconds.com/issue/030604.htm>

چنانچه می خواهید بدانید قبل از حذف یک رکورد چه اتفاقی در GridView رخ می دهد ، در MSDN موضوع "GridView.RowDeleting Event" را جستجو کنید .

۴-۴- نمایش یک رکورد

یادداشت : از کنترل Details View جهت نمایش یک رکورد می توانید استفاده کنید.

علاوه بر GridView کنترل های جدید دیگری جهت نمایش رکوردها وجود دارند . به جای نمایش چندین رکورد در یک صفحه ، کنترل details view یک رکورد را نمایش می دهد . این کنترل نیز در بخش Data از Toolbox قرار دارد . (شکل ۴-۳۳)



شکل ۴-۳۳- کنترل Details View

۴-۴-۱- مراحل انجام کار :

برای اینکه با تفاوت های میان GridView و Details View بیشتر آشنا شوید کنترل GridView برنامه بخش قبل را با یک کنترل Details View عوض می کنیم . همچنین خواهید دید که چگونه کنترل Details View امکان افزودن رکورد به پایگاه داده را فراهم می آورد در حالیکه در GridView این امکان را نداشتید .

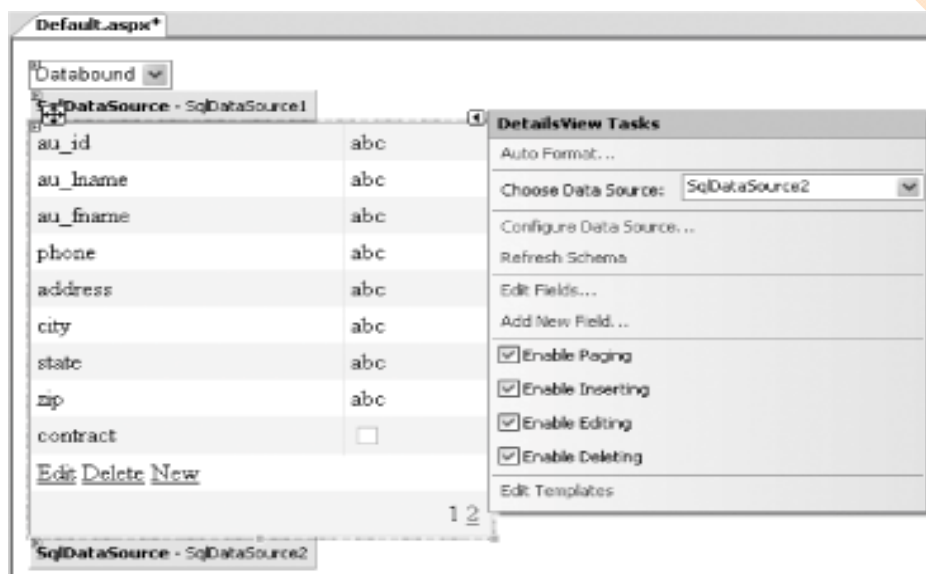
1. پروژه بخش را باز کنید و یک کنترل Details View را جایگزین کنترل GridView نمایید .

2. شما Mocha را بر روی آن اعمال کنید (از طریق لینک Auto format... در پنجره DetailsView Tasks)

3. در منوی DetailsView Tasks کنترل **SqlDataSource2** را به عنوان Data Source انتخاب کنید و گزینه های زیر را تیک بزنید . (شکل ۴-۳۴)

- Enable Paging
- Enable Inserting
- Enable Editing
- Enable Deleting

4. وجود این گزینه ها به نحوه پیکر بندی Data Source بستگی دارد . به عنوان مثال ، گزینه های Enable Editing و Enable Deleting زمانی وجود دارند که Data Source ای که از آن استفاده می کنید آنها را پشتیبانی کند . همچنین اگر Data Source را به گونه ای تنظیم کنید که از درج (Insert) پشتیبانی کند ، آنگاه چک باکس Enable Inserting نیز وجود خواهد داشت .



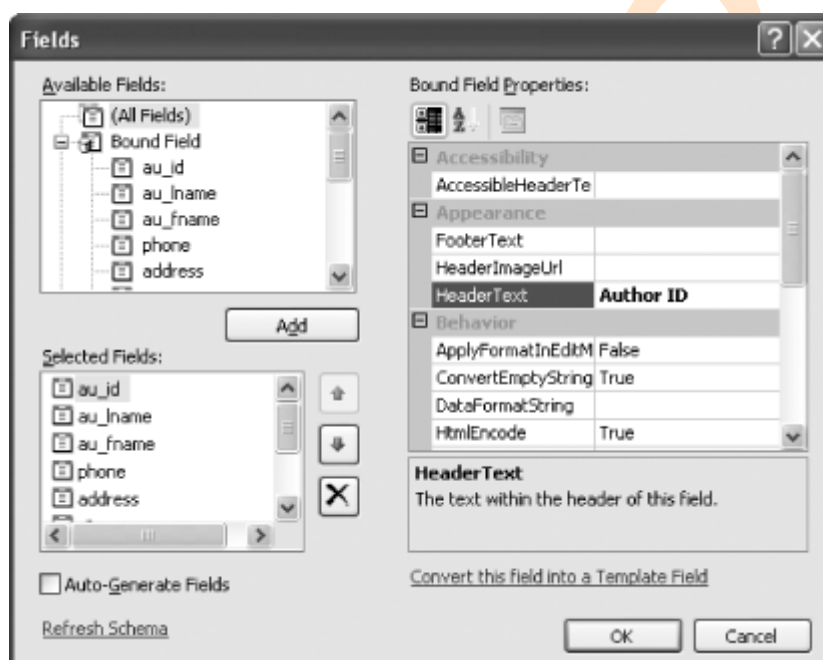
شکل ۴-۳۴- پیکر بندی کنترل برای paging و Inserting و Deleting

5. کلید F5 را فشار دهید . شکل ۴-۳۵ خروجی Details View را نمایش می دهد همچنین در این کنترل می توانید رکورد را ویرایش و یا رکورد جدیدی اضافه کنید .



شکل ۴-۳۵. نمایش رکورد در Details View

نکته: بطور پیش فرض کنترل‌های Gridview و Details View از نام فیلدهای جدول به عنوان متن هدر (Header Text) استفاده می‌کنند. به عنوان مثال Au_id و Au_fname به عنوان هدر در کنترل‌های Gridview و Details View در این بخش و بخش قبل نمایش داده شدند. برای سفارشی (Customize) کردن متن هدر کنترل Details View بر روی لینک Edit Fields... در پنجره Details View Tasks کلیک کنید سپس با استفاده از خاصیت Header Text در پنجره Fileds، هدر فیلدهای مورد نظر را ویرایش کنید. (شکل ۴-۳۶) جهت تغییر متن هدر کنترل Gridview بر روی Edit Columns... در منوی Gridview Tasks کلیک کنید و همانند Details View با استفاده از خاصیت Header Text درون پنجره Fileds، هدر فیلدهای مورد نظر را ویرایش کنید.



شکل ۴-۳۶- سفارشی کردن Header Text کنترل Details View

۴-۴-۲- پاورقی بخش ...

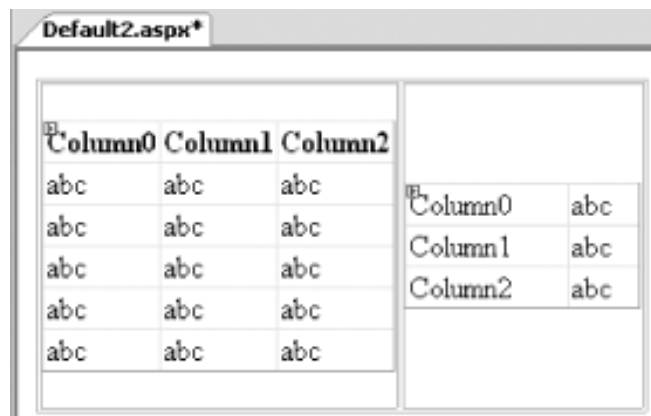
... طراحی ساختار کل به جزء (Master / Details) از رکوردها

با استفاده از ترکیب Gridview و Details View می‌توانید رکوردهای جدول را به شکل جالب تری نمایش دهید. به عنوان مثال می‌توانید از طریق Gridview تمامی ID های کارمندان را نمایش دهید و با کلیک بر روی هر ID مشخصات مربوط به آن ID را در یک Details View به نمایش درآورید.

در ادامه به بررسی این روش می‌پردازیم.

1. فرم جدیدی با نام Default2.aspx به پروژه اضافه کنید .

2. یک جدول با یک سطر و دو ستون به فرم اضافه کنید. (Layout → Insert Table). یک GridView به ستون سمت چپ و یک Details View به ستون سمت راست اضافه کنید. (شکل ۴-۳۷ را ببینید)



شکل ۴-۳۷- افزودن GridView و Details View به فرم

3. کنترل GridView را طوری پیکر بندی کنید تا از طریق یک کنترل جدید SqlDataSource فیلد au_id را از جدول authors در پایگاه داده pubs نمایش دهد. (نحوه انجام کار در بخش ۱-۴ توضیح داده شد) Source View مربوط به GridView و کنترل SqlDataSource جدید مشابه مثال ۴-۲ می باشد .

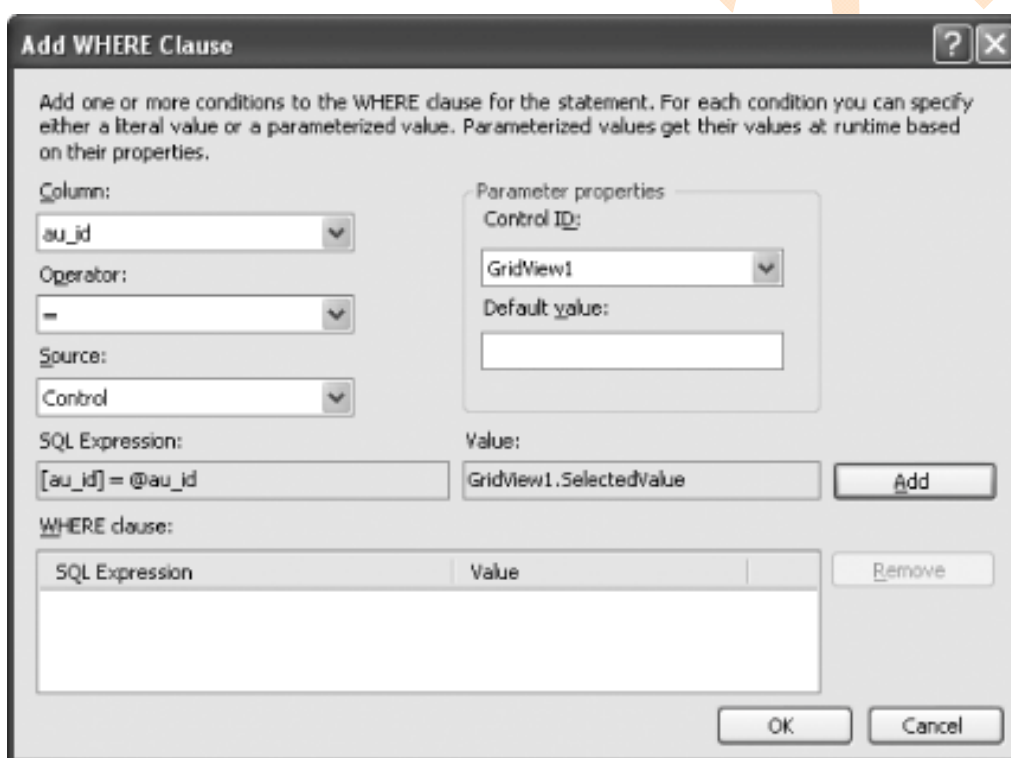
نکته : توصیه می شود این مثال را از آدرس <http://www.oreilly.com/catalog/aspnetadn/> دانلود کنید .

```
<asp:GridView ID="GridView1" runat="server"
    Width="158px"
    DataSourceID="SqlDataSource1"
    AutoGenerateColumns="False"
    DataKeyNames="au_id"
    AllowPaging="True"
    PageSize="4"
    AllowSorting="True">
    <Columns>
        <asp:CommandField ShowSelectButton="True">
        </asp:CommandField>
        <asp:BoundField ReadOnly="True"
            HeaderText="au_id"
            DataField="au_id"
            SortExpression="au_id">
        </asp:BoundField>
    </Columns>
</asp:GridView>
```

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    SelectCommand="SELECT [au_id] FROM [authors]"
    ConnectionString="<%%$ ConnectionStrings:
        pubsConnectionString %>">
</asp:SqlDataSource>
```

مثال ۴-۲. سورس مربوط به GridView و SqlDataSource

4. کنترل Details View را جهت نمایش تمامی فیلدهای جدول Authors از پایگاه داده pubs توسط یک کنترل جدید Sql Data Source پیکر بندی کنید. در واقع فیلد au_id به رکورد انتخاب شده در GridView وابسته است. (شکل ۴-۳۸)



شکل ۴-۳۸. انتخاب یک رکورد در کنترل GridView

5. Source View کنترل‌های Details View و Sql Data Source دوم همانند مثال ۴-۳ خواهد بود.

```
<asp:DetailsView ID="DetailsView1" runat="server"
    Height="50px" Width="286px"
    DataSourceID="SqlDataSource2"
    AutoGenerateRows="False"
    DataKeyNames="au_id">
<Fields>
    <asp:BoundField ReadOnly="True" HeaderText="au_id"
```



```

        DataField="au_id" SortExpression="au_id">
</asp:BoundField>
<asp:BoundField HeaderText="au_lname" DataField="au_lname"
    SortExpression="au_lname">
</asp:BoundField>
<asp:BoundField HeaderText="au_fname" DataField="au_fname"
    SortExpression="au_fname">
</asp:BoundField>
<asp:BoundField HeaderText="phone" DataField="phone"
    SortExpression="phone">
</asp:BoundField>
<asp:BoundField HeaderText="address" DataField="address"
    SortExpression="address">
</asp:BoundField>
<asp:BoundField HeaderText="city" DataField="city"
    SortExpression="city">
</asp:BoundField>
<asp:BoundField HeaderText="state" DataField="state"
    SortExpression="state">
</asp:BoundField>
<asp:BoundField HeaderText="zip" DataField="zip"
    SortExpression="zip">
</asp:BoundField>
<asp:CheckBoxField HeaderText="contract"
    SortExpression="contract" DataField="contract">
</asp:CheckBoxField>
</Fields>
</asp:DetailsView>

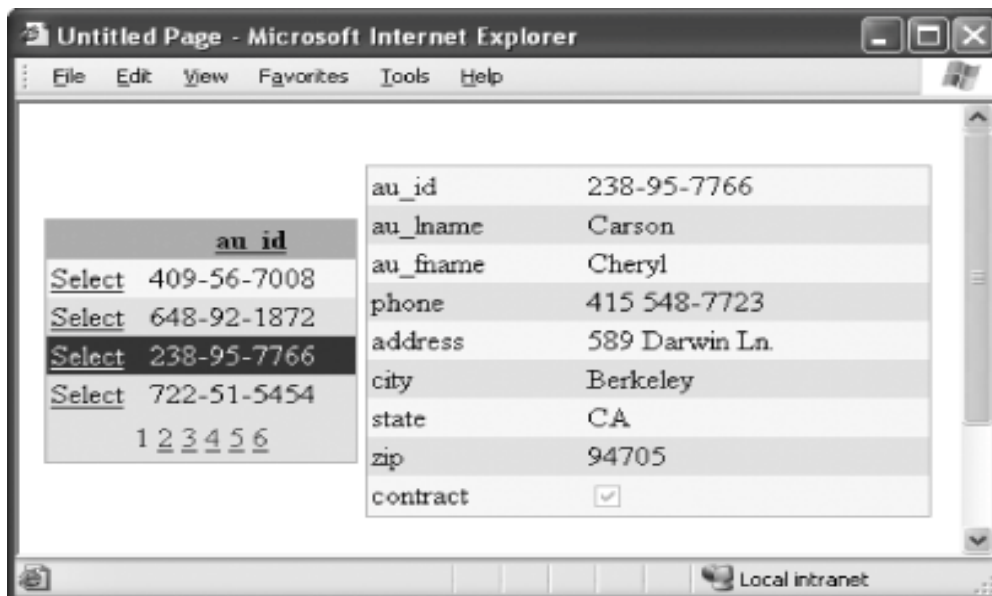
<asp:SqlDataSource ID="SqlDataSource2" runat="server"
    SelectCommand="SELECT * FROM
    [authors] WHERE ([au_id] = @au_id)"
    ConnectionString="<%%$ ConnectionStrings:
    pubsConnectionString %>">
<SelectParameters>
    <asp:ControlParameter Name="au_id" ControlID="GridView1"
        PropertyName="SelectedValue"
        Type="String" />
</SelectParameters>
</asp:SqlDataSource>

```

مثال ۴-۳. Source View مربوط به Details View و Sql Data Source

6. شمای Sand & sky را بر روی هر دو کنترل (Gridview و Details View) اعمال کنید .

7. کلید F5 را فشار دهید . اکنون با انتخاب هر رکورد در Gridview جزئیات آن در Details View نمایش داده می شوند . (شکل ۴-۳۹ را ببینید)



شکل ۴-۳۹- جزئیات رکورد انتخاب شده

۴-۳-۴- منابع اطلاعاتی :

یکی از اعمال مخصوص کنترل‌های Gridview و Details View امکان حذف رکوردهاست . حال چنانچه کاربر اشتباهاً اقدام به حذف رکوردی بکند ، کنترل‌های Gridview و Details View بدون هشدار (Warning) به کاربر آن رکورد را حذف می کنند . اما بهتر است قبل از حذف رکورد از کاربر پرسیده شود که آیا واقعاً قصد پاک کردن رکورد را دارد یا خیر ! این حالت را می توانید از طریق اسکریپت‌های سمت مشتری (Client Side Scripts) چک کنید . برای آگاهی از شیوه نوشتن این نوع اسکریپت‌ها به آدرسهای زیر مراجعه کنید :

- <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndive/html/data01102002.asp>
- <http://www.dotnetbips.com/displayarticle.aspx?id=108>
- <http://www.dotnetjunkies.com/Article/17F2328F-ECB8-478F-A393-3D0E0F024ECO.dcik>

برای کسب اطلاعات بیشتر در مورد کنترل Details View نیز می توانید در MSDN موضوع "DetailsView Class" را جستجو کنید .

۴-۵- کش (Cache) کردن Data Source

یادداشت: به منظور بهبود سرعت اجرای برنامه می توانید کنترل Data Source را کش نمایید.

دسترسی به پایگاه داده عملی زمان بر (time-consuming) است. از این رو کم کردن میزان اتصالات به پایگاه داده امری ضروریست. کنترل Sql Data Source (مانند کنترل AccessDataSource) از کش کردن (Caching) داده ها پشتیبانی می کند. کش کردن داده ها در زمانیکه اطلاعات دائم تغییر نمی کنند بسیار مفید است. به عنوان مثال زمانیکه شما می خواهید اطلاعات لیست مربوط به یک محصول را نمایش دهید.

۴-۵-۱- مراحل انجام کار:

برای اینکه با نحوه کش کردن داده ها توسط Data Source ها آشنا شوید. در این بخش برنامه ای طراحی می کنیم که از طریق SqlDataSource سطرهای مربوط به جدول Authors در پایگاه داده pubs را بازیابی و در یک GridView نمایش می دهد. کنترل Sql Data Source داده ها را به مدت ۶۰ ثانیه کش می کند.

یادداشت: به طور کلی هر چه داده ها ایستاتر (Static) باشند زمان کش کردن آنها بیشتر خواهد بود.

1. VS 2005 را باز کنید و یک وب سایت جدید باز کنید. نام آن را C:\ASPNET20\chap04- *DataSourceCache* قرار دهید.

2. یک GridView به فرم اضافه کنید و یک کنترل Sql Data Source جهت اتصال GridView به پایگاه داده pubs پیکربندی کنید. (مراحل انجام این کار را در بخش ۴-۱ آموختید) فیلدهای au _ id ، au _ fname و au _ lname را از پایگاه داده بازیابی کنید.

Source View کنترلهای GridView و Sql Data Source به صورت مثال ۴-۴ خواهد بود.

```
<asp:GridView ID="GridView1" runat="server"
    DataSourceID="SqlDataSource1"
    AutoGenerateColumns="False"
    DataKeyNames="au_id">
    <Columns>
        <asp:BoundField ReadOnly="True" HeaderText="au_id"
            DataField="au_id" SortExpression="au_id">
        </asp:BoundField>
```

```

<asp:BoundField HeaderText="au_lname"
    DataField="au_lname" SortExpression="au_lname">
</asp:BoundField>
<asp:BoundField HeaderText="au_fname"
    DataField="au_fname" SortExpression="au_fname">
</asp:BoundField>
</Columns>
</asp:GridView>

```

```

<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    SelectCommand="SELECT [au_id], [au_lname],
        [au_fname] FROM [authors]"
    ConnectionString="<%%$ ConnectionStrings:
        pubsConnectionString %>">
</asp:SqlDataSource>

```

مثال ۴-۴. Source View مربوط به کنترل‌های Gridview و Sql Data Source

3. برای اینکه کنترل Sql Data Source قابلیت کش کردن داشته باشد، خواص Enable Caching و Cache Duration را همانند کد زیر تنظیم نمایید :

```

<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    SelectCommand="SELECT [au_id], [au_lname],
        [au_fname] FROM [authors]"
    ConnectionString="<%%$ ConnectionStrings: pubsConnectionString %>"
    CacheDuration="60"
    EnableCaching="True"
    CacheExpirationPolicy="Absolute" >
</asp:SqlDataSource>

```

4. کلید F5 را فشار دهید . برای اینکه کش شدن اطلاعات را به چشم خود مشاهده کنید . یکی از سطرهای جدول

Authors را تغییر دهید و صفحه را Refresh کنید ، متوجه می شوید که اطلاعات سطر مربوطه در Gridview تغییر نمی

کنند تا اینکه حدود ۱ دقیقه (۶۰ ثانیه) از اولین نمایش اطلاعات توسط Gridview بگذرد .

زمان انقضای (expiration) پیش فرض عبارت **absolute** می باشد و معنی آن این است که داده های کش شده در Data Source پس از n ثانیه (زمانی که در خاصیت Cache Duration مشخص شده است) از زمان بارگذاری از بین می روند .

۴-۵-۲- پاورقی بخش ...

... استفاده از حالت Sliding برای کش کردن داده ها

در این حالت ، چنانچه طول مدت زمان خاصی هیچ گونه درخواستی صورت نگیرد ، اطلاعات کش شده از بین می روند. به عبارت دیگر پس از کش شدن داده ها به Data Source مادامیکه در طول مدت زمان تعیین شده به Data Source درخواست فرستاده می شود از اطلاعات کش شده استفاده می کند . به عنوان مثال قطعه کد زیر Sliding duration را یک دقیقه مشخص می کند . حال چنانچه درخواست ها در بازه مشخص شده صورت گیرند ، اعتبار اطلاعات کش شده مجدداً به یک دقیقه تنظیم می شود .

Sliding برای زمانی مفید است که آیتمهای زیادی جهت کش داشته باشید و باعث می شود تنها اطلاعاتی که دسترسی به آنها زیاد است در حافظه باقی بمانند .

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    SelectCommand="SELECT [au_id], [au_lname],
        [au_fname] FROM [authors]"
    ConnectionString="<%%$ ConnectionStrings:pubsConnectionString %>"
    CacheDuration="60"
    EnableCaching="True"
    CacheExpirationPolicy="Sliding" >
</asp:SqlDataSource>
```

۴-۵-۳. منابع اطلاعاتی :

برای بررسی بیشتر Caching در Asp.Net به مقاله زیر توجه کنید .

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpatterns/html/ImpPageCacheInASP.asp>

همچنین در فصل شش کتاب می توانید راجع به چگونگی کش کردن صفحه به صورت قطعه بندی (fragments) مطالب بیشتری بیابید .

۴-۶- کشینگ وابسته

یادداشت: به جای از بین بردن داده های کش شده بر اساس مدت زمان بهتر است داده ها هنگام اعمال تغییرات در پایگاه داده از بین بروند.

مشخص کردن زمان برای کش کردن داده ها همیشه کار آیی ندارد. به عنوان مثال فرض کنید داده ها برای مدت ۱۰ دقیقه کش کرده اید، اما ممکن است طی همین مدت اطلاعات پایگاه داده چندین بار به روز (update) شوند. به همین دلیل بهتر است کش کردن داده به تغییرات صورت گرفته در پایگاه داده وابسته باشد.

در Asp.Net 2.0 با استفاده از SQL Cache Dependency می توانید اطلاعات کش شده را هنگام رخ دادن تغییرات در پایگاه داده، باطل (Invalidate) کنید. ضمناً اطمینان داشته باشید که اطلاعات، کش شده همزمان با تغییرات پایگاه داده به روز می شوند.

۴-۶-۱- مراحل انجام کار:

به منظور استفاده از SQL Cache Dependency ابتدا بایستی پایگاه داده و جدولهای آن را برای تغییرات آماده سازید. (بطوریکه برنامه Asp.net از تغییرات پایگاه داده آگاهی یابد) برای این کار از برنامه `aspnet_regsql.exe` استفاده می کنیم و هنگامی که پایگاه داده آماده شد، می توانید SQL Cache Dependency را با استفاده از کنترلهای `Gridview` و `SqlDataSource` آزمایش کنید.

نکته: چنانچه از SQL Server 2005 استفاده می کنید نیازی به انجام مراحل ۱ تا ۴ نیست.

1. برای استفاده از `aspnet_regsql.exe` از طریق Comand prompt به آدرس زیر بروید:

`C:\WINDOWS\Microsoft.NET\Framework\<version>`

نکته: با فرض اینکه windows و VS 2005 را بر روی درایو C:\ نصب کرده باشید.

2. `aspnet_regsql.exe` (که هم به صورت گرافیکی عمل می کند و هم دستوری) ابزار است جهت پیکربندی Sql server برای استفاده برنامه های Asp.net. پارامترهای مختلف این ابزار به شرح زیرند:

C:\WINDOWS\Microsoft.NET\Framework\version>aspnet_regsql /?

پارامترهای aspnet_regsql.exe

در این بخش خلاصه ای از پارامترهای این ابزار را معرفی می کنیم :

-S	نام سرور
-E	تأیید هویت بر اساس تأیید هویت فعلی ویندوز
-d	نام پایگاه داده
-t	نام جدول
-ed	فعال کردن پایگاه داده
-et	فعال کردن جدول

3. برای فعال سازی SQL Cache Dependency دو کار باید انجام دهید :

الف) فعال سازی پایگاه داده

ب) فعال سازی جدول (جدولها)

4. برای فعال سازی پایگاه داده از پارامتر ed- استفاده کنید :

C:\WINDOWS\Microsoft.NET\Framework\version>aspnet_regsql

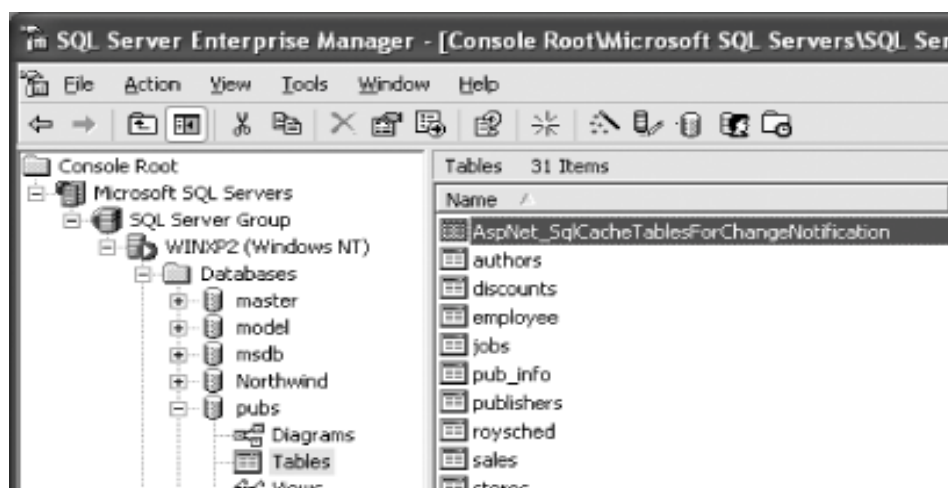
-S localhost -E -d Pubs -ed

5. برای فعال سازی جدول (جدولها) از پارامتر et- استفاده کنید :

C:\WINDOWS\Microsoft.NET\Framework\version>aspnet_regsql

-S localhost -E -t Authors -d Pubs -et

مراحل ۳ و ۴ جدول جدیدی به نام **AspNet_SqlCacheTablesForChangeNotification** در پایگاه داده مورد نظر ایجاد می کنند. (شکل ۴-۴۰ را ببینید)



شکل ۴-۴۰- جدول **AspNet_SqlCacheTablesForChangeNotification**

جدول **AspNet_SqlCacheTablesForChangeNotification** دارای سه فیلد است که در شکل ۴-۴۱ نمایش داده شده اند. که آخرین زمان اعمال تغییر در جدول مورد استفاده (از طریق فیلد **notificationCreated**) و تعداد تغییرات را نگهداری می کند.

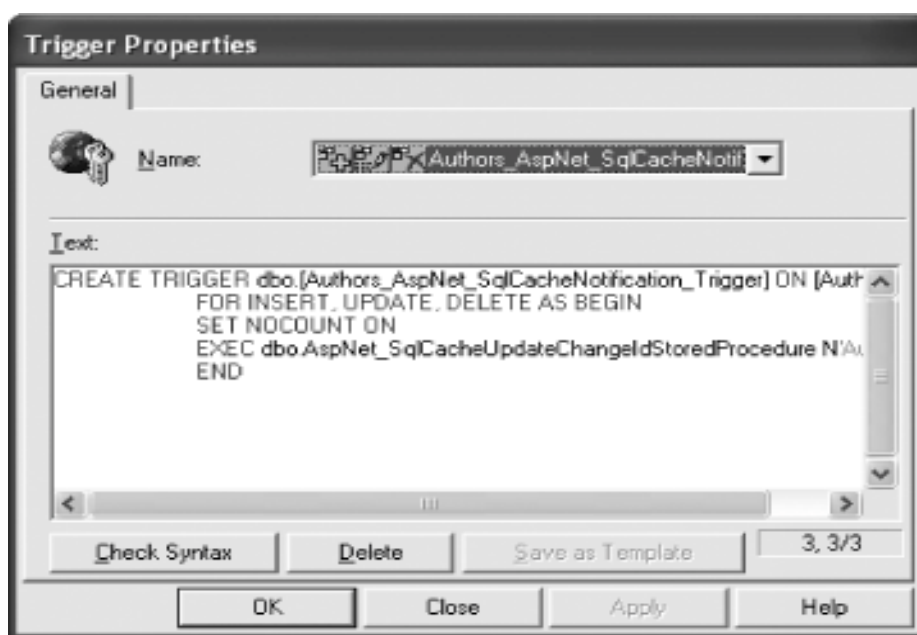
هر زمان که تغییری در جدول صورت می گیرد یک واحد به مقدار فیلد **changeId** اضافه می شود.

 A screenshot of the 'Data in Table' window for the 'AspNet_SqlCacheTablesForChangeNotification' table. The table has three columns: 'tableName', 'notificationCreated', and 'changeId'. There is one row of data with 'Authors' in the 'tableName' column, '1/20/2005 12:02:40' in the 'notificationCreated' column, and '0' in the 'changeId' column.

tableName	notificationCreated	changeId
Authors	1/20/2005 12:02:40	0

شکل ۴-۴۱- محتویات جدول **AspNet_SqlCacheTablesForChangeNotification**

Sql server از طریق یک راه انداز (Trigger) به نام **AspNet_SqlCacheNotification_Trigger** (که توسط *aspnet_regsql* نصب شده است) زمان افزایش فیلد جدول **AspNet_SqlCacheTablesForChangeNotification** را تشخیص می دهد. در واقع هنگام تغییر جدول Trigger فراخوانی می شود. (شکل ۴-۴۲ را ببینید)



شکل ۴-۴۲ - AspNet_SqlCacheNotification_Trigger

6. اکنون که پایگاه داده شما برای SQL Cache Dependency فعال شد ، در VS 2005 پروژه جدیدی با نام *C:\ASPNET20\Chap04-SQLCacheDep* ایجاد کنید .
7. یک فایل *web.config* به پروژه اضافه کنید ، برای این کار بر روی نام پروژه در Solution Explorer راست کلیک کنید گزینه های *Add New Item...* و *web.configuration File* را انتخاب کنید .
8. مرحله بعدی افزودن رشته اتصال پایگاه داده (Data Base Conection String) به برنامه است . کد مثال ۴-۵ را به فایل *web.config* اضافه کنید .

```
<?xml version="1.0"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <appSettings/>
  <connectionStrings>
    <add name="PubsDatabase"
      connectionString="Server=(local); Integrated Security=True;
      Database=pubs;Persist Security Info=True"
      providerName="System.Data.SqlClient" />
  </connectionStrings>

  <system.web>
```

```

<compilation debug="true"/>
<authentication mode="Windows"/>
<casting>
  <sqlCacheDependency enabled="true">
    <databases>
      <add
        name="Pubs"
        connectionStringName="PubsDatabase"
        pollTime="10000" />
    </databases>
  </sqlCacheDependency>
</casting>
</system.web>
</configuration>

```

مثال ۴-۵ - ویرایش web.config برای پشتیبانی از SQL Cache Dependency

نکته : اگر از Sql Server 2005 استفاده می کنید نیازی به افزودن < Caching > و زیر مجموعه های آن به web.config نیست .

9. در فایل web.config مثال ۴-۵ یک رشته اتصال برای اتصال به پایگاه داده pubs ایجاد کردید . همچنین مشخص کردیم که می خواهیم از SQL Server caching با زمان نمونه برداری (polling time) ۱۰ ثانیه (زمان بر حسب میلی ثانیه است) استفاده کنیم .

این زمان همان زمانی است که Asp.Net از پایگاه داده (خصوصاً جدولAspNet_SqlCacheTablesForChangeNotification) نمونه برداری می کند .

از آنجا که این جدول بسیار کوچک است نمونه برداری از این جدول عملی کارا و مؤثر بوده و سرعت سیستم را کاهش نمی دهد . از همین رو هر چه زمان مشخص شده کمتر باشد اطلاعات نمایش داده شده به روزتر خواهند بود .

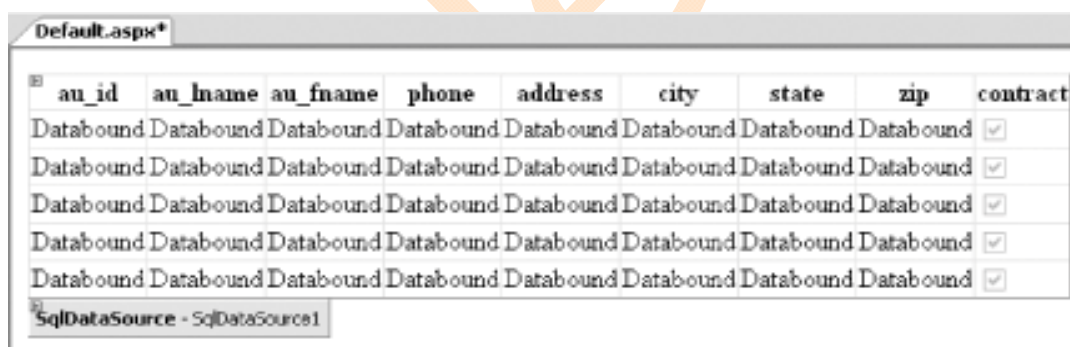
10. یک Gridview به فرم پروژه اضافه کنید . Gridview را برای استفاده از یک کنترل SqlDataSource پیکربندی کنید . در واقع کنترل SqlDataSource از رشته اتصال ذخیره شده در web.config استفاده می کند view Source کنترل Gridview به صورت زیر خواهد بود :

```
<asp:GridView ID="GridView1" runat="server"
    DataSourceID="SqlDataSource1"
    AutoGenerateColumns="False"
    DataKeyNames="au_id">
    ...
    ...
</asp:GridView>

<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    SelectCommand="SELECT * FROM [authors]"
    ConnectionString="<%%$ ConnectionStrings:PubsDatabase %>">
</asp:SqlDataSource>
```

نکته: پیگر بندی Gridview برای استفاده از SqlDataSource در بخش ۴-۱ همین فصل توضیح داده شد.

11. فرم وب ایجاد شده مشابه شکل ۴-۴۳ می باشد.



شکل ۴-۴۳. کنترل های Gridview و SqlDataSource

12. به view Source صفحه بروید و کد مربوط به SqlDataSource را جهت استفاده از Sql Cach Dependency به صورت زیر ویرایش کنید:

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    EnableCaching="true" SqlCacheDependency="Pubs:Authors"
    SelectCommand="SELECT * FROM [authors]"
    ConnectionString="<%%$ ConnectionStrings:PubsDatabase %>">
</asp:SqlDataSource>
```

سطوح کش کردن اطلاعات در Sql server 7,2000,2005

در Sql server 7,2000 صرف نظر از تعداد صفحات کش شده تغییر در هر رکورد از یک جدول باعث نامعتبر شدن داده های کش شده می شود . اما در Sql server 2005 تنها داده های کش شده مربوط به صفحاتی که دچار تغییر می شوند ، منقضی می شوند و اطلاعات سایر صفحات همچنان کش شده باقی می ماند و همچنان می توانند مورد استفاده قرار گیرند .

در Sql server 2005 نیازی به نمونه برداری پیوسته از پایگاه داده نیست .

در عوض Sql server 2005 بطور خود کار هر گونه تغییرات داده ها را به Asp.Net اعلام می کند . Sql server 2005 این کار را از طریق سرویسهای گزارش دهی (Notification Delivery Services) انجام می دهد که بطور مستقیم با IIS در تقابل هستند .

نکته : بمنظور نمونه برداری SQL Server 2000 ؛ خاصیت SQLCache Depebdency قالب database:table دارد اما اگر از SQL Server 2005 استفاده می کنید، خاصیت SQLCache Depebdency را برابر CommandNotification قرار دهید.

همچنین کد زیر را در رویداد Page_Load صفحه اضافه کنید:

```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles Me.Load
    Dim settings As ConnectionStringSettings
    settings = _
        ConfigurationManager.ConnectionStrings("PubsDatabase")
    System.Data.SqlClient.SqlDependency.Start( _
        settings.ConnectionString)

End Sub
```

13. کلید F5 را فشار دهید . اطلاعات جدول Authors را تغییر دهید . (برای این کار می توانید از Enterprise manager یا Query analyzer استفاده کنید) و مرورگر را Refresh کنید . مشاهده می کنید که با تغییر داده های جدول Authors اطلاعات Gridview نیز (تقریباً) هر ۱۰ ثانیه به روز می شوند .

۴-۶-۲- پاورقی بخش ...

... کش کردن خروجی صفحه

شما می توانید از page output caching به همراه SQL Cache Dependency استفاده کنید . به عنوان مثال فرض کنید یک کنترل Gridview در یک فرم دارید که به یک کنترل SqlDataSource متصل است . می توانید با استفاده از رهنمود OutputCache (directive) از کش کردن خروجی صفحه بهره مند شوید :

```
<%@ OutputCache Duration="15000" VaryByParam="none"
    SqlDependency ="Pubs:Authors" %>
```

فرض کنید SQL Cache Dependency با زمان ۱۰ ثانیه تنظیم شده باشد ، در این صورت Gridview هر ۱۰ ثانیه بازسازی (Refresh) می شود (چنانچه تغییراتی صورت پذیرد) از طرف دیگر صفحه هر ۱۵ ثانیه یک بار Expire می شود و هر گونه تغییر هنگام بازسازی صفحه در هر ۱۵ ثانیه ، نمایش داده می شود .

... فعال سازی پایگاه داده و جدولهای آن از طریق برنامه نویسی

به جای استفاده از *aspnet_regsql.exe* می توانید از طریق برنامه نویسی و با استفاده از کلاس *SqlCacheDependencyAdmin* پایگاه داده و جدولهای آن را فعال کنید .

مثال ۴-۶ نحوه انجام این کار را نشان می دهد :

```
Public Sub EnableNotification(ByVal tableName As String)
```

```
Dim connStr As String = "Server=.\SQLEXPRESS; Integrated
```

```
Security=True; Database=pubs; Persist Security
```

```
Info=True"
```

```
Dim mustEnable As Boolean = True
```

```
Try
```

```
Dim tablesEnabled( ) As String
```

```
'---Retrieve all tables enabled for notifications---
```

```
tablesEnabled = _
```

```
    SqlCacheDependencyAdmin. _
```

```
    GetTablesEnabledForNotifications(connStr)
```

```
If (tablesEnabled IsNot Nothing) Then
```

```
    Dim table As String
```

```

Response.Write("<b>Tables Enabled For " & _
                "Notification</b><br/>")
For Each table In tablesEnabled
    Response.Write(table & "<br>")
    If (table.ToString.Equals(tableName)) Then
        mustEnable = False
    End If
Next
End If
Catch ex As Exception
    mustEnable = True
End Try
If mustEnable Then
    '--enables the database for notification
    SqlCacheDependencyAdmin.EnableNotifications(connStr)
    '--enables the table for notification
    SqlCacheDependencyAdmin. _
        EnableTableForNotifications(connStr, tableName)
    Response.Write(tableName & "<br>")
End If
End Sub

```

مثال ۴-۶. فعال سازی پایگاه داده و جدول های آن

کلاس **SqlCacheDependencyAdmin** مدیریت **Sql Server** را جهت فعال سازی **SQL Cache Dependency** بر عهده دارد. متد (**GetTablesEnabledForNotifications**) لیست تمامی جدولهای فعال شده در پایگاه داده را بازبایی می کند و نام جدول ها را به صورت آرایه ای از رشته های نشان می دهد. چنانچه نام جدولی در این آرایه وجود داشته باشد به این معنی است که آن جدول فعال شده است در غیر این صورت فعال نیست و بایستی آن را فعال کنید.

برای این منظور ابتدا بایستی پایگاه داده و سپس جدول را فعال کنید. از متد (**EnableNotifications**) جهت فعال سازی پایگاه داده و از متد (**EnableTableForNotifications**) جهت فعال سازی هر یک از جدول ها استفاده کنید.

۴-۶-۳- منابع اطلاعاتی :

برای بررسی نحوه اجرای SQL Cache Dependency در Net 1.1. مقاله زیر را مطالعه کنید :

<http://www.ondotnet.com/pub/a/dotnet/2005/01/17/sqlcachedependency.html>

برای آگاهی بیشتر از کش کردن داده ها در Asp.Net 1.x , 2.0 به مقاله زیر مراجعه کنید :

<http://www.devx.com/asp/Article/21751>

و نهایتاً برای شناختن تفاوت‌های Sql server 2000 و Sql server 2005 در Sql Cache Dependency به لینک زیر مراجعه کنید :

<http://beta.asp.net/GuidedTour/s20.aspx>

۴-۷- به رمز در آوردن رشته های اتصال (Encrypt Connection Strings)

یادداشت : با ذخیره رشته اتصال (Connection Strings) در فایل web.config از آن محافظت کنید .

بجای ذخیره رشته اتصال پایگاه داده در صفحات برنامه بهتر (و راحت تر) این است که آن را در فایل web.config ذخیره کنید . Asp.Net 2.0 این قابلیت را از طریق رمز دار کردن رشته اتصال و ذخیره در web.config فراهم آورده است .

۴-۷-۱- مراحل انجام کار :

برای آشنایی با این قابلیت از یک کنترل Gridview به همراه SqlDataSource استفاده می کنیم و رشته اتصال استفاده شده توسط SqlDataSource را در فایل web.config ذخیره می کنیم . سپس با استفاده از سرویس دهنده های پیکر بندی موجود در Asp.Net 2.0 رشته اتصال را رمز دار می کنیم .

1. در VS 2005 پروژه جدیدی با نام `C:\ASPNET20\chap04-EncryptConfig` ایجاد کنید .

2. یک Gridview به فرم اضافه کنید و آن را جهت استفاده از یک SqlDataSource پیکربندی کنید . کنترل SqlDataSource را برای اتصال به پایگاه داده pubs و استفاده از جدول Authors پیکربندی کنید . دقت کنید که حتماً رشته اتصال را در web.config ذخیره کنید . (نحوه انجام این کار قبلاً توضیح داده شد)

نکته: برای پیکربندی Gridview و SqlDataSource به صورت گفته شده به بخش ۴-۱ مراجعه کنید.

3. کدهای زیر به فایل web.config افزوده شده است:

```
<configuration
  xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <appSettings/>
  <connectionStrings>
    <add name="pubsConnectionString" connectionString="Data
      Source=.\SQLEXPRESS; Initial Catalog=pubs;
      Integrated Security=True"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
  <system.web>
  ...
```

4. به Code – behind فرم بروید و متد () Encrypt را به آن اضافه کنید. متد () Encrypt ابتدا فایل web.config را باز می کند و سپس بخش خالی از فایل (در اینجا بخش < Connection Strings >) را به رمز درمی آورد. این متد از سرویس دهنده حفاظتی (protection provider) استفاده می کند:

```
Imports System.Configuration
Imports System.Web.Security
```

```
Public Sub Encrypt(ByVal protectionProvider As String)
  '---open the web.config file
  Dim config As Configuration = _
    System.Web.Configuration. _
      WebConfigurationManager.OpenWebConfiguration( _
        Request.ApplicationPath)
  '---indicate the section to protect
  Dim section As ConfigurationSection = _
    config.Sections("connectionStrings")
  '---specify the protection provider
  section.SectionInformation.ProtectSection(protectionProvider)
  '---Apply the protection and update
  config.Save( )
End Sub
```

5. برای استفاده از رشته رمز دار شده نیازمند متدی برای بازگشایی آن هستید. سپس متد Decrypt() را نیز به صفحه اضافه کنید:


```
Public Sub Decrypt( )
    Dim config As Configuration = _
        System.Web.Configuration. _
            WebConfigurationManager.OpenWebConfiguration( _
                Request.ApplicationPath)
    Dim section As ConfigurationSection = _
        config.Sections("connectionStrings")
    section.SectionInformation.UnProtectSection( )
    config.Save( )
End Sub
```

نکته : همانطور که مشاهده می کنید متد () UnProtectSection بر خلاف () ProtectSection به نام سرویس دهنده نیاز ندارد . هنگامی که یک بخش به رمز در می آید اطلاعات مربوط به سرویس دهنده رمز دار کننده در فایل web.config ذخیره می شود و متد () UnProtectSection با استفاده از همین اطلاعات می تواند سرویس دهنده مربوطه را تشخیص دهد .

6. از دو سرویس دهنده حفاظتی می توانید استفاده کنید :

- DataProtectionConfigurationProvider
- RSAProtectedConfigurationProvider

برای اجرای متد () Encrypt آن را در رویداد Page – Load فراخوانی کنید :

```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load
    Encrypt("DataProtectionConfigurationProvider")
    '---or---
    ' Encrypt("RSAProtectedConfigurationProvider")
End Sub
```

نکته : RSAProtectedConfigurationProvider از الگوریتم public-key کلاس RSACryptoServiceProvider جهت رمز گذاری استفاده می کند. و DataProtectionConfigurationProvider از DPAPI ویندوز. ضمناً متد () Encrypt را تنها یکبار میتوانید فراخوانی کنید .

7. اگر از `DataProtectionConfigurationProvider` استفاده کنید رشته اتصال شما شبیه مثال ۴-۷ خواهد بود .

...

```
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
```

```
  <connectionStrings
    configProtectionProvider="DataProtectionConfigurationProvider">
    <EncryptedData>
      <CipherData>
        <CipherValue>AQAAANCMnd.....gioRQROy/n</CipherValue>
      </CipherData>
    </EncryptedData>
  </connectionStrings>
</system.web>
```

...

مثال ۴-۷ رشته اتصال با `Data Protection Configuration Provider`

8. چنانچه از `RSAProtectedConfigurationProvider` برای رمز گذاری استفاده کنید رشته اتصال مانند مثال ۴-۸

می شود .

...

```
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
```

```
  <connectionStrings
    configProtectionProvider="RsaProtectedConfigurationProvider">
    <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
      xmlns="http://www.w3.org/2001/04/xmlenc#">
      <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc" />
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
          <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
```

```

<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
  <KeyName>Rsa Key</KeyName>
</KeyInfo>
<CipherData>
  <CipherValue>CgnpVe87ISd3.....N4vqWOUbIKNo=</CipherValue>
</CipherData>
</EncryptedKey>
</KeyInfo>
<CipherData>
  <CipherValue>rIRKN2QkGZM.....ScwFDL2IRjM=</CipherValue>
</CipherData>
</EncryptedData>
</connectionStrings>
<system.web>
...

```

مثال ۴-۸ رشته اتصال با RSAProtectedConfigurationProvider

9. یکی از مزایای رمز گذاری فایل web.config این است که عملیات رمز گشایی رشته اتصال کلاً از دید برنامه نویس پنهان است. کنترلها و کدی که به رشته اتصال نیاز دارند به طور خود کار می دانند چگونه اطلاعات رمزدار را رمز گشایی کنند. به هر حال چنانچه می خواهید فایل web.config را رمز گشایی کنید (مثلاً تغییری در آن ایجاد کنید) تنها کافیسست متد Decrypt() را فراخوانی کنید.

10. با استفاده از خاصیت Isprotected() می توانید تشخیص دهید که آیا از یک بخش خاص حفاظت می شود یا خیر مانند زیر:

```

If Not section.SectionInformation.IsProtected Then
  section.SectionInformation.ProtectSection(protectionProvider)
  config.Save()
End If

```

حفاظت خوددار (Self-Contained Protection)

دقت کنید که خاصیت configProtectionProvider اضافه شده به عنصر connectionStrings شامل اطلاعات مورد نیاز جهت رمز گشایی رشته اتصال است.

مهمتر از آن در فایل web.config هیچ گونه کلید رمزگشایی وجود ندارد. در حالیکه به عنوان مثال اگر از DataProtectionConfigurationProvider ویندوز استفاده کنید، کلید رمزگشا بصورت خودکار تولید و در LSA (Local Security Authority) ذخیره میشود.

۴-۷-۲- پاورقی بخش ...

... افزودن رشته اتصال جدید به فایل web.config رمز شده از طریق برنامه نویسی

متد () AddConnString نحوه افزودن رشته اتصال به فایل web.config را نشان می دهد :

```
Public Sub AddConnString ( )
    '---add a connection string to Web.config
    Dim config As Configuration = _
        System.Web.Configuration. _
            WebConfigurationManager.OpenWebConfiguration( _
                Request.ApplicationPath)
    config.ConnectionStrings.ConnectionStrings.Add _
        (New ConnectionStringSettings("NorthwindConnectionString", _
            "server=.\\SQLEXPRESS;database=northwind;integrated security=true"))
    config.Save( )
End Sub
```

... حفاظت سایر قسمتهای web.config

تقریباً تمامی بخشهای فایل web.config را می توانید رمز دار کنید به استثنای بخشهایی که در ASP.NET مدیریت نمی شوند از جمله <httpRuntime> و <processModel>

... بازیابی رشته اتصال از طریق برنامه نویسی

همانند کد زیر می توانید رشته اتصال را از فایل web.config بازیابی کنید:

```
Dim connect As String = _
    ConfigurationManager.ConnectionStrings _
    ("NorthwindConnectionString").ConnectionString
Response.Write(connect)
```

۴-۷-۳- منابع اطلاعاتی :

برای کسب اطلاعات بیشتر راجع به کلاس ProtectedConfigurationProvider در MSDN موضوع "ProtectedConfigurationProvider Class" را جستجو کنید .

برای آگاهی از چگونگی حفاظت اطلاعات توسط ویندوز به آدرس زیر مراجعه کنید :

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsecure/html/windataprotection-dpapi.asp>

برای آشنایی بیشتر در مورد کلاسهای رمزگذار در دات نت مقاله زیر را مطالعه کنید :

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnhcvs03/html/vs03l1.asp>

۴-۸- اتصال به Business Object

یادداشت : به عنوان برنامه نویس وب می توانید بسیاری از اعمال مورد نیازتان را به صورت کامپوننت و شیء در آورید و سپس کنترلرها (مانند Gridview و ...) را به این اشیاء مقید (bind) سازید .

دریک طراحی خوب اعمال مختلف درون اشیاء مختلف قرار می گیرند و هنگام نیاز به هر عمل تنها به شیء مربوطه دسترسی صورت می گیرد . به عنوان مثال برنامه نویسان اعمال مورد نیازشان را در کلاسهای جداگانه می نویسند و از طریق آنها با کاربر ارتباط متقابل دارند .

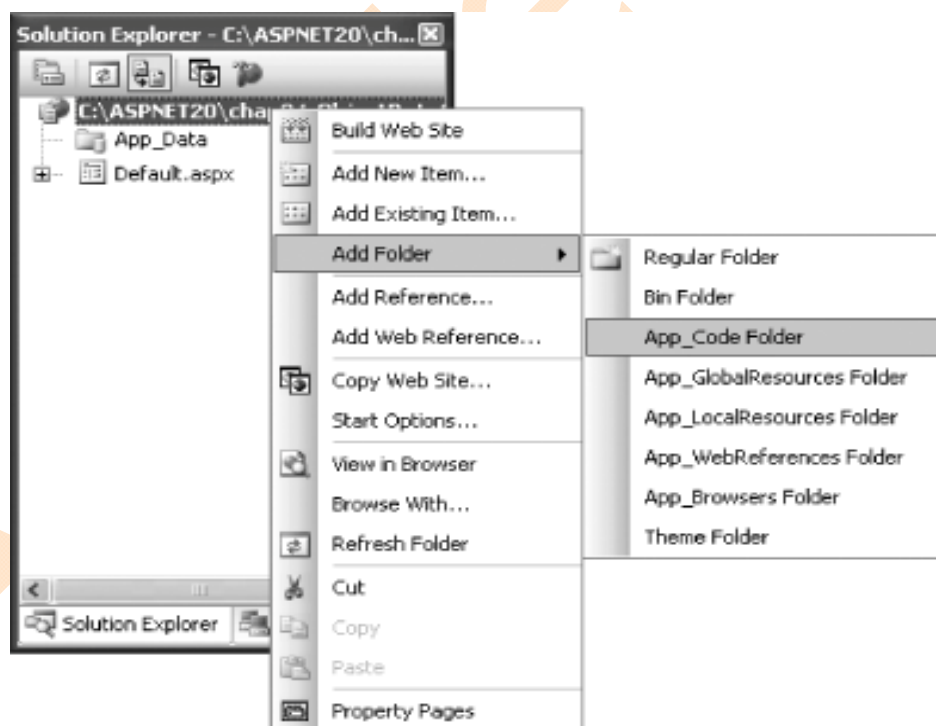
نکته: برای آشنایی بیشتر با مباحث مربوط به برنامه نویسی شی گرا یا (Object Oriented Programming) OOP به کتابهای نوشته شده در این زمینه مراجعه کنید.

در Asp.Net 2.0 نیز می توانید اطلاعات را درون کلاسها نگهداری کنید و با مقید ساختن کنترلها (مثل Gridview) از طریق ObjectDataSource به داده ها دسترسی داشته باشید .

۴-۸-۱. مراحل انجام کار :

برای اینکه ببینید چگونه می توان از طریق ObjectDataSource یک کنترل Gridview را به یک شیء (object) مقید ساخت ، کلاسی جهت دسترسی به داده ها طراحی می کنیم سپس یک کنترل ObjectDataSource را به آن مقید می سازیم و در نهایت Gridview با استفاده از ObjectDataSource اطلاعات را نمایش می دهد .

1. در VS 2005 پروژه جدیدی با نام `C:\ASPNET20\chap04-ObjectDataSource` ایجاد کنید.
2. بر روی نام پروژه Solution Explorer راست کلیک کنید و گزینه های `Add Folder` → `App_Code` را انتخاب کنید . (شکل ۴-۴۵) از این شاخه برای نگهداری کلاسها استفاده می کنیم .



شکل ۴-۴۵. افزودن شاخه جدید به پروژه

3. یک فایل `web.config` نیز به پروژه اضافه کنید . برای این کار بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های `Add New Item` → `Web Configuration File` را انتخاب کنید .

4. رشته اتصال به پایگاه داده را به فایل web.config اضافه کنید :

```
<connectionStrings>
  <add name="pubsConnectionString" connectionString="Data
    Source=.\SQLEXPRESS; Initial Catalog=pubs;
    Integrated Security=True"
    providerName="System.Data.SqlClient" />
</connectionStrings>
```

5. بر روی شاخه *App_Code* راست کلیک کنید و گزینه ... Add New Item را انتخاب کنید . در پنجره افزودن آیتم گزینه Class را انتخاب کنید و نام آن را Authors.vb قرار دهید .

6. کد مثال ۴-۹ را درون فایل Authors.vb بنویسید . این کلاس تنها یک تابع () get Authors دارد که یک Dataset باز می گرداند .

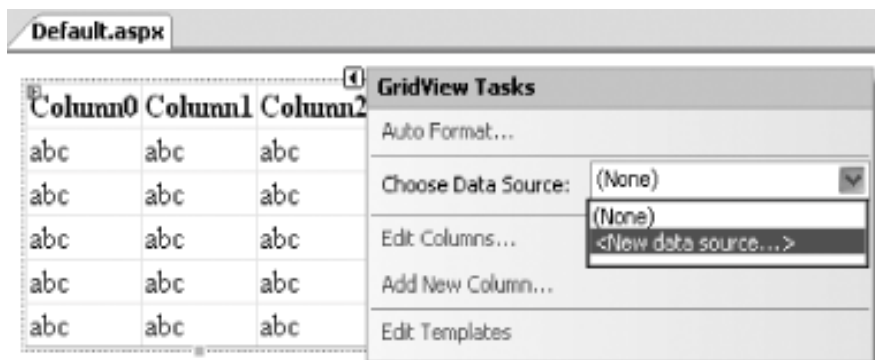
```
Imports System.Data.SqlClient
Imports System.Data
Imports Microsoft.VisualBasic
```

```
Public Class Authors
  Public Function getAuthors( ) As DataSet
    Dim conn As New SqlConnection( _
      ConfigurationManager.ConnectionStrings _
      ("pubsConnectionString").ConnectionString)
    Dim adapter As New SqlDataAdapter( _
      "SELECT au_id, au_fname, au_lname FROM Authors", conn)
    Dim ds As New DataSet
    adapter.Fill(ds, "Authors")
    Return ds
  End Function
End Class
```

مثال ۴-۹- کلاس Authors.vb

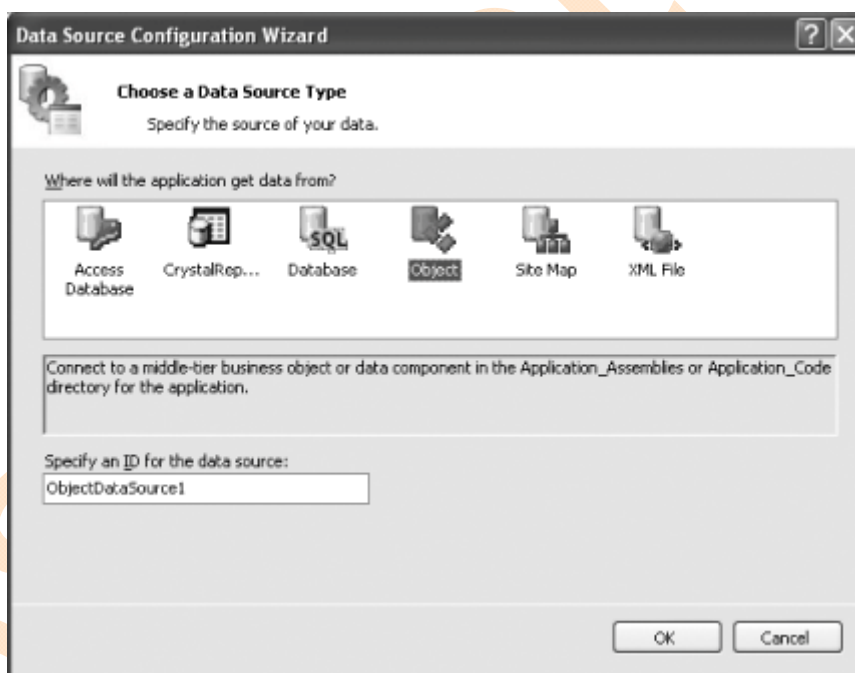
نکته : فراموش نکنید قبل از رفتن به مرحله بعد فایل Authors.vb را ذخیره کنید .

7. یک GridView به فرم اضافه کنید . در پنجره GridView Tasks گزینه <New data source...> را انتخاب کنید . (شکل ۴-۴۶)



شکل ۴-۴۶- انتخاب منبع اطلاعاتی جدید

8. در پنجره باز شده گزینه object را انتخاب نمایید . (شکل ۴-۴۷)
ID پیش فرض (ObjectDataSource1) را تغییر ندهید و ok کنید .



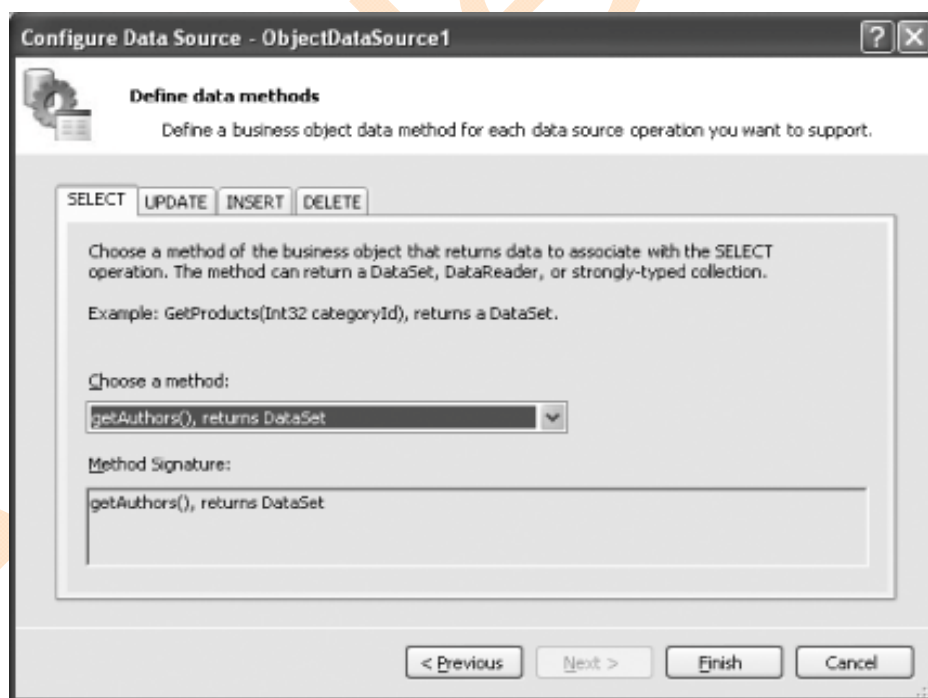
شکل ۴-۴۷- انتخاب object به عنوان منبع اطلاعاتی

9. در drop-down listbox Authors ؛ را به عنوان business object انتخاب کنید . (شکل ۴-۴۸) گزینه Next را کلیک کنید . در واقع با این کار کنترل Object Data Source را به کلاس Authors مقید کردید .



شکل ۴-۴۸. انتخاب object برای Object Data Source

10. در برگه SELECT از پنجره جدید "getAuthors(), returns DataSet" را انتخاب کنید و finish را کلیک کنید. (شکل ۴-۴۹)



شکل ۴-۴۹. انتخاب متد

11. کلید F5 را فشار دهید. همانگونه که مشاهده می کنید کنترل Gridview رکوردهای جدول Authors را نمایش می دهد.

نکته : چنانچه متد () getAuthors را مشاهده نمی کنید. احتمالاً فایل Authors.vb را ذخیره نکرده اید!

۴-۸-۲- پاورقی بخش ...

... ویرایش سطرهای Gridview از طریق کنترل Object Data Source

1. متد جدیدی با نام () updateAuthors به کلاس Authors اضافه کنید . (همانند مثال ۴-۱۰)

```
Public Sub updateAuthors(ByVal au_id As String, _
                        ByVal au_fname As String, _
                        ByVal au_lname As String)
    Dim conn As New SqlConnection( _
        ConfigurationManager.ConnectionStrings _
        ("pubsConnectionString").ConnectionString)

    Dim adapter As New SqlDataAdapter( _
        "SELECT au_id, au_fname, au_lname " & _
        "FROM Authors WHERE au_id=@au_id", conn)

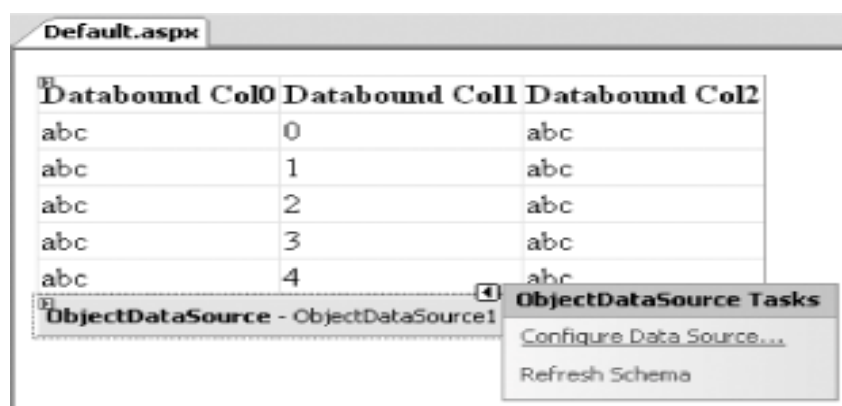
    Dim ds As New DataSet
    adapter.SelectCommand.Parameters.Add( _
        "@au_id", SqlDbType.NVarChar, 11).Value = au_id
    adapter.Fill(ds, "Authors")

    With ds.Tables(0).Rows(0)
        .Item("au_fname") = au_fname
        .Item("au_lname") = au_lname
    End With
    Dim cb As New SqlCommandBuilder(adapter)
    adapter.Update(ds, "Authors")
End Sub
```

مثال ۴-۱۰- متد () UpdateAuthors

2. بر روی لینک . . . Configure Data Source در پنجره ObjectDataSource Tasks کنترل Object

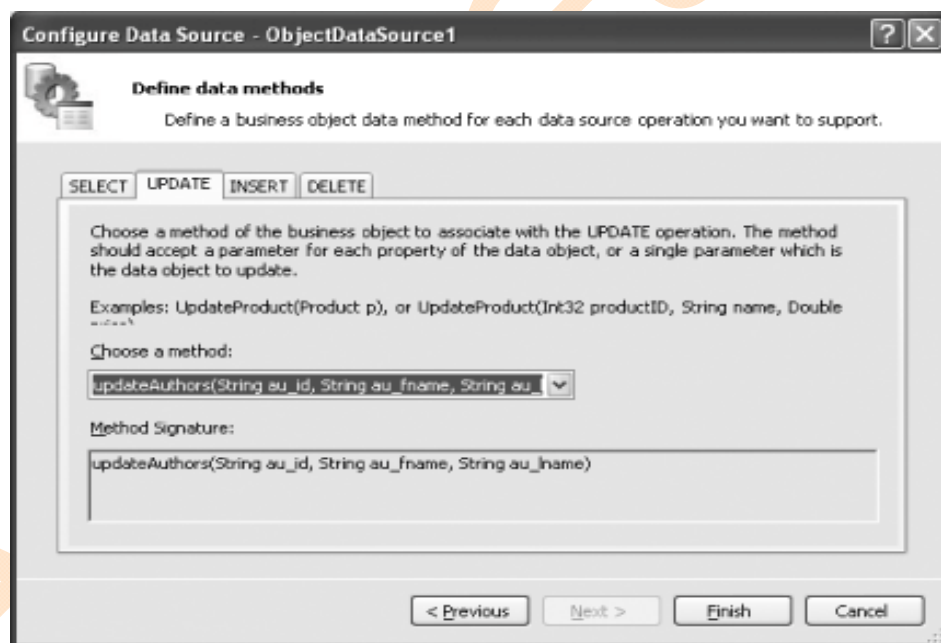
Data Source کلیک کنید . (۴-۵۰)



شکل ۴-۵۰- پیکربندی کنترل Object Data Source

3. گزینه Next را کلیک کنید و در برگه UPDATE متد (UpdateAuthors) را انتخاب کنید . (شکل ۴-۵۱) این متد تغییرات انجام شده در Gridview را به پایگاه داده اعمال می کند .

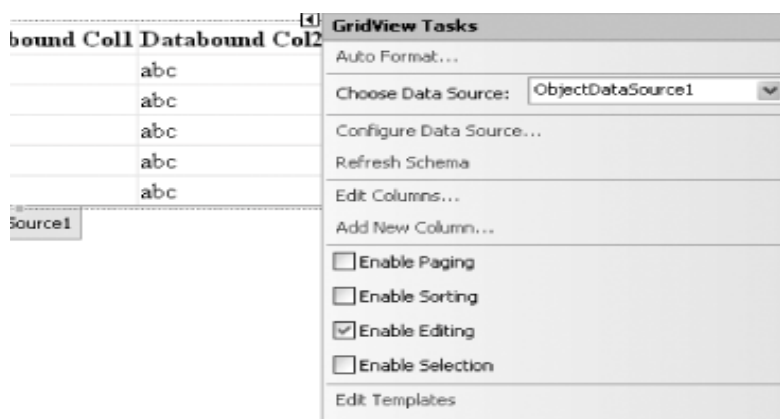
نکته : برای افزودن و حذف رکوردها کفایت متدهای مربوطه را به کلاس Authors اضافه کنید و در برگه های INSERT و DELETE آنها را انتخاب نمایید .



شکل ۴-۵۱ . انتخاب متد در برگه Update

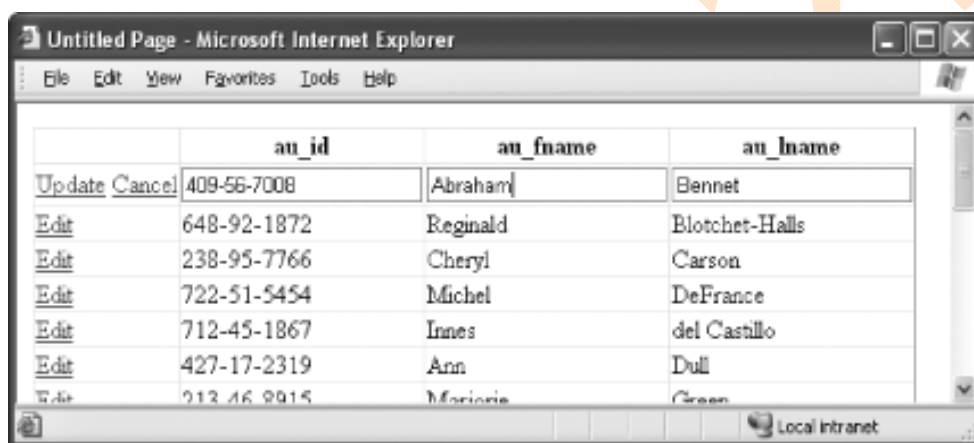
نکته : توجه کنید که متد update authors دارای پارامتر ورودی است ، این پارامترها همان ستونهای Gridview هستند .

4. در پنجره Gridview Tasks کنترل Gridview گزینه Enable Editing را انتخاب نمایید (شکل ۴-۵۲) تا Gridview لینک Edit را در اولین ستون سمت چپ نمایش دهد .



شکل ۴-۵۲- فعال سازی Edit در Gridview

5. کلید F5 را فشار دهید. همانطور که در شکل ۴-۵۳ مشاهده می کنید. سطرهای Gridview قابلیت ویرایش دارند.



شکل ۴-۵۳. ویرایش یک سطر از Gridview

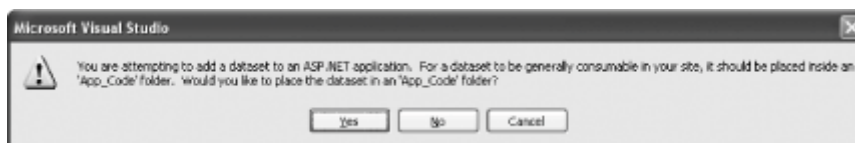
... استفاده از Data Component به همراه Object Data Source

به جای ذخیره متدهای دسترسی به داده درون یک کلاس، بهتر است اعمال استاندارد بودن انتخاب (selecting)، درج (Inserting)، حذف (Deleting) و به روزرسانی (updating) را Visual Studio به صورت خودکار تولید کند. در واقع این کار به کمک Data Component امکان پذیر است.

1. فرم جدیدی به نام Default2.aspx به پروژه قبلی اضافه کنید. (بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های Add New Item... سپس web form را انتخاب کنید)

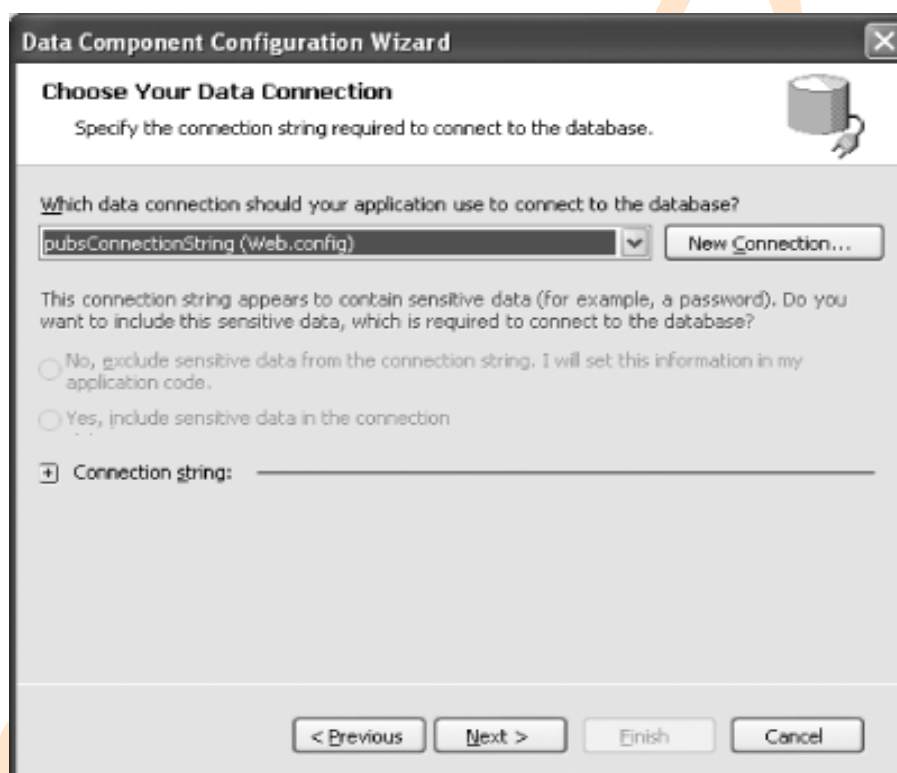
2. بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه Add New Item... را انتخاب کنید. در پنجره افزودن آیتم Data Set را انتخاب کنید و نام آن را Dataset.XSD قرار دهید.

3. از شما پرسیده می شود که آیا Data Set در شاخه App-code ذخیره شود یا خیر؟ گزینه yes را انتخاب کنید. (شکل ۴-۵۴)



شکل ۴-۵۴. ذخیره فایل XSD در شاخه App-code

4. پنجره "Data Component Configuration Wizard" به نمایش درمی آید. دکمه Next را کلیک کنید.
5. رشته اتصال موجود (pubsConnectionString) را انتخاب و Next را کلیک کنید. (شکل ۴-۵۵)



شکل ۴-۵۵. انتخاب رشته اتصال

6. در پنجره بعدی گزینه "Use SQL statements" را انتخاب و Next را کلیک کنید. (شکل ۴-۵۶)

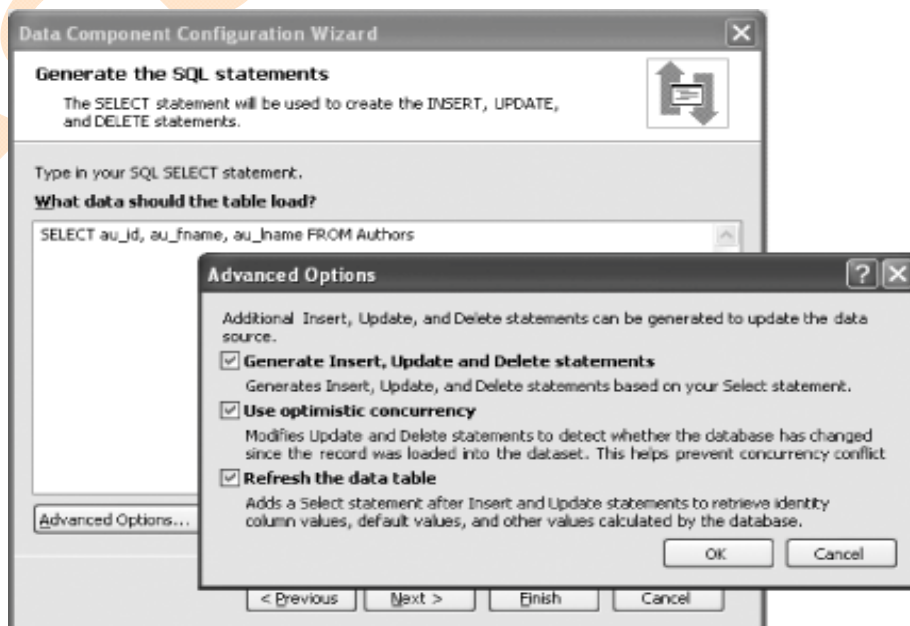


شکل ۴-۵۶- انتخاب نحوه دسترسی datacomponent به پایگاه داده

7. در کادر "What data should the table load?" دستور sql زیر را وارد نمایید :

`SELECT au_id, au_fname, au_lname FROM Authors`

8. گزینه Advanced Option... انتهای کادر را کلیک کنید و تمام چک باکس های موجود در پنجره را تیک بزنید . (شکل ۴-۵۷) در اینجا مشخص می کنید که دستورات درج ، به روزرسانی و حذف به طور خود کار تولید شوند . همچنین با استفاده از گزینه " Use optimistic concurrency " از به روزرسانی همزمان رکوردها جلوگیری کنید . گزینه ok و سپس Next را کلیک کنید .



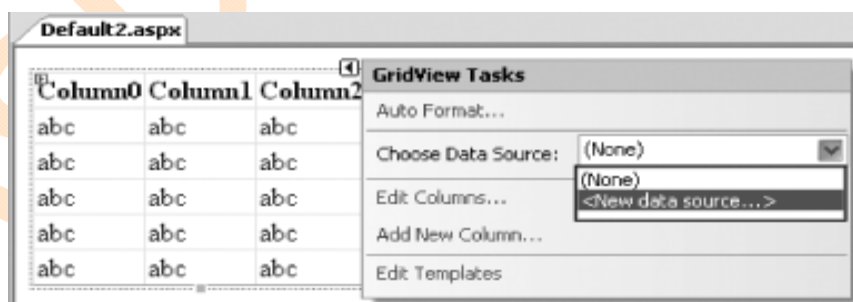
شکل ۴-۵۷- تولید تمامی دستورات SQL مورد نیاز

9. چک باکس های موجود در پنجره بعدی را نیز تیک بزنید . (شکل ۴-۵۸ را ببینید) در این مرحله متدهای مورد استفاده توسط data componet را مشخص می کنیم .مقادیر پیش فرض را تغییر ندهید و گزینه Next و نهایتاً Finish را انتخاب کنید .



شکل ۴-۵۸- تولید متدهای مورد نیاز

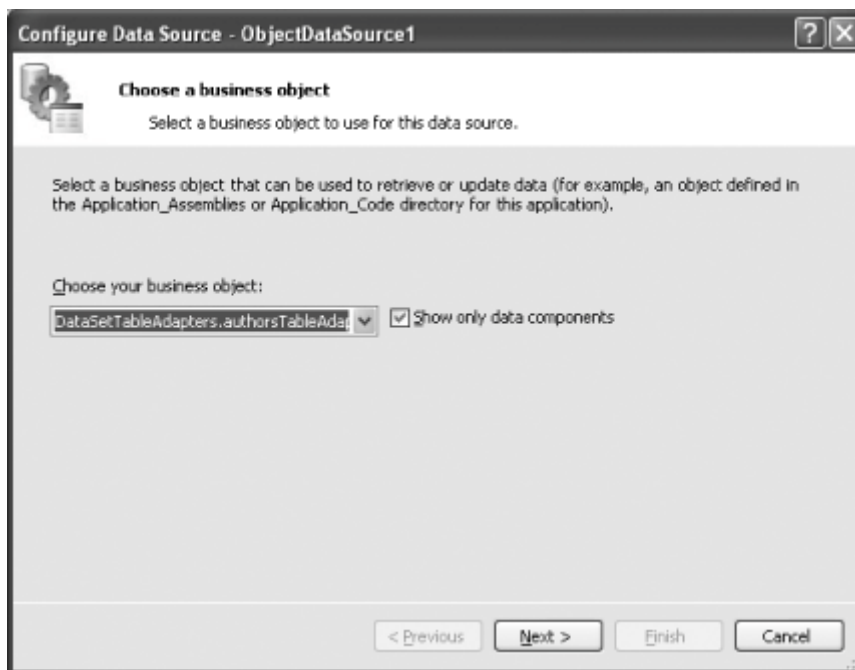
10. پروژه را ذخیره کنید . کنترل Gridview را جهت استفاده از منبع اطلاعاتی جدید پیکربندی کنید . (شکل ۴-۵۹)



شکل ۴-۵۹- پیکربندی Gridview جهت استفاده از منبع اطلاعاتی جدید

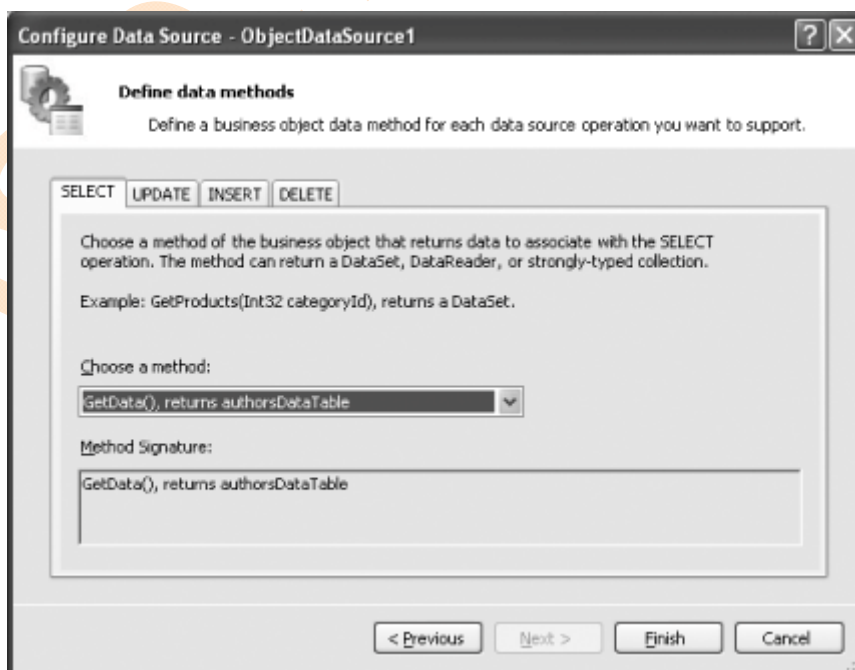
11. در پنجره بعدی object را با ID پیش فرض **ObjectDataSource1** به عنوان data source انتخاب و ok کنید .

12. در قسمت "choose your business object" گزینه `DataSetTableAdapter.authorsTableAdapter` را انتخاب کنید. (شکل ۴-۶۰) این `table adapter` براساس `Dataset` اضافه شده در قبل (`Dataset.Xsd`) ایجاد شده است. گزینه `Next` را کلیک کنید.



شکل ۴-۶۰- انتخاب business object

13. هر یک از برگه های `SELECT`، `UPDATE`، `INSERT` و `DELETE` از متد تعیین شده خود استفاده می کنند. (شکل ۴-۶۱)



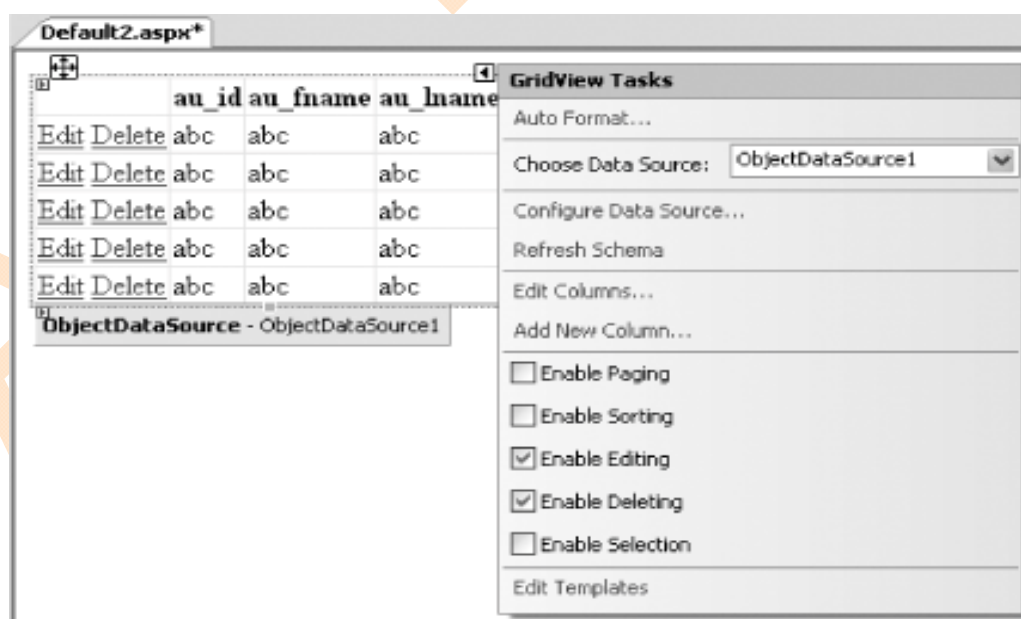
شکل ۴-۶۱- انتخاب متدهای مربوط به اعمال مختلف

14. نهایتاً در Gridview Tasks گزینه های Enable Editing و Enable Deleting انتخاب کنید .
(شکل ۴-۶۲)

نکته : بر اساس طراحی میکروسافت (Microsoft) می بایست خاصیت Readonly فیلد au_id از کنترل Gridview را False کنید . همانند زیر : (در Source View)

```
<asp:BoundField
    ReadOnly="false" HeaderText="au_id"
    DataField="au_id" SortExpression="au_id">
</asp:BoundField>
```

یادداشت : علت این است که بطور پیش فرض کنترل Gridview کلیدهای اصلی (primary keys) را غیر قابل ویرایش و به روز رسانی در نظر می گیرد . در حالیکه Data component به کلیدهای اصلی اجازه به روز رسانی می دهد. بنابراین کنترل Gridview مقادیر کلید اصلی را به Data component ارسال نمی کند و هنگامی که می خواهید Gridview را به روز رسانی (update) کنید خطایی مبنی بر اینکه کنترل object data source نمی تواند متد update با پارامترهای صحیح را پیدا کند نمایش داده می شود !



شکل ۴-۶۲. ویرایش و حذف در gridview

15- کلید F5 را فشار دهید هم اکنون می توانید رکوردها را در کنترل gridview ویراش و حذف نمایید.

۴-۸-۳. منابع اطلاعاتی :

جهت کسب اطلاع از نحوه دسترسی به داده ها از طریق ADO.NET مقاله زیر را مطالعه کنید:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbcon/html/vbconFundamentalDataConceptsInVisualStudioNET.asp>

چنانچه می خواهید با قابلیت های جدید ADO.NET 2.0 آشنا شوید به آدرس زیر مراجعه کنید :

<http://msdn.microsoft.com/data/DataAccess/Whidbey/default.aspx>

اگر با data components آشنایی کافی ندارید لینک زیر به شما کمک می کند تا آنها را بهتر بشناسید :

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbcon/html/vbconadonetdatasetcreationinvisualstudio.net.asp>

۴-۹- اتصال به اسناد XML :

یادداشت : با استفاده از کنترل *XmlDataSource* می توانید به آسانی به فایل های XML دسترسی داشته باشید.

همانند کنترل SQL Data source ؛ کنترل جدید *XmlDataSource* امکان کار با اسناد Xml را بدون نوشتن کد برایتان فراهم آورده است خصوصا این کنترل برای مقید ساختن کنترلهایی مانند Data list و Treeview بسیار مناسب است.

۴-۹-۱. مراحل انجام کار :

در این بخش یک Rss Reader جهت خواندن فایل های RSS طراحی می کنیم سپس با استفاده از کنترلهای Data list و XML Data source محتویات Rss را در یک فرم نمایش می دهیم .

VS 2005.1 را باز کنید و پروژه جدیدی با نام *C:\ASPNET20\chap04-XMLDataSource* ایجاد کنید.

2. بر روی نام پروژه در Solution Explorer راست کلیک و گزینه های Add New Item... سپس XML File را انتخاب کنید و نام آنرا Rss.Xml قرار دهید.

3. فایل Rss.Xml را باز کنید و کد مثال ۴.۱۱ را به آن اضافه کنید.

```
<?xml version="1.0"?>
```

```
<rss version="2.0">
```

```
<channel>
```

```
<title>Liftoff News</title>
```

```
<link>http://liftoff.msfc.nasa.gov/</link>
```

```
<description>Liftoff to Space Exploration.</description>
```

```
<language>en-us</language>
```

```
<pubDate>Tue, 10 Jun 2003 04:00:00 GMT</pubDate>
```

```
<lastBuildDate>Tue, 10 Jun 2003 09:41:01 GMT</lastBuildDate>
```

```
<docs>http://blogs.law.harvard.edu/tech/rss</docs>
```

```
<generator>Weblog Editor 2.0</generator>
```

```
<managingEditor>editor@example.com</managingEditor>
```

```
<webMaster>webmaster@example.com</webMaster>
```

```
<item>
```

```
<title>Star City</title>
```

```
<link>http://liftoff.msfc.nasa.gov/news/2003/news-starcity.asp</link>
```

```
<description>How do Americans get ready to work with Russians aboard the International Space Station? They take a crash course in culture, language and protocol at Russia's <a href="http://howe.iki.rssi.ru/GCTC/gctc_e.htm">Star City</a>.</description>
```

```
<pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>
```

```
<guid>http://liftoff.msfc.nasa.gov/2003/06/03.html#item573</guid>
```

```
</item>
```

```
<item>
```

```
<title>The Engine That Does More</title>
```

```
<link>http://liftoff.msfc.nasa.gov/news/2003/news-VASIMR.asp</link>
```

```
<description>Before man travels to Mars, NASA hopes to design new engines that will let us fly through the Solar System more quickly. The proposed VASIMR engine would do that.</description>
```

```
<pubDate>Tue, 27 May 2003 08:37:32 GMT</pubDate>
<guid>http://liftoff.msfc.nasa.gov/2003/05/27.html#item571</guid>
</item>
<item>
  <title>Astronauts' Dirty Laundry</title>
  <link>http://liftoff.msfc.nasa.gov/news/2003/news-laundry.asp</link>
  <description>Compared to earlier spacecraft, the International
Space Station has many luxuries, but laundry facilities are not one of them.
Instead,
astronauts have other options.</description>
  <pubDate>Tue, 20 May 2003 08:56:02 GMT</pubDate>
  <guid>http://liftoff.msfc.nasa.gov/2003/05/20.html#item570</guid>
</item>
</channel>
</rss>
```

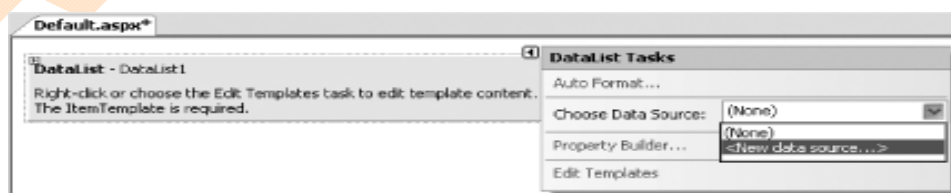
مثال ۱۱.۴ یک فایل RSS نمونه

نکته: می توانید از RSS دلخواهتان نیز استفاده کنید. در اینجا چند نمونه RSS آورده شده است:

- http://www.cnet.com/4520-6022_1-5115040-1.html?tag=cnetfd.ft
- <http://msdn.microsoft.com/aboutmsdn/rss/>

4. یک کنترل DataList (از برگه Data در Toolbox) به فرم اضافه کنید.

5. در پنجره DataList Tasks از کنترل DataList گزینه <New data source . . .> را انتخاب کنید و Ok کنید. (شکل ۴-۶۳ را ببینید)



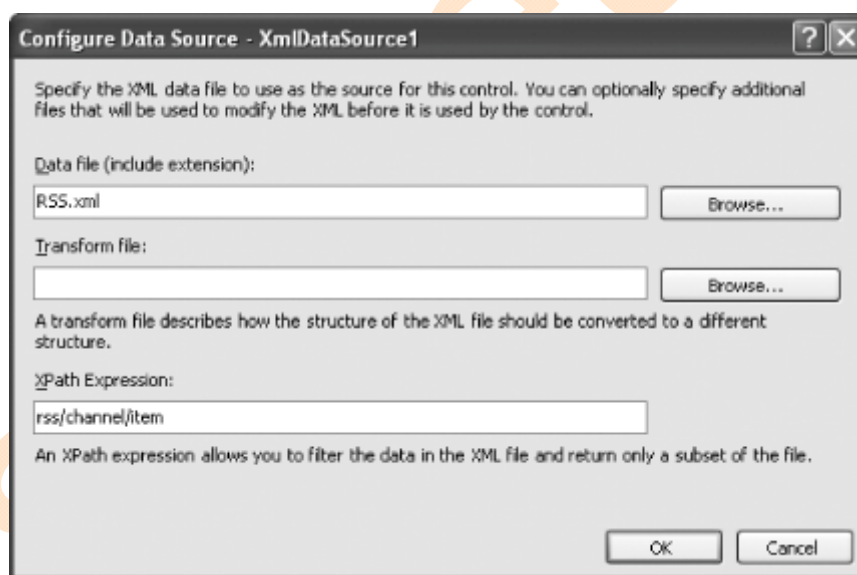
شکل ۴-۶۳. ساخت منبع اطلاعاتی جدید برای کنترل DataList

6. گزینه Xml File با نام پیش فرض XmlDataSource1 را انتخاب و Ok کنید. (شکل ۴-۶۴)



شکل ۴-۶۴. انتخاب فایل Xml بعنوان منبع اطلاعاتی

7. در قسمت DataFile نام فایل (Rss.Xml) و در قسمت XPath Expression عبارت rss/channel/item را وارد و نهایتا OK کنید. (شکل ۴-۶۵)



شکل ۴-۶۵. پیکر بندی XmlDataSource

نکته: گزینه XPath Expression بخشهایی از فایل xml را که به آنها نیاز دارید نمایش می دهد. در این مثال می خواهیم عناصر <item> را نمایش دهیم پس عبارت rss/channel/item را وارد کردیم.

8- در source view فرم کد زیر را اضافه کنید، این کد کنترل Data list را بصورتی پیکر بندی می کند که بخشهای مناسب فایل Xml را نمایش دهد:

```

<asp:DataList ID="DataList1" runat="server"
DataSourceID="XmlDataSource1">
  <itemtemplate>
    <b><XPath("title")%> </b><br />
    <i><XPath("description")
%></i>&nbsp;<XPath("pubDate")%><br />
    <a href='<XPath("link") %>'>Link</a><br />
  </itemtemplate>
</asp:DataList>

```

9- از طریق لینک Auto format... در Data list Tasks شمای sand & sky را به Data list اعمال کنید.

10- کلید F5 را فشار دهید. سند XML با قالبی که در شکل ۴-۶۶ مشاهده می کنید نمایش داده می شود. چنانچه بر روی Link کلیک کنید می توانید منبع خبر را مشاهده کنید.



شکل ۴-۶۶ نمایش یک وبلاگ (weblog) بدون نوشتن کد

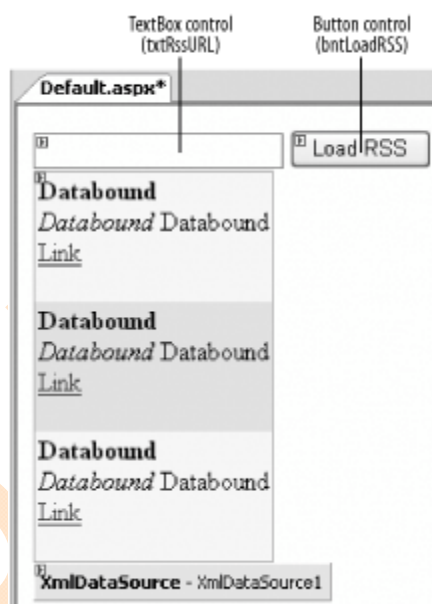
نکته: وبلاگ یا تارنما یادداشتهای روزانه افراد (هر فرد از جامعه) هستند که برای نمایش همگانی در اینترنت نوشته می شوند. سایتهای زیادی وجود دارند که امکان وبلاگ نویسی را فراهم می آورند از جمله PersianBlog(www.persianblog.com), Blogfa(www.blogfa.com), و ...

پاورقی بخش ...

... بارگذاری Rss از وب

بجای بارگذاری سند Xml از حافظه محلی می توانید با استفاده از خاصیت DataFile یک فایل Xml را از وب بارگذاری کنید.

1- به پروژه یک text Box و یک Button همانند شکل ۴-۶۷ اضافه کنید.

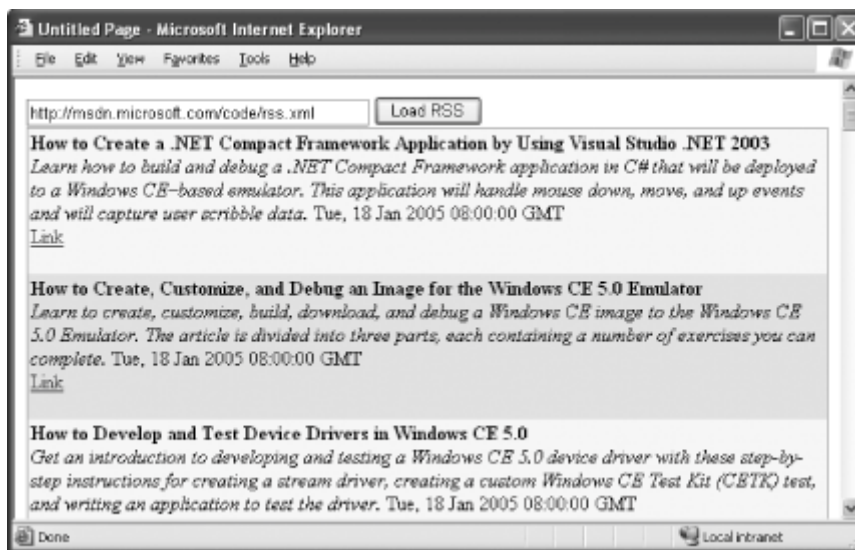


شکل ۴-۶۷. افزودن text Box و Button به فرم

2- در رویداد Click دکمه load Rss خاصیت DataFile از کنترل XmlDataSource را برابر با محتویات text Box قرار دهید :

```
Protected Sub btnLoadRSS_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles btnLoadRSS.Click
    XmlDataSource1.DataFile = txtRssURL.Text
End Sub
```

3- کلید F5 را فشار دهید و ادرس مربوط به یک Rss را در Text Box وارد کنید. چنانچه چنانچه کلمه load Rss را کلیک کنید محتویات سند Rss بازایی و در Data list نمایش داده می شوند. (شکل ۴-۶۸ را ببینید)



شکل ۴-۶۸. XmlDataSource از فایل XML

۴-۹-۳. منابع اطلاعاتی :

چنانچه هنوز با مفاهیم Rss و weblog مشکل دارید مقاله زیر را مطالعه کنید:

http://www.oreillynet.com/pub/a/javascript/2002/12/30/weblog_intro.html

برای آگاهی بیشتر در مورد XmlDataSource و DataList در MSDN عناوین "XmlDataSource Members" و "DataList Members" را جستجو کنید.

فصل پنجم

امنیت

(Security)

کنترل‌های حفاظتی جدید اضافه شده در ASP.NET 2.0 طراحی و برنامه نویسی را برای برنامه نویسان وب بسیار آسان نموده اند، با استفاده از کنترل‌های حفاظتی جدید می توانید اعمال ورود (login)، ثبت نام (register) تغییر کلمه عبور (password change) بدون افزودن کنترل‌های مختلف به فرمتان اضافه کنید.

۵-۱- ساخت صفحه ورود (login) با استفاده از کنترل‌های حفاظتی جدید

در ASP.NET 1.x می توانید با استفاده از تایید هویت مبنی بر فرم (form-based authentication) کاربران را در یک صفحه سفارشی شناسایی و تایید هویت کنید. گرچه این تکنیک روش مفید و کارایی است اما نیازمند کد نویسی جهت شناسایی کاربران است که اطلاعات مربوط آنها اغلب در منابعی همچون SQL Server ذخیره می شوند. اما در ASP.NET 2.0 با استفاده از کنترل‌های حفاظتی جدید شناسایی و تایید هویت کاربران بسیار آسانتر شده است.

۵-۱-۱- مراحل انجام کار:

در این بخش برای آشنایی با کنترل‌های جدید ASP.NET 2.0 صفحه ورودی مخصوص کاربران طراحی خواهیم کرد. در این صفحه با استفاده از کنترل‌های `Login`, `LoginView`, `LoginStatus` و `LoginName` اعمالی همچون شناسایی کاربران، تعیین وضعیت ورود کاربر، نام کاربری و... را شبیه سازی می کنیم.

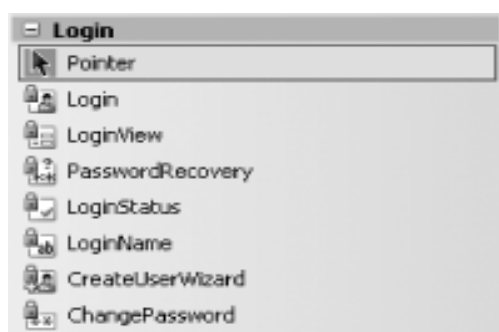
برای این منظور دو فرم طراحی می کنیم: یکی برای نمایش وضعیت ورود (login status) کاربر و دیگری برای ورود کاربر به سایت (login).

یادداشت: با استفاده از انواع مختلف کنترل‌های login به آسانی کاربران را شناسایی کنید.

فرمهای طراحی شده در این بخش پایه و اساس بخش بعد (بخش ۵-۲) هستند که با افزودن کاربران به سایت عملکرد کنترل‌های حفاظتی را نمایش می دهند.

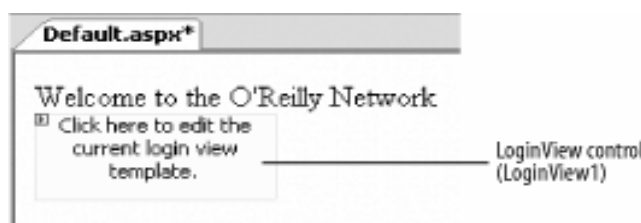
1- در VS 2005 وب سایت جدیدی با نام `C:\ASPNET20\chap-5-SecurityControls` ایجاد کنید.

2- کنترل‌های حفاظتی مورد استفاده در این فصل را در برگه Login از، ToolBox می توانید پیدا کنید. (شکل ۵-۱)



شکل ۵-۱. کنترل‌های حفاظتی در ASP.NET 2.0

3- همانند شکل ۵-۲ کنترل‌های مورد نیاز را به فرم اضافه کنید. کنترل login view در واقع مکانی است جهت نمایش اطلاعاتی در مورد کاربر. (بسته به اینکه کاربر به سایت وارد شده باشد یا خیر) از این صفحه جهت نمایش وضعیت کاربر استفاده می‌کنیم.

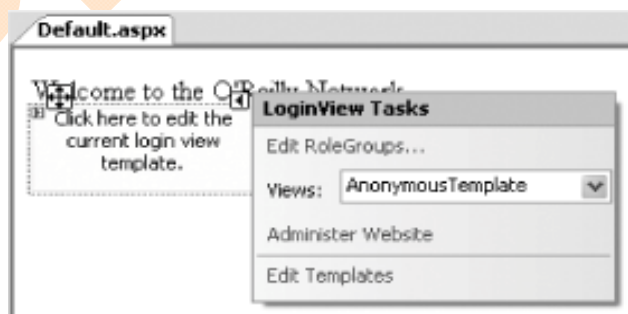


شکل ۵-۲. کنترل login view

4- برای پیکر بندی کنترل LoginView به LoginView Tasks بروید و در قسمت Views گزینه AnonymousTemplate را انتخاب کنید. متن زیر را کنترل LoginView تایپ کنید:

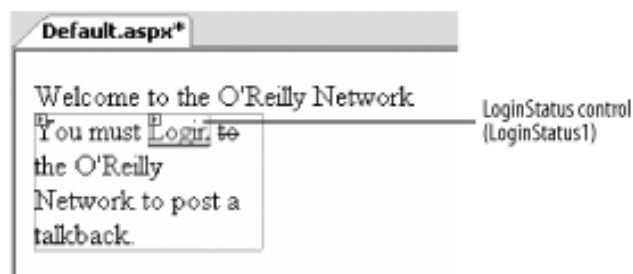
"You must Login to the O'Reilly Network to post a talkback."

هنگامیکه کاربر ناشناس (anonymous) از صفحه Default.aspx دیدن می‌کند متن فوق در login view نمایش داده می‌شود.



شکل ۵-۳ پیکر بندی LoginView برای حالت ناشناس

5- همانند شکل ۴-۵ یک کنترل Login Status به LoginView اضافه کنید. کنترل LoginStatus لینکی را نشان می دهد که قبل از ورود کاربر "Login" و پس از ورود کاربر "Logout" خواهد بود. در واقع از طریق این لینک کاربر می تواند به سایت وارد شود و یا اینکه از سایت خارج شود.(logout)



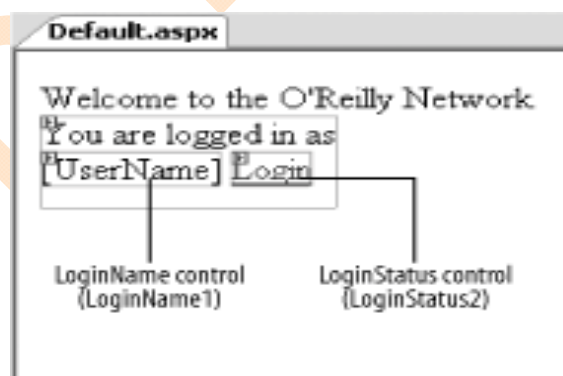
شکل ۴-۵. استفاده از کنترل LoginStatus

6- برای پیکربندی کنترل LoginView در حالتی که کاربر به سایت وارد شده است در قسمت view از LoginView Tasks (شکل ۳-۵) گزینه LoggedInTemplate را انتخاب کنید و متن زیر را در LoginView وارد کنید:

"You are logged in as."

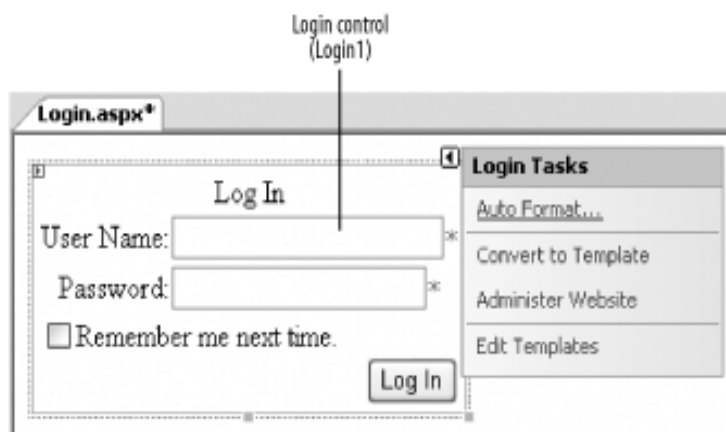
این متن برای کاربر شناسایی شده (authenticated) نمایش داده می شود.

7- یک کنترل LoginName و یک LoginStatus دیگر به LoginView در صفحه Default.aspx اضافه کنید (شکل ۵-۵) هنگامیکه کاربر به سایت وارد می شود نام کاربری که با آن به سایت وارد شده است توسط LoginName نمایش داده می شود همچنین کاربر از طریق کنترل LoginStatus می تواند از سایت خارج شود.



شکل ۵-۵. استفاده از کنترل LoginName جهت نمایش نام کاربر

8- یک فرم جدید به پروژه اضافه کنید و نام آنرا *Login.aspx* قرار دهید. این صفحه به کاربران اجازه ورود به سایت را می دهد. یک کنترل login به صفحه اضافه کنید. (شکل ۵-۶) در واقع کنترل Login واسط بین کاربر و سایت است.



شکل ۵-۶. افزودن کنترل login به فرم

صفحه ورود (Login Page) پیش فرض در ASP.NET 2.0

در Asp.Net 2.0 بطور پیش فرض صفحه *Login.aspx* صفحه ورود است. (این نام از طریق فایل *machine.config.comments* تعریف شده است)

اما شما می توانید در فایل web config صفحه ورود با نام دیگری تعریف کنید. کد زیر صفحه ورود *Login.aspx* را به *Authenticate.aspx* تغییر می دهد:

```
<system.web>
  <authentication mode="Forms">
    <forms name=".ASPXAUTH"
      loginUrl="Authenticate.aspx"
      protection="Validation"
      timeout="999999" />
  </authentication>
```

...

خواص عنصر <forms> به شرح زیر میباشد:

name

نام کوکی مورد استفاده جهت شناسایی را مشخص می کند.

loginUrl

نام صفحه login را مشخص می کند.

protection

نوع رمز گذاری کوکی ها را مشخص می کند .

timeout

طول عمر کوکی را مشخص می کند، (بر حسب دقیقه) کوکی پس از این مدت از بین می رود.

توجه داشته باشید که شناسایی مبتنی بر فرمها در ASP.NET هر نوع رمز گذاری را انجام نمی دهد و هر صفحه ورود بایستی توسط رمز گذاری SSL محافظت شود.

نکته: برای اینکه بدانید چگونه SSL را در IIS پیکر بندی به آدرس زیر مراجعه کنید:

http://www.microsoft.com/resources/documentation/iis/6/all/proddocs/en-us/nntp_sec_securing_connections.msp

9. شمای Elegant را از طریق لینک Auto format... در پنجره login tasks به کنترل login اعمال کنید
کنترل login هم اکنون شبیه شکل ۵-۷ شده است.



شکل ۵-۷ اعمال شما به کنترل login

10- یک فایل *Web.config* به پروژه اضافه کنید. (بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه Add New Item... سپس Web Configuration File را انتخاب کنید)

11- در فایل *Web.config* حالت authentication را از Windows به Forms تغییر دهید. با استفاده از اعتبار سنجی مبتنی بر فرم می توانید کاربران را به سایت اضافه کنید بدون اینکه نیازی به ساخت حساب کاربری در ویندوز باشد:

```
<system.web>
```

```
<authentication mode="Forms"/>
```

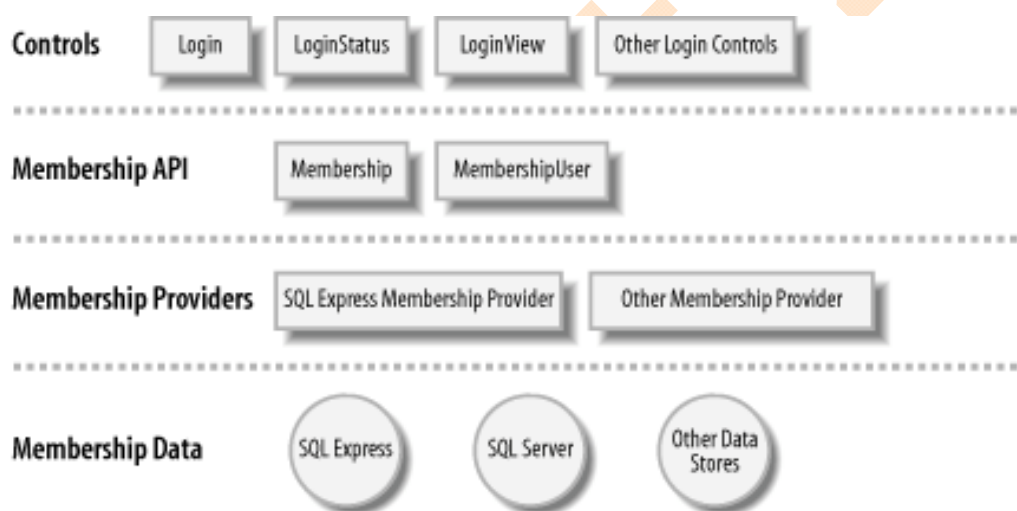
12- هم اکنون می توانید پروژه را تست کنید اما قبل از آن باید کاربری را برای ورود به سایت تعریف کنید این کار را در بخش بعد یاد خواهید گرفت .

نکته: بایستی خاصیت `DestinationPageUrl` کنترل `Login` را برابر با آدرس یکی از صفحات سایت قرار دهید تا پس از ورود کاربر، وی مستقیماً به این صفحه هدایت شود. (رفتن به آدرس صفحه مورد نظر منوط به بارگذاری مستقیم صفحه `login.aspx` است)

۵-۱-۲. تشریح عملکرد برنامه:

ASP.NET 2.0 از مدل حفاظتی جدیدی به نام مدل سرویس دهنده عضویتی `Membership Provider Model` استفاده می کند، این مدل سرویس دهنده بیشترین انعطاف و سادگی را جهت افزودن قابلیت امنیتی به برنامه ها را برای برنامه نویسان فراهم آورده است.

بعنوان نمونه گوشه ای از قابلیت این مدل مجموعه کنترلهایی است که در این بخش با آنها آشنا شدید. کنترلها (controls)، API ها و سرویس دهنده های تشکیل دهنده این مدل هستند. (شکل ۵-۸)



شکل ۵-۸. Membership Provider Model

در بالاترین سطح کنترلهای مختلف از جمله `login`، `loginstatus` و `loginview` قرار دارند در لایه بعدی API ها هستند که اعمال لازم جهت اجرای قابلیت های کنترلها را فراهم می کنند کلاس `membership` اعمال افزودن و حذف کاربران را برعهده دارد در حالیکه کلاس `membership user` مسئول مدیریت اطلاعات کاربر از جمله کلمه عبور `password`، سئوالات امنیتی کلمه عبور `passwords questions` و ... می باشد. `Membership API` ها از `membership provider` ها جهت نمایش و ذخیره اطلاعات در منابع اطلاعاتی استفاده می کنند. VS 2005 از `membership provider` پیش فرض با نام `SQL Express 2005 Membership Provider` استفاده می کند در واقع `membership provider` نقش پل میان `member API` و منبع ذخیره اطلاعات (`data stores`) را ایفا می کند بطوریکه برنامه نویس را قادر می سازند بدون نوشتن کدهای اضافه به داده ها دسترسی باشد و با آنها کار کند.

چنانچه سرویس دهنده هایی که توسط میکروسافت فراهم شده اند نیازهای شما را برآورده نمی کنند می توانید سرویس دهنده بخصوص خود را بنویسید مثلاً اگر می خواهید اطلاعات کاربران را بجای ذخیره در پایگاه داده رابطه ای (نظیر SQL sever) در یک سند Xml ذخیره کنید می توانید سرویس دهنده ای جهت دسترسی و ارتباط با فایل xml بنویسید.

۵-۱-۳. پاورقی بخش ...

... تایید هویت کاربران بدون استفاده از کوکی ها

شناسایی مبتنی بر فرمها در ASP.NET 1.x از یک HTTP کوکی جهت رمزگذاری نام کاربری (username) استفاده می کند.

در ASP.NET 2.0 حالت جدیدی از شناسایی اضافه شده است به نام: شناسایی مبتنی بر فرم بدون کوکی. (cookieless forms authentication) برای استفاده از این حالت تنها کافی است خاصیت Cookieless را به فایل web.config اضافه کنید:

```
<authentication mode="Forms">
  <forms name=".ASPXAUTH"
    loginUrl="Login.aspx"
    protection="Validation"
    timeout="999999"
    cookieless="UseUri"/>
</authentication>
```

خاصیت Cookie less چهار حالت را پشتیبانی می کند:

AutoDetect

این حالت بطور خودکار چک می کند که آیا مرورگر از کوکی ها پشتیبانی می کند یا خیر؟

چنانچه مرورگر کوکی ها را پشتیبانی کند ASP.NET نیز از کوکی ها برای شناسایی کاربران بهره می گیرد.

UseCookies

از کوکی ها جهت شناسایی استفاده می کند. (حالت پیش فرض)

UseDeviceProfile

در این حالت استفاده از یا عدم استفاده از کوکی ها به تنظیمات فایل *machine.config* وابسته است.

UseUri

استفاده از حالت Cookie less

یادداشت : چنانچه از Cookie less استفاده کنید یک رشته بزرگ متنی در آدرس URL شما جاسازی می شود .

... پیکر بندی login view در source View صفحه

اگر تمایلی به استفاده از wizard کنترل login view ندارید می توانید در source view صفحه خواص مورد نیاز تان را جهت پیکر بندی این کنترل وارد کنید. کد زیر مربوط به کنترل login view ساخته شده در همین بخش را نشان می دهد :

```
<asp:LoginView ID="LoginView1" runat="server">
  <LoggedInTemplate>
    You are logged in as
    <asp:LoginName ID="LoginName1" runat="server" />
    <asp:LoginStatus ID="LoginStatus2" runat="server" />
    &nbsp;
  </LoggedInTemplate>
  <AnonymousTemplate>
    You must&nbsp;<asp:LoginStatus ID="LoginStatus1"
    runat="server" />&nbsp;to the O'Reilly Network to
    post a talkback.
  </AnonymousTemplate>
</asp:LoginView>
```

۵-۱-۴ منابع اطلاعاتی :

برای کسب اطلاعات بیشتر راجع به پارامترهای عنصر <forms> در Web.config در MSDN عبارت <forms> Element را جستجو کنید .

برای آشنایی بیشتر با قابلیت های Membership در ASP.NET 2.0 در MSDN موضوع "Introduction to Membership" را جستجو کنید.

۵-۲- افزودن کاربر توسط WAT

هنگامیکه شناسایی کاربران را در سایت ایجاد می کنید نیازمند راهی برای ایجاد و افزودن حساب کاربری هستید در ASP.NET این کار را می توانید از طریق ابزار مدیریت وب سایت (Web Site Administration Tool) معروف به WAT انجام دهید. حساب کاربران بطور پیش فرض درون یک پایگاه داده SQL Server 2005 Express (این فایل در شاخه App-Data پروژه قرار دارد) که توسط Membership Provider پیش فرض مشخص شده است ذخیره می شوند. نیازی هم به ساخت پایگاه داده مخصوص برای ذخیره اطلاعات کاربران در *Web.config* (که نا امن هم هست) ندارید.

نکته: از این پس بجای ASP.NET Web Application Administration Tool از مخفف آن یعنی Asp.Net WAT استفاده می کنیم.

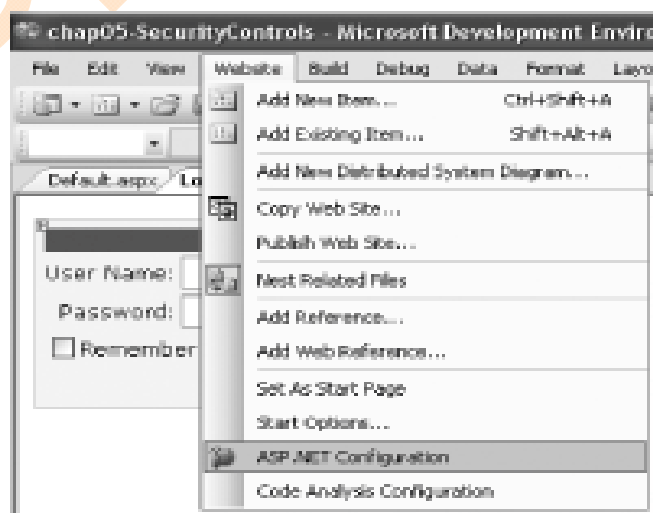
یادداشت: Asp.Net WAT امکان ایجاد حساب کاربری جدید و ذخیره اطلاعات در پایگاه داده را فراهم می کند.

۵-۲-۱- مراحل انجام کار:

در این بخش می خواهیم با استفاده از WAT ASP.NET حساب کاربری جدیدی به برنامه اضافه کنیم. حسابهای کاربران در پایگاه داده مشخص شده توسط Membership Provider ذخیره می شود. بطور پیش فرض WAT کاربران جدید را به یک پایگاه داده SQL 2005 Express اضافه می کند

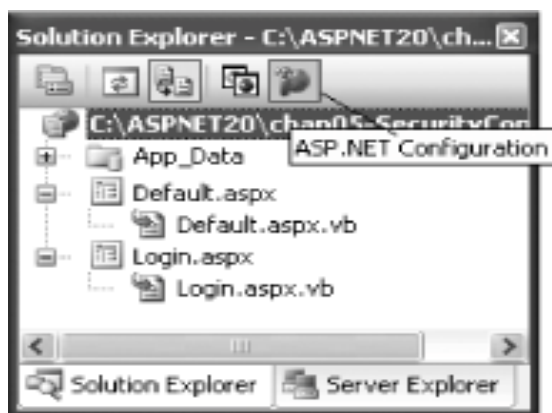
1- پروژه ای در بخش قبل ایجاد کردیم باز کنید. (*C:\ASPNET20\chap-5-SecurityControls*)

2- برای افزودن کاربر جدید از WAT استفاده کنید. (ASP.NET Configuration → Website) شکل ۵-۹ را ببینید.



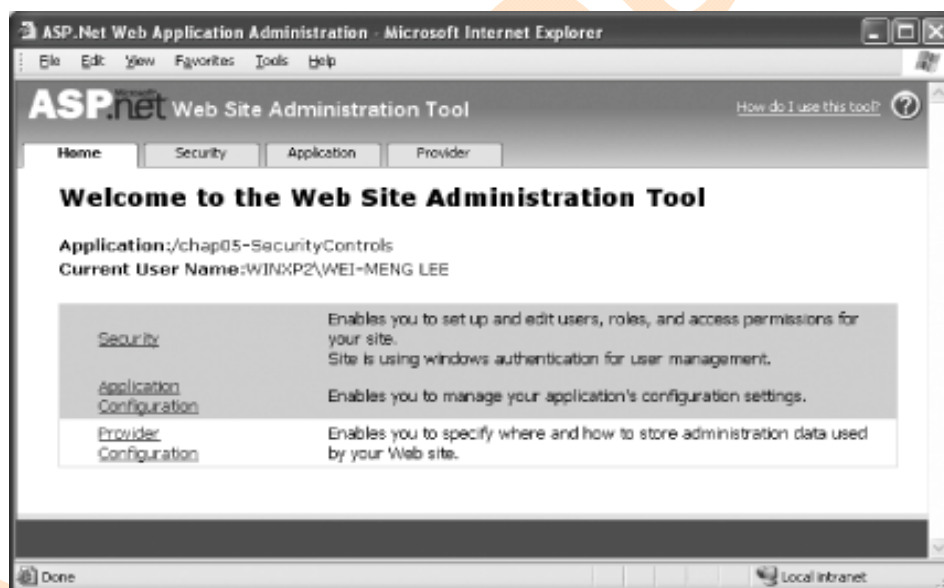
شکل ۵-۹ استفاده از ابزار ASP.NET Configuration

همچنین می توانید با کلیک بر روی آیکن این ابزار در solution Explorer آنرا اجرا کنید. (شکل ۵-۱۰ را ببینید)



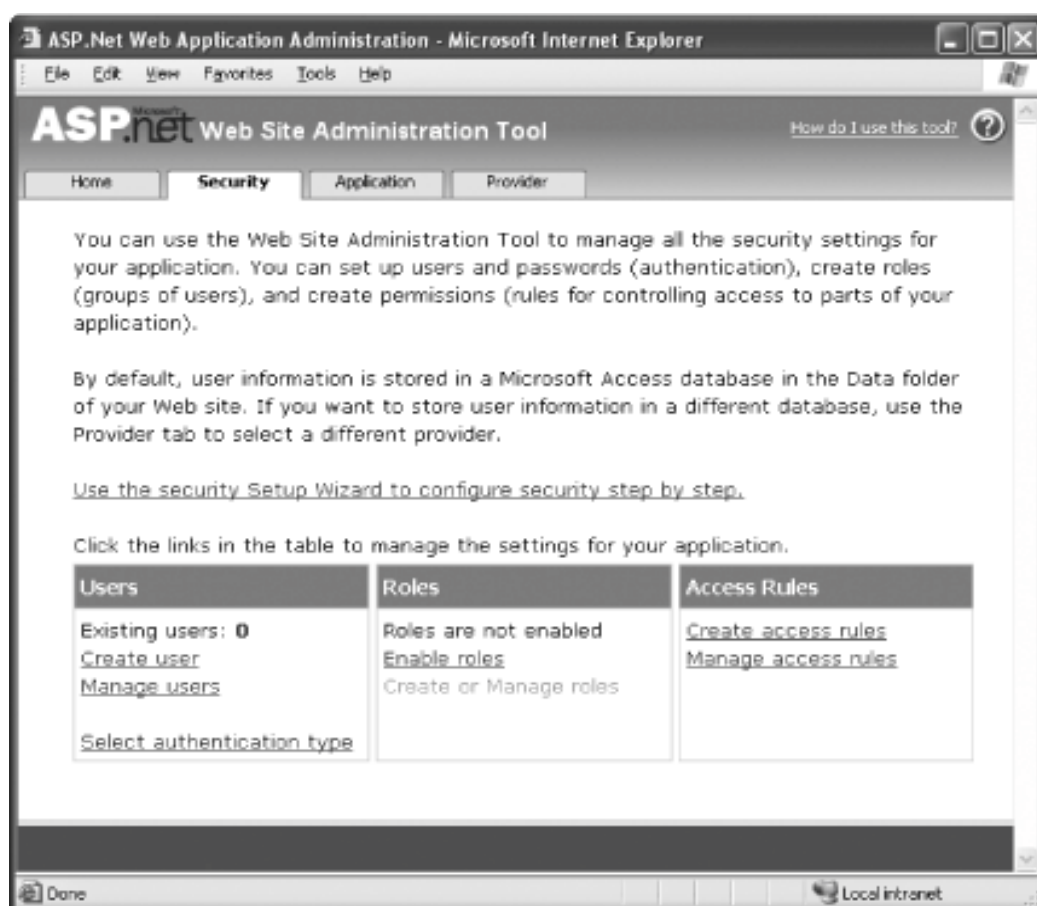
شکل ۵-۱۰. اجرای ASP.NET Configuration از solution Explorer

4- برای افزودن کاربر جدید برگه security را کلیک کنید. (شکل ۵-۱۱)



شکل ۵-۱۱. ابزار ASP.NET Web Site Configuration Tool

5- در قسمت Users لینک "Create user" را کلیک کنید. (شکل ۵-۱۲)



شکل ۵-۱۲ ایجاد کاربر جدید

6- اطلاعات مورد نیاز را وارد و Create user را کلیک کنید. (شکل ۵-۱۳)

Create User	Roles
<p>Sign Up for Your New Account</p> <p>User Name: <input type="text" value="wml"/></p> <p>Password: <input type="password" value="*****"/></p> <p>Confirm Password: <input type="password" value="*****"/></p> <p>E-mail: <input type="text" value="wei_meng_lee@hotmail.com"/></p> <p>Security Question: <input type="text" value="what is your favorite movie?"/></p> <p>Security Answer: <input type="text" value="Black Hawk Down"/></p> <p><input type="button" value="Create User"/></p> <p><input checked="" type="checkbox"/> Active User</p>	<p>Roles are not enabled.</p>

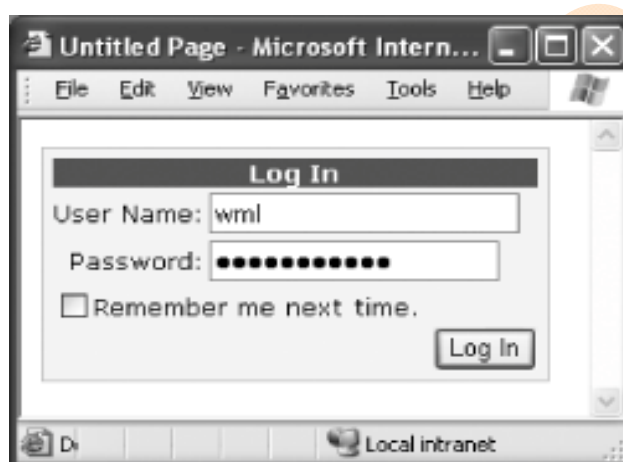
شکل ۵-۱۳. ورود اطلاعات کاربر جدید

7- نهایتاً در solution Explorer صفحه Default.aspx را انتخاب و کلید F5 را فشار دهید (شکل ۵-۱۴)



شکل ۵-۱۴. اجرای برنامه قبل از ورود کاربر

8- بر روی login کلیک کنید و در صفحه login اطلاعات مربوط به کاربر ساخته شده را وارد کنید (شکل ۵-۱۵)



شکل ۵-۱۵. ورود کاربر

9- چنانچه اطلاعات وارد شده صحیح باشند صفحه ای مشابه شکل ۵-۱۶ نمایش داده می شود.



شکل ۵-۱۶. کاربر شناسایی شده

۵-۲-۲- تشریح عملکرد برنامه :

هنگامیکه کاربر جدیدی به برنامه اضافه می کنید اطلاعات مربوط به کاربر در دو جدول از پایگاه داده SQL 2005 Express ذخیره می شوند (این پایگاه داده در شاخه *App_Data* پروژه ذخیره می شود) :

- *aspnet_Membership*
- *aspnet_Users*

نکته : جدول *aspnet_Membership* جزئیات اطلاعات کاربر را نگهداری می کند (مانند کلمه عبور (*password*) (پست الکترونیکی (*email*) (آخرین ورود و...) در حالیکه جدول *aspnet_Users* اطلاعات برنامه ای که کاربر به آن تعلق دارد را نگه می دارد .

جدول های ۵-۱ و ۵-۲ مقادیر مربوط به کاربر ایجاد شده را نشان می دهند:

Field	Value
ApplicationId	{ B6677867-19E8-409B-B5CD-9D584B5F5647 }
UserId	{ 03D7299A-2C80-4A3C-B6B7-818369526BEE }
Password	TyPAwmdjkASXxZAn+Kxy3U59qQ0=
PasswordFormat	1
PasswordSalt	ZsmfGb76Jflh2+xmxcyXcA==
MobilePIN	
Email	wei_meng_lee@hotmail.com
LoweredEmail	wei_meng_lee@hotmail.com
PasswordQuestion	What is your favorite movie?
PasswordAnswer	Black Hawk Down
IsApproved	1
IsLockedOut	0
CreateDate	2/5/2005 6:58:05 PM
LastLoginDate	2/5/2005 6:58:05 PM
LastPasswordChangedDate	2/5/2005 6:58:05 PM
LastLockoutDate	1/1/1753
FailedPasswordAttemptCount	0

Field	Value
FailedPasswordAttemptWindowStart	1/1/1753
FailedPasswordAnswerAttemptCount	0
FailedPasswordAnswerAttemptWindowStart	1/1/1753
Comment	

جدول ۵-۱- محتویات جدول aspnet_Membership

Field	Value
ApplicationId	{ B6677867-19E8-409B-B5CD-9D584B5F5647 }
UserId	{ 03D7299A-2C80-4A3C-B6B7-818369526BEE }
UserName	wml
LoweredUserName	wml
MobileAlias	
IsAnonymous	0
LastActivityDate	2/5/2005 6:58:05 PM

جدول ۵-۲- مقادیر موجود در جدول aspnet_Users

توجه کنید که کلمه عبور در جدول aspnet_Membership قبل از ذخیره به صورت *salt hashed* کد شده است.

در هم سازی کلمه عبور (Password Hashing)

روشهای مختلفی جهت ذخیره کلمه عبور کاربران در پایگاه داده وجود دارد ساده ترین روش ذخیره کلمه عبور بدون هرگونه تغییر است. این روش امنیت پایینی دارد.

روش بهتر رمزگذاری کلمه عبور است (با استفاده از کلید رمز و یا بهتر از آن بدون کلید). بهترین گزینه استفاده از یک تابع یکطرفه (one-way function) است بطوریکه به کلید نیاز نداشته باشد و بدین ترتیب خط کشف کلید توسط هکرها (heders) نیز وجود ندارد. درهم سازی (hashing) یک الگوریتم پنهانی است که یک رشته از ورودی گرفته و یک رشته درهم را بعنوان خروجی می دهد. این روش به کلید نیاز ندارد و احتمال تولید خروجی یکسان هم بسیار کم است.

به همین دلیل معمولاً مقادیر درهم (hashed value) کلمات عبور در پایگاه داده ذخیره می شوند. هنگامیکه کاربر به سایت وارد می شود کلمه عبور وارد شده hash شده و با مقدار موجود در پایگاه داده مقایسه می شود چنانچه این مقادیر یکسان باشند به کاربر اجازه دسترسی داده می شود.

یکی از معایب درهم سازی این است که یک هکر حرفه ای می تواند از یک لیست جهت حمله استفاده کند در واقع هکر می تواند لیستی از مقادیر درهم که معمولاً در کلمات عبور استفاده می شوند تهیه کند و آنها را با مقادیر ذخیره شده در پایگاه مقایسه کند و به این ترتیب کلمه عبور کاربر لو رود.

به منظور کاهش حملات و کاهش درسر لو رفتن کلمات عبور یک رشته تصادفی (random string) از مقادیر که به salt معروف است، برای هر کاربر تولید شده و قبل از درهم سازی کلمه عبور به آن اضافه می شود سپس مقدار حاصل به همراه salt در پایگاه ذخیره می شوند. مزیت این کار این است که هکر مجبور است دو لیست متفاوت تولید کند یکی برای کلمه عبور درهم شده یکی هم برای salt که این کار بسیار سخت و وقتگیر است. علاوه بر این استفاده از saled hash باعث می شود برای کاربران با کلمات عبور یکسان مقادیر متفاوتی در پایگاه ذخیره شود چرا که هر کاربر salt مخصوصی برای خود دارد که به صورت تصادفی ایجاد شده است.

۵-۳-۲. پاورقی بخش ...

... پیدا کردن صفحات دیده شده توسط کاربر شناسایی شده

با استفاده از متد () GetUser از کلاس Membership می توانید تشخیص دهید که آیا صفحه ای توسط یک کاربر دیده شده است یا خیر. بعنوان مثال فرض کنید می خواهید هنگام ورود کاربر به او خوش آمد گویی کنید برای این کار کافست در رویداد Form-load متد () GetUser را فراخوانی کنید چنانچه کاربر هنوز به سایت وارد نشده باشد در واقع صفحه توسط کاربر ناشناس بازدید شود. تابع () GetUser مقدار null بر می گرداند. کد زیر در صورت درخواست صفحه توسط کاربر شناسایی شده زمان فعلی سیستم را برای او نمایش می دهد:

```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load
    If Membership.GetUser IsNot Nothing Then
        Response.Write("Time is " & Now)
    End If
End Sub
```


... محدود کردن تعداد دفعات ورود کلمه عبور جهت جلوگیری از حمله هکرها

شما می توانید با استفاده از خواص `passwordAttemptThreshold` و `passwordAttemptWindow` تعداد دفعات ورود ناموفق کلمه عبور کاربران را کنترل نمایید. از این خواص در فایل `Web.config` استفاده کنید همانند مثال ۱-۵.

...

```
<membership defaultProvider="SqlProvider"
  userIsOnlineTimeWindow="20">
  <providers>
    <add name="SqlProvider"
      type="System.Web.Security.SqlMembershipProvider"
      connectionStringName="LocalSqlServer"
      requiresQuestionAndAnswer="true"
      passwordAttemptThreshold="5"
      passwordAttemptWindow="30"
      applicationName="MyApplication" />
  </providers>
</membership>
```

مثال ۱-۵ - محدود کردن تعداد دفعات ورود کلمه عبور

نکته: رشته اتصال `LocalSqlServer` به پایگاه داده `SQL Server 2005 Express` اشاره می کند. این رشته در فایل `machine.config.comments` تعریف شده است.

مثال ۱-۵ به کاربر اجازه می دهد در طول ۳ دقیقه ۵ بار به سایت `login` کند چنانچه کاربر از این ارقام تجاوز کند حسابش قفل می شود.

۱-۲-۴- منابع اطلاعاتی :

برای کسب اطلاع از چگونگی استفاده از `salted hashes` در برنامه های دات نت مقاله زیر را مطالعه کنید:

<http://msdn.microsoft.com/msdnmag/issues/03/08/SecurityBriefs/>

برای آشنایی بیشتر با خواص passwordAttemptThreshold و passwordAttemptWindow در MSDN به موضوعات "Membership.passwordAttemptThreshold property" و "Membership.passwordAttemptWindow property" مراجعه کنید.

۵-۳- جلوكيری از دسترسی غير مجاز به صفحات

یادداشت: کنترل‌های login در صورتی مفید هستند که بتوانند دسترسی به صفحات را کنترل نمایند.

تا اینجا یاد گرفتید چگونه صفحه ورود برای سایت طراحی کنید و کاربر جدید به سایت اضافه کنید برای اطمینان از شناسایی کاربران قبل از اینکه به آنها اجازه دسترسی به قسمت‌های مختلف سایت را بدهید، بایستی Asp.Net را بگونه ای پیکربندی کنید که کاربران را وادار به ورود کنید

۵-۳-۱- مراحل انجام کار:

در بخش ۵-۱ آموختید که چگونه با استفاده از کنترل login ورود کاربران به سایت را مدیریت کنید در این بخش می آموزید که چگونه بر اساس ورود کاربران از دسترسی آنها به برخی صفحات خاص جلوگیری کنید در ادامه شاخه جدیدی به پروژه اضافه می کنیم و با اصلاح فایل Web.config دسترسی به این شاخه را محدود می کنیم هنگامیکه صفحه ای از صفحات موجود در این شاخه درخواست می شود صفحه login بطور خود کار برای شناسایی کاربر لود می شود.

1- پروژه ایجاد شده در ابتدای فصل را باز کنید (C:\ASPNET20\chap-5-SecurityControls) شاخه جدیدی با نام Members به آن اضافه کنید بر روی نام پروژه Solution Explorer راست کلیک کنید و گزینه های Regular → Add Folder را انتخاب کنید.

2- یک فرم جدید با نام MemberDefault.aspx به شاخه Members اضافه کنید شاخه Members در Solution Explorer راست کلیک کنید و گزینه های Add New Item... و سپس Web Form را انتخاب کنید.

3- یک فایل Web.config نیز به شاخه Members اضافه کنید.(بر روی شاخه Members در Solution Explorer راست کلیک کنید و گزینه های Add New Item... و سپس Web Configuration File را انتخاب کنید) کدهای زیر را به آن اضافه کنید :

```

<!-- Remove this line
  <authentication mode="Windows" />
-->
<authorization>
  <deny users="?" />
</authorization>

```

4- عنصر deny از دسترسی به شاخه جاری (در اینجا شاخه member) جلوگیری می کند، همچنین با استفاده از عنصر allow می توانید به کاربران اجازه دسترسی به شاخه جاری را بدهید علامت سؤال (?) به کاربران ناشناس (anonymous) و تایید هویت نشده (nonauthenticated) اجازه دسترسی به صفحات شاخه members را نمی دهد در مقابل استفاده از علامت ستاره (*) باعث دسترسی تمامی کاربران به شاخه member می شوند.

5- اکنون Solution Explorer پروژه شبیه شکل ۵-۱۷ شده است.



شکل ۵-۱۷ Solution Explorer پروژه

6- صفحه memberdefault.aspx را انتخاب و کلید F5 را فشار دهید. چنانچه مشاهده می کنید به صفحه Login.aspx هدایت می شوید به این معنی که برای دسترسی به صفحه MemberDefault.aspx ابتدا باید به سایت وارد شوید. با شناسه کاربری که در بخش قبل ساختم به سایت وارد شوید. چنانچه در ورود به سایت مشکلی نداشته باشید صفحه MemberDefault.aspx نمایش داده می شود.

۵-۳-۲- پاروقی بخش ...

... استفاده از یک فایل *Web.config* جهت تعیین دسترسی ها برای کل سایت

بجای افزودن فایل *Web.config* مجزا برای هر شاخه می توانید از طریق عنصر `<location>` در فایل *web.config* ریشه (Root) دسترسی به قسمتهای مختلف را مشخص کنید قطعه کد زیر که در *Web.config* شاخه ریشه اضافه شده معادل مرحله ۳ است :

```
...
</system.web>

<location path="/Members">
  <system.web>
    <authorization>
      <deny users="?" />
    </authorization>
  </system.web>
</location>
</configuration>
```

با استفاده از این روش نیازی به *Web.config* مخصوص برای هر شاخه ندارید با استفاده از عناصر `<location>` می توانید دسترسی هر شاخه را مشخص کنید.

۵-۳-۳- منابع اطلاعاتی :

برای آشنایی بیشتر با عنصر `<location>` به MSDN مراجعه کنید.

۵-۴- بازیابی کلمات عبور کاربران

یادداشت : می توانید از طریق پست الکترونیکی کلمات عبور فراموش شده کاربران را برایشان ارسال کنید .

کاربران معمولاً در اکثر مواقع کلمات عبورشان را فراموش می کنند ! و شما به مکانیزمی جهت بازیابی کلمات عبور آنها نیاز دارید. یکی از معمولی ترین روشها ؛ دریافت پست الکترونیکی و ارسال کلمه عبور به آن است. در ASP.NET 2.0 این قابلیت توسط کنترل PasswordRecovery ارائه شده است.

این کنترل کاربران را در بازیابی کلمه عبورشان از طریق پست الکترونیکی کمک می کند.

بازیابی کلمه عبور (Password Recovery)

بازیابی کلمه عبور زمانی امکان پذیر است که کلمه عبور کاربران به همان صورت وارد شده در پایگاه داده ذخیره شده باشد و بازیابی کلمه عبوری که الگوریتمهای درهم سازی (و یا الگوریتم های مشابه) بر روی آن اعمال شده باشد بی معنی است! به هر حال بر اساس تنظیمات پیش فرض فایل *machine.config* کلمات عبور پس از درهم سازی در پایگاه داده ذخیره می شوند همچنین بازیابی کلمه عبور بصورت پیش فرض غیر فعال است.

برای ذخیره کلمات عبور به همان صورت وارد شده توسط کاربر کدهای زیر را به *web.config* اضافه کنید:

```
...
<system.web>
  <membership
    defaultProvider="SqlProvider"
    userIsOnlineTimeWindow="15">
    <providers>
      <clear />
      <add
        name="SqlProvider"
        type="System.Web.Security.
          SqlMembershipProvider"
        connectionStringName="LocalSqlServer"
        applicationName="chap05-SecurityControls"
        enablePasswordRetrieval="true"
        enablePasswordReset="true"
        requiresQuestionAndAnswer="true"
        requiresUniqueEmail="true"
        passwordFormat="Clear" />
      </providers>
    </membership>
  </system.web>
...
```

در واقع با این کد membership provider ها را پاک و *SqlMembershipProvider* جدیدی اضافه کردیم. توجه کنید برای بازیابی کلمات عبور بایستی خواص *enablePasswordRetrieval* و *passwordFormat* را به ترتیب با مقادیر *True* و *Clear* تنظیم کنید.

چنانچه *passwordFormat* را با مقدار *Hashed* تنظیم کنید *enablePasswordReset* باید *false* شود.

۵-۴-۱. مراحل انجام کار :

در این بخش از کنترل PasswordRecover جهت بازیابی کلمات عبور فراموش شده کاربران استفاده می کنیم. صفحه Default.aspx ابتدای فصل را گسترش می دهیم بگونه ای که کاربرانی که کلمات عبور شان را فراموش کرده اند از همین صفحه برای بازیابی کلمات عبور شان استفاده کنند.

1- پروژه بخش اول را باز کنید (C:\ASPNET20\chap-5-SecurityControls) ، یک کنترل PasswordRecovery به صفحه اضافه کنید. (شکل ۵-۱۸)



شکل ۵-۱۸ افزودن کنترل PasswordRecovery به صفحه Default.aspx

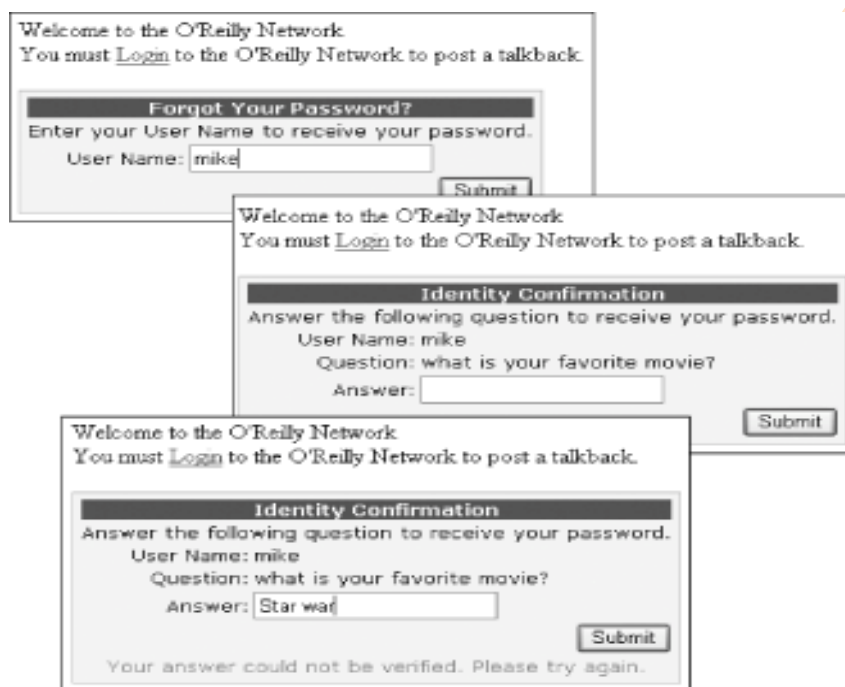
2- در پنجره properties کنترل PasswordRecovery به خواص from و subject مقدار بدهید. (شکل ۵-۱۹)



شکل ۵-۱۹. پیکر بندی کنترل PasswordRecovery

3-شمای Elegant را به PasswordRecovery اعمال کنید. (از طریق لینک Auto Format... در PasswordRecovery Tasks)

4- کلید F5 را فشار دهید. ابتدا باید نام کاربری user name را وارد کنید در قسمت بعد سوالی که هنگام ثبت نام وارد کردید از شما پرسیده می شود، چنانچه پاسخ شما با آنچه در پایگاه ذخیره شده یکسان باشد نامه ای به پست الکترونیک شما ارسال می شود که کلمه عبورتان را برایتان می فرستد.



شکل ۵-۲۰. بازیابی کلمه عبور فراموش شده

نکته : برای کار با پست الکترونیکی (ارسال و دریافت) بایستی سرویس SMTP (simple mail transfer protocol) بر روی سیستمتان پیکر بندی شده باشد برای پیکر بندی این سرویس به آدرس زیر مراجعه کنید :

ASP.NET Web Application Administration → Application → Configure SMTP

به دلایل امنیتی ارسال کلمه عبور از طریق پست الکترونیکی ایده مناسبی نیست از اینرو در استفاده از این گزینه قدری تامل کنید.

۵-۴-۲. پاورقی بخش...

...مقدار دهی مجدد (Resetting) کلمه عبور

در بخش ۵-۸ همین فصل به این مطلب خواهیم پرداخت.

۵-۴-۳- منابع اطلاعاتی :

برای آشنایی با خواص و متدهای مختلف کنترل PasswordRecovery در MSDN موضوع "PasswordRecovery Members" را جستجو کنید.

چنانچه می خواهید راجع به SMTP اطلاعات بیشتری کسب کنید به ادرس زیر مراجعه کنید :

<http://www.microsoft.com/technet/prodtechnol/exchange/guides/E2k3TransnRouting/1bee564f-a307-4b65-83f4-61c1b5dcc744.msp>

۵-۵- تغییر کلمه عبور کاربر

یادداشت : از طریق کنترل ChangePassword به کاربران اجازه دهید کلمه عبورشان را تغییر دهند .

تغییر کلمه عبور نیز یکی از معمولی ترین اعمال کاربران سایت است. ASP.NET 2.0 نیز توسط کنترل جدید ChangePassword این کار را آسان ساخته است .

۵-۵-۱. مراحل انجام کار :

در این بخش صفحه ای برای تغییر کلمه عبور کاربران طراحی می کنیم این صفحه جدید را به شاخه Members اضافه می کنیم ، بطوریکه کاربران تایید هویت شده بتوانند در صورت تمایل کلمه عبور شان تغییر دهند .

1- پروژه بخش قبل را باز کنید (C:\ASPNET20\chap-5-SecurityControls) یک کنترل ChangePassword به صفحه MemberDefault.aspx اضافه کنید.

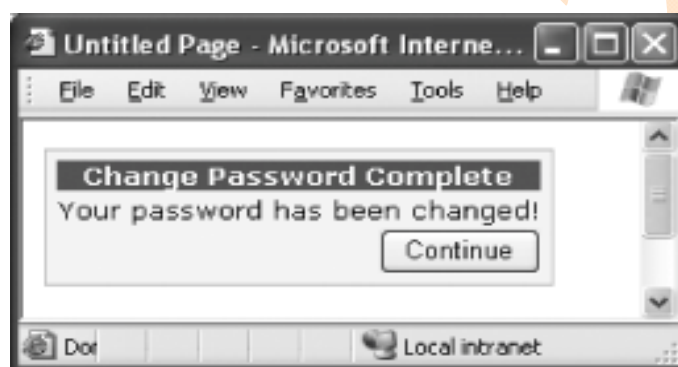
2- شمای Elegant را به ChangePassword اعمال کنید. (از طریق لینک Auto Format... در ChangePassword Tasks). اکنون کنترل طراحی شده شبیه شکل ۵-۲۱ شده است .

شکل ۵-۲۱ کنترل change password

3- کنترل change password را باید در صفحاتی استفاده کنید که توسط کاربران شناسایی شده درخواست می شوند همانند صفحه memberDefault.aspx در شاخه member .

4- کلید F5 را فشار دهید. ابتدا جهت شناسایی به صفحه ورود (login.aspx) هدایت می شوید ، از حساب کاربری user Account ساخته شده در بخش ۵-۲ جهت ورود با سایت استفاده کنید. پس از ورود به صفحه MemberDefault.aspx بازگشت داده می شوید ، اکنون کلمه عبور فعلی و جدید را وارد کنید و دکمه change password را کلیک کنید.

5- چنانچه کلمه عبور با موفقیت تغییر یابد شکل ۵-۲۲ را مشاهده خواهید کرد.



شکل ۵-۲۲ تغییر کلمه عبور با استفاده از کنترل change password

نکته: باید خاصیت ContinueDestinationPageUrl کنترل ChangePassword را برابر با آدرس یکی از صفحات سایت قرار دهید تا پس از تغییر کلمه عبور و کلیک دکمه Continue کاربر آن به این صفحه هدایت شوند ؛ بهتر است آدرس home page را وارد کنید.

6- در صورتیکه کلمات عبور جدید یکسان نباشند پیغام خطایی بر همین مبنا نمایش داده می شود. (شکل ۵-۲۳)



شکل ۵-۲۳ خطا حین تغییر کلمه عبور

۵-۲-۵ پاورقی بخش ...

.... استفاده از عبارات با قاعده (Regular Expression) جهت کنترل کلمه عبور وارد شده توسط کاربر

با افزودن خاصیت NewPasswordRegularExpression به عنصر <asp:changepassword> در source view می توانید محدودیت هایی را به کلمه عبور کاربر اعمال کنید:

```
<asp:changepassword id="ChangePassword2" runat="server"
    PasswordHintText = "Password must be 8 characters long
    and includes two numbers and two
    special character."
    NewPasswordRegularExpression =
    '@\"(?={8,})(?=(.*\d){2,})(?=(.*\W){2,})'
    NewPasswordRegularExpressionErrorMessage=
    "Error: Your password must be 8 characters long
    and includes two numbers and two special character." >
</asp:changepassword>
```

به این ترتیب چنانچه کلمه عبور وارد شده توسط کاربر با آنچه شما تعریف کرده اید تطابق نداشته باشد از طریق پیغام خطا کاربر را مطلع می سازید. چنانچه از شمای Elegant استفاده کرده باشید کنترل ChangePassword در زمان اجرا شکل ۵-۲۴ خواهد بود.

شکل ۵-۲۴ کنترل change به همراه پیغامهای خطا

۵-۵-۳ منابع اطلاعاتی :

برای آشنایی بیشتر با عبارات با قاعده و نحوه کار آنها به کتاب *Mastering Regular Expressions* تألیف Jeffrey E. F. Friedl از انتشارات O'Reilly مراجعه کنید.

۵-۶- ساخت حساب کاربری توسط کنترل Creat user wizard

یادداشت: کنترل Creat user Wizard ساخت حساب کاربری برای کاربران را بسیار آسان نموده است.

از آنجا که ساخت حساب کاربری برای کاربران از جمله معمولی ترین اعمال در اکثر سایتها می باشد میکروسافت کنترل جدیدی در ASP.NET 2.0 ارائه داده که دریافت اطلاعات کاربران و ساخت حساب مخصوص آنها را آسانتر نموده است. این کنترل اعمالی چون دریافت اطلاعات اولیه کاربران از جمله کلمه عبور، پست الکترونیکی و... را انجام می دهد کاربر جدید از طریق Membership Provider مشخص شده اضافه می کند.

۵-۶-۱- مراحل انجام کار :

در این بخش صفحه ای جهت ثبت نام کاربران در سایت طراحی می کنیم.

1- پروژه بخش قبل از باز کنید (C:\ASPNET20\chap-5-SecurityControls)، یک فرم جدید به آن اضافه کنید. (بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های Add New Item... و سپس Web Form را انتخاب کنید)، نام آنرا *NewUser.aspx*.

2- یک کنترل CreateUserWizard به صفحه اضافه کنید.

3- شمای Elegant را به کنترل اعمال کنید (از طریق لینک Auto Format... در CreateUserWizard tasks) صفحه ساخته شده همانند شکل ۵-۲۵ است.



شکل ۵-۲۵ استفاده از کنترل Create user wizard

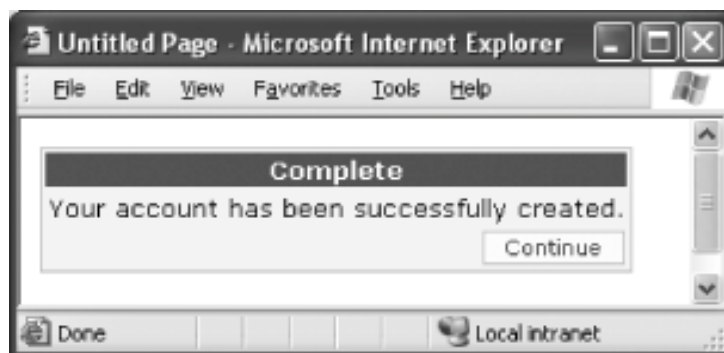
می توانید ظاهر کنترل creat user wizard را از پنجره creat user wizard tasks به شکل دلخواه خود تغییر دهید مثلا از قالبهای برای آن استفاده کنید و یا یک دکمه Cancel به آن اضافه کنید (از طریق خاصیت DisplayCancelButton)

4- کلید F5 را فشار دهید ، اطلاعاتتان را وارد کنید و دکمه creat user را کلیک کنید. (شکل ۵-۲۶)



شکل ۵-۲۶ ورود اطلاعات کاربر جدید

5- چنانچه در ورود اطلاعات مشکلی بوجود نیاید پیغام اتمام کار نمایش داده می شود. (شکل ۵-۲۷)



شکل ۵-۲۷ ساخت حساب کاربری جدید

نکته : می بایست خاصیت `ContinueDestinationPageURL` کنترل `CreateUserWizard` را برابر با آدرس صفحه ای قرار دهید که کاربر پس از کلیک گزینه `continue` به آن صفحه هدایت شود.

۶-می توانید با استفاده از خاصیت `EmailRegularExpression` و عبارات با قاعده قالب پست الکترونیکی کنترل کنید.

۵-۶-۲- پاورقی بخش...

... افزودن کاربر به سایت از طریق برنامه نویسی

بجز افزودن کاربر از طریق ASP.NET WAT ، یا کنترل `Create user wizard` می توانید با استفاده از کلاس `member ship` و از طریق برنامه نویسی اعمال مدیریت کاربران از جمله افزودن کاربر، حذف کاربر و تغییر کلمات عبور را انجام دهید.

1- پروژه بخش قبل (`C:\ASPNET20\chap-5-SecurityControls`) را باز کنید و فرم جدیدی به آن اضافه کنید. (بر روی نام پروژه در `Solution Explorer` راست کلیک کنید و گزینه های `Add New Item...` و سپس `Web Form` را انتخاب کنید)، نام آنرا `AddUser.aspx` قرار دهید.

2- یک جدول با ۲ ستون و ۶ سطر به فرم اضافه کنید (`Layout → Insert Table`) و کنترل های موجود در شکل ۵-۲۸ را به آن اضافه کنید.

شکل ۵-۲۸ طراحی فرم

3- خاصیت TextMode کنترل‌های txtPassword1 و txtPassword2 را Password قرار دهید این خاصیت باعث می‌شود کلمه عبور وارد شده قابل نمایش نباشد.

4- بر روی دکمه Add User کلیک مضاعف کنید و کد زیر را اضافه کنید:

Imports System.Web.Security

Partial Class AddUser_aspx

Inherits System.Web.UI.Page

Protected Sub btnAddUser_Click(ByVal sender As Object, _

ByVal e As System.EventArgs) _

Handles btnAddUser.Click

Dim IsCreated As MembershipCreateStatus

If (txtPassword1.Text = txtPassword2.Text) Then

Membership.CreateUser(txtUserID.Text, txtPassword1.Text, _

txtEmail.Text, txtQn.Text, txtAns.Text, True, IsCreated)

Response.Write("Add user: " & IsCreated.ToString)

Else

Response.Write("Passwords do not match")

End If

End Sub

End Class

5- MembershipCreateStatus وضعیت پروسه ساخت را بر می گرداند. جهت اجرای برنامه کلید F5 را فشار دهید. اطلاعات کاربر در جداول aspnet_Users و aspnet_Membership پایگاه داده مشخص شده توسط Membership provider (بصورت پیش فرض SQL 2005 Express)

... حذف کاربر

برای حذف کاربر می توانید از متد (DeleteUser) از کلاس Membership استفاده کنید:

```
Dim username as String
```

```
Dim deleted as Boolean= _
```

```
Membership.DeleteUser(username)
```

این متد یک نتیجه منطقی (Boolean) جهت تشخیص حذف کاربر باز می گرداند.

۵-۳- منابع اطلاعاتی :

برای آشنایی بیشتر با متدهای کلاس membership در MSDN عبارت "Membership Methods" را جستجو کنید.

۵-۷- گروه بندی کاربران در نقشها (Roles)

یادداشت : با گروه بندی کاربران به آسانی آنها را مدیریت کنید.

بجای محدود کردن دسترسی برخی کاربران خاص، در بسیاری موارد محدود کردن دسترسی بر اساس وظایف و نقش کاربران ساده تر و موثرتر است. با استفاده از ASP.NET WAT می توانید به کاربران نقش هایی بدهید و بر اساس کارکرد آن نقش برای کاربران سطح دسترسی تعیین کنید. بعنوان مثال کاربری کد مدیریت سایت (administrator) را برعهده دارد بایستی به نقش Admin متعلق باشد. ضمناً می توانید در فایل web.cong به نقش و کاربری که به آن تعلق دارد اجازه دسترسی به بخشهای غیر قابل دسترسی توسط دیگر کاربران را بدهید.

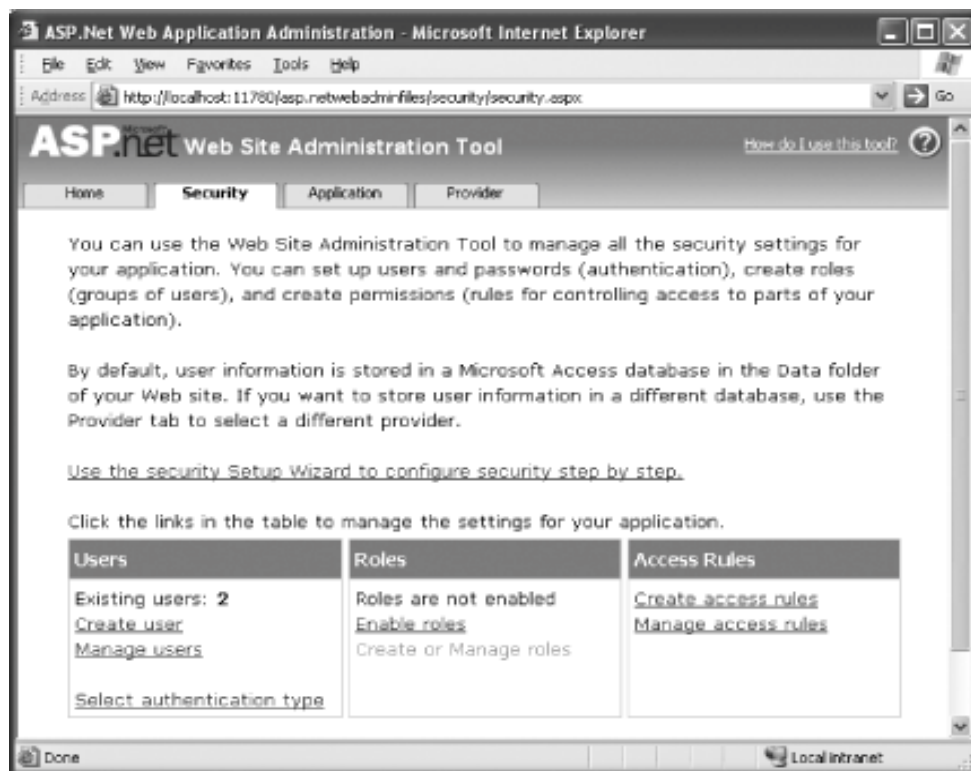
۵-۷-۱. مراحل انجام کار :

در این بخش می آموزید چگونه توسط Asp.Net WAT برای سایتتان نقش تعریف کنید. در ادامه نقشی با نام admin می سازیم و کاربر جدیدی به نام administrator را به آن انتساب می دهیم پس از آن دسترسی به شاخه های سایت را بر اساس نقش کاربر محدود می کنیم.

1- برنامه ابتدای فصل را باز کنید. (C:\ASPNET20\chap-5-SecurityControls)

2- به ASP.NET WAT بروید. (Website → ASP.NET Configuration)

3- در ASP.NET WAT برگه Security را کلیک کنید. (شکل ۵-۲۹)



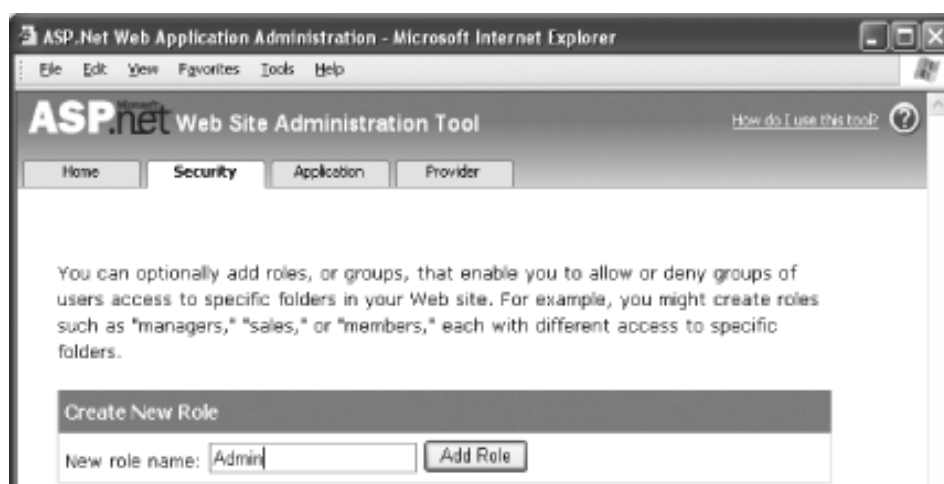
شکل ۵-۲۹ برگه Security در ابزار ASP.NET WAT

4- ابتدا کاربری جدیدی به نام Administrator به سایت اضافه کنید (بر روی لینک creat user کلیک کنید)

5- در قسمت Rules لینک "Enable roles" را کلیک کنید این گزینه باعث فعال شدن "Create or Manage roles" می شود.

6- پس از فعال شدن بر روی لینک "Create or Manage roles" کلیک کنید.

7- نام نقش جدید را وارد کنید: Admin. دکمه Add Role را کلیک کنید. (شکل ۵-۳۰) نقش مورد نظر ساخته و در همان صفحه نمایش داده می شود.



شکل ۵-۳۰ نامگذاری نقش جدید

8- از گزینه manage؛ برای افزودن و حذف کاربران به نقش استفاده کنید. (شکل ۵-۳۱)



شکل ۵-۳۱ مدیریت نقش

9- اکنون می توانید کاربران را برای افزودن به نقش جستجو کنید. بر روی لینک All کلیک کنید و چک باکس روبروی Administrator را تیک بزنید با این کار کاربر به نقش Admin منتسب شد. (شکل ۵-۳۲)



شکل ۵-۳۲ انتساب کاربران به نقشها

10- اکنون کاربر Administrator عضوی از نقش Admin است. کد زیر را به فایل web.config شاخه members اضافه کنید :

<authorization>

```
<allow roles="Admin" />
<deny users="*" />
</authorization>
```

11- با این کد تنها کاربران با نقش Admin به صفحات members دسترسی دارند (سایر تقاضاها رد (Deny) می شوند)

12- در Solution Explorer صفحه *MemberDefault.aspx* را انتخاب و F5 را فشار دهید. همانطور که مشاهده می کنید به صفحه login هدایت می شوید در صفحه login با نام کاربری **wml** که در بخش قبل ایجاد کردیم به سایت وارد شوید، متوجه می شوید که از شما درخواست ورود مجدد می شود. چرا که کاربر WML به نقش Admin تعلق ندارد و امکان دسترسی برایش وجود ندارد. بار دوم با شناسه Administrator وارد شوید این بار صفحه *MembersDefault.aspx* لود می شود.

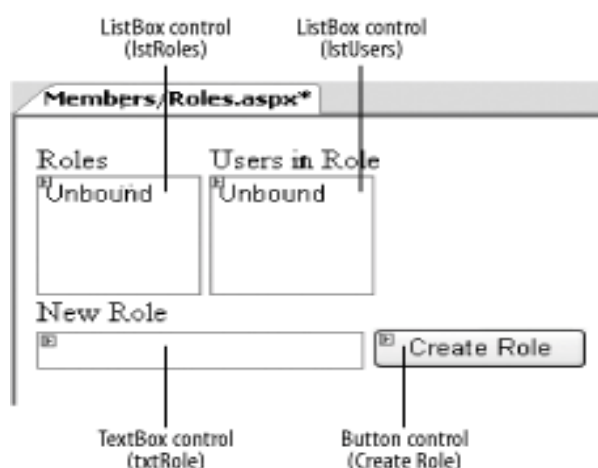
۵-۷-۲- پاورقی بخش ...

... ساخت نقش ها و انتساب کاربران به آنها از طریق برنامه نویسی

بجای استفاده از ASP.NET WAT می توانید با استفاده از کلاس Roles و از طریق برنامه نویسی نقش ایجاد کنید و یا کاربران را به نقشها انتساب دهید.

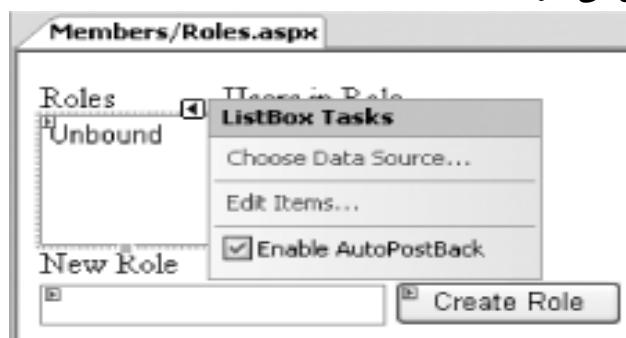
1- برای اینکه با نحوه کار با کلاس Roles آشنا شوید فرم جدیدی به شاخه members اضافه کنید (بر روی members در Solution Explorer راست کلیک کنید و گزینه های add new item و web form را انتخاب کنید) فرم را Roles.aspx نامگذاری کنید.

3- کنترل های موجود در شکل ۵-۳۳ را به فرم اضافه کنید. در این صفحه نقشها و کاربران مربوط به هر نقش تعریف شده را لیست می کنیم.



شکل ۳۳-۵ طراحی فرم جهت نمایش نقشها و کاربران

4- در پنجره **ListBox Tasks** کنترل **IstRoles** گزینه **Enable AutoPostBack** را تیک بزنید (شکل ۳۴-۵). با این کار هنگام انتخاب نقش در **listbox** یک **postback** رخ می دهد که باعث نمایش تمامی کاربران متعلق به آن نقش می شود.



شکل ۳۴-۵ تنظیم گزینه **Enable AutoPostBack** کنترل **IstRoles**

4- بر روی فرم کلیک مضاعف کنید و کد زیر را در رویداد **Form_load** وارد کنید متد **(GetAllRoles)** کلاس **Roles** برای بازیابی تمامی نقشهای تعریف شده در سایت هنگام بارگذاری صفحه بکار می رود. هنگام انتخاب هر نقش کاربران متعلق به آن نقش توسط متد **(GetUsersInRole)** بازیابی و نمایش داده می شوند :

```
Protected Sub Page_Load(ByVal sender As Object, _  
    ByVal e As System.EventArgs) _  
    Handles Me.Load
```

```
'---display roles when the page is first loaded
```

```
If Not IsPostBack Then
```

```
    Dim allRoles( ) As String
```

```
    allRoles = Roles.GetAllRoles( )
```

```
    IstRoles.Items.Clear( )
```

```
    For i As Integer = 0 To allRoles.Length - 1
```

```
        IstRoles.Items.Add(allRoles(i).ToString)
```

```
    Next
```

```
End If
```

```
'---displays users in selected role
```

```
IstUsers.Items.Clear( )
```

```
If IstRoles.SelectedItem IsNot Nothing Then
```

```
    Dim allUsersInRole( ) As String = _
```

```
        Roles.GetUsersInRole(IstRoles.SelectedItem.ToString)
```

```

For i As Integer = 0 To allUsersInRole.Length - 1
    IstUsers.Items.Add(allUsersInRole(i).ToString)
Next
End If
End Sub

```

5- از متد () CreateRole برای ساخت نقش جدید استفاده می شود بر روی دکمه Create Role کلیک مضاعف کنید و کد زیر را به آن وارد کنید :

```

Protected Sub btnCreateRole_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles btnCreateRole.Click
    Roles.CreateRole(txtRole.Text)
    IstRoles.Items.Add(txtRole.Text)
End Sub

```

6- کلید F5 را فشار دهید. پس از اجرا جهت ورود به سایت به صفحه login هدایت می شوید با کاربر administrator به سایت وارد شوید. پس از ورود می توانید نقش Admin را در بخش roles مشاهده کنید حال چنانچه بر روی Admin کلیک کنید .کاربر (کاربران) متعلق به نقش Admin در قسمت USER ROLES لیست می شوند. (شکل ۳۵-۵)



شکل ۳۵-۵ نمایش نقشها و کاربران

7- علاوه بر این می توانید در قسمت New Role نقش های جدید را ایجاد کنید. کلاس Roles متدهای دیگری نیز دارد که برخی از آنها عبارتند از :

- Roles.GetRolesForUser
- Roles.GetUsersInRole
- Roles.GetUsersInRole

- Roles.IsUserInRole
- Roles.RemoveUserFromRole
- Roles.RemoveUserFromRoles
- Roles.RemoveUsersFromRole
- Roles.RemoveUsersFromRoles
- Roles.RoleExists
- Roles.FindUsersInRole
- Roles.AddUsersToRole
- Roles.AddUsersToRoles
- Roles.AddUserToRole
- Roles.AddUserToRoles

۵-۷-۳- منابع اطلاعاتی :

برای آشنایی با چگونگی ذخیره اطلاعات نقشها توسط ASP.NET 2.0 در MSDN موضوع "ASP.NET Role Management Providers" را جستجو کنید.

همچنین موضوع "Implementing a Role Provider" به نحوه ایجاد role provider توسط کاربر می پردازد . برای آشنایی با سایر متدهای کلاس Roles در MSDN به موضوع "Roles Methods" را مراجعه کنید.

۵-۸- مدیریت تنظیمات کاربر

یادداشت : از طریق کلاس Membership می توانید اطلاعات مربوط به کاربران را از طریق برنامه نویسی ویرایش کنید.

تا اینجا آموختید چگونه از طریق کنترلهای ASP.NET 2.0 اطلاعات کاربران را ویرایش کنید ، اما کلاس Membership امکان اصلاح اطلاعات کاربران را حین برنامه نویسی فراهم آورده است. در واقع این کلاس منشا (source) بسیاری از کنترلهای جدید ASP.NET 2.0 است .

۵-۸-۱- مراحل انجام کار :

در این بخش می آموزید چگونه از طریق کد نویسی به تنظیمات کاربران دسترسی پیدا کنید و آنها را ویرایش کنید. برای این منظور صفحه ای با چندین TextBox و Button طراحی می کنیم که امکان تغییر کلمه عبور، سؤالات امنیتی و پست الکترونیکی را برای کاربری فراهم می آورد.

1- پروژه بخش قبل (*C:\ASPNET20\chap-5-SecurityControls*) را باز کنید و فرم جدیدی به آن اضافه کنید (بر روی فرم پروژه در Solution Explorer راست کلیک کنید و گزینه های Add New Item... سپس web form را انتخاب کنید) نام فرم جدید را UserInfo.aspx قرار دهید.

2- یک جدول با ۳ ستون و ۶ سطر به فرم اضافه کنید. (Layout → Insert Table) همانند شکل ۵-۳۶ کنترل های textbox و Button را به جدول اضافه کنید.

شکل ۵-۳۶ کنترل های مختلف صفحه UserInfo.aspx

3- هنگام بارگذاری صفحه متد () GetUser از کلاس Membership اطلاعات کاربر را بازیابی می کند. در code behind_ فرم کد زیر را وارد کنید :

```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load
    Dim user As MembershipUser
    '---get info about current user
    user = Membership.GetUser
    Response.Write("Your last login was on: " & _
        user.LastLoginDate & "<br/>")
    Response.Write("Your account was created on: " & _
        user.CreationDate & "<br/>")
    Response.Write("Your password was last changed on: " & _
        user.LastPasswordChangedDate & "<br/>")
    If Not IsPostBack Then
```

```
txtUserID.Text = user.UserName.ToString
txtPasswordQuestion.Text = user.PasswordQuestion
txtEmail.Text = user.Email
```

```
End If
```

```
End Sub
```

4- دقت کنید کلاس MembershipUser شامل اطلاعات گوناگون کاربر است از قبیل تاریخ آخرین ورود (last login)، تاریخ ایجاد (creation date) و آخرین تاریخ تغییر کلمه عبور (last changed date) همچنین می توانید سؤال امنیتی کلمه عبور را نیز باز یابی کنید (البته جواب آنرا نمی توانید باز یابی کنید)

نکته: چنانچه کاربر تایید هویت نشده باشد متد Membership.GetUser مقدار NULL باز می گرداند .

5- زیر روال () displayMessage را جهت نمایش یک پیغام در صفحه به برنامه اضافه کنید :

```
Private Sub displayMessage(ByVal str As String)
```

```
Response.Write("<br/><b>" & str & "</b>")
```

```
End Sub
```

6- هنگام کلیک دکمه با استفاده از متد () ChangePassword از کلاس MembershipUser می توانید کلمه عبور را تغییر دهید این متد یک مقدار منطقی (Boolean) باز می گرداند که مشخص می کند کلمه عبور تغییر یافت یا خیر :

```
Protected Sub btnChangePassword_Click(ByVal sender As Object, _
```

```
ByVal e As System.EventArgs) _
```

```
Handles btnChangePassword.Click
```

```
Dim user As MembershipUser = Membership.GetUser
```

```
'==change password=='
```

```
If txtCurrentPassword.Text <> "" And _
```

```
(txtNewPassword.Text <> txtCurrentPassword.Text) And _
```

```
txtNewPassword.Text <> "" Then
```

```
'---means the user wants to change password
```

```
If user.ChangePassword(txtCurrentPassword.Text, _
```

```
txtNewPassword.Text) Then
```

```
displayMessage("Password Changed.")
```

```
Else
```

```
displayMessage("Password Changed Failed.")
```

```

End If
Else
    displayMessage("Required fields missing")
End If
End Sub

```

7- از متد () `ChangePasswordQuestionAndAnswer` نیز برای تغییر سؤال مربوط به کلمه عبور استفاده می شود این متد نیز یک مقدار منطقی برمی گرداند :

```

Protected Sub btnChangePasswordQuestion_Click( _
    ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles btnChangePasswordQuestion.Click
    Dim user As MembershipUser = Membership.GetUser
    If txtCurrentPassword.Text <> "" And _
        txtPasswordQuestion.Text <> "" And _
        txtPasswordAnswer.Text <> "" Then
        If user.ChangePasswordQuestionAndAnswer( _
            txtCurrentPassword.Text, _
            txtPasswordQuestion.Text, _
            txtPasswordAnswer.Text) Then
            displayMessage("Password Question Changed.")
        Else
            displayMessage("Password Question Failed.")
        End If
    Else
        displayMessage("Required fields missing")
    End If
End Sub

```

8- هنگامیکه کاربر دکمه `Update Email` را کلیک می کند از طریق متد () `UpdateUser` از کلاس `Membership` پست الکترونیکی کاربر به روز می شود :

```

Protected Sub btnUpdateEmail_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles btnUpdateEmail.Click

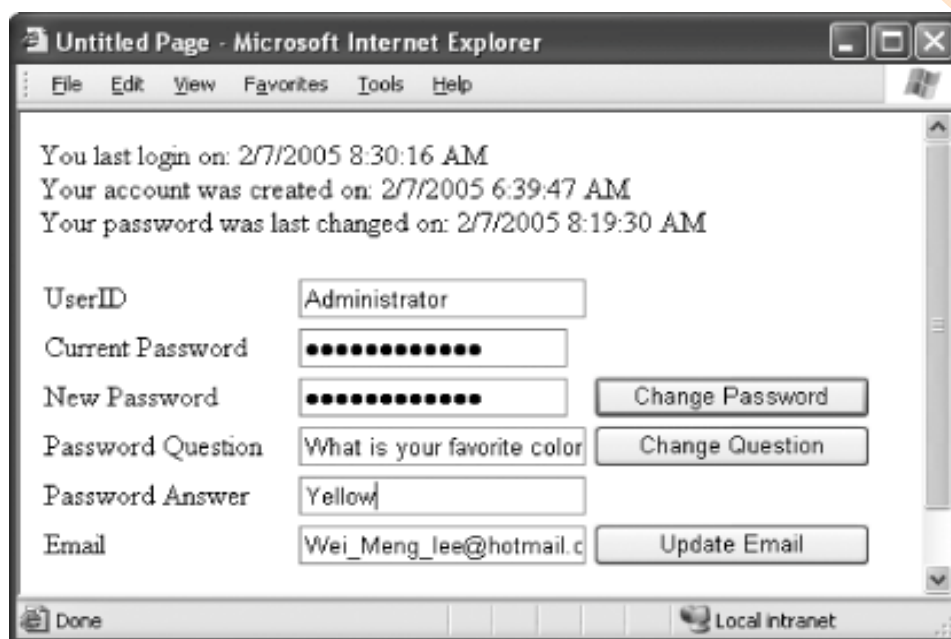
```



```
Dim user As MembershipUser = Membership.GetUser  
user.Email = txtEmail.Text  
Membership.UpdateUser(user)  
displayMessage("Email Updated.")
```

End Sub

9- کلید F5 را فشار دهید ، پس از اجرا به صفحه ورود هدایت می شوید به سایت وارد شوید. از نام کاربری Administrator استفاده کنید. شکل ۵-۳۷ نتیجه صفحه را در عمل نشان می دهد برای تغییر هر کدام از گزینه ها Textbox مربوطه را با اطلاعات جدید پر کنید و کلید مربوطه را کلیک کنید.



شکل ۵-۳۷ ویرایش اطلاعات کاربر

دقت کنید ؛ از صفحه بایستی حفاظت شود (مثلاً با استفاده از SSL) چرا که کلمات عبور بصورت صریح به این صفحه انتقال داده می شوند.

۵-۸-۲- پاورقی بخش...

... بازیابی اطلاعات تمامی کاربران

با استفاده از متد (GetAllUsers) از کلاس MembershipUser می توانید اطلاعات تمامی کاربران بصورت یکجا بازیابی کنید:

```
Dim users As MembershipUserCollection  
users = Membership.GetAllUsers
```

اطلاعات کاربران در قالب یک `MembershipUserCollection` بازگردانده می شوند که می توانید با مقید ساختن آن به یک `GridView` آنها را نمایش دهید :

```
GridView1.DataSource = users
```

```
GridView1.DataBind( )
```

برای مقدار دهی مجدد (Resetting) کلمه عبور کاربر می توانید از متد (`ResetPassword`) از کلاس `MembershipUser` استفاده کنید البته بایستی جواب کلمه عبور را بعنوان ورودی به این متد بدهید :

```
Dim newPassword As String = user.ResetPassword(txtPasswordAnswer.Text)
```

متد (`ResetPassword`) یک کلمه عبور تصادفی (random) تولید می کند و باز می گرداند.

۵-۸-۳. منابع اطلاعاتی :

برای آشنایی با تمامی خواص و متدهای کلاسهای `Membership` , `MembershipUser` به MSDN و موضوعات "Membership Members" و "MembershipUser Members" مراجعه کنید.

در فصل چهارم نیز اطلاعات جامعی راجع به کنترل `GridView` ارائه شده است.

برای آموختن نحوه پیکربندی SSL در IIS به آدرس زیر مراجعه کنید :

http://www.microsoft.com/resources/documentation/iis/6/all/proddocs/en-us/nntp_sec_securing_connections.msp

فصل ششم

کار آیی های جدید

Asp.Net 2.0

ASP.NET 2.0 قابلیت‌های جدیدی برای بالا بردن کارایی برنامه های وب به همراه آورده است.

بعنوان مثال در ASP.NET 1.x صفحات بصورت پویا (Dynamic) ترجمه (Compile) و با اولین بارگزاری توسط کاربر کش (Cache) میشوند. در نتیجه هنگامیکه برنامه برای بار اول بارگزاری میشود سرعت آن پائین است. در ASP.NET 2.0 از طریق پیش ترجمه (PreCompile) سایت، هنگامیکه اولین درخواست به سایت صورت می پذیرد سایت شما از قبل ترجمه شده است در نتیجه با سرعت بالا تری در اختیار کاربر قرار میگیرد.

در ASP.NET 1.x تمامی صفحات Code-Behind پس از ترجمه در قالب یک فایل dll. درون پوشه bin/ قرار میگیرند، در حالیکه ترجمه صفحات aspx. بر اساس تقاضا صورت می پذیرد.

از اینرو هر گونه تغییر در فرمها بصورت خودکار به روز میشوند، در حالیکه تغییرات صفحات Code-Behind تا ترجمه مجدد آنها اعمال نمی شوند.

در ASP.NET 2.0 با استفاده از ترجمه پویا در زمان اجرا (Dynamic Runtime Compile) هر گونه تغییر در فایل های aspx و vb. (یا cs.) باعث ترجمه مجدد صفحه بصورت خودکار میشود.

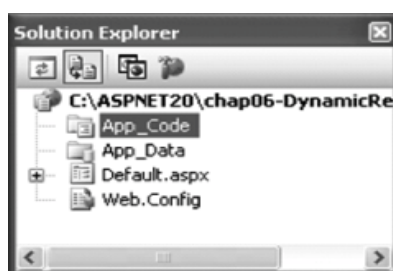
۱-۶. ترجمه صفحات بصورت پویا

برای اینکه ببینید ASP.NET 2.0 چگونه فایل های aspx و Code_Behind را بصورت پویا ترجمه مجدد می کند برنامه ای می نویسیم که از یک کلاس در شاخه خاصی به نام App_Code استفاده می کند. چنانچه در این کلاس تغییری ایجاد کنید هنگام در خواست این کلاس توسط صفحات ASP.NET 2.0 بطور خودکار آنرا ترجمه مجدد میکند.

1. VS 2005 را باز کنید و پروژه جدیدی ایجاد کنید، نام آنرا C:\ASPNET20\chap06-DynamicRecompilation قرار دهید.

2- برای اینکه از ترجمه پویا در زمان اجرا استفاده کنید بایستی شاخه ای به نام App_Code به پروژه تان اضافه کنید؛ بر روی نام پروژه در Solution Explorer راست کلیک کنید، و گزینه های Add New Folder → Application Code Folder را انتخاب کنید. این شاخه را در عکس ۱.۶ می توانید

ببینید.



شکل ۱-۶ شاخه App_Code در Solution Explorer

شاخه های خاص ASP. Net 2.0

در ASP.Net 2.0 شاخه های زیادی وجود دارد که هر کدام استفاده خاصی دارند. این شاخه ها عبارتند از:

App_Code

این شاخه محل نگهداری کلاسها، اسناد WSDL و XSD است، تمامی فایل های موجود در این شاخه در زمان اجرا بصورت خودکار ترجمه می شوند.

App_Data

از این شاخه برای نگهداری فایل های پایگاه داده، XML و شماها (schema) استفاده می شود.

App_LocalResources

این شاخه محل ذخیره فایل های منبع (resource) تولید شده توسط Visual Studio برای بومی سازی ضمنی (implicit localization) است. برای کسب اطلاعات بیشتر به فصل ۸ مراجعه کنید.

App_GlobalResources

این شاخه برای ذخیره فایل های منبع (resource) مخصوص بومی سازی صریح (Explicit localization) بکار می رود. توجه داشته باشید که فایل های Resource (resx) موجود این شاخه بصورت خودکار توسط کلاس Resources تولید می شوند. برای کسب اطلاعات بیشتر به فصل ۸ کتاب مراجعه کنید.

App_Themes

این شاخه فایل های Theme, skin را نگهداری می کند.

App_Browsers

از این شاخه جهت نگهداری فایل های مرورگرها و پشتیبانی از کلاینت های مختلف استفاده میشود.

App_WebReferences

این شاخه فایل های wsdl و discomap. مخصوص وب سرویس ها (web services) را نگهداری می کند. شاخه Bin/ نیز برای پشتیبانی از برنامه های ASP.NET 1.x همچنان وجود دارد.

3. بر روی شاخه *App_Code* راست کلیک کنید، گزینه Add New Item... و سپس آیتم class را انتخاب و نام آنرا *TimeClass.vb* قرار دهید.

اگر بر روی نام پروژه راست کلیک کنید و بخواهید یک فایل class به پروژه اضافه کنید. از شما خواسته می شود کلاس را به شاخه *App_Code* اضافه کنید و چنانچه این شاخه وجود نداشته باشد، بطور خودکار ساخته می شود.

4. کد زیر را به فایل *Timeclass.Vb* اضافه کنید :

```
Imports Microsoft.VisualBasic
```

```
Public Class TimeClass
```

```
    Public Function getCurrentTime( ) As String
```

```
        Return "The current time is " & Now
```

```
    End Function
```

```
End Class
```

5. به *code-behind* فرم بروید و کد زیر را به رویداد *From_Load* اضافه کنید :

```
Protected Sub Page_Load(ByVal sender As Object, _
```

```
    ByVal e As System.EventArgs) _
```

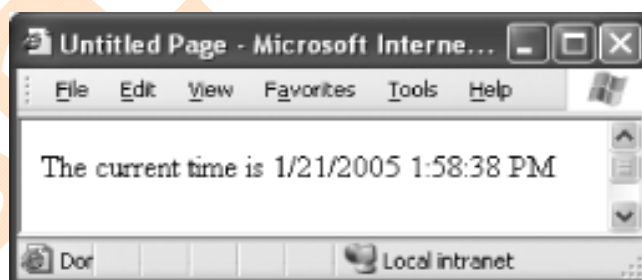
```
    Handles Me.Load
```

```
    Dim time As New TimeClass
```

```
    Response.Write(time.getCurrentTime( ))
```

```
End Sub
```

6. کلید *F5* را فشار دهید، خروجی را در شکل ۲-۶ می توانید مشاهده کنید.



شکل ۲-۶. تست برنامه

7. بدون اینکه مرورگر را ببندید، با استفاده از Notepad از فایل *TimeClass.vb* را از آدرس *C:\ASPNET20\chap06-DynamicRecompilation\App_Code* باز کنید، تغییرات زیر را به آن اعمال کنید و فایل را ذخیره کنید.

```
Imports Microsoft.VisualBasic
```

Public Class TimeClass

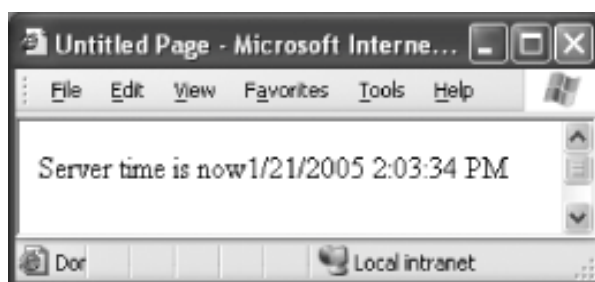
Public Function getCurrentTime() As String

Return "Server time is now" & Now

End Function

End Class

8- در مرورگر صفحه را Refresh کنید. اگر دقت کنید مدت زمانی برای بارگذاری مجدد (Reload) صفحه سپری می شود. علت این است که ASP.Net ایجاد تغییر در فایل Timeclass.Vb را شناخته و در حال ترجمه مجدد (recompile) آن است. نتیجه را در شکل ۶-۳ میتوانید مشاهده کنید.



شکل ۶-۳. نمایش صفحه به روز شده

۶-۱-۱. پاورقی بخش ...

... استفاده از چند زبان در یک پروژه

همانطور که مشاهده شد؛ می توانید فایل های class را در شاخه *App_Code* قرار دهید تا ASP.Net بصورت پویا آنها را ترجمه نماید. اما از آنجا که تمامی این فایلها تبدیل به یک فایل می شوند، نمی توانید فایل های چند زبانه داشته باشید. (مثلاً یکی VB 2005 و دیگری C#).

یادداشت: ASP.Net از مترجم زبان (language compiler) مناسب برای ترجمه فایل های شاخه *App_Code* استفاده می کند.

چنانچه می خواهید از فایل های چند زبانه در شاخه *App_Code* استفاده کنید. بایستی فایل را با استفاده از زیر شاخه (subfolder) از یکدیگر جدا کنید.

نکته: همانطور که در فصل اول هم بیان شد، اگر چه پشتیبانی از چند زبانی قابلیت مفیدی است، اما برنامه نویسان بایستی بر اساس مزایا و معایب این قابلیت از آن استفاده کنند. مزیت چند زبانی این است که می توانید از چند برنامه نویس (با تخصص های مختلف) در یک پروژه بهره گیرید اما نگهداری و پشتیبانی پروژه نیز به مراتب دشوارتر خواهد بود.

برای ساخت زیر شاخه در *App_Code* عنصر `<codeSubDirectories>` را به *web.Config* اضافه کنید:

```
<compilation debug="true">
  <codeSubDirectories>
    <add directoryName="VB" />
    <add directoryName="CS" />
  </codeSubDirectories>
</compilation>
```

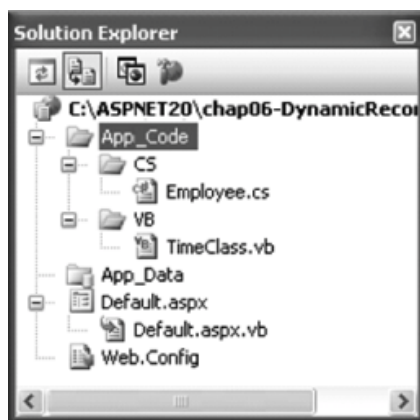
نکته: نام زیر شاخه های تعریف شده کاملاً اختیاری است و نیازی نیست حتماً همنام زبان برنامه نویسی مورد نظر باشد. ASP.Net فایل های هر زیر شاخه را بررسی می کند و بر اساس نوع آنها از مترجم مناسب استفاده می کند. نکته مهم این است که فایل های مخصوص هر زبان همگی درون یک شاخه قرار گیرند.

بر روی شاخه *App_code* راست کلیک کنید و گزینه های *Add Folder* → *Regular Folder* را انتخاب کنید، نام شاخه را *VB* قرار دهید، شاخه دیگری با نام *CS* به همین شیوه به *App_code* اضافه کنید. اکنون شاخه *App_code* مانند شکل ۴-۶ شده است.



شکل ۴-۶. زیر شاخه های *App_Code*

اکنون می توانید فایل *Timeclass.VB* را به شاخه *VB* انتقال دهید. یک *class* جدید (با نام *Employee.CS* و به زبان C#) به شاخه *CS* اضافه کنید. (شکل ۵-۶)



شکل ۶-۵. انتقال فایلها به شاخه های مربوطه

۶-۱-۲. منابع اطلاعاتی :

انتخاب زبان برنامه نویسی همیشه بحثی داغ و تازه بوده است. چنانچه شما هم نمی دانید از VB.NET یا C# استفاده کنید. مقالات زیر را مطالعه کنید؛ سپس خودتان تصمیم بگیرید.

- <http://blogs.msdn.com/csharpfaq/archive/2004/03/11/87816.aspx>
- <http://dotnetjunkies.com/WebLog/sahilmalik/archive/2004/05/07/13071.aspx>
- <http://www.developerdotstar.com/community/node/61>

۶-۲. تولید کلاسهای web service proxy بصورت پویا

یادداشت: هم اکنون با انتقال یک فایل WSDL به شاخه App_Code، می توانید بصورت پویا وب پراکسی (web proxy) تولید کنید.

در Visual Studio.Net 2003 برای تولید کلاس web proxy مناسب می بایست بصورت دستی یک web Refrence به یک web service اضافه کنید. اما در VS 2005 این عمل بسیار آسانتر شده است و تنها کافیست یک سند WSDL که سرویس مورد نظر را تشریح می کند. به شاخه App_code اضافه کنید، سپس کلاس مورد نیاز برای دسترسی به سرویس بصورت خودکار تولید می شود.

وب سرویس (Web Service) چیست؟

از لحاظ فنی، web service یک Business object محسوب می شود که بر روی سرور (server) قرار گرفته است و شما می توانید از طریق برنامه نویسی به آن دسترسی داشته باشید، بعنوان مثال بورس سهام را در نظر بگیرید، بورس میتواند از طریق یک وب سرویس به مشتریان این امکان را بدهد تا از آخرین تغییرات سهام آگاه شوند. بعلاوه مشتریان می

توانند با دسترسی به داده های بورس از طریق web server، برنامه های سفارشی مخصوص خود بنویسند سپس با یکی کردن نتایج، یک برنامه جامع تولید کنند. برای ایجاد رابطه متقابل میان وب سرویس ها و کاربران آن (که به عنوان مشتریان وب سرویس شناخته می شوند) اکثر وب سرویس ها از استانداردهایی مانند: HTTP, XML, WSDL استفاده می کنند:

XML زبان تبادل پیام میان وب سرویس و مشتریان آن است.

HTTP پروتکل (protocol) انتقال پیام به وب سرویس است.

WSDL هم قراردادی است که مشخص می کند وب سرویس چه چیزی ارائه می دهد.

در (2003, 2005) VS برای برقراری ارتباط با وب سرویس به فایل WSDL مربوط به وب سرویس نیازمندید. مزیت استفاده از VS برای وب سرویس ها این است که VS بطور خودکار یک کلاس پراکسی (بر اساس سند WSDL) برایتان تولید می کند و از آن پس با وب سرویس می توانید همانند یک شی (object) معمولی رفتار کنید.

۶-۲-۱. مراحل انجام کار:

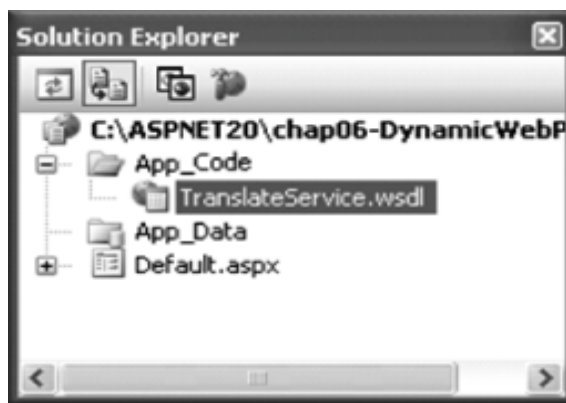
در این بخش می بینید که تولید وب سرویس به کمک سند WSDL آن چقدر ساده است. در این بخش یک وب سرویس مترجم تولید کنیم. تنها کاری که انجام می دهیم ذخیره سند WSDL مربوطه در شاخه App_Code است. سپس مشاهده می کنید که چگونه وب پراکسی ؛ بر اساس WSDL تولید می شود.

1- VS 2005 را باز کنید و پروژه جدیدی با نام *C:\ASPNET20\chap06-DynamicWebProxy* ایجاد کنید.

2- بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های Add Folder → App_Code Folder را انتخاب کنید.

3- در IE (Internet Explorer) سند WSDL موجود در آدرس <http://www.webservices.com/TranslateService.asmx?WSDL> را باز کنید و با نام *C:\ASPNET20\chap06-DynamicWebProxy\App_Code\TranslateService.wsdl* در آدرس ذخیره کنید.

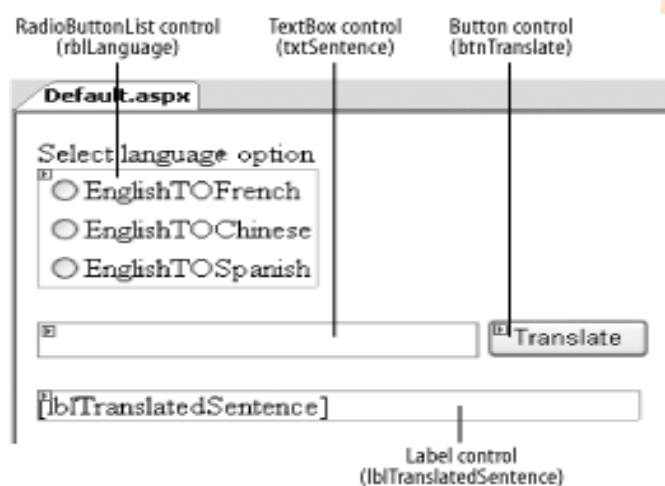
4- برای اینکه سند WSDL را در Solution Explorer مشاهده کنید، بر روی شاخه App_Code راست کلیک کنید و شاخه را Refresh کنید. (شکل ۶-۶)



شکل ۶-۶. ذخیره فایل WSDL درون شاخه App_code

5. به این ترتیب یک کلاس وب پراکسی بصریت پویا تولید خواهد شد.

6. برای تست web proxy class، کنترل های موجود در شکل ۶-۷ را به فرم اضافه کنید.



شکل ۶-۷ طراحی فرم

7. بر روی دکمه Translate کلیک مضاعف کنید و در رویداد click آن کد زیر را وارد کنید:

```
Protected Sub btnTranslate_Click(ByVal sender As Object, _  
    ByVal e As System.EventArgs) _  
    Handles btnTranslate.Click  
  
    Dim ws As New TranslateService  
    Dim targetLanguage As String  
    Select Case rblLanguage.SelectedIndex  
        Case 0 : targetLanguage = _  
            ws.Translate(Language.EnglishTOFrench, _  
                txtSentence.Text)
```

```
Case 1 : targetLanguage = _
ws.Translate(Language.EnglishTOChinese, _
txtSentence.Text)
```

```
Case 2 : targetLanguage = _
ws.Translate(Language.EnglishTOSpanish, _
txtSentence.Text)
```

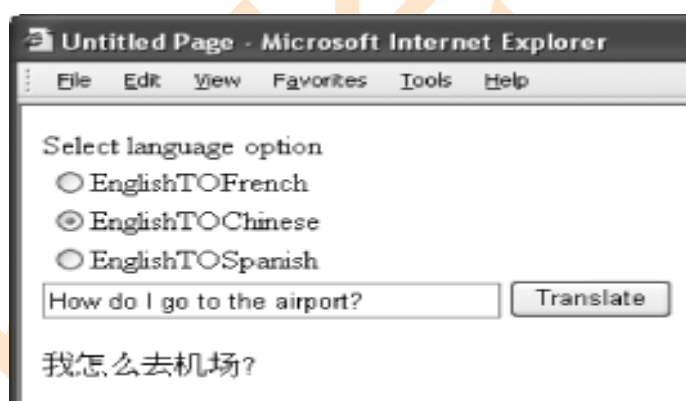
End Select

```
lblTranslatedSentence.Text = targetLanguage
```

End Sub

8. کلید F5 را فشار دهید ؛ جمله ای را در Text box وارد کنید و دکمه Translate را کلیک کنید. (شکل ۸-۶)

نکته: برای نمایش صحیح متن ترجمه شده بایستی زبان مورد نظر بر روی مرورگر نصب شده باشد، برای نمایش صحیح جملات در IE به آدرس View → Encoding → More بروید و زبان مورد نظر را انتخاب کنید (مثلاً زبان چینی) چنانچه زبان مورد نظر از قبل بر روی سیستم نصب نشده باشد مرورگر شروع به نصب زبان می کند. (برای نصب زبانها به CD ویندوز نیاز دارید)

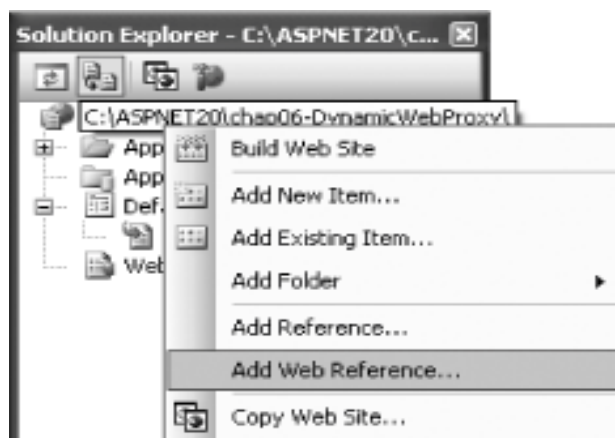


شکل ۸-۶. تست برنامه

۶-۲-۲. پاورقی بخش...

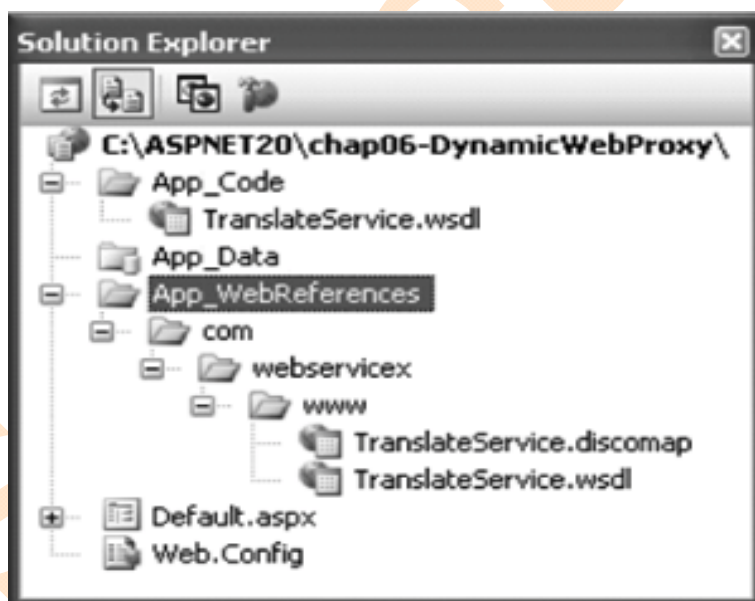
... تولید web service proxy به روش قدیمی و با استفاده از Add web Refrence

در ASP.Net 2.0 همچنان از روش قبلی که در VS 2003 استفاده می کردید می توانید برای تولید کلاس web استفاده کنید، برای این کار بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه Add web Refrence را انتخاب کنید. (شکل ۹-۶)



شکل ۹-۶. افزودن Web Reference

آدرس URL وب سرویس از شما خواسته می شود، چنانچه آدرس <http://www.webservices.com/TranslateService.asmx?WSDL> را وارد کنید، وب سرویس درون شاخه *App_WebReferences* به پروژه اضافه میشود. (شکل ۱۰-۶)



شکل ۱۰-۶ web Refrence اضافه شده به پروژه

از دستور زیر برای دسترسی به وب سرویس استفاده کنید :

`Dim ws As New com.webservices.www.TranslateService`

۳.۲.۶. منابع اطلاعاتی :

برای کسب اطلاعات بیشتر راجع به تولید وب سرویسها از طریق Visual Studio به آدرس زیر مراجعه کنید :

<http://www.programmingtutorials.com/webservices.aspx>

برای آشنایی با معماری وب سرویسها و خصوصیات آنها مقاله زیر را مطالعه کنید :

<http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnwebsrv/html/introwsa.asp>

این مقاله اطلاعات جامع و کاملی راجع به وب سرویس ها در اختیار شما قرار می دهد، بخصوص چنانچه تازه با وب سرویس ها آشنا شده اید، پیشنهاد می شود حتماً این مقاله را مطالعه کنید.

۶-۳. پیش ترجمه سایت (Precompile Web Site)

در Asp.Net 1.x فایل های با پسوند .aspx را بر روی سرور میزبان (Host) قرار می دهید، برای حفاظت بیشتر نیازی به ارسال فایل های code_behind فرمها نیست. اما همچنان واسط کاربری (User Interface) سایت شما درون فایل های .aspx قرار دارد و بایستی آنها را به سرور ارسال کنید. (Uploading) هنوز هم افرادی بجز شما هستند که به فرمها دسترسی دارند و می توانند UI (user Interface) سایتان را بخوانند. Asp.Net 2.0 با یک مرحله پیشرفت؛ به شما امکان پیش ترجمه (Precompile) سایتان را می دهد، به این ترتیب هنگام قرار گرفتن سایت بر روی سرور:

یادداشت: به کمک Asp.Net 2.0 می توانید سایتان را پیش ترجمه نمایید و بدون قرار دادن هرگونه source بر روی سرور، سایت را به اجرا درآورید، و از این طریق زمان پاسخگویی به اولین کاربر را بهبود ببخشید. بعلاوه از آدرس IP (Internet Protocol) خود محافظت کنید.

- هیچگونه کدی (نه .aspx و نه .vb (یا .CS)) بر روی سرور وجود نخواهد داشت.
- زمان اولین پاسخگویی کاهش می یابد.

نکته: چنانچه برنامه شما از اسکریپت های سمت مشتری (Client Side Scripts) استفاده می کند، بایستی برای مرورگر کاربر قابل دسترسی و نمایش باشند. چرا که برای اجرا باید به مرورگر ارسال شوند.

۶-۳-۱. مراحل انجام کار:

در این بخش مراحل پیش ترجمه سایت را با هم مرور می کنیم.

1- VS 2005 را باز کنید، پروژه جدیدی با نام `C:\ASPNET20\chap06-Precompile` ایجاد کنید.

2- برای پیش ترجمه سایت از ابزار `aspnet_compiler` موجود در آدرس `C:\WINDOWS\Microsoft.NET\Framework\<version>` استفاده می کنیم.

3- در پنجره خط فرمان (command-line) دستور زیر را تایپ کنید :

```
C:\Windows\Microsoft.NET\Framework\<version>aspnet_compiler -v /Precompile -p C:\ASPNET20\chap06-Precompile c:\Precompile_Target
```

ابزار `aspnet_compiler`

قالب دستوری (syntax) این ابزار به صورت زیر است:

```
aspnet_compiler [-?] [-m metabasePath | -v virtualPath [-p physicalDir]] [[-u] targetDir]
```

-m

آدرس برنامه در IIS را مشخص می کند. این سوئیچ (switch) را نمی توانید به همراه سوئیچ های -V یا -P بکار ببرید.

-v

آدرس مجازی (Virtual path) برنامه مورد نظر برای ترجمه را مشخص می کند. (مثلاً "/MyApp"). چنانچه از سوئیچ -P هم استفاده شده باشد، از آدرس فیزیکی مشخص شده برای نگهداری برنامه استفاده می کند. در غیر این صورت از IIS استفاده می شود و فرض گرفته می شود که برنامه در آدرس پیش فرض قرار دارد. (آدرس "/LM/W3SVC/1/Root") از این سوئیچ به همراه سوئیچ -m نمی توانید استفاده کنید.

-p

آدرس فیزیکی برنامه توسط این سوئیچ مشخص می شود، چنانچه آدرس مشخص شده توسط -P موجود نباشد، برنامه در IIS قرار می گیرد. این سوئیچ بایستی با سوئیچ -V بصورت ترکیبی بکار رود.

-u

چنانچه از این سوئیچ استفاده شده باشد، برنامه ترجمه شده قابلیت به روز رسانی خواهد داشت. (updatable).

-f

این سوئیچ باعث بازنویسی فهرست (Directory) مقصد می شود، در صورتیکه قبلاً ایجاد شده باشد مقادیر قبلی نیز از بین خواهند رفت.

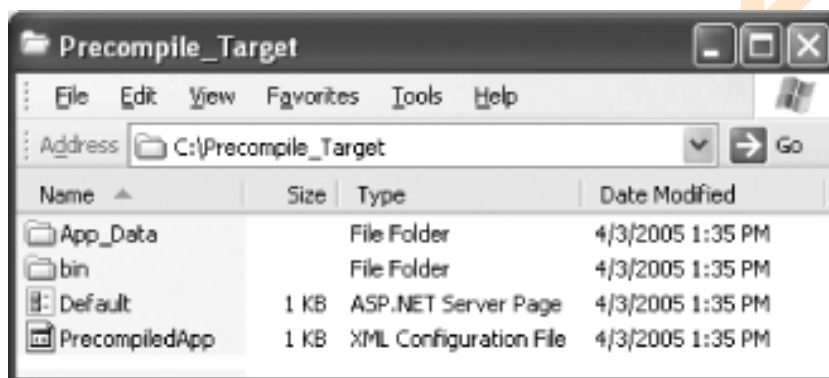
-nologo

از نمایش پیام Copy Right مترجم جلوگیری می کند.

targetDir

آدرس فیزیکی جهت ذخیره برنامه ترجمه شده را مشخص می کند.

4- وب سایت ترجمه شده را در آدرس *C:\Precompile_Target* می توانید پیدا کنید. (شکل ۱۱-۶)



شکل ۱۱-۶. شاخه مربوط به صفحه ترجمه شده

5- چنانچه محتویات فایل Default.aspx را مشاهده کنید، می بینید که این فایل، تنها شامل یک خط می باشد. (تمامی برنامه درون فایل های شاخه /bin ترجمه شده است)

This is a marker file generated by the precompilation tool, and should not be deleted!

6. برای نمایش سایت کافیس در IIS یک شاخه مجازی برای برنامه ایجاد کنید. برای دسترسی به سایت نیز از قالب URL مشابه زیر استفاده کنید.

Http: // webserver/virtualdir/default.aspx

۶-۳-۲. پاورقی بخش...

... به روز رسانی برنامه پیش ترجمه شده

پس از پیش ترجمه برنامه ، تنها می توانید آنرا نمایش دهید، چنانچه می خواهید تغییری در برنامه ایجاد کنید ، بایستی در برنامه اصلی تغییرات را اعمال کنید و مجدداً سایت را پیش ترجمه (precompile) کنید.

۶-۳-۳. منابع اطلاعاتی :

اگر می خواهید با چگونگی پیش ترجمه برنامه های 1.X Asp.NET آشنا شوید ، مقاله زیر را مطالعه کنید :

<http://www.codeproject.com/aspnet/PreCompileAspx.asp>

۶-۴-۶. کش (chache) کردن بخشهای یک صفحه

یادداشت: می توانید با کش کردن بخشی از صفحه بجای تمام آن سرعت به روز رسانی صفحات را افزایش دهید.

اگر چه در Asp.Net 1.x می توانید تمام و یا بخشی از صفحه را به کمک User control ها، کش کنید اما در ASP.NET 2.0 می توانید خروجی صفحه را کش نمایید، بطوریکه در هر درخواست (Request) بخشی از صفحه به روز شود.

۶-۴-۱. مراحل انجام کار:

در این بخش صفحه ای بر اساس fragment caching طراحی می کنیم. صفحه اصلی از کشیک خروجی استفاده می کند بطوریکه در هر درخواست بخشی از صفحه به روز می شود.

1- VS را باز کنید و پروژه جدیدی با نام *C:\ASPNET20\chap06-CacheFragments* ایجاد کنید.

2- به Source view فرم رفته و رهنمود (Directive) `@OutputCache` را به ابتدای صفحه اضافه کنید. خروجی صفحه را به مدت ۵ ثانیه کش خواهیم کرد.

```
<%@ Page Language="VB" AutoEventWireup="false"
CodeFile="Default.aspx.vb"
```

```
Inherits="Default_aspx" %>
```

```
<%@ OutputCache Duration="5" VaryByParam = "none" %>
```

3- در رویداد page_Load صفحه، کد زیر را اضافه کنید :

```
Protected Sub Page_Load(ByVal sender As Object, _
                        ByVal e As System.EventArgs) _
    Handles Me.Load
    Response.Write("Current (cached) Time is : " & Now)
End Sub
```

4- خروجی به مدت ۵ ثانیه کش خواهد شد، به این معنی که زمان نمایش داده شده هر ۵ ثانیه، به هنگام می شود. اکنون تابع زیر را به صفحه اضافه کنید، این تابع زمان های سیستم را نمایش می دهد :

```
Public Shared Function getServerTime( _
    ByVal context As System.Web.HttpContext) _
    As String
    Return "Current Time is : " & Now.ToString
End Function
```

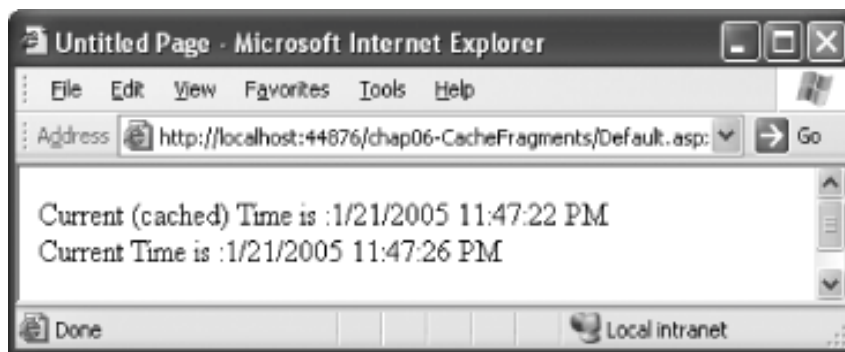
نکته: به پارامتر تابع فوق دقت کنید. کلاس `HttpResponseSubstitutionCallback` که در ادامه خواهد آمد، به این پارامتر نیاز دارد.

5- بجای استفاده از متد `Response.Write()` برای نمایش زمان ارسال شده توسط تابع از متد جدید `Response.WriteSubstitution()` استفاده کنید :

```
Protected Sub Page_Load(ByVal sender As Object, _
                        ByVal e As System.EventArgs) _
    Handles Me.Load
    Response.Write("Current (cached) Time is : " & Now)
    Response.Write("<br/>")
    Response.WriteSubstitution(New _
        HttpResponseSubstitutionCallback(AddressOf _
            getServerTime))
End Sub
```

6- در واقع متد `Response.WriteSubstitution()` تابع `getServerTime()` را در هر درخواست فراخوانی می کند و خروجی جایگزین پاسخ کش شده می شود.

7- کلید F5 را فشار دهید. صفحه را چندین بار Refresh کنید، اگر دقت کنید متوجه می شوید که زمان نمایش داده شده در خط اول هر 5 ثانیه یکبار تغییر می کند ولی زمان خط دوم به ازای هر درخواست. (شکل ۶-۱۲)



شکل ۶-۱۲. نمایش زمان در دو حالت مختلف

۶-۴-۲. پاورقی بخش...

... ذخیره خروجی کش شده بر روی دیسک

بطور پیش فرض Asp.Net 2.0 خروجی کش شده را بر روی دیسک ذخیره می کند، مزیت این کار این است که اولاً می توانید اطلاعاتی بیش از فضای حافظه ذخیره کنید دوماً با راه اندازی مجدد (Restart) وب سرور می توانید اطلاعات قبلی را بازیابی کنید.

به هر حال می توانید این خاصیت را در رهنمود Out putCache صفحه اصلاح کنید :

```
<%@ OutputCache Duration="120" VaryByParam="name"
    DiskCacheable="false" %>
```

برای اعمال تنظیمات کشینگ به تمامی صفحات، عنصر <outputCache> را به Web.Config اضافه کنید. همچنین در صورت تمایل می توانید اندازه فضای مورد نیاز برنامه را مشخص کنید :

```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <caching>
      <outputCache>
        <diskCache enabled="true" maxSizePerApp="5" />
      </outputCache>
    </caching>
  </system.web>
```

</configuration>

چنانچه صفحات زیادی دارید که هر کدام به مدیریت جداگانه ای نیاز دارند، بهتر است با تعریف profile صفحات را بصورت مرکزی مدیریت کنید. به این ترتیب بجای افزودن تنظیمات کشینگ هر صفحه بطور مجزا، مجموعه ای از تنظیمات را در قالب profile کنار هم قرار می دهید. تعریف profile ها در فایل web. Config صورت می گیرد.

کد زیر چندین profile مختلف تعریف می کند:

<system.web>

<outputCache>

<outputCacheSettings>

<outputCacheProfiles>

<add name="Cache30sec" duration="30" />

<add name="Cache5min" duration="300" />

<add name="Cache1hr" duration="3600" />

<add name="ShortTerm" duration="1800" />

<add name="LongTerm" duration="7200" />

</outputCacheProfiles>

</outputCacheSettings>

</outputCache>

</system.web>

برای استفاده از هر profile در صفحه از خاصیت cache profile بصورت زیر استفاده نمائید:

<%@ OutputCache CacheProfile="ShortTerm" VaryByParam="name" %>

از دیگر مزایای تعریف profile ها این است که می توانید مدت زمان کشینگ را بدون ویرایش source صفحه یا صفحات از درون فایل web.config تنظیم نمائید. بعنوان مثال چنانچه فکر می کنید مدت زمان کش در پروفایل "short Term" باید بیش از نیم ساعت (۱۸۰۰ ثانیه) باشد، کافیت این مقدار را با مقدار بیشتری عوض کنید.

۳-۴-۶. منابع اطلاعاتی:

برای آشنایی با شیوه های مختلف بهبود بخشی برنامه های وب با استفاده از کشینگ مقاله زیر را مطالعه کنید.

<http://www.devx.com/asp/Article/21751>

۵-۶- کاهش ارسال PostBack به سرور

یادداشت: با استفاده از Client Call Back Manager داده ها را بدون post back به سرور ارسال کنید.

یکی از محدودیت های برنامه های وب تأخیرهایی هستند که هنگام ارسال داده ها از صفحات به سمت سرور، رخ می دهند چرا که در بسیاری موارد به هنگام سازی بخشی از صفحه که عناصر آن نیازمند تغییر هستند، باعث می شود تمامی صفحه به هنگام (Refresh) می شود. بعنوان مثال فرض کنید در یک صفحه ثبت نام ؛ کاربر "US" را بعنوان کشور انتخاب می کند، در این حالت در یک Drop down ایالت های مربوط به این کشور نمایش داده می شوند تا کاربر ایالت موردنظر خود را انتخاب کند، حال اگر کاربر کشور دیگری را انتخاب کند، drop down باید ایالت های همان کشور را نمایش دهد، برای این کار تمامی صفحه به سرور ارسال شده و نتیجه برای کاربر بازگردانده می شود. (یک روش دیگر این است که ایالت ها به سمت مشتری (client) ارسال شوند و با انتخاب هر کشور به کمک Java script به نمایش در آیند، اگر چه این روش نیاز به Post back را از بین می برد اما در مقابل حجم صفحات را بالا می برد و باعث کندی بارگذاری صفحه می شود)

یکی از تکنیک های که برنامه نویسان ASP.Net 1.x برای غلبه بر post back بکار می برند، استفاده از شیء Microsoft XMLHTTP ActiveX برای ارسال درخواست های سمت سرور از سوی مشتری است.

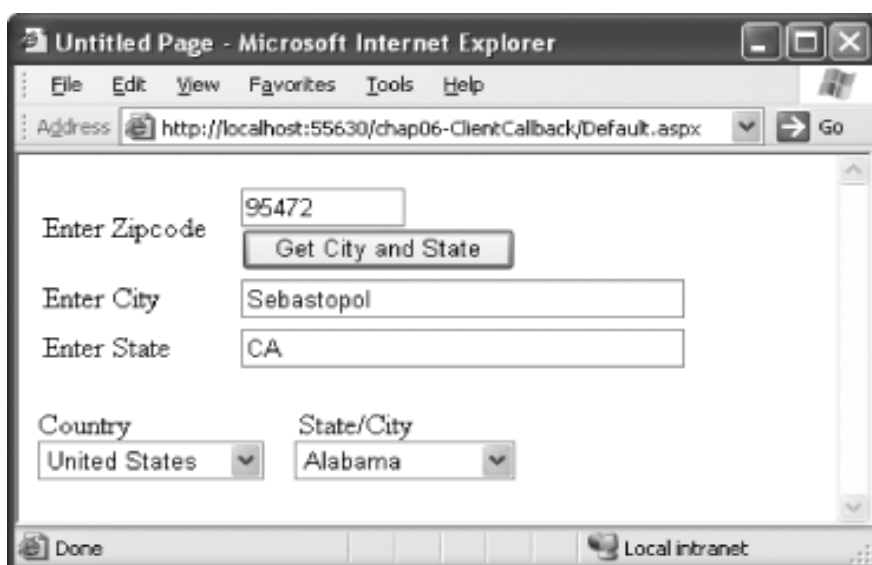
در ASP.Net 2.0 این عمل به صورت ساده تری توسط تابعی به نام client callback manager صورت می پذیرد.

XML HTTP

ASP. Net client callback manager از XMLHTTP جهت ارسال داده ها به سرور و دریافت از مشتری استفاده می کند. برای اینکه callback manger به خوبی کار کند به مرور گری که از XMLHTTP پشتیبانی کند نیاز دارید. Internet Explorer یکی از آنها ست البته نسخه 5.0 یا بالاتر آن.

۵-۶-۱ مراحل انجام کار:

برای آشنایی با نحوه کار callback manager در ASP.Net 2.0 صفحه شکل ۶-۱۳ را طراحی می کنیم.



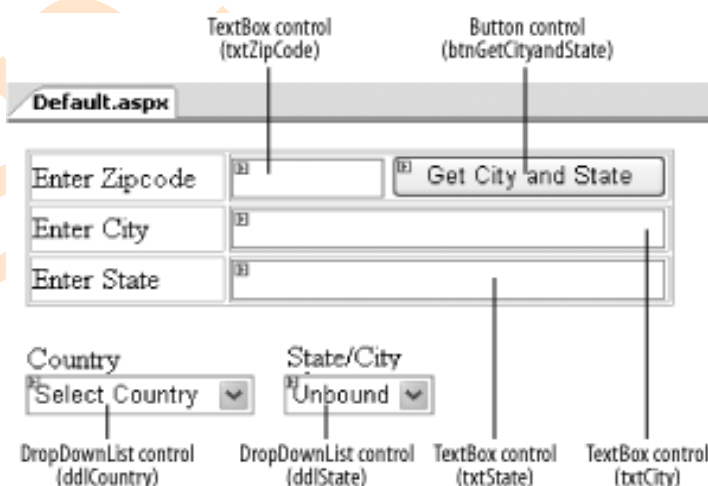
شکل ۶-۱۳ طراحی یک صفحه ساده جهت کار با client call back

در این مثال کاربر zip code را درون یک text box وارد می کند سپس شهر و ایالت مربوط به آن کد بدون post back به سرور بازایی می شوند.

همچنین دو کنترل drop_down list در فرم وجود دارند و هنگامی که کاربر در drop_down list اول کشوری را انتخاب می کند، ایالتها (و یا شهرهایی) متعلق به آن کشور در drop_down list دوم نمایش داده می شوند.

1. VS 2005 راه باز کنید و پروژه جدیدی با نام *C:\ASPNET20\chap06-ClientCallback* ایجاد کنید.

2. کنترلهای شکل ۶-۱۴ را به فرم اضافه کنید.



شکل ۶-۱۴ طراحی فرم

3. سه آیتم به Drop_down list اول (ddlCountry) اضافه کنید. آیتنها را می توانید در Source view فرم بصورت زیر اضافه کنید:

```
<asp:DropDownList ID="ddlCountry" Runat="Server" >
  <asp:ListItem>Select Country</asp:ListItem>
  <asp:ListItem Value="US">United States</asp:ListItem>
  <asp:ListItem Value="Sing">Singapore</asp:ListItem>
  <asp:ListItem Value="UK">United Kingdom</asp:ListItem>
</asp:DropDownList>
```

4. به code_behind صفحه بروید. صفحه ای که post_back را دریافت می کند، نیازمند استفاده از واسط ICallbackEventHandler و یک رشته عمومی (public string) است. (بعدها از این رشته استفاده خواهد شد)، بطوریکه از یک رشته خصوصی (Private) برای نگهداری نتیجه به مشتری ارسال می شود:

Partial Class Default_aspx

Inherits System.Web.UI.Page

Implements ICallbackEventHandler

Public callbackStr As String

Private _result As String

رابط ICallbackEventHandler interface دو متد دارد: RaiseCallbackEvent() و Getcallback () متد RaiseCallbackEvent() هنگام ارسال یک post back از مشتری به سرور فراخوانی می شود و شهر و ایالت مربوط به zip code داده شده چک می شوند.

پس از اتمام پروسه رویداد () Getcallback soult برای باز گرداندن نتیجه به مشتری فراخوانده می شود.

بهتر است تمامی اطلاعات مورد نیاز از یک وب سرویس بازیابی شوند. کد مربوط به متد () RaiseCallbackEvent در مثال ۶-۱ آورده شده است.

Public Sub RaiseCallbackEvent(_

ByVal eventArgument As String) _

Implements _

System.Web.UI.ICallbackEventHandler.RaiseCallbackEvent

If eventArgument.StartsWith("1:") Then

'---strips away the command

eventArgument = eventArgument.Substring(2)

'---get city and state based on Zipcode

```

' System.Threading.Thread.Sleep(5000)
Select Case eventArgument
    Case "95472" : _result = "Sebastopol,CA"
    Case "02140" : _result = "Cambridge,MA"
    Case Else
        _result = "error"
End Select
Elseif eventArgument.StartsWith("2:") Then
    '---strips away the command
    eventArgument = eventArgument.Substring(2)
    '---get states and cities related to country
    Select Case eventArgument
        Case "Sing" : _result = "Singapore,"
        Case "US" : _result = _
            "Alabama,California,Maryland,Massachusetts,New
York,Oklahoma,Wisconsin,"
        Case "UK" : _result = _
            "Birmingham,Cambridge,Christchurch,Leeds,Sheffield,"
        Case Else
            _result = ""
    End Select
Else
    _result = "Command not recognized"
End If
End Sub

```

مثال ۶-۱. متد RaiseCallbackEvent()

5- توجه کنید تابع RaiseCallbackEvent() پس از اجرا یک رشته (String) را باز می گرداند. بنابراین چنانچه می خواهید داده های مرکب (complex) از کلاینت به سرور (و بالعکس) انتقال دهید، بایستی داده مرکب را به رشته تبدیل کنید و سپس ارسال کنید.

پارامتر EventArgument از سمت مشتری ارسال می شود. برای بازیابی ایالت و شهر بر اساس Zip Code پارامتر EventArgument مشابه زیر می باشد:

1:02140

که در آن 1 دستور و 02140 Zip Code را مشخص می کند . برای بازیابی تمامی ایالتها و شهرهای کشور بر اساس نام کشور Event Agument بصورت زیر خواهد بود :

2: US

مشابه قبلی 2 مشخص کننده دستور و US مشخص کننده کشور است .

نتیجه در متغیر خصوصی _result ذخیره می شود . برای بازگرداندن نتیجه به مشتری از تابع (getcallback result) استفاده می کنیم :

```
Public Function GetCallbackresult() As String _
    Implements System.Web.UI.ICallbackEventHandler.GetCallbackResult
    If _result = "error" Then
        Throw (New Exception("ZipCode not valid!"))
    Else
        Return _result
    End If
End Function
```

نکته : توجه کنید که در دستور اول ، نتیجه (شهر و ایالت) توسط کاما از هم جدا شده اند . مانند : Scbastpol,CA . همچنین در دستور دوم نیز خروجی (شهرها و ایالتها) توسط کاما از هم جدا شده اند ، مانند : Alabama,California, Maryland,Massachusetts,New York,Oklahoma,Wisconsin

6-در رویداد Page-load صفحه ویداد onclick دکمه "GetCity and State" و رویداد onchange کنترل County drop-dow list را بصورت زیر تعریف کنید ، همچنین از متد (GetCallbackEventReference در کلاس (ClientScriptmanager جهت انجام Callback استفاده کنید .

کلاس (ClientScriptmanager اعمالی همچون لیستهای سمت مشتری را کنترل میکند . از طریق خاصیت Client Script صفحه می توانید به کلاس ClientScriptmanager دسترسی داشته باشید .

```
Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles Me.Load
```

```
'---prevent the Get City & State button from postback
btnGetCityandState.Attributes.Add("onClick", "return false")
```

```
ddlCountry.Attributes.Add("onChange", "GetStatesFromServer( )")
callbackStr = Page.ClientScript.GetCallbackEventReference(_
    Me, "Command", "CallBackHandler", _
    "context", "onError", False)
End Sub
```

7- پارامترهای () getcallback Event Refrence عبارتند از :

Control (control)

ارجاعی به شی فراخواننده متد

Argument (String)

رشته ارسالی به سرور

ClientCallBack (String)

ارجاعی به کنترل کننده Call back در سمت سرور

Context (String)

نشانه ای جهت شناسایی آغازگر Call back

ErrorCallback (String)

ارجاعی به کنترل کننده Call back هنگام رخ دادن خطا

Async (Boolean)

زمانی که بایستی CallbackScript بصورت غیر همزمان (asynchron) فراخوانی شود از این پارامتر استفاده می شود .

متغیر CallbackStr رشته زیر را هنگام لود شدن فرم نگهداری می کند :

```
WebForm_DoCallback('_ _Page',Command,
    CallBackHandler,context,onError,false)
```

آنچه در اینجا مهم است این است که **Command** به رشته ای اشاره می کند که قرار است به سرور ارسال شود ، در حالیکه **CallBackHandler** تابعی است که هنگام بازگشت نتیجه از سرور (در سمت کلانیت) فراخوانی می شود .

8- در این مرحله توابع مورد نیاز در سمت مشتری را تعریف می کنیم . به Source View صفحه بروید و اسکرپت های مثال ۶-۲ را به صفحه اضافه کنید :

```
...
<head runat="server">
    <title>Untitled Page</title>
</script>
function GetStateFromZip( ){
    var Command = "1:" + document.forms[0].elements['txtZipCode'].value;
    var context = new Object( );
    context.CommandName = "GetStateFromZip";
    <%=callbackStr%>
}

function GetStatesFromServer( ) {
    var Command = "2:" + document.forms[0].elements['ddlCountry'].value;
    var context = new Object( );
    context.CommandName = "GetStatesFromCountry";
    <%=callbackStr%>
}

function CallBackHandler(result, context) {
    if (context.CommandName == "GetStateFromZip" ) {
        var indexofComma = result.indexOf(",");
        var City = result.substring(0,indexofComma);
        var State = result.substring(indexofComma+1,result.length);
        document.forms[0].elements['txtState'].value = State;
        document.forms[0].elements['txtCity'].value = City;
    } else
    if (context.CommandName == "GetStatesFromCountry")
    {
        document.forms[0].elements['ddlState'].options.length=0;
        while (result.length>0) {
            var indexofComma = result.indexOf(",");
            var State = result.substring(0,indexofComma);
            result = result.substring(indexofComma+1)
```

```

    opt = new Option(State,State);
    document.forms[0].elements['ddlState'].add(opt);
  }
}
}

function onError(message, context) {
  alert("Exception :\n" + message);
}

</script>
</head>
...

```

مثال ۶-۲. توابع مورد نیاز در سمت مشتری

۹-متدهای () getStateFromZip و () getStateFromServer در واقع درخواستهایی را برای ارسال به سرور تنظیم می کنند، در این حالت مقدار موجود در کنترل text Box (و کنترل drop-dow list) گرفته شده و درون رشته CallbackStr قرار می گیرد . عبارت <%=CallbackStr %> رشته تولید شده را درون تابع قرار می دهد ، بطوریکه در زمان اجرا به این صورت در می آید :

```

function GetStateFromZip( ){
  var Command = "1:" + document.forms[0].elements['txtZipCode'].value;
  var context = new Object( );
  context.CommandName = "GetStateFromZip";
  WebForm_DoCallback('__ _Page',Command,CallBackHandler,context,onError,
false)
}

function GetStatesFromServer( ){
  var Command = "2:" + document.forms[0].elements['ddlCountry'].value;
  var context = new Object( );

```

```
context.CommandName = "GetStatesFromCountry";
WebForm_DoCallback('_ _Page',Command,CallBackHandler,context,onError,
false)
}
```

10- دقت کنید که هر دو تابع فراخوانی را به متد () CallBackHandler باز می گردانند. متد () CallBackHandler نیز هنگامیکه سرور نتیجه را به کلاینت ارسال میکند، فراخوانی می شود. از اینرو باید بدانید کدام فراخوان باعث بازگشت شده است. از متغیر Context جهت تعیین نام برای هر کدام فراخوانیها (GetStateFromZip یا GetStatesFromCountry) استفاده می شود.

نتیجه توسط متغیر Result بازگشت داده می شود، سپس نتیجه به کنترلهای روی صفحه ارسال و نمایش داده می شود.

11- برای تکمیل برنامه تابع () GetStateFromZip را در کنترل Button فراخوانی کنید :

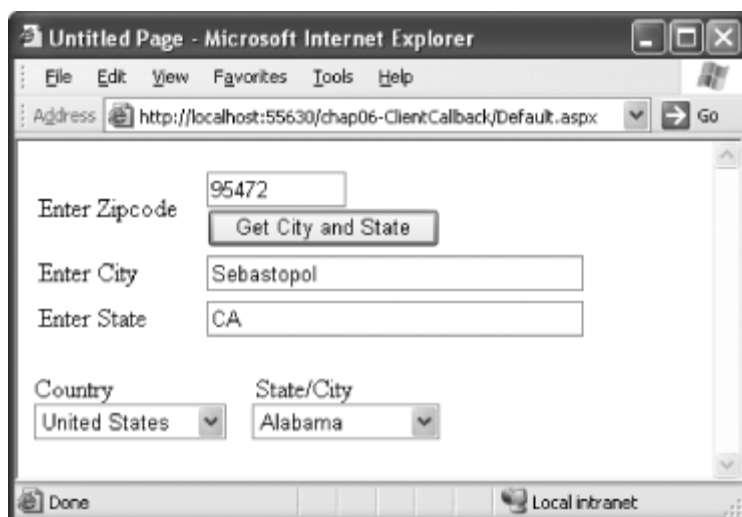
```
<asp:Button ID="btnGetCityandState"
OnClientClick="GetStateFromZip( )"
runat="server" Text="Get City and State"
Width="144px" />
```

12- تابع () GetStateFromZip را قبلا در رویداد Page_Load بصورت زیر فراخوانی کردیم :

```
ddlCountry.Attributes.Add("onChange", "GetStatesFromServer( )")
```

13- توسط این کد با تغییر آیتمهای Drop-Down List تابع () GetStateFromZip فراخوانی میشود.

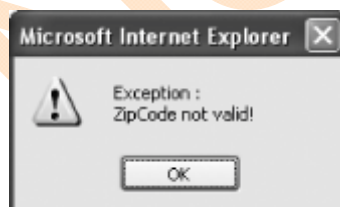
14- کلید F5 را فشار دهید. اکنون بدون Post Back به سرور دسترسی دارید. (شکل ۶-۱۵)



شکل ۶-۱۵. استفاده از Callback Manager برای اجتناب از Post Back

نکته: **Java Script** نسبت به کوچک و بزرگ بودن حروف حساس است، از اینرو در نوشتن کدها، خصوصا نام کنترلها دقت کنید.

چنانچه بدون وارد کردن Zip Code دکمه "Get City and State" را کلیک کنید، پیام خطایی برایتان نمایش در می آید. (شکل ۶-۱۶) این عمل توسط تابع `OnError()` در سمت مشتری کنترل می شود.



شکل ۶-۱۶. نمایش پیغام خطا

۶-۵-۲. پاورقی بخش...

...شناسایی مرورگر هایی که از Client Callback پشتیبانی میکنند

چنانچه قبلا اشاره شد برای استفاده از Client Callback به IE 5.0 یا بالاتر نیاز دارید. با استفاده از خاصیت `SupportsCallback` می توانید بفهمید مرورگر از Client Callback پشتیبانی میکند یا خیر؟

```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load
```

If Request.Browser.SupportsCallback Then

'---prevent the Get City & State button from postback

btnGetCityandState.Attributes.Add("onClick", "return false")

ddlCountry.Attributes.Add("onChange", "GetStatesFromServer()")

callbackStr = Page.ClientScript.GetCallbackEventReference(Me,

"Command", _ "CallBackHandler", "context",

"onError", True)

Else

Response.Write("Your browser does not support Client Callback.")

End If

End Sub

۶-۵-۳. منابع اطلاعاتی :

برای آشنایی بیشتر با Client CallBack در Asp.Net 2.0 مقاله زیر را مطالعه کنید :

<http://msdn.microsoft.com/msdnmag/issues/04/12/CuttingEdge/>

برای آشنایی با نحوه کارکرد XMLHTTP ActiveX object به آدرس زیر مراجعه کنید :

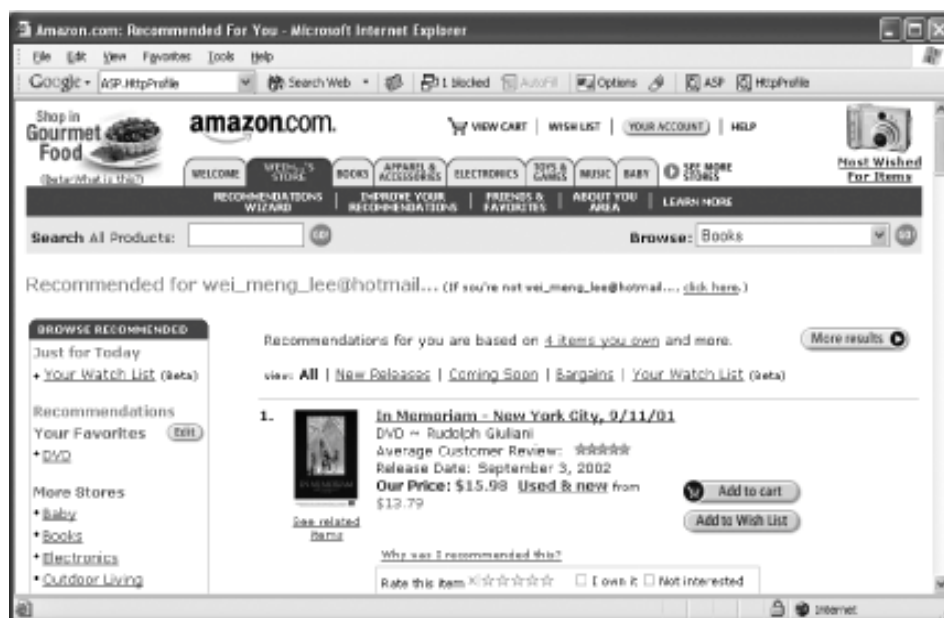
<http://www.devx.com/getHelpOn/10MinuteSolution/20358>

فصل هفتم

پروفایلها

(Profiles)

شخصی سازی (Personalizing) راهی است برای افزایش کاربران سایت . شخصی سازی این امکان را به بازدید کنندگان می دهد تا اطلاعاتشان را حفظ نمایند و در رجوع مجدد به سایت از آنها استفاده کنند ؛ به عنوان مثال Amazon.com اطلاعات کاربران از جمله لیست کالاهای خریداری شده آنها را نگهداری می کند تا ظاهری مناسب از سایت بر اساس علائق کاربر برایش فراهم آورد . (شکل ۷-۱ را ببینید)



شکل ۷-۱- سایت Amazon.com

۷-۱- شخص سازی برنامه های وب

یادداشت : از پروفایلها (profiles) جهت نگهداری تنظیمات کاربران سایت استفاده کنید .

در فصل پنجم با قابلیت جدید Asp.net Membership Provider آشنا شدید و دیدید که چگونه با استفاده از آن می توانید کاربران سایت را به آسانی مدیریت کنید .

Membership Provider مدیریت ورود کاربران (User Login) را بسیار آسان نموده است . اما این قابلیت برخی اطلاعات کاربران از جمله ، تنظیمات آنها برای تمها (Themes) یا خریدهای قبلی آنها از سایت را نگهداری نمی کند .

برای نگهداری تنظیمات کاربر ، می توانید از سریس جدید پروفایل در ASP.NET 2.0 استفاده نمایید .

سرویس پروفایل را مکانیزمی برای نگهداری اطلاعات کاربر در نظر بگیرید همانند : شیء session ؛ با این تفاوت که session مدت زمان خاصی اطلاعات را نگهداری می کند و با منقضی (Expire) شدن session و یا پاک شدن آن

اطلاعات همراه آن نیز از بین می روند ، در حالیکه اطلاعات موجود در پروفایل ها تازمانیکه خودتان آنها را از منبع اطلاعاتی پاک نکنید باقی می ماند .

نکته : چنانچه قبلاً برنامه نویس Asp.Net 1.x بوده اید ، با شیء session به اندازه کافی آشنا هستید، از session ها برای انتقال مقادیر بین صفحات استفاده می شود .

علاوه بر این شیء پروفایل نسبت به شیء session چندین برتری دیگر نیز دارد از جمله :

- پایداری (Nonvolatility)

پروفایلها در منابع اطلاعاتی (Data Store) ذخیره می شود در حالیکه متغیرهای نوع session در حافظه قرار می گیرند .

- پیاده سازی مؤثر و مفید

پروفایلها تنها در هنگام نیاز لود می شوند ، در حالیکه تمامی session ها هنگامیکه به یکی از آنها دسترسی صورت می گیرد بارگذاری می شوند .

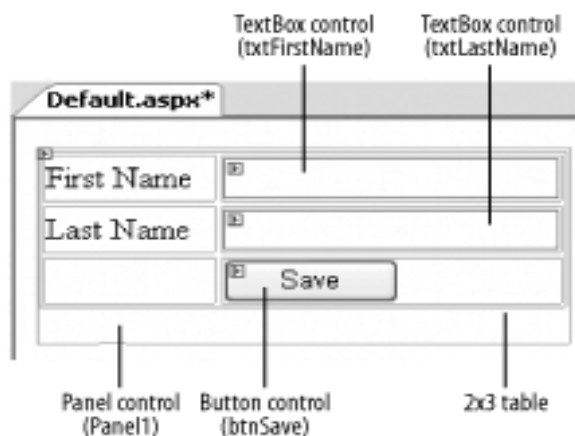
در Asp.Net 20 شیء جدید پروفایل اجازه می دهد تا داده های کاربر به شیوه بهتری نگهداری شوند .

۷-۱-۱- مراحل انجام کار :

جهت استفاده از اشیاء پروفایل برای نگهداری اطلاعات کاربران به روشی جهت شناسایی کاربران نیاز دارید ، به عنوان مثال نام کاربری (User Name) کاربران می تواند به عنوان کلیدی برای بازیابی اطلاعات ذخیره شده کاربران ، بکار رود . در این بخش برنامه ای طراحی می کنیم ، که در اولین ورود کاربر به سایت نام و نام خانوادگی کاربر را گرفته و ذخیره می کند . متعاقباً ، هنگام ورود مجدد کاربر به سایت می توانید نامش را از پروفایل او بازیابی کنید .

1. VS 2005 را باز کنید و پروژه وب جدیدی با نام C:\ASPNET20\chap07-Profile ایجاد کنید .

2. یک کنترل panel به صفحه اضافه کنید . سپس یک جدول با ۳ سطر و ۲ ستون به panel اضافه کنید . (Insert layout → table) نهایتاً کنترلهای موجود در شکل ۷-۲ را به جدول اضافه کنید .



شکل ۷-۲- طراحی صفحه به کمک کنترل‌های مختلف

3. برای استفاده از شیء پروفایل ابتداء بایستی یک فایل پیکربندی (Web Configuration File) به پروژه اضافه کنید . بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های... Add new item سپس Web Configuration File را انتخاب کنید .

4. عنصر < profile > را به صورت زیر به فایل Web.Config اضافه کنید :

```
<system.web>
  <profile>
    <properties>
      <add name="FirstName" type="System.String"/>
      <add name="LastName" type="System.String"/>
    </properties>
  </profile>
```

...

5. در اینجا دو خاصیت First name و Last name به پروفایل اضافه کردید ؛ که از آنها به ترتیب برای ذخیره نام و نام خانوادگی کاربر استفاده می شود . همان طور که مشاهده می کنید در این خواص نوع داده (Data Type) هم مشخص شده است ، شما می توانید از تمامی داده های موجود در Framework دات نت استفاده کنید. برای دسترسی به هر کدام از این خواص کافست کلمه کلیدی (keyword) profile را به ابتدای آنها اضافه کنید. به صورت Profile.FirstName و Profile.LastName

خواص شیء profile

به جز نام و نوع (هر نوع داده در دات نت؛ به طور پیش فرض string) از خواص زیر نیز می توانید بهره گیرید :

readOnly

مشخص می کند که خاصیت مربوطه Readonly است یا خیر.

serializeAs

مشخص کننده نحوه ذخیره خاصیت در پایگاه داده است. مقادیر ممکن، مقدار String (پیش فرض)، Binary و ProviderSpecific هستند.

Provider

نام سرویس دهنده (provider) پروفایل را مشخص می کند.

defaultValue

مقدار پیش فرض خاصیت را نگهداری می کند.

allowAnonymous

به کاربران ناشناس اجازه ذخیره مقادیر را می دهد.

6. بر روی فرم کلیک مضاعف کنید و به Code_behind صفحه بروید. در رویداد page_Load ابتدا چک کنید که خاصیت First name خالی نباشد، اگر خالی نیست، نام و نام خانوادگی را از خواص ذخیره شده در پروفایل بازیابی کنید. اگر First name خالی باشد، فرض بر این است که کاربر اولین باری است که از صفحه دیدن می کند، پس می توانید برای ورود نام و نام خانوادگی، کنترل panel را نمایش دهید:

```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load
    If Profile.FirstName <> "" Then
        Panel1.Visible = False
        Response.Write ("Welcome back, " & _
            Profile.FirstName & ", " & _
            Profile.LastName)
    Else
        Panel1.Visible = True
    End If
End Sub
```

7. هنگامی که کاربر دکمه "save" را کلیک می کند، نام و نام خانوادگی کاربر در خواص تعریف شده پروفایل ذخیره می شوند. کد مربوط به رویداد کلیک دکمه "save" به این صورت است:

```
Protected Sub btnSave_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles btnSave.Click
```

```
' save the profile
```

```
Profile.FirstName = txtFirstName.Text
```

```
Profile.LastName = txtLastName.Text
```

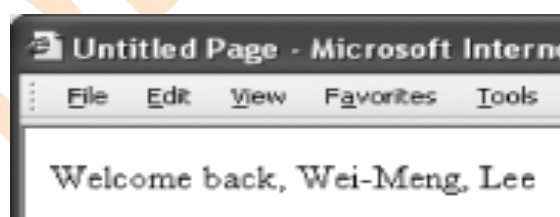
```
Response.Write("Thank you, " & _
    txtFirstName.Text & ", " & txtLastName.Text)
```

```
End Sub
```

8. کلید F5 را برای تست برنامه فشار دهید . شکل ۷-۳ صفحه را در اولین ورود کاربر به سایت نشان می دهد . کاربر می تواند نام و نام خانوادگی خود را وارد کند و توسط دکمه "save" آنها را در پروفایل مخصوص خودش ذخیره کند . هنگامیکه کاربر مجدداً به سایت بازمی گردد نام و نام خانوادگی او بازیابی شده به همراه پیغام خوش آمد گویی نمایش داده می شوند . (شکل ۷-۴)



شکل ۷-۳. نمایش صفحه برای بار اول



شکل ۷-۴. صفحه مراجعات بعدی کاربر

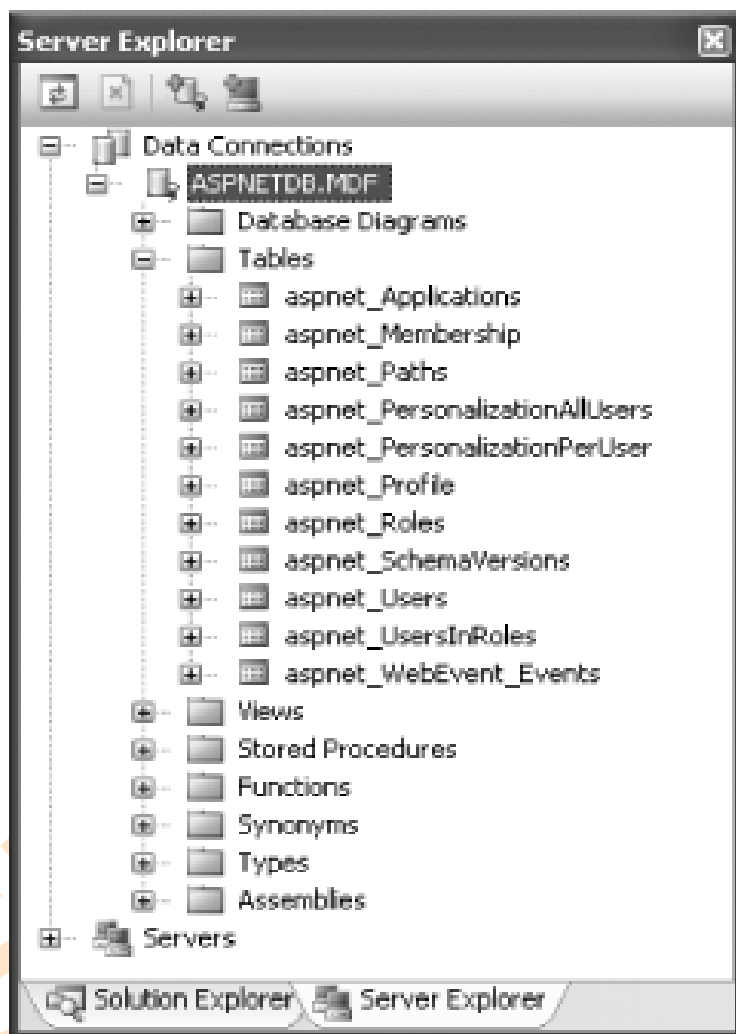
۷-۱-۲- تشریح عملکرد برنامه :

هنگامیکه برنامه را اجرا می کنید ، نام و نام خانوادگی کاربر در خواص تعریف شده پروفایل ذخیره می شوند . اما در واقع این مقادیر در کجا نگهداری می شوند ؟

Asp.net 2.0 از یک profile provider برای ذخیره داده ها در شیء پروفایل استفاده می کند . بطور پیش فرض این سرویس دهنده sql server 2005 Express است .

1. برای بررسی مقادیر ذخیره شده توسط پرو فایل ، بر روی فایل *ASPNETDB.MDF* در شاخه *App_Data* پروژه کلیک مضاعف کنید .

2. پایگاه داده *ASPNETDB.MDF* را از بخش *Data Connections* در *Server Explorer* می توانید ببینید (شکل ۵-۷) علامت (+) کنار *tables* را کلیک کنید تا جدولها نمایش داده شوند .



شکل ۵-۷- بررسی پایگاه داده aspnetdb

3. بر روی جدول *aspnet_Profile* راست کلیک کنید و گزینه "Show Table Data" را انتخاب کنید . مقادیر ذخیره شده در جدول را در شکل ۶-۷ می توانید مشاهده کنید .

aspnet_Profile...\ASPNETDB.MDF)				
UserId	PropertyNames	Property/ValuesS...	Property/ValuesBi...	LastUpdatedDate
2b922fb5-1c81-48c...	FirstName:S:0:8...	Wei-MengLee	<Binary data>	4/7/2005 8:22:5...

شکل ۷-۶- بررسی جدول aspnet_Profile

4. به فیلد (filed) UserId از جدول aspnet_Profile دقت کنید . این فیلد به پروفایل مخصوص هر کاربر در برنامه اشاره می کند . چنانچه جدول aspnet_Users را باز کنید ، (شکل ۷-۷) مشاهده می کنید که فیلد UserId با فیلدهای ApplicationID (که مشخص کننده برنامه است) و فیلد Username (که کاربر را مشخص می کند) در ارتباط است .

aspnet_Users..ASPNET20.MDF						
ApplicationId	UserId	UserName	LoweredUserName	MobileAlias	IsAnonymous	LastActivityDate
70a62b0b-c237-...	2b922fb9-1c81-...	WINXP2\Wei-Me...	winxp2\wei-men...	/NULL	False	4/7/2008 8:22:5...

شکل ۷-۷- بررسی جدول aspnet_Users

5. دقت کنید که مقدار فیلد Username برابر است با " WINXP2\Wei-Meng Lee " . از آنجا که در این برنامه از تأیید هویت پیش فرض (Windows authentication) استفاده کردیم . نام کاربری مربوط به ویندوز به ابتدای نام کاربر در سایت اضافه شده است .

۷-۱-۳- پاورقی بخش ...

... گروه بندی خواص مرتبط (وابسته) به یکدیگر

فرض کنید می خواهید آدرس مربوط به یک کاربر را ذخیره کنید ؛ به جای استفاده از ساختار زیر :

Profile.Line1 = ""

Profile.Line2 = ""

Profile.State = ""

Profile.Zip = ""

منطقی تر است که تمامی اطلاعات مربوط به هم را در یک گروه قرار دهید ، مانند زیر :

Profile.Address.Line1 = ""

Profile.Address.Line2 = ""

Profile.Address.State = ""

Profile.Address.Zip = ""

برای این کار کافیت از عنصر <group> در فایل web.config استفاده کنید :

```

<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <system.web>
    <profile>
      <properties>
        <add name="FirstName" type="System.String" />
        <add name="LastName" type="System.String" />
        <add name="LastLoginTime" type="System.DateTime" />
        <group name="Address">
          <add name="Line1" type="System.String" />
          <add name="Line2" type="System.String" />
          <add name="State" type="System.String" />
          <add name="Zip" type="System.String" />
        </group>
      </properties>
    </profile>
  </system.web>
</configuration>

```

۷-۱-۴- منابع اطلاعاتی :

ممکن است بخواهید به جای استفاده از سرویس دهنده های پیش فرض ، از provider دیگری استفاده کنید ، مثلاً به جای پایگاه داده sql server اطلاعات پروفایل ها را در اسناد XML نگهداری کنید ؛ در این راستا مقالات زیر می توانند به شما کمک کنند :

- <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspnet/html/asp02182004.asp>
- <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspnet/html/asp04212004.asp>

۷-۲- شناسایی کاربران از طریق Forms Authentication

یادداشت: **Forms authentication** بهترین شیوه شناسایی کاربران در برنامه های وب است. این حالت با سرویس پروفایل نیز به خوبی کار می کند.

در بخش ۷-۱ نحوه استفاده از windows authentication در برنامه های وب Asp.net را مشاهده کردید. اگرچه این روش در برنامه های تحت اینترنت (Interanet) مفید است اما روش بهتر برای شناسایی کاربری خارجی استفاده از شناسایی مبتنی بر فرمها (forms authentication) است. در این بخش از اشیاء پروفایل به همراه forms authentication استفاده خواهیم کرد.

forms authentication در مقابل windows authentication

در تأیید هویت مبتنی بر فرمها کاربر از طریق نام کاربری و کلمه عبور مخصوص خود به سایت وارد (login) می شود (forms authentication) و هر گونه درخواست ناشناس باعث هدایت درخواست کننده به یک صفحه خاص می شود. (این صفحه را برنامه نویس طراحی می کند و یک صفحه معمولی است مثل صفحه login.aspx). پس از ورود یک کوکی برای کاربر ساخته می شود که در واقع حکم اعتبار نامه کاربر را دارد و مادامیکه این کوکی وجود دارد کاربر برای Asp.net شناخته شده است. درخواستها بر اساس کوکی ایجاد شده و در سرآیندهای درخواست (request headers) کنترل می شوند. سپس این درخواستها توسط یک ASP.NET event handler شناسایی می شوند. چنانچه به هر دلیلی کوکی ایجاد شده از بین برود (مثل اتمام زمان حیات و یا حذف توسط برنامه نویس) کاربر ناشناس محسوب شده و با اولین درخواست به صفحه ای که در ابتداء اشاره شد هدایت می شود.

در تأیید هویت مبتنی بر ویندوز (Asp.net (windows authentication) بر اساس مدل شناسایی IIS عمل میکند. شناسایی ایجاد شده توسط IIS به یکی از این سه شیوه است: basic، digest و یا Integrated Windows Authentication. هنگامی که شناسایی IIS کامل شد Asp.Net از شناسه تأیید شده برای دادن اجازه دسترسی استفاده می کند.

ساخت چندین حساب کاربری در اینترنت برای دسترسی کاربران به سایت غیر ممکن است، از این رو در برنامه هایی که در اینترنت قرار می گیرند بایستی از forms authentication استفاده کنید.

۷-۲-۱- مراحل انجام کار :

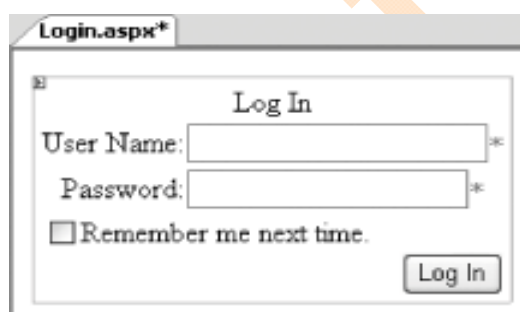
در این بخش با نحوه شناسایی کاربران با استفاده forms authentication آشنا می شوید . سپس با استفاده از WAT (ASP.NET Web Site Administration Tool) کاربر به سایت اضافه می کنید و نهایتاً از طریق شیء پروفایل تنظیمات کاربر را بر اساس نام کاربری وارد شده ذخیره می کنید .

1. پروژه بخش قبل (C:\ASPNET20\chap07-Profile) را باز کنید و شاخه جدیدی به آن اضافه کنید . (برروی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های New folder → regular folder را انتخاب کنید) نام شاخه را Members قرار دهید .

2. صفحه Default.aspx را به شاخه Members انتقال دهید .

3. یک فایل web.config جدید به شاخه Members اضافه کنید .

4. در نهایت فرم جدیدی با نام login.aspx به پروژه اضافه کنید (شکل ۷-۸) و یک کنترل login در آن قرار دهید .



شکل ۷-۸. صفحه login به همراه کنترل login

5. هم اکنون Solution Explorer پروژه شبیه شکل ۷-۹ شده است .



شکل ۷-۹. پروژه به همراه دو فایل web.config

6. در web.config برنامه (فایل شماره ۱ در شکل ۷-۹) حالت authentication را از windows به forms تغییر

دهید :

```
<authentication mode="Forms"/>
```

مقادیر پیش فرض در فایل Machine.Config

در Asp.net 1.x هنگام استفاده از شناسایی مبتنی بر فرمها می بایست نام صفحه login را به صورت واضح در فایل web.config مشخص کنید . همانند کد زیر :

```
<authentication mode="Forms">
  <forms name=".ASPXAUTH"
    loginUrl="login.aspx"
    protection="Validation"
    timeout="999999" />
</authentication>
```

پس از آن اگر کاربر ناشناسی صفحه حفاظت (protect) شده ای را درخواست می کرد، برای شناسایی کاربر را به صفحه login.aspx هدایت می کردید .

در Asp.net 2.0 نیازی به انجام این مرحله نیست ، تنظیمات بالا (به علاوه بسیاری تنظیمات دیگر) بطور پیش فرض توسط Asp.net اعمال می شوند ، در Asp.net 2.0 هر برنامه سه فایل *machine.config* متفاوت دارد که هر سه در آدرس *C:\WINDOWS\Microsoft.NET\Framework\<version>\CONFIG* قرار دارند :

machine.config

machine.config.default

machine.config.comments

فایل *machine.config.default* شامل تمامی تنظیمات پیکربندی پیش فرض سیستم است . برای مشاهده جزئیات تنظیمات پیش فرض تعریف شده در *machine.config.default* ، فایل *machine.config.comments* را باز کنید .

بعنوان مثال تنظیمات پیش فرض مربوط به forms authentication در فایل *machine.config.comments* به صورت زیر است :

```
<forms
```

```

name=".ASPXAUTH"
loginUrl="login.aspx"
protection="All"
timeout="30"
path="/"
requireSSL="false"
slidingExpiration="true"
defaultUrl="default.aspx"
cookieless="UseCookies"
enableCrossAppRedirects="false" >

```

برای بازنویسی مقادیر پیش فرض باید مقادیر موجود در فایل Machine.Config و یا web.config را تغییر دهید . اساس کار فایل‌های پیکربندی به این صورت است که چنانچه فایل web.config در برنامه وجود نداشته باشد برنامه تنظیماتش را از سطح بالاتر آن یعنی Machine.Config ارث می برد . دلیل تقسیم فایل Machine.Config به سه فایل در Asp.net 2.0 کاهش حجم فایل Machine.Config و بهبود عملکرد است .

7. در فایل web.config شاخه Members (فایل شماره ۲ در شکل ۷-۸) عنصر <authorization> را اضافه کنید ، تا از دسترسی کاربران ناشناس به صفحات این شاخه جلوگیری کنید . همچنین خط مربوط به عنصر <authentication> را حذف کنید :

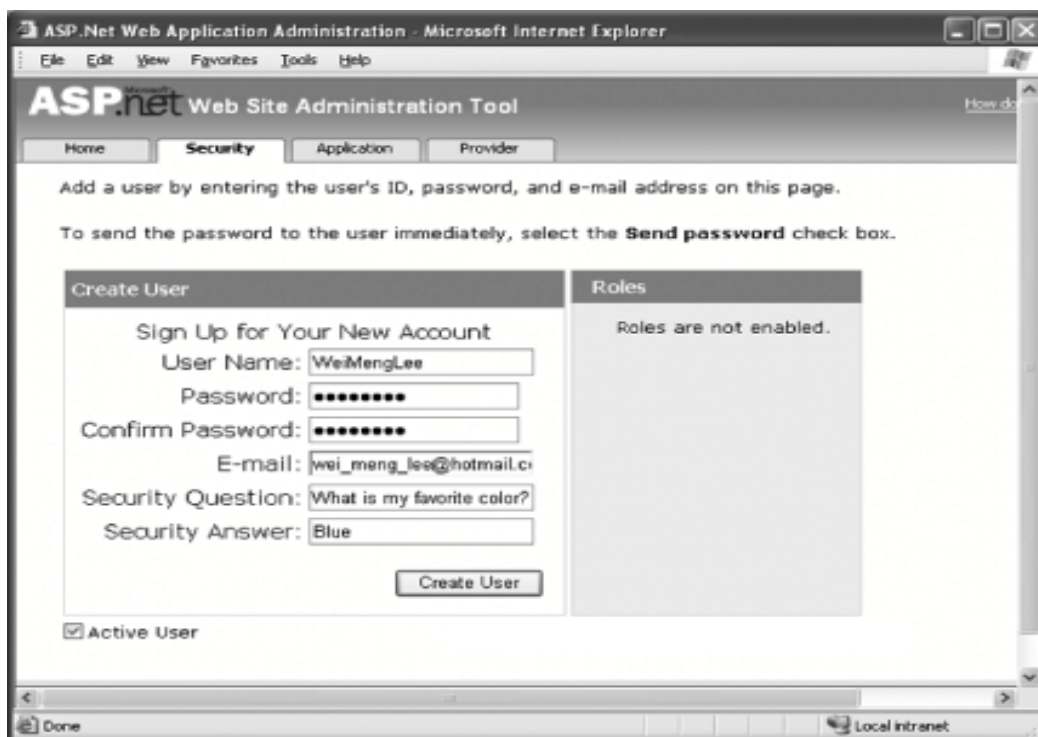
```

<system.web>
  <authorization>
    <deny users="?" />
  </authorization>
  ...
  <!--
  <authentication mode="Windows"/>
  -->

```

نکته : برای بیان توضیحات در فایل‌های پیکربندی آنها را داخل بلوک <-- comments --> قرار دهید . خطوطی که در این بلوک قرار می گیرند قابلیت اجرا ندارند .

8. با استفاده از WAT (website → Asp.net configuration → Security Tab) یک کاربر جدید به برنامه اضافه کنید . (شکل ۷-۱۰)



شکل ۷-۱۰ افزودن کاربر جدید به برنامه

نکته: در فصل پنجم نحوه افزودن کاربر به برنامه بطور کامل تشریح شد.

9. برای اجرای برنامه کلید F5 را فشار دهید. در مرورگر صفحه Default.aspx شاخه Member را لود کنید. تا زمانیکه توسط سایت ناشناس (Anonymous) شناخته شوید به صفحه login.aspx هدایت می شوید.
10. با نام کاربری ایجاد شده در مرحله ۸ به سایت وارد شوید. اکنون می توانید صفحه Default.aspx را لود کنید. همانند قبل نام و نام خانوادگی تان را وارد کنید و دکمه save را کلیک کنید.
11. بار دیگر جداول aspnet_Profile و aspnet_Users را بررسی کنید. همان طور که مشاهده می کنید کاربر جدیدی با نام "WeiMengLee" یعنی همان شناسه کاربری که با آن وارد شدید به جدول اضافه شده است. این شناسه کاربری را با آنچه در بخش قبل به جدول اضافه شد مقایسه کنید. ("WINXP2\Wei-Meng Lee"). در بخش قبل از windows authentication استفاده شده بود و در اینجا از Forms authentication.

۷-۲-۲- پاورقی بخش ...

... دسترسی به صفحه Default.aspx خارج از شاخه Members

شکل ۷-۱۱ فایل Default.aspx را خارج از شاخه Members نشان می دهد.



شکل ۷-۱۱. انتقال Default.aspx به خارج از شاخه Members

حال چنانچه صفحه Default.aspx را لود کنید؛ دیگر به صفحه login.aspx هدایت نمی شوید و صفحه برایتان نمایش داده می شود، اگر نام و نام خانوادگی را وارد کنید و دکمه Save را کلیک کنید، هنگامیکه برنامه قصد مقدار دهی به خواص پروفایل را دارد توسط خطا متوقف می شود. علت هم این است که Asp.net به یک کلید یکتا (unique) برای شناسایی هر کاربر احتیاج دارد. استفاده از شیء پروفایل برای کاربر ناشناس به عنوان anonymous profiling شناخته می شود. در بخش بعد (۷-۳) بیشتر به این مطلب می پردازیم.

بازیابی پروفایل سایر کاربران

با استفاده از متد () Getprofile می توانید به پروفایلهای سایر کاربران دسترسی داشته باشید.

قطعه کد زیر، ابتدا تمامی کاربران سایت را بازیابی می کند، سپس با استفاده از خاصیت UserName و متد () Getprofile پروفایل کاربر را بازیابی می کند:

```
Dim member As MembershipUser
Dim members As MembershipUserCollection = _
    Membership.GetAllUsers
For Each member In members
    Dim userProfile As ASP.HttpProfile
    userProfile = _
        Profile.GetProfile(member.UserName.ToString)
    Response.Write(member.UserName & " - ")
    Response.Write(userProfile.FirstName & ", " & _
        userProfile.LastName & "<br/>")
Next
```

Next

۷-۲-۳- منابع اطلاعاتی :

برای آشنایی با شیوه های ایمن سازی برنامه های وب Asp.net مقاله زیر را مطالعه کنید :

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secmod/html/secmod98.asp>

۷-۳- ذخیره پروفایل کاربران ناشناس (Anonymous Users)

یادداشت : با استفاده از سرویس پروفایل می توانید ، پروفایل مربوط به کاربران ناشناس را نیز نگهداری کنید .

هر چند Asp.Net 2.0 از یک شناسه کاربری (ID) یکتا برای پروفایل هر کاربر استفاده می کند . اما در بسیاری مواقع نگهداری اطلاعات کاربران ناشناس پروفایل ها ، استفاده زیادی دارد . به عنوان مثال در یک فروشگاه الکترونیکی ممکن است کاربران قبل از ورود به سایت بخواهند ، کالا یا کالاهایی را به سبد خریدشان اضافه کنند و پس از ورود به سایت سایر اطلاعات مانند شماره کارت اعتباری ، آدرس و ... را به سیستم وارد کنند .

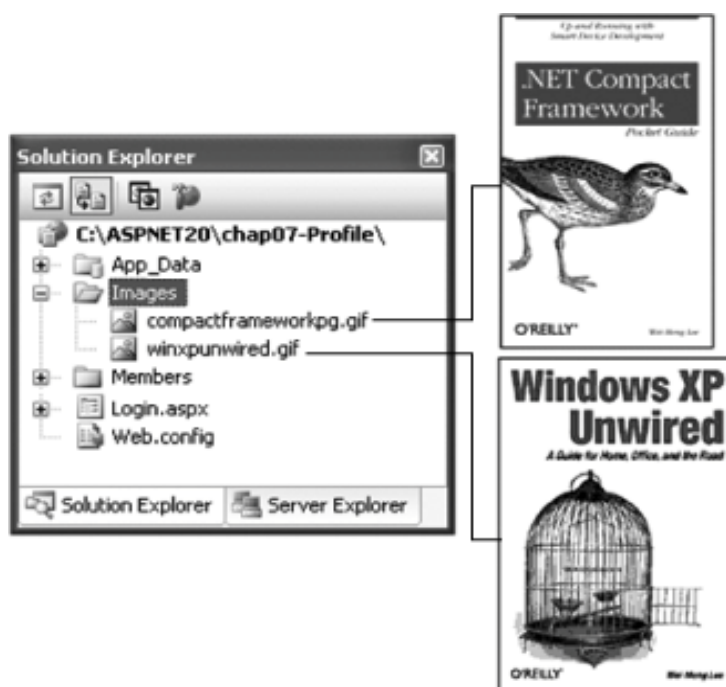
در Asp.net 2.0 با استفاده از یک شناسه یکتای سراسری (Globally Unique Identifier) با نام مستعار GUID می توانید اطلاعات کاربران ناشناس را نگهداری کنید .

۷-۳-۱- مراحل انجام کار :

در این بخش با نحوه ذخیره اطلاعات کاربران ناشناس و همچنین ذخیره داده های مرکب (Complex) در خواص پروفایل آشنا می شوید .

1. پروژه بخش قبل را باز کنید . (C:\ASPNET20\chap07-Profile) شاخه ای با نام Images به پروژه اضافه کنید و عکسهای شکل ۷-۱۲ را به آن اضافه کنید . (عکسها را می توانید از آدرس <http://www.oreilly.com/catalog/aspnetadn> دانلود کنید)

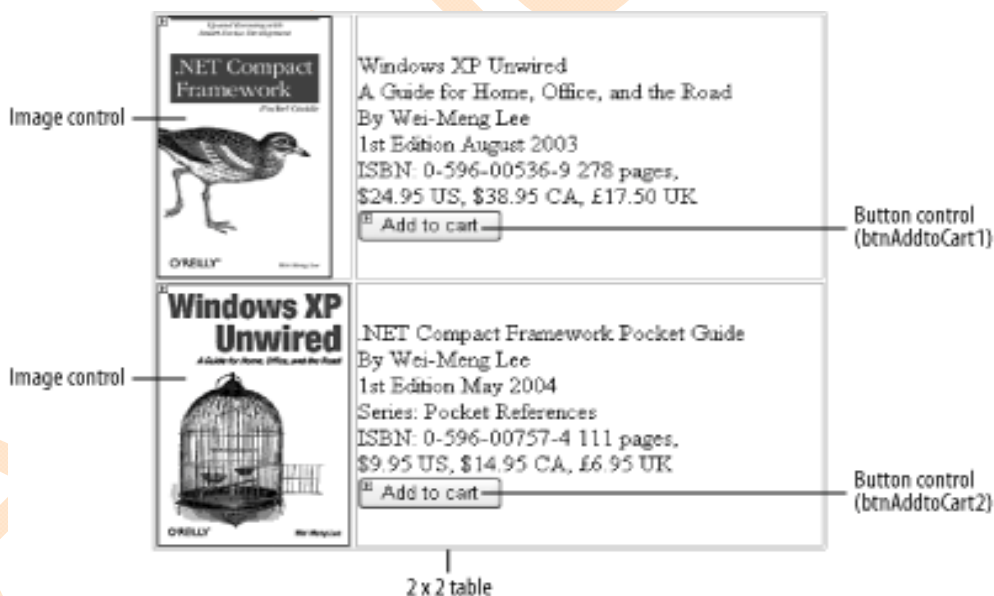
از این عکسها برای نمایش دو کتاب برای فروش استفاده می کنیم .



شکل ۷-۱۲ عکسهای ذخیره شده در شاخه Images

2. فرم جدیدی به پروژه اضافه کنید و نام آن را products.aspx قرار دهید .

3. صفحه products.aspx را همانند شکل ۷-۱۳ طراحی کنید .



شکل ۷-۱۳ طراحی صفحه products.aspx

4. اکنون دو دکمه با نام " Add to cart " به صفحه اضافه کنید . با استفاده از این دکمه ها کاربر می تواند آیتم های مورد نظرش را به سبد خرید اضافه کند . صرف نظر از اینکه به سایت وارد شده است یا خیر . بهترین مکان جهت ذخیره سبد خرید شیء پروفایل است .

5. کلاس جدیدی با نام *ShoppingCart.vb* به پروژه اضافه کنید . از شما خواسته می شود فایل را در شاخه *App_Code* ذخیره کنید . yes را کلیک کنید . این کلاس (*ShoppingCart.vb*) سبد خرید را پیاده سازی می کند .

6. کد کلاس *ShoppingCart.vb* در مثال ۷-۱ آمده است . این کلاس آیتمها را به یک آرایه اضافه می کند .

Imports Microsoft.VisualBasic

Imports System.Xml.Serialization

Namespace OReilly

Public Structure itemType

Dim isbn As String

Dim qty As Integer

End Structure

<XmlInclude(GetType(itemType))> _

Public Class Cart

'---use public for Xml serialization---

Public items As New ArrayList

Public Sub AddItem(ByVal isbn As String, ByVal qty As Integer)

Dim cartItem As New itemType

cartItem.isbn = isbn

cartItem.qty = qty

items.Add(cartItem)

End Sub

End Class

End Namespace

مثال ۷-۱ *ShoppingCart.vb*

7. به منظور دسترسی کاربران ناشناس کد زیر را به فایل *web.config* پروژه اضافه کنید.(شکل ۷-۱۴) در واقع بایستی عنصر *<anonymousIdentification>* را تعریف کنید :



شکل ۷-۱۴- اصلاح فایل web.config پروژه

یادداشت: برای شناسایی ItemType توسط Xmlserializer به خاصیت xmlInclude نیاز دارید.

```
<system.web>
  <anonymousIdentification enabled="true"/>
  <profile>
    <properties>
      <add name="FirstName" type="System.String"/>
      <add name="LastName" type="System.String"/>
      <add name="shoppingcart" allowAnonymous="true"
        type="OReilly.Cart" serializeAs="Xml"/>
    </properties>
  </profile>
  ...
```

یادداشت: دقت کنید که از یک خاصیت ویژه برای دسترسی ناشناس استفاده شده است.

8. نوع خاصیت shoppingcart را OReilly.Cart تعریف کردیم، این نوع به کلاس cart تعریف شده در ShoppingCart.vb اشاره می کند. نحوه ذخیره خاصیت shoppingcart در پایگاه داده نیز xml تعریف شده است.

9. به code_behind صفحه product.aspx بروید و کد مربوط به متد (btnAddToCart_Click) را همانند زیر وارد کنید. این متد سبد خرید کاربر جاری (شناخته شده یا ناشناس) را بازیابی می کند، و سپس آیتم های انتخاب شده را به آن اضافه می کند. نهایتاً سبد خرید در شیء پروفایل ذخیره می شود:

```
Protected Sub btnAddToCart_Click(ByVal sender As Object, _
```

```

        ByVal e As System.EventArgs) _
        Handles btnAddtoCart1.Click, _
        btnAddtoCart2.Click

Dim myCart As OReilly.Cart
'---retrieve the existing cart
myCart = Profile.shoppingcart
If myCart Is Nothing Then
    myCart = New OReilly.Cart
End If

Dim isbn As String
Select Case CType(sender, Button).ID
    Case "btnAddtoCart1" : isbn = "0-596-00536-9"
    Case "btnAddtoCart2" : isbn = "0-596-00757-4"
End Select
myCart.AddItem(isbn, 1)
'---save the cart back into the profile
Profile.shoppingcart = myCart
End Sub

```

10. کد مربوط به رویداد Page_Load را نیز به صورت زیر وارد کنید. از طریق کلاس Membership می توانید چک کنید که کاربر شناخته شده هست یا خیر:

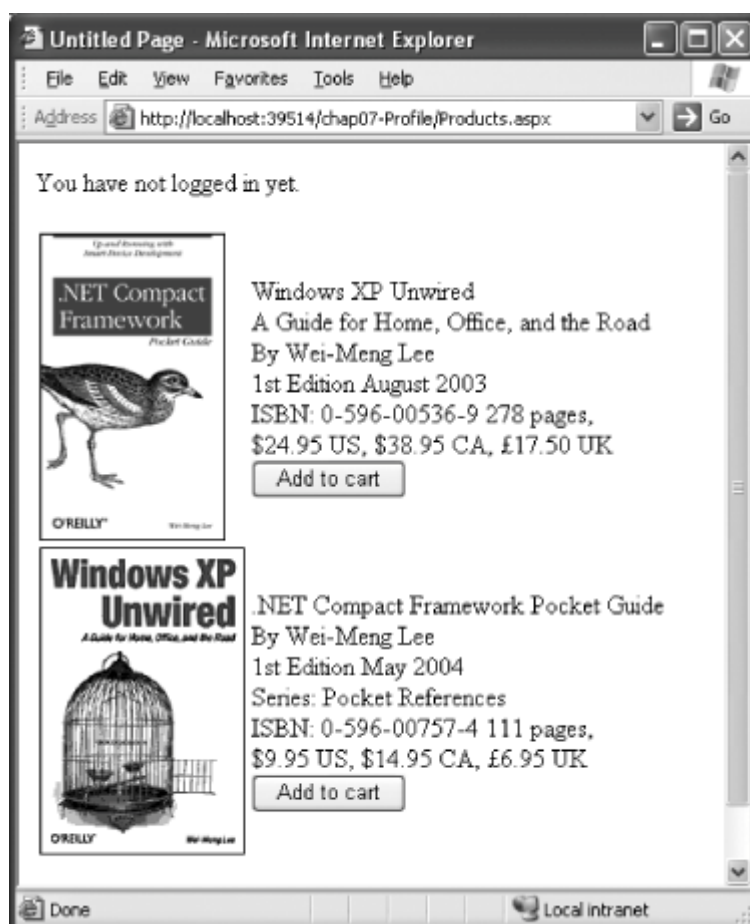
```

Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load

Dim user As MembershipUser = Membership.GetUser
If user Is Nothing Then
    Response.Write("You have not logged in yet.")
End If
End Sub

```

11. برای اجرای برنامه کلید F5 را فشار دهید. در صفحه products.aspx پیغام "You have not logged in yet" نمایش داده می شود. (شکل ۷-۱۵)



شکل ۷-۱۵- نمایش صفحه products.aspx

12. هر دو دکمه "add to cart" را کلیک کنید. آیتم های کلیک شده به پروفایل اضافه می شوند. برای مشاهده اطلاعات ذخیره شده از طریق Server Explorer محتویات جدول aspnet_Profile از فایل ASPNETDB.MDF را مشاهده کنید. دقت کنید که خاصیت ShoppingCart بصورت یک رشته xml ذخیره شده است:

```
<?xml version="1.0" encoding="utf-16"?>
<Cart xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <items>
    <anyType xsi:type="itemType">
      <isbn>0-596-00536-9</isbn>
      <qty>1</qty>
    </anyType>
    <anyType xsi:type="itemType">
      <isbn>0-596-00757-4</isbn>
```

```
<qty>1</qty>
</anyType>
</items>
</Cart>
```

13. حالا محتویات جدول aspnet_Users را چک کنید ، همان طور که مشاهده می کنید کاربر جدیدی به جدول اضافه شده است . این کاربر برای شناسایی کاربر ناشناس ساخته شده است .

شناسه (ID) ناشناس و GUID

چنانچه anonymous identification فعال باشد ، هنگامی که کاربر تأیید هویت نشده اطلاعاتی را در پروفایل ذخیره می کند ، یک شناسه کاربری (userID) مخصوص برایش ساخته می شود . این شناسه همان GUID (Globally Unique Identifier) است که برای هر کاربر به صورت یکتا ساخته می شود .

از طریق Request.AnonymousId می توانید به شناسه کاربر ناشناس دسترسی داشته باشید .

۷-۳-۲- پاورقی بخش ...

... ذخیره ShoppingCart به صورت binary به جای xml

برای ذخیره خاصیت ShoppingCart به صورت باینری ، از خاصیت (serializable) بصورت زیر استفاده کنید :

```
Imports Microsoft.VisualBasic
Imports System.Xml.Serialization
<Serializable(> _
Public Structure itemType
    Dim isbn As String
    Dim qty As Integer
End Structure
```

Namespace OReilly

```
<Serializable(>, _
    XmlInclude(GetType(itemType))> _
Public Class Cart
    '---use public for Xml serialization---
```

Public items As New ArrayList

Public Sub AddItem(ByVal isbn As String, ByVal qty As Integer)

Dim cartItem As New itemType

cartItem.isbn = isbn

cartItem.qty = qty

items.Add(cartItem)

End Sub

End Class

End Namespace

با این کد بجای فیلد PropertyValuesString، فیلد PropertyValuesBinary در جدول aspnet_Profile ذخیره می شود.

۷-۳-۳- منابع اطلاعاتی:

اگر به کلاس cart بیشتر دقت کنید، متوجه می شوید که آرایه مربوط به آیتم های اضافه شده به صورت public تعریف شده است. علت هم این است که (در اینجا) فقط متغیرهایی که به صورت public تعریف شده باشند می توانند به صورت xml ذخیره شوند. اما چنانچه همانند آنچه در بخش ۷-۳-۲ گفته شد اطلاعات را به جای xml به صورت binary ذخیره کنید. برای کسب اطلاعات بیشتر در مورد Binary و XML Serialization به مقالات زیر مراجعه کنید:

- <http://www.ondotnet.com/pub/a/dotnet/2003/10/13/serializationpt1.html>
- <http://www.ondotnet.com/pub/a/dotnet/2003/11/03/serializationpt2.html>

۷-۴- تبدیل پروفایل ناشناس به پروفایل شناخته شده

یادداشت: هنگامیکه یک کاربر ناشناس به سایت وارد می شود، بایستی به صورت دستی خواص پروفایل او را تغییر دهید.

اگر چه Asp.net 2.0 امکان نگهداری اطلاعات کاربران شناخته شده (Authenticated) و ناشناس (anonymous) را می دهد، اما نیازمند راهی برای نگهداری اطلاعات پس از ورود کاربر ناشناس به سایت هستید. به عنوان مثال ممکن است یک کاربر ناشناس چندین آیتم را به سبد خریدش اضافه کرده باشد؛ حال چنانچه این کاربر به سایت وارد شود به اطلاعات وی که قبل از ورود ذخیره شده بودند دسترسی ندارید.

به همین دلیل بایستی اطلاعات ذخیره شده در پروفایل ناشناس را به پروفایل مخصوص کاربر انتقال دهید ، در واقع بایستی به موازات ورود به سایت اطلاعات کاربر نیز به پروفایلش انتقال داده شوند .

۷-۴-۱- مراحل انجام کار :

در این بخش ، که مکمل بخش قبل (بخش ۷-۳) است ، یاد می گیرید چگونه اطلاعات پروفایل ناشناس را به پروفایل شناخته شده انتقال دهید .

1. پروژه بخش قبل را باز کنید . (C:\ASPNET20\chap07-Profile) ، فرم جدیدی به نام Checkout.aspx به شاخه Members اضافه کنید .

2. یک فایل Global.asax به پروژه اضافه کنید . (بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های... Add new Item ... و سپس Global Application Class را انتخاب کنید .

3. اکنون Solution Explorer پروژه همانند شکل ۷-۱۶ شده است .



عکس ۷-۱۶ افزودن فایل Global.asax به پروژه

4. متد زیر را به فایل Global.asax اضافه کنید . رویداد MigrateAnonymous هنگام تغییر وضعیت کاربر از ناشناس به شناخته شده به اجرا درمی آید .

با استفاده از خاصیت AnonymousId میتوانید شناسه کاربری مربوط به کاربر ناشناس را پیدا کنید؛ سپس از طریق متد () GetProfile() پروفایل کاربر ناشناس را بازیابی نمائید. پس از بازیابی پروفایل کاربر کافیهست آن را به پروفایل کاربر شناخته شده منتسب (assign) کنید:

```
Sub Profile_MigrateAnonymous(ByVal sender As Object, _
    ByVal e As ProfileMigrateEventArgs)
    Dim anonymousProfile As ProfileCommon = _
        Profile.GetProfile(e.AnonymousId)
    If anonymousProfile.shoppingcart IsNot Nothing Then
        Profile.shoppingcart = anonymousProfile.shoppingcart
    End If

    '---delete the items associated with the anonymous user
    ProfileManager.DeleteProfile(e.AnonymousID)
    '---clear the anonymous identifier from the request
    ' so that this event will not fire for an authenticated user
    AnonymousIdentificationModule.ClearAnonymousIdentifier( )

End Sub
```

5. در code-behind صفحه checkout.aspx کد زیر را به رویداد form_load اضافه کنید. صفحه checkout.aspx تمامی آیتمهای موجود در سبد خرید را لیست می کند.

```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.Load
    Dim myCart As OReilly.Cart
    myCart = Profile.shoppingcart
    Dim item As OReilly.itemType
    Response.Write("Your cart contains: <br/>")
    For Each item In myCart.items
        Response.Write(item.isbn & " - " & item.qty & "<br/>")
    Next
End Sub
```


6. یک کنترل Button به صفحه products.aspx اضافه کنید و خاصیت Text آن را checkout قرار دهید .
(شکل ۷-۱۷) پس از اینکه کاربر آیتمهای مورد نظرش را به سبد خرید اضافه کرد این دکمه را کلیک می کند تا به صفحه checkout.aspx هدایت شوید . خاصیت PostBackUrl کنترل Button را Members/checkout.aspx قرار دهید .



شکل ۷-۱۷- افزودن دکمه check out

7. برای اینکه برنامه را تست کنید ، ابتدا صفحه products.aspx را لود کنید (بدون ورود به سایت) تعدادی آیتم به سبد خریدتان اضافه کنید و دکمه checkout را کلیک کنید ؛ از آنجا که به دلیل ناشناس بودن اجازه دسترسی به صفحه checkout.aspx را ندارید جهت ورود به سایت به صفحه login.aspx هدایت می شوید .

پس از ورود به سایت رویداد MigrateAnonymous به اجرا درآمده و خواص پروفایل را به پروفایل اصلی انتقال می دهد . در صفحه checkout.aspx نیز تمامی آیتمهای موجود در سبد خرید نمایش داده می شوند .

نکته : برای اینکه ببینید پروفایلها به درستی تعویض شده اند جدول aspnet_profile را چک کنید .

۷-۴-۲ پاورقی بخش ...

... تکلیف داده های ذخیره شده توسط کاربر ناشناس در جدول aspnet_Profile چه می شود ؟ آیا با تغییر پروفایل آنها نیز حذف می شوند ؟

پاسخ منفی است . داده های قبلی جدول را بایستی دستی پاک کنید . خوشبختانه کلاس ProfileManager (که شامل متد DeleteProfile() است) در Asp.net 2.0 شما را در مدیریت پروفایل ها کمک می کند .

۷-۴-۳ منابع اطلاعاتی :

برای آشنایی با کلاس ProfileManager و نحوه گزارش گیری در پروفایلها به MSDN مراجعه کنید . به علاوه برای آشنایی تکمیلی با سرویس پروفایل مقاله زیر را مطالعه کنید .

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvs05/html/userprofiles.asp>

Asp.Net 2.0

فصل هشتم

تمها (Themes)

اسکین ها (skins)

و بومی سازی (localization)

طراحی ظاهر مناسب و زیبا برای سایت و ایجاد واسط کاربری (User Interface) راحت برای کاربر، جزء لاینفک وظایف طراحان وب است. با این حال این کار به ظاهر آسان نیازمند برنامه ریزی و تلاش زیادی است تا در نهایت UI (User Interface) مورد پسند کاربران در اختیار آنها قرار گیرد.

یادداشت: با استفاده از تمها و اسکین ها ظاهر مناسب و زیبایی برای سایتتان طراحی کنید.

۸-۱- ساخت تمها و اسکین ها

اکثر طراحان وب از CSS ها (Cascading Style Sheets) برای ایجاد ظاهر مناسب برای سایت استفاده می کنند.

در Asp.net 2.0 با استفاده از تمها و اسکین ها می توانید ظاهر زیبایی به کنترلها بدهید. در این بخش یاد می گیرید چگونه برای کنترلها اسکین بسازید و چگونه کنترلها را به تمها گروه بندی کنید.

تمها در مقابل CSS ها

اگرچه تمها و CSS ها هر دو اجازه اعمال تنظیمات و خواص مشترکی را به صفحات می دهند، اما تفاوتهایی هم دارند:

- توسط تمها می توانید مشخصات کنترلها را تعریف کنید، مانند عکسهای استفاده شده در یک کنترل ImageMap، در حالیکه توسط CSS شما تنها برخی مشخصات خاص را می توانید تنظیم کنید.
- مقادیر موجود در خواص تمها همیشه جایگزین مقادیر محلی می شوند؛ در حالیکه CSS ها می توانند به صورت آبشاری عمل کنند.

توجه داشته باشید که می توانید از تمها و CSS ها به همراه یکدیگر نیز استفاده کنید.

تمها و اسکین ها به شما اجازه می دهند تا UI برنامه را به چندین لایه تفکیک کنید؛ که در نتیجه قابلیت اداره پروژه افزایش می یابد.

۸-۱-۱. مراحل انجام کار :

در این بخش تعدادی اسکین طراحی می کنیم و آنها در یک تم گروه بندی می کنیم . سپس این تم را به صفحه اعمال می کنیم .

1. VS 2005 را باز کنید و پروژه وب جدیدی ایجاد کنید . نام آن را *C:\ASPNET20\chap08-Themes* قرار دهید .

2. به فرم پیش فرض (*Default.aspx*) یک کنترل label ، Calendar ، و یک Button اضافه کنید . (شکل ۸-۱)



شکل ۸-۱- طراحی فرم توسط کنترل‌های مختلف

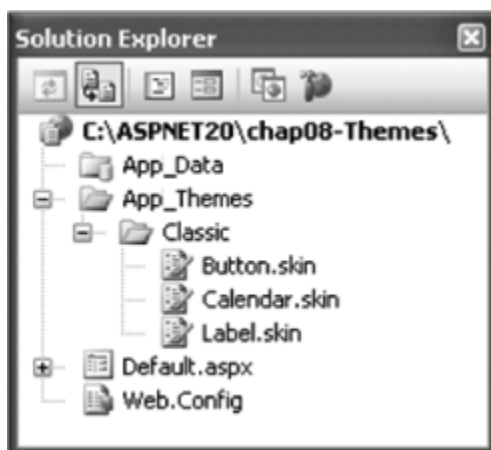
3. پوشه جدیدی به پروژه اضافه کنید ، برای این کار بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های *Add Folder → Themes Folder* را انتخاب کنید . این پوشه محل نگهداری تم‌های برنامه است .

4. چنانچه دقت کنید ، پوشه دیگری به نام *Theme1* درون شاخه *App_Themes* ساخته شده است ، نام این پوشه را به classic تغییر نام دهید . (این نام اختیاری است) پوشه *classic* فایل‌های اسکین را نگهداری می کند . (نام پوشه همان نام تم مورد نظر (classic) محسوب می شود)

یادداشت : تمامی پوشه های موجود در *App_Themes* همنام تم‌های مورد استفاده هستند .

5. سه فایل *skin* به پوشه *classic* اضافه کنید . برای این کار بر روی پوشه *classic* راست کلیک کنید و گزینه های *Add New Item....* سپس *Skin File* را انتخاب کنید . نام فایل‌های جدید را به ترتیب *Button.skin* ، *Label.skin* و *Calendar.skin* قرار دهید . هر فایل *Skin* دارای اطلاعات قالب مورد نظر برای تغییر ظاهر کنترل است .

اکنون Solution Explorer پروژه مشابه شکل ۸-۲ شده است .



شکل ۸-۲- افزودن فایل‌های اسکین به یک تم

6. قطعه کد زیر را به فایل Button.skin اضافه کنید :

```
<asp:Button runat="server" Width="90px" Font-Bold="True" />
```

7. کد زیر را نیز درون فایل Calendar.skin تایپ کنید :

```
<asp:Calendar runat="server" Width="400px"
    Font-Names="Times New Roman" Font-Size="10pt"
    Height="220px" BorderColor="Black" ForeColor="Black"
    BackColor="White" NextPrevFormat="FullMonth"
    TitleFormat="Month" DayNameFormat="FirstLetter">
    <SelectedDayStyle ForeColor="White" BackColor="#CC3333" />
    <SelectorStyle ForeColor="#333333" Font-Names="Verdana"
        Font-Size="8pt" Font-Bold="True"
        Width="1%" BackColor="#CCCCCC" />
    <OtherMonthDayStyle ForeColor="#999999" />
    <TodayDayStyle BackColor="#CCCC99" />
    <DayStyle Width="14%" />
    <NextPrevStyle ForeColor="White" Font-Size="8pt" />
    <DayHeaderStyle ForeColor="#333333" Height="10pt"
        Font-Size="7pt" Font-Bold="True"
        BackColor="#CCCCCC" />
    <TitleStyle ForeColor="White" Height="14pt"
```

```
Font-Size="13pt" Font-Bold="True"
```

```
BackColor="Black" />
```

```
</asp:Calendar>
```

8. نهایتاً کد زیر را به فایل Label.skin اضافه کنید :

```
<asp:Label runat="server" Width="256px"
```

```
Font-Bold="True" Font-Names="Arial Narrow"
```

```
Font-Size="Medium" />
```

یک نکته راجع به ساخت فایل‌های اسکین

چنانچه دقت کنید مشاهده می کنید که فایل‌های skin. چیزی بیش از خواص تعریف شده کنترل‌ها در source view نیستند. (البته به جز خواصی مانند Id) از این رو می توانید ظاهر و خواص کنترل‌ها را در Design view تغییر دهید؛ سپس به source view رفته و کد تولید شده برای کنترل را درون فایل skin. کپی نمایید. نهایتاً خاصیت Id مربوط به کنترل را حذف کنید. به این ترتیب برای ساخت فایل اسکین زمان کمتری صرف می کنید.

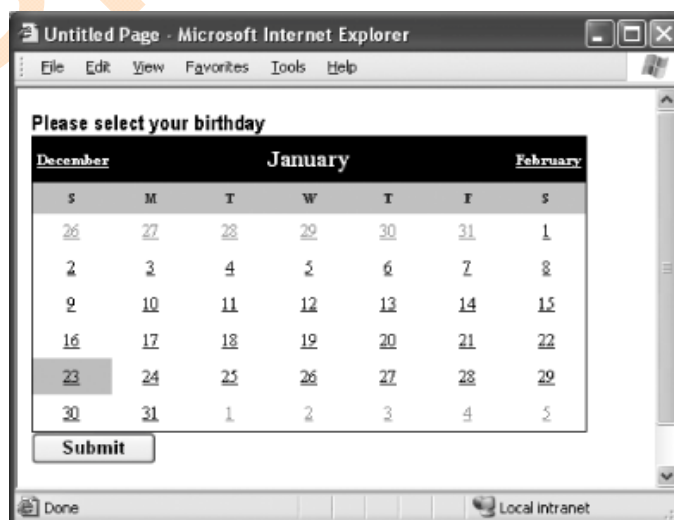
9. به source view صفحه بروید و خاصیت theme را به رهنمود (Directive) آن اضافه کنید :

```
<%@ Page Language="VB" AutoEventWireup="false"
```

```
CodeFile="Default.aspx.vb" Inherits="Default_aspx"
```

```
Theme="Classic" %>
```

10. برای اجرای برنامه کلید F5 را فشار دهید. همانطور که در شکل ۸-۳ نیز مشاهده می کنید؛ ظاهر کنترل‌ها نسبت به حالت اصلی شان تغییر کرده است. در واقع فایل‌های اسکین طراحی شده در پوشه classic بر روی تک تک کنترل‌ها اعمال شده است.



شکل ۸-۳- صفحه Default.aspx پس از اعمال تم

۸-۱-۲- تشریح عملکرد برنامه :

تنها کاری که انجام دادیم ساخت سه فایل اسکین بود که آنها را به عنوان یک تم واحد گروه بندی کردیم . هر فایل اسکین برای ایجاد ظاهر مناسب برای یک کنترل مشخص بکار می رود .

یا دداشت : یک تم شامل یک یا چند فایل اسکین است و هر فایل اسکین ظاهر یک کنترل را مشخص می کند .

بعنوان مثال : فایل Button.skin موجود در تم classic قالب (style) یک کنترل Button را تعریف می کند :

```
<asp:Button runat="server" Width="90px" Font-Bold="True" />
```

یادداشت : به خاطر داشته باشید که یک اسکین نمی تواند شامل خاصیت Id کنترل مربوطه باشد . وجود این خاصیت در فایل اسکین باعث خطا در زمان ترجمه (compile-time error) می شود .

فایل Button.skin مورد اشاره در این بخش شامل یک اسکین تعریف شده برای کنترل Button بود ، شما می توانید در صورت تمایل در یک فایل چندین اسکین برای کنترل Button تعریف نمایید . همانند کد زیر :

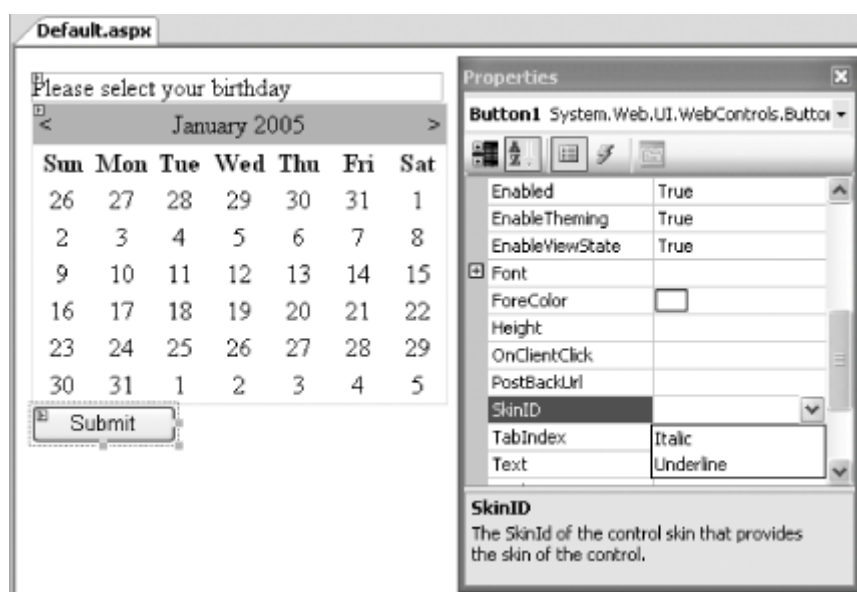
```
<asp:Button runat="server" Width="90px" Font-Bold="True" />
```

```
<asp:Button skinid="Italic" runat="server" Width="90px"
Font-Bold="True" Font-Italic="True" />
```

```
<asp:Button skinid="Underline" runat="server" Width="90px"
Font-Bold="True" Font-Italic="True" Font-Underline="True" />
```

هر اسکین در فایل skin . توسط مقدار موجود در خاصیت skinId شناسایی می شود . اسکین اول در فایل skin . خاصیت skinId ندارد و اسکین پیش فرض کنترل Button محسوب می شود .

حال اگر به پنجره مشخصات (properties window) کنترل Button نگاه کنید ؛ خاصیتی به نام skinId را مشاهده می کنید . بطور پیش فرض این خاصیت مقدار دهی نشده است ، اما چنانچه در فایل skin . مربوط به این کنترل چند اسکین تعریف کرده باشید (همانند آنچه در کد بالا آورده شد) می توانید به تمامی آنها دسترسی داشته باشید و هر کدام را که تمایل دارید انتخاب نمایید . (شکل ۸-۴ را ببینید) چنانچه خاصیت skinId مقدار دهی نشده باشد ، در زمان اجرا اسکین پیش فرض به کنترل اعمال می شود .



شکل ۸-۴- تعیین skinId برای کنترل

هنگامی که یک تم برای صفحه تعیین می کنید (از طریق رهنمود صفحه) اسکین های پیش فرض در فایل های skin. به کنترل های درون صفحه اعمال می شوند. مگر اینکه توسط خاصیت skinId ؛ اسکین خاصی برای کنترل تعریف کرده باشید.

نکته: اگرچه نیازی به ساخت چند فایل اسکین برای پروژه تان نیست، (در واقع اسکین های مربوط به تمامی کنترل ها را می توانید درون یک فایل skin. نیز قرار دهید) اما پیشنهاد می شود برای هر کنترل موجود در صفحه یک فایل skin. جداگانه تعریف نمائید. بهترین حالت این است که مطابق با نام هر کنترل یک فایل skin. طراحی کنید. (همانند آنچه در این بخش دیدید)

جلوگیری از اعمال اسکین به کنترل

هنگامی که یک تم بر روی یک صفحه اعمال می شود، قالب تمامی کنترل ها بر اساس اسکین های تعریف شده در فایل skin. تغییر می کند. برای جلوگیری از اعمال تم بر روی کنترل خاصیت Enable Theming آن کنترل را False کنید. مانند زیر:

```
<asp:Button ID="btnGetTitle"
    Runat="server"
    Text="Get Title"
    EnableTheming="False" />
```

۸-۱-۳- پاورقی بخش ...

... اعمال تم از طریق برنامه نویسی

برای اعمال تم به صفحه در برنامه از خاصیت Theme موجود در کلاس page استفاده کنید. البته باید خاصیت Theme را قبل از بارگذاری صفحه مقدار دهی کنید. (قبل از اجرای رویداد page_Load) در واقع این خاصیت در رویداد PreInit و قبل از لود شدن کنترلها، مقدار دهی می شود:

```
Protected Sub Page_PreInit(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.PreInit
    Page.Theme = "Classic"
End Sub
```

۸-۱-۴- منابع اطلاعاتی:

برای یادگیری سریع تمها و اسکین ها به آدرس زیر مراجعه کنید:

<http://beta.asp.net/quickstart/aspnet/>

۸-۲- اعمال تم ها در زمان اجرا

یادداشت: می توانید به کاربران سایت امکان انتخاب تم دلخواهشان را بدهید.

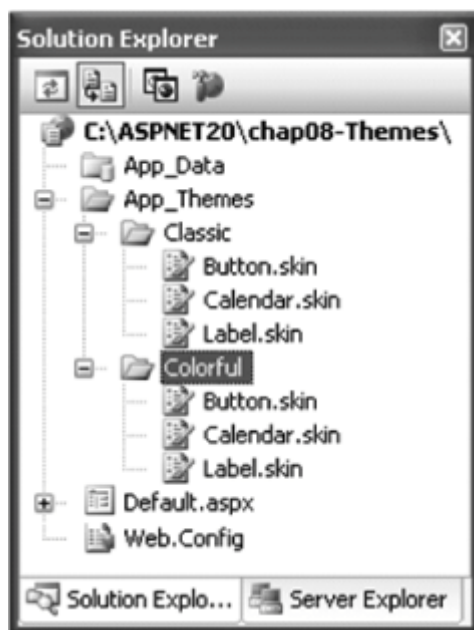
در بخش قبل (بخش ۸-۱) آموختید در زمان طراحی (Design Time) تمها را به وب سایت اعمال کنید. اما بهتر است که به کاربران اجازه دهید تم مورد علاقه شان را در زمان اجرا انتخاب نمایند؛ در این بخش به نحوه انجام این کار می پردازیم.

۸-۲-۱- مراحل انجام کار:

برای انجام آنچه گفته شد؛ به پروژه بخش قبل تم جدیدی به نام colorful اضافه می کنیم و به کاربران اجازه انتخاب تمهای classic و colorful را می دهیم.

1. پروژه بخش قبل را باز کنید (C:\ASPNET20\chap08-Themes)، پوشه جدیدی به App_Themes اضافه کنید و نام آن را colorful قرار دهید.

2. سه فایل اسکین به پوشه colorful اضافه کنید (شکل ۸-۵ را ببینید). همانند تم classic از این اسکین ها برای تغییر ظاهر کنترل های موجود در صفحه استفاده می شود.



شکل ۸-۵- افزودن تم جدید به پروژه

3. کد زیر را به فایل Button.skin اضافه کنید :

```
<asp:Button runat="server" Width="90px" Font-Bold="True"
    BackColor="#C0C0FF" />
```

4. کد فایل Calendar.skin را به صورت زیر تایپ کنید :

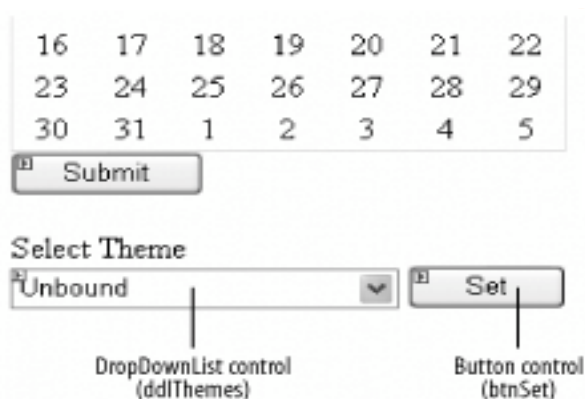
```
<asp:Calendar runat="server" Width="220px"
    Font-Names="Verdana" Font-Size="8pt"
    Height="200px" BorderColor="#FFCC66"
    ForeColor="#663399" BackColor="#FFFFCC"
    DayNameFormat="FirstLetter" BorderWidth="1px"
    ShowGridLines="True">
    <SelectedDayStyle BackColor="#CCCCFF" Font-Bold="True" />
    <SelectorStyle BackColor="#FFCC66" />
    <OtherMonthDayStyle ForeColor="#CC9966" />
    <TodayDayStyle BackColor="#FFCC66" ForeColor="White" />
    <NextPrevStyle ForeColor="#FFFFCC" Font-Size="9pt" />
    <DayHeaderStyle Height="1px" Font-Bold="True"
        BackColor="#FFCC66" />
```

```
<TitleStyle ForeColor="#FFFFCC" Font-Size="9pt"
Font-Bold="True" BackColor="#990000" />
</asp:Calendar>
```

5. نهایتاً کد زیر را نیز به فایل lable.skin اضافه کنید :

```
<asp:Label runat="server" Width="256px"
Font-Bold="True" Font-Names="Arial Narrow"
Font-Size="Medium" BackColor="#FFFFC0" />
```

6. یک کنترل DropDownList و Button همانند شکل ۸-۶ به فرم اضافه کنید .



شکل ۸-۶ افزودن کنترل‌های جدید به صفحه

7. برای اینکه تم‌های موجود در برنامه را از طریق DropDownList در اختیار کاربر قرار دهید ؛ کفایت پوشه‌های موجود در App_Themes را به Dropdowlist اضافه کنیم ؛ برای این منظور کد زیر را به رویداد page_load صفحه اضافه کنید :

```
Protected Sub Page_Load(ByVal sender As Object, _
ByVal e As System.EventArgs) _
Handles Me.Load
If Not IsPostBack Then
'---get all folders under the App_Themes folder
Dim themes As String( ) = _
Directory.GetDirectories( _
Request.PhysicalApplicationPath & "App_Themes")
'---add the themes into the DropDownList
ddlThemes.Items.Clear( )
For Each theme As String In themes
```

```
'---add only the theme name and not full path
ddlThemes.Items.Add( _
    theme.Substring(theme.LastIndexOf("\") + 1))
```

```
Next
```

```
End If
```

```
End Sub
```

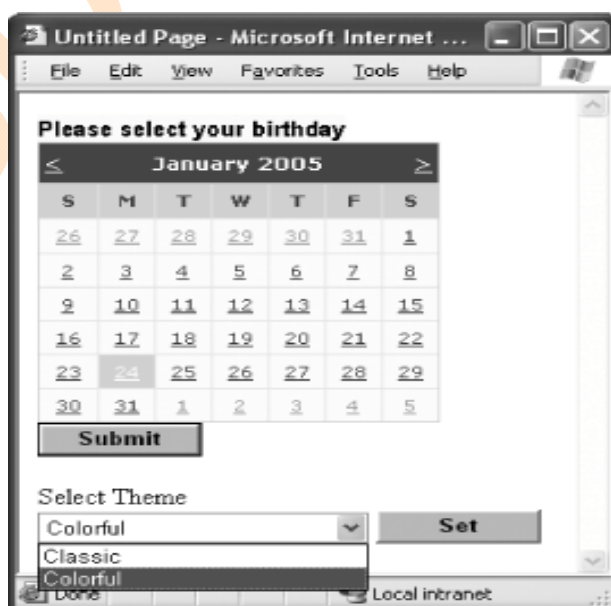
8. در رویداد Page_PreInit؛ با استفاده از شیء Request (object)، تم انتخاب شده در Dropdownlist را به دست آوردید و آن را به صفحه اعمال کنید:

```
Protected Sub Page_PreInit(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.PreInit

    Dim Theme As String = Request("ddlThemes")
    Page.Theme = Theme
End Sub
```

نکته: دقت کنید که نمی توانید به خواص کنترل Dropdownlist بطور مستقیم دسترسی داشته باشید. چرا که رویداد PreInit قبل از postback اجرا می شود از این رو هنگامی که دکمه Set کلیک می شود؛ و یک postback رخ می دهد؛ تم انتخاب شده هنوز در رویداد PreInit در دسترس نیست.

9. کلید F5 را فشار دهید. از طریق Dropdownlist تم دلخواهتان را انتخاب با کلیک بر روی دکمه set آن را به صفحه اعمال کنید. (شکل ۸-۷)



شکل ۸-۷- اعمال تم به صفحه به صورت پویا

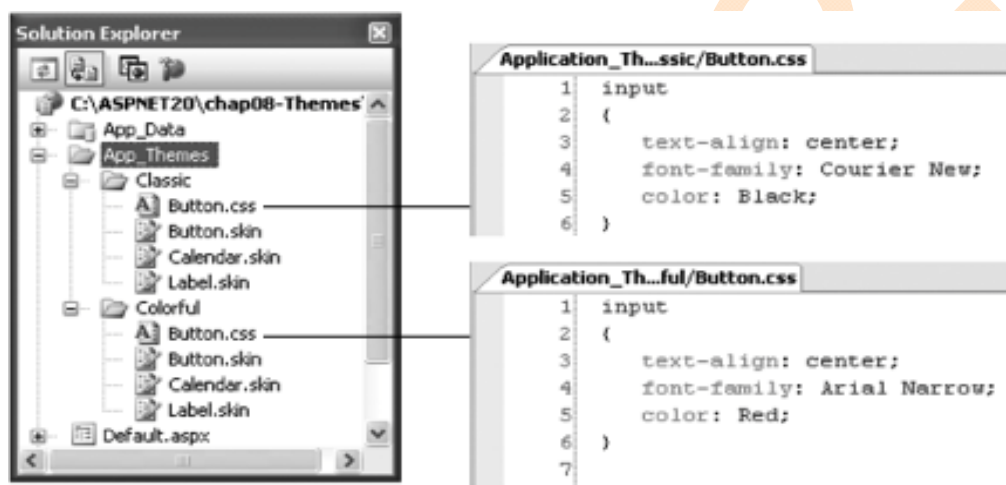
۸-۲-۲- پاورقی بخش ...

... ترکیب تمها با CSS

علاوه اعمال جداگانه تمها به صفحات سایت ؛ می توانید آنها را با CSS ها ترکیب نمائید .

1. بر روی پوشه classic پروژه (همین بخش) راست کلیک کنید و گزینه های ... Add new folder و سپس Style Sheet را انتخاب کنید ، نام فایل را Button.css قرار دهید . به همین شیوه فایل CSS . دیگری نیز به پوشه colorful اضافه کنید .

2. محتویات فایل های اضافه شده را در شکل ۸-۸ می توانید مشاهده کنید .



شکل ۸-۸- محتویات دو فایل Button.css

یادداشت : CSS ها در سمت مشتری (client side) به کنترل های اعمال می شوند . به همین دلیل در اینجا از عنصر `< Input >` در فایل CSS استفاده شده است . این عنصر کد HTML تولید شده توسط کنترل Button است .

3. کلید F5 را فشار دهید . هنگامی که تم classic را انتخاب کنید . فونت (font) کنترل های Button ، به caure " new " تغییر می کند ، و با انتخاب تم colorful فونت کنترل ها "Arial Narrow" می شود، به علاوه اینکه رنگ متن آنها نیز قرمز می شود . (شکل ۸-۹ را ببینید)



شکل ۸-۹- نتیجه اعمال CSS در صفحه

۸-۲-۳- منابع اطلاعاتی :

برای آشنایی بیشتر با CSS ها و کار با آنها به آدرس زیر مراجعه کنید :

<http://www.w3schools.com/css/default.asp>

برای آشنایی بیشتر با کاربرد تمها در برنامه های وب مبحث زیر را مطالعه کنید :

<http://mgemmons.com/blog/archive/0001/01/01/156.aspx>

۸-۳- ذخیره تمها در پروفایلهای کاربران

یادداشت : می توانید به کاربران امکان ذخیره تمهای انتخابیشان را بدهید تا در مراجعات بعدی نیز بتوانند از آنها استفاده کنند.

در بخش قبل (بخش ۸-۲) دیدید که چگونه کاربر می تواند بطور پویا تم مورد نظرش را انتخاب و بر صفحات اعمال کنند. این تکنیک زمانی مناسب است که کاربر تنها یک بار به سایت مراجعه کند ؛ چنانچه کاربر ، عضو دائمی سایت باشد ؛ بایستی از مکانیزمی استفاده کنید تا هر بار که کاربر به سایت مراجعه می کند ؛ مجبور نباشد تم مورد علاقه اش را انتخاب کند. اولین چیزی که به ذهن می رسد استفاده از پروفایل ها است .

۸-۳-۱- مراحل انجام کار :

یادداشت : از پروفایل ها جهت شخصی سازی سایت برای کاربران استفاده می شود . هر شیء پروفایل می تواند اطلاعات و تنظیمات یک کاربر را در خود نگهداری کند .

در این بخش خواهید دید که چگونه تم انتخاب شده کاربر در پروفایل شخصی اش ذخیره می شود و با مراجعه مجدد کاربر به سایت تم مورد نظر از پروفایل کاربر بازیابی و به صفحه اعمال می شود .

1. پروژه بخش قبل (C:\Asp.net20\chap08-theme) را باز کنید و یک فایل web.config به آن اضافه کنید.

2. عنصر < profile > را به فایل web.config اضافه کنید به همراه یک خاصیت به نام theme ، که برای ذخیره تم انتخابی کاربر مورد استفاده قرار خواهند گرفت :

```
<system.web>
  <profile>
    <properties>
      <add name="Theme" type="System.String" />
    </properties>
  </profile>
```

...

3. اگر می خواهید پروفایل کاربران ناشناس را نیز ذخیره کنید ؛ فراموش نکنید که عنصر < anonymousIdentification > و خاصیت allowAnonymous را به هر پروفایل اضافه کنید . (در بخش ۷-۳ به این بحث پرداختیم)

4. برای ذخیره تم در پروفایل ، به code_behind فرم بروید و کد مربوط به رویداد click دکمه set را به صورت زیر وارد کنید . تم انتخاب شده در dropdown list در خاصیت theme پروفایل کاربر ذخیره می شود :

```
Protected Sub btnSet_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles btnSet.Click
    Profile.Theme = ddlThemes.SelectedValue
End Sub
```

5. ابتدا در رویداد Page_PreInit چک می شود که آیتمی از dropdownlist انتخاب شده است یا خیر، اگر انتخابی صورت نگرفته باشد ؛ تم ذخیره شده در پروفایل بازیابی و به صفحه اعمال می شود . در غیر این صورت تم انتخاب شده در dropdownlist به صفحه اعمال می شود :


```
Protected Sub Page_PreInit(ByVal sender As Object, _
    ByVal e As System.EventArgs) _
    Handles Me.PreInit
    Dim Theme As String = Request("ddlThemes")
    '---theme is not set in the DropDownList...
    If Theme = "" Then
        '---get it from profile
        Page.Theme = Profile.Theme
    Else
        '---get it from DropDownList
        Page.Theme = Theme
    End If
End Sub
```

6. کلید F5 را فشار دهید . شمای colorful را انتخاب و اجرای برنامه را متوقف کنید . مجدداً برنامه را اجرا (Run) کنید . اگر دقت کنید مشاهده می کنید که تم colorful همچنان بر صفحات اعمال شده است . علت هم این است که در اجرای اول برنامه این تم در پروفایل کاربر ذخیره شد و دفعات بعد از پروفایل بازیابی و بر صفحات اعمال می شود .

۸-۳-۲- پاورقی بخش ...

... اعمال تم به کل برنامه

می توانید یک تم را به تمام (صفحات) سایت اعمال کنید ؛ برای این کار کافست عنصر < pages > را به فایل web.config اضافه کنید :

```
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
    <system.web>
        <pages theme="Colorful" />
    </system.web>
</configuration>
```

در مثال قبل صفحاتی که در رهنمود (Directive) آنها از خاصیت theme استفاده نشده است از تم colorful استفاده می کند . اما چنانچه خاصیت theme در آنها مقداردهی شده باشد ، مقدار تعریف شده در web.config بر آنها اثر نمی گذارد .

۸-۳-۳- منابع اطلاعاتی :

برای کسب اطلاعات بیشتر راجع به سرویس پروفایل به فصل هفتم مراجعه کنید . در این بخش جزئیات بیشتری از این سرویس مورد بررسی قرار گرفت .

۸-۴- بومی سازی (Localization)

یادداشت : Asp.net 2.0 با قابلیت جدید Auto-culture ساخت وب سایتهای عمومی و بومی را آسان نموده است .

امروزه اکثر شرکتها و کمپانیها در حال توسعه وب سایتهای خود برای کسب جایگاه بهتر در دنیای وب و اینترنت هستند ، یکی از جنبه های این رقابت ساخت وب سایتهای است که سراسری و فراگیر باشند ؛ بطوریکه کاربر در هر نقطه جهان قرار دارد بتواند از سایت استفاده کند و نکته غیر قابل فهم در سایت وجود نداشته باشد . به عنوان مثال برنامه طراحی شده برای یک سوپر مارکت آمریکایی در سوپرمارکت آسیایی قابل استفاده نیست . برای این که یک برنامه مشتریان زیادی را جذب کند ، بهتر است صفحات مخصوص هر کاربر را برایش بومی سازی کنید .

Auto-culture در Asp.net 2.0

Asp.net 2.0 با قابلیت Auto-culture امکان بومی سازی صفحات وب را فراهم آورده است .

قابلیت Auto-culture را از طریق خواص " auto " culture=" " و " Auto " UIculture در رهنمود (Directive) صفحه می توانید فعال کنید .

پس از فعال سازی ؛ Asp.net بطور خود کار زبان مربوطه را به اشیاء cultureInfo نگاشت (map) می کند . به عنوان برنامه نویس وب تنها کفایت منابع (resources) مربوط به زبانهای مختلف را فراهم کنید . سپس Asp.net منبع مربوط به هر زبان در موقع نیاز بارگذاری می کند .

بعنوان طراح وب بایستی با مفاهیم زیر آشنا باشید :

- عمومی سازی (Globalization)

هنگام طراحی برنامه ، منابع مورد نیاز برنامه را به گونه ای طراحی می کنید تا به آسانی هنگام درخواست ، زبانهای مختلف اصلاح شوند .

- بومی سازی (Localization)

اعمال تغییرات به برنامه بطوریکه کاربر برنامه را به زبانی که انتخاب کرده است مشاهده کند .

بومی سازی برنامه اعمال دیگری نیز به همراه دارد :

- قالب تاریخ مناسب

به عنوان مثال مردم در ایالات متحده از قالب تاریخ متفاوتی نسبت به مردم انگلستان استفاده می کنند . نمایش قالب مناسب تاریخ به محل زندگی کاربر بستگی دارد ، مثلاً 2/7/2007 یا 2 July 2007 و ...

- ترجمه محتوای صفحات از زبانی به زبان دیگر

به عنوان مثال اگر قرار است برنامه در سوپر مارکتی در چین اجرا شود باید متون برنامه به زبان چینی تبدیل شوند .

- تنظیم جهت متن .

نمایش متون از چپ به راست یا از راست به چپ نیز به زبان مورد نیاز کاربر وابسته است .

در این بخش ؛ کار با قابلیت بومی سازی در Asp.net 2.0 را یاد می گیرید.

مفاهیم پایه ای بومی سازی

چنانچه با مفهوم بومی سازی غریبه اید ؛ اجازه دهید ابتداء به توضیح برخی مفاهیم بپردازیم :

Culture (فرهنگ) : روشی برای تعیین تنظیمات مربوط به یک مکان یا کشور خاص می باشد . از Culture Code برای نمایش Culture استفاده می شود .

natural Culture (بی طرف) : یک مدل از Culture که با زبان خاصی پیاده سازی شده اما برای مکان مشخصی طراحی نشده ؛ مانند " en " که زبان انگلیسی است اما نمونه مشخصی را برای استفاده تعیین نمی کند .

Specific Culture (مشخص) : یک Culture مشخص شده برای منطقه و یا کشور خاص . مانند " en-us " که معرف زبان انگلیسی مخصوص ایالات متحده است .

invariant Culture (ثابت): نه از نوع مشخص است و نه از نوع بی طرف. به زبان انگلیسی است اما به مکان خاصی اختصاص ندارد. از این نوع Culture برای نمایش داده هایی که به کاربران داده نمی شوند استفاده می شود، برای مثال از *invariant culture* برای ذخیره اطلاعات در یک فایل استفاده می شود.

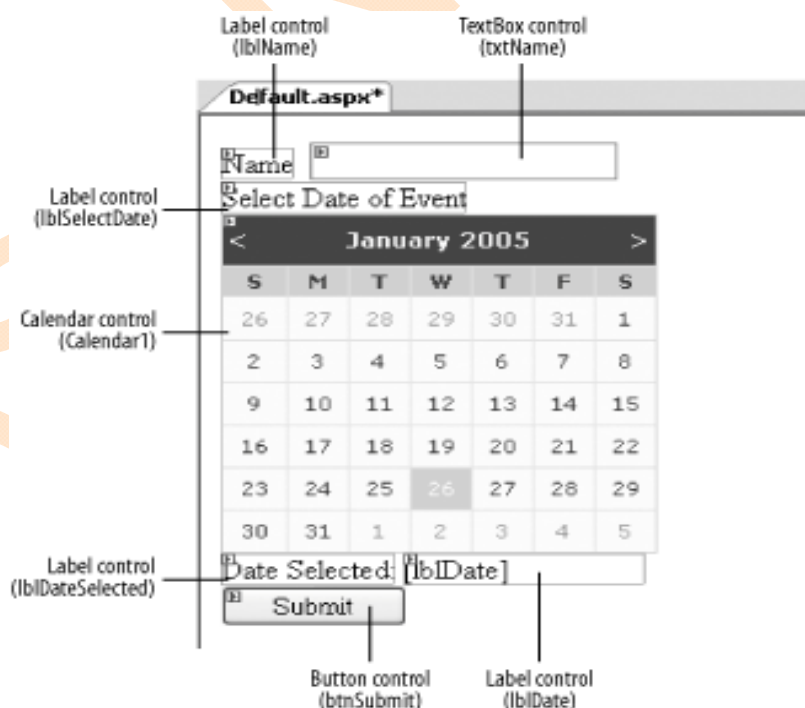
۸-۴-۱- مراحل انجام کار:

در این بخش برنامه ای را طراحی می کنیم که از دو زبان پشتیبانی می کند، انگلیسی و چینی. در این بخش یاد می گیرید چگونه منابع مورد نیاز زبان های مربوطه را فراهم کنید، بطوریکه با انتخاب زبان توسط کاربر برنامه به همان زبان در IE به نمایش درآید.

ASP.net 2.0 با استفاده از بومی سازی ضمنی (Implicit localization) مقادیر موجود در کنترلها را بر اساس فایل منبع مربوط به زبان خاص را تنظیم می کند.

1. برای اینکه نحوه کار Implicit localization را ببینید؛ VS 2005 را باز کنید و پروژه وب جدیدی ایجاد کنید. نام پروژه را *C:\ASPNET20\chap08-Localization* قرار دهید.

2. کنترلهای موجود در شکل ۸-۱۰ را به فرم اضافه کنید. تم 1 colorful را به کنترل calendar اعمال کنید. (از طریق لینک... Auto Format در منوی Calendar Tasks)

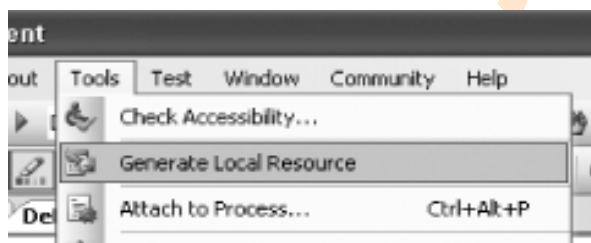


شکل ۸-۱۰- طراحی صفحه Default.aspx

3. بر روی کنترل calendar کلیک مضاعف کنید و کد زیر را به آن اضافه کنید . با این کد تاریخ انتخاب شده در calendar در کنترل label (lblDate) نمایش داده می شود :

```
Protected Sub Calendar1_SelectionChanged(ByVal sender As Object, _  
    ByVal e As System.EventArgs)  
    lblDate.Text = Calendar1.SelectedDate.ToShortDateString  
End Sub
```

4. پس از طراحی فرم به منوی Tools بروید و گزینه Generate Local Resource را انتخاب کنید . VS 2005 بطور خودکار فایل منبعی شامل تمام متون مورد استفاده در کنترلها را تولید می کند . (شکل ۸-۱۱ را ببینید)



شکل ۸-۱۱- تولید منبع محلی (local resource) در VS 2005

5. مرحله ۴ باعث ایجاد شاخه جدیدی با نام *App_LocalResources* می شود که درون آن فایل منبع تولید شده توسط VS 2005 نگهداری می شود . نام فایل منبع از نام فرم (Default.aspx) پیروی می کند و به پسوند *.resx* ختم می شود . (شکل ۸-۱۲)



شکل ۸-۱۲- شاخه App_LocalResources در Solution Explorer

6. برای مشاهده فایل منبع بر روی شاخه *App_LocalResources* کلیک مضاعف کنید تا ویرایشگر مربوط به منبع را مشاهده کنید . (شکل ۸-۱۳)

App_Localized.aspx.resx

Categories: Add Remove

Name	Value	Comment
ButtonResource1.AccessKey		
ButtonResource1.Text	Submit	
ButtonResource1.ToolTip		
CalendarResource1.AccessKey		
CalendarResource1.Caption		
CalendarResource1.NextMonthText	>	
CalendarResource1.PrevMonthText	<	
CalendarResource1.SelectMonthText	>>	
CalendarResource1.SelectWeekText	>	
CalendarResource1.ToolTip		
LabelResource1.AccessKey		
LabelResource1.Text	Name	
LabelResource1.ToolTip		
LabelResource2.AccessKey		
LabelResource2.Text	Select Date of Event	
LabelResource2.ToolTip		
LabelResource3.AccessKey		
LabelResource3.Text	Date Selected:	
LabelResource3.ToolTip		
LabelResource4.AccessKey		
LabelResource4.Text		
LabelResource4.ToolTip		
PageResource1.Title	Untitled Page	
TextBoxResource1.AccessKey		
TextBoxResource1.Text		
TextBoxResource1.ToolTip		

شکل ۸-۱۳- ویرایشگر فایل منبع

7. اگر به Source View صفحه بروید؛ مشاهده می کنید که خاصیت جدیدی به نام `meta:resourcekey` به تمامی کنترلها اضافه شده است. مقدار این خاصیت با هر فیلد در فایل منبع برابر است:

...

```
<asp:Label ID="lblName" runat="server"
    Text="Name"
    meta:resourcekey="LabelResource1">
</asp:Label>
```

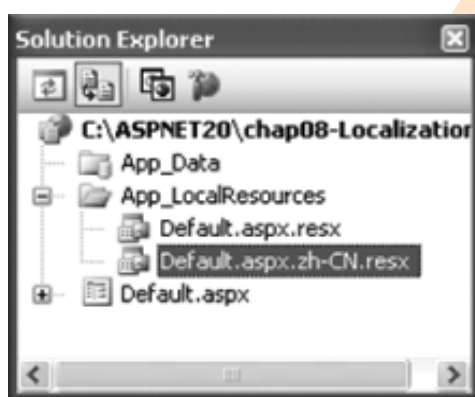
...

```
<asp:TextBox ID="txtName" runat="server"
    meta:resourcekey="TextBoxResource1">
</asp:TextBox>
```

...

8. از آنجا که می خواهیم برنامه را به دو زبان مختلف نمایش دهیم ، به یک فایل منبع دیگر نیز نیاز داریم .به همین دلیل یک کپی از فایل *Default.aspx.resx* ایجاد کنید . (بر روی *Default.aspx.resx* راست کلیک کنید ، گزینه کپی را انتخاب و سپس بر روی *App_Resources* راست کلیک و *past* را انتخاب کنید) فایل جدید را به *Default.aspx.zh-CN.resx* تغییر نام دهید . (شکل ۸-۱۴) Culture code بایستی بعنوان بخشی از نام فایل منبع بکار رود . بطور کلی فایل های *.resx* . با قالب زیر نامگذاری می شوند :

< file name > .aspx . < Culture code > .resx



شکل ۸-۱۴- ایجاد نسخه دوم از فایل منبع

9. فایل منبع جدید را باز کنید و مطابق شکل ۸-۱۵ مقادیر جدیدی به آن اضافه کنید .

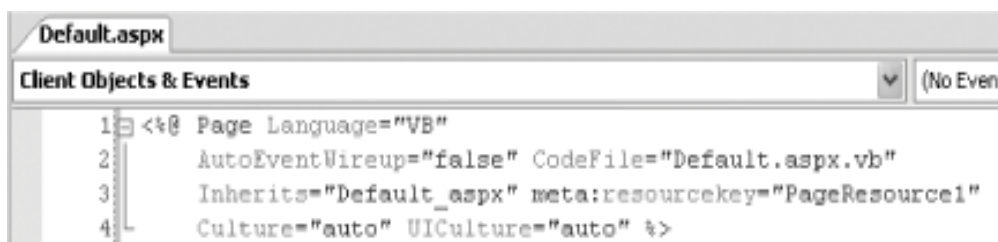
ButtonResource1.Text	提交表格
LabelResource1.Text	姓名
LabelResource2.Text	请选择日期
LabelResource3.Text	日期:

شکل ۸-۱۵- محتویات فایل منبع برای culture چینی

نکته : کاراکترهای چینی را از طریق ویندوز وارد کنید . اما ابتدا بایستی از طریق CD ویندوز زبان چینی را بر روی سیستم نصب کنید . برای پیکربندی کامپیوتر برای این منظور به بخش " منابع اطلاعاتی " همین بخش مراجعه کنید . بعلاوه می توانید از نرم افزار سومی برای ورود کاراکترهای چینی استفاده کنید .

10. برای استفاده از قابلیت Auto-culture در Asp.net 2.0 (همان طور که در ابتدای بخش گفته شد) خواص زیر را به رهنمود (Directive) صفحه اضافه کنید . (شکل ۸-۱۶)

- Culture="auto"
- UICulture="auto"



شکل ۸-۱۶- خواص Culture و UICulture

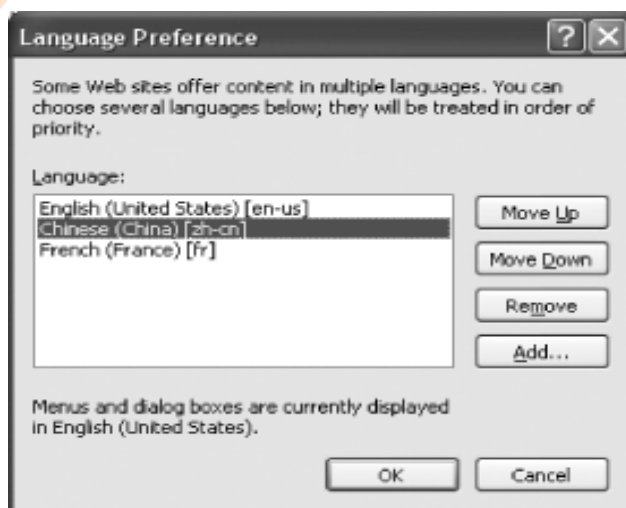
نکته: برای اعمال Auto culture به کل برنامه ؛ عنصر <globalization> را به web.config اضافه کنید.

<system.web>

<globalization culture="auto" uiCulture="auto" />

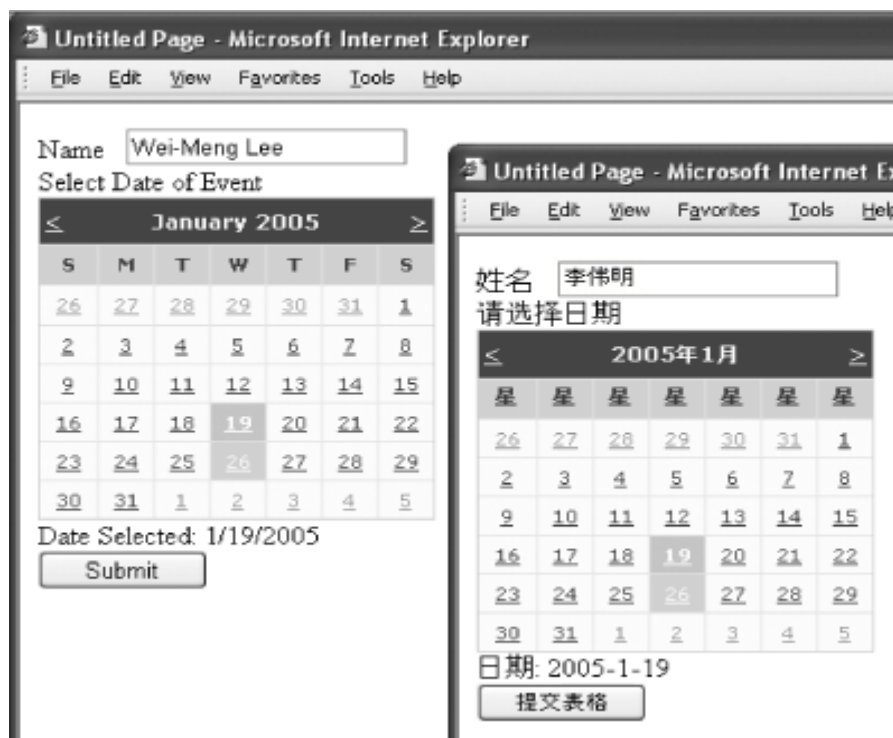
11. خواص Culture و UICulture اطلاعات مربوط به زبان انتخابی را به طور خود کار به خواص CurrentCulture و CurrentUICulture انتقال می دهند و بدین ترتیب کنترل‌های موجود در صفحه را برای بومی سازی آگاه می کنند .

12. برای اجرای برنامه ، کلید F5 را فشار دهید . صفحه Default.aspx به زبان انگلیسی نمایش داده می شود . برای نمایش برنامه به زبان چینی از منوی Tools گزینه Internet options را انتخاب کنید . (در Internet Explorer) سپس دکمه Languages را کلیک کنید . در پنجره language preference (شکل ۸-۱۷) دکمه add را کلیک و زبان جدید را اضافه کنید . در اینجا گزینه "Chinese (China) [zh-cn]" را انتخاب کنید . زبان اضافه شده را به بالای لیست انتقال دهید و نهایتاً ok کنید .



شکل ۸-۱۷- تغییر زبان Internet Explorer

13. برای دیدن صفحه به زبان چینی، IE را Refresh کنید. شکل ۸-۱۸ صفحه Default.aspx را به دو زبان انگلیسی و چینی نمایش می دهد.



شکل ۸-۱۸. نمایش صفحه Default.aspx به زبانهای انگلیسی و چینی

۸-۴-۲- تشریح عملکرد برنامه :

در زمان اجرا، Asp.net بطور خودکار تنظیمات Culture درخواست شده را شناخته و بر اساس فایل منبع مربوطه تنظیمات تعریف شده را اعمال می کند. فایل Default.aspx.resx بعنوان منبع پیش فرض شناخته می شود. بعلاوه چنانچه فایل منبع Culture درخواست شده در پوشه App_LocalResources موجود نباشد؛ از فایل Default.aspx.resx استفاده می شود.

دقت داشته باشید که برخی کنترلها مانند Calendar، خودشان از بومی سازی پشتیبانی می کنند. در این حالت نیاز به انجام عمل خاصی بر روی کنترل نیست چرا که خودش تاریخ را به زبان چینی نمایش می دهد. بعلاوه توجه داشته باشید که قالب نمایش تاریخ در انگلیسی و چینی متفاوت است.

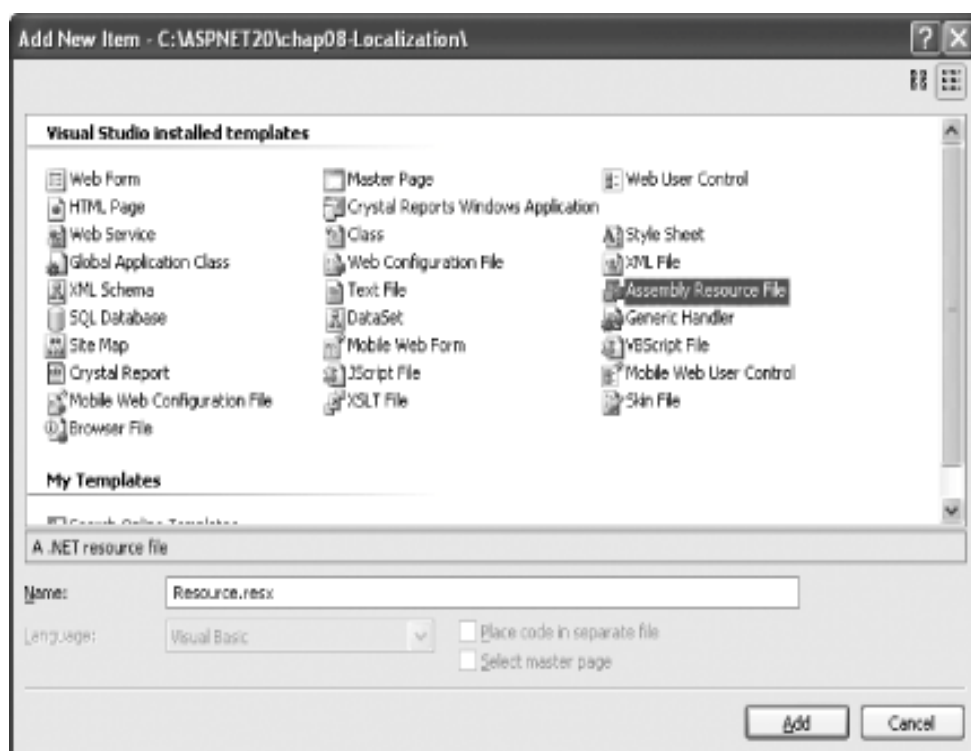
بومی سازی صریح (Explicit localization)

در بومی سازی ضمنی هر فرم وب مجموعه ای از فایل های منبع جداگانه برای خود دارد. این متد زمانی مفید است که بخواهید خروجی کنترلها را بومی سازی کنید و در حالیکه متون زیادی دارید و مرتب از آنها استفاده می کنید ؛ (مثل پیغام خوش آمد گویی یا پیغامهای خطا) استفاده از این نوع بومی سازی منطقی نیست . در این حالت بهتر است از بومی سازی صریح (Explicit) استفاده نمائید . بومی سازی صریح امکان تعریف مجموعه ای از منابع برای استفاده تمامی صفحات پروژه را فراهم می آورد. (یک مجموعه برای تمامی صفحات بجای یک مجموعه برای یک صفحه)

از پروژه ساخته شده در همین بخش (*C:\ASPNET20\chap08-Localization*) برای نمایش بومی سازی صریح استفاده می کنیم . در این قسمت با استفاده از بومی سازی صریح یک پیغام نوشته شده به زبانهای چینی و انگلیسی را ذخیره می کنیم . با استفاده از کلاس Resources برنامه پیغام مورد نظر را به زبان انتخاب شده نمایش می دهد . (بر مبنای تنظیمات اعمال شده در IE)

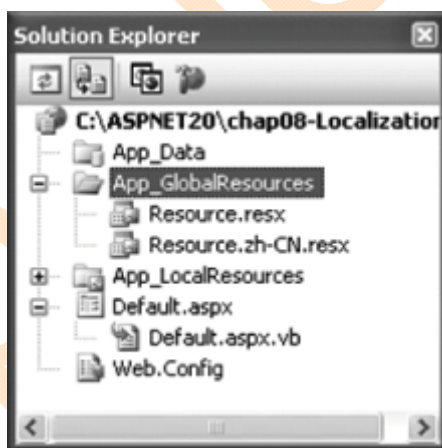
یادداشت : بومی سازی صریح به شما امکان تعریف یک مجموعه از منابع برای استفاده تمامی صفحات را می دهد .

1. پروژه ساخته شده در همین بخش (*C:\ASPNET20\chap08-Localization*) را باز کنید . و پوشه جدیدی به نام *App_GlobalResources* به Solution Explorer پروژه اضافه کنید . (بر روی نام پروژه در Solution Explorer راست کلیک کنید و گزینه های *App_GlobalResources* → *Add new folder* را انتخاب کنید)
2. فایل منبع جدیدی به پروژه اضافه کنید . برای این کار بر روی پوشه *App_GlobalResources* راست کلیک و گزینه های *Add new Item...* سپس *Assembly Resource File* را انتخاب کنید . (شکل ۸-۱۹) نام فایل را *Resource.resx* قرار دهید . یک کپی از آن نیز به نام *Resource.zh-CN.resx* به پوشه *App_GlobalResources* اضافه کنید .



شکل ۸-۱۹- افزودن فایل Resource به پروژه

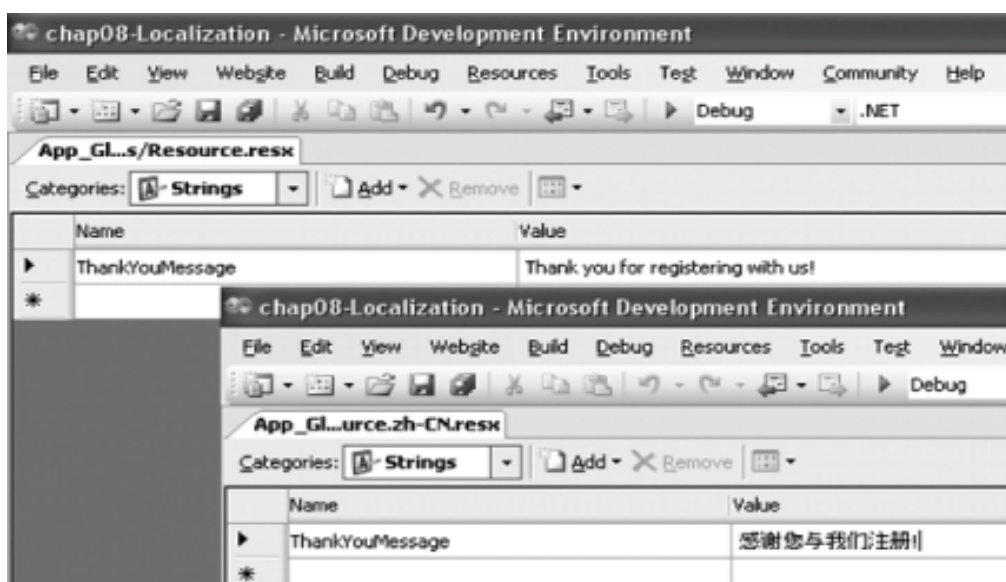
3. Solution Explorer پروژه هم اکنون همانند شکل ۸-۲۰ شده است .



شکل ۸-۲۰- Solution Explorer پروژه

4. به فایل *Resource.resx* رشته با مقادیر موجود در شکل ۸-۲۱ اضافه کنید .

همین کار را در فایل *Resource.zh-CN.resx* انجام دهید با این تفاوت که این بار مقدار (value) را به زبان چینی وارد کنید .



عکس ۸-۲۱- ساخت رشته در فایل های منبع

5. کد زیر را به رویداد click دکمه submit در صفحه Default.aspx.vb اضافه کنید :

```
Protected Sub btnSubmit_Click(ByVal sender As Object, _  
    ByVal e As System.EventArgs) _  
    Handles btnSubmit.Click  
  
    Dim message As String = _  
        Resources.Resource.ThankYouMessage.ToString( )  
  
    Dim script As String = "alert('" & _  
        message & "');"   
  
    Page.ClientScript.RegisterClientScriptBlock(Me.GetType, _  
        "MyKey", script, True)  
  
End Sub
```

6. کلاس Resources دسترسی پویا به مقادیر موجود در فایل های منبع را فراهم می کند.

7. کلید F5 را فشار دهید . هنگامی که در IE زبان چینی را انتخاب کنید . (شکل ۸-۱۷) و دکمه submit انتهای فرم را کلیک کنید . پیغام خوش آمد گویی از فایل منبع بازایی شده و با استفاده از متد RegisterClientScriptBlock() توسط پنجره ای در سمت مشتری نمایش داده می شود . (شکل ۸-۲۲)

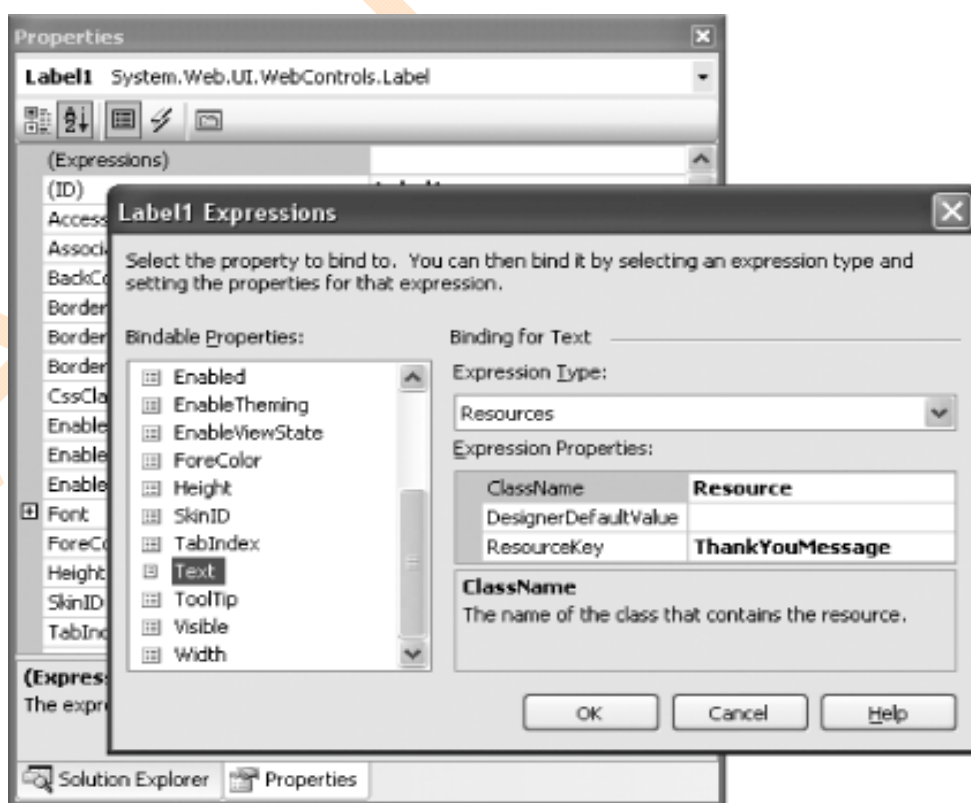


شکل ۸-۲۲- نمایش رشته بومی سازی شده

بازیابی مقادیر از فایل‌های منبع

بجای بازیابی مقادیر از طریق برنامه نویسی می توانید از طریق تعریف نیز آنها را بازیابی کنید . بعنوان مثال می توانید خاصیت text یک label را به فایل منبع مقید سازید .

فیلد Expressions در پنجره مشخصات (properties window) این امکان را به شما می دهد .



همچنین می توانید در source view این کار را انجام دهید :

```
<asp:Label ID="Label1"
  Runat="server"
  Text="<%%$ Resources:Resource,
    ThankYouMessage %>">
</asp:Label>
```

۸-۴-۴- منابع اطلاعاتی :

برای آشنایی بیشتر با پیکر بندی ویندوز XP جهت دریافت کارکترهای یک زبان خارجی این مقاله را مطالعه کنید :

http://www.ondotnet.com/pub/a/dotnet/2003/09/29/globalization_pt2.html?page=3

من شکست نخورده ام ؛

من کاشف ۹۹۹ راه هستم

که هیچکدام به هدف نرسیده اند !

توماس ادیسون

ASP.NET 2.0

