

معرفی

Visual Basic 2005

برای برنامه نویسان

مترجم: میر محمد جواد معتمد هاشمی

انتشارات زانیس

با همکاری

انتشارات ناقوس

معرفی Visual Basic 2005 [ویژوال بیسیک ۲۰۰۵] برای برنامه‌نویسان / [شون کمبل .
و دیگران]؛ مترجم: میرمحمد جواد معتمد هاشمی. -- تهران: زانئیس، ناقدوس، ۱۳۸۵.
۳۳۶ ص. : مصور- یک لوح فشرده.

ISBN : 964-96899-5-8 : ۳۵۰۰۰ ریال

فهرست‌نویسی بر اساس اطلاعات فیپا.

۱. ویژوال بیسیک مایکروسافت. ۲. بیسیک (زبان برنامه‌نویسی کامپیوتر). الف. کمبل.
شون، ۱۹۷۰-م. Campbell, Sean. ب. معتمد هاشمی، میرمحمدجواد، ۱۳۵۳-ج. عنوان.

۰۰۵/۲۷۶۸

QAVP/۷۳/۱۹۱۵

۱۳۸۵

۸۵-۱۱۰۶۲

کتابخانه ملی ایران



www.naghoospress.ir



چاپ اول

معرفی Visual Basic 2005 برای برنامه‌نویسان

انتشارات زانئیس

انتشارات ناقدوس

میرمحمد جواد معتمد هاشمی

۱۳۸۵

جلد ۲۰۰۰

شاد

فرنو

فرنو

۳۵۰۰۰ ریال

۹۶۲-۹۶۸۹۹-۵-۸

964-96899-5-8

نام کتاب

ناشر

ناشر همکار

مترجم

چاپ اول

تیراژ

لیتوگرافی

چاپ

صحافی

قیمت به همراه CD

شابک

ISBN

کتابه حقوق برای ناشر محفوظ
است. تکثیر تمامی یا قسمتی از
این اثر به صورت حرفه‌ای یا
چاپ مجدد، چاپ انست، پلی‌کپی،
فتوکپی و انواع دیگر چاپ ممنوع
است و پیگرد قانونی دارد.

مرکز پخش: انتشارات ناقدوس

تهران: خیابان انقلاب- خیابان ۱۲ فروردین- کوچه نوروز- پلاک ۲۷

تلفن: ۶۶۹۶۶۷۲۹-۶۶۹۶۶۷۵۰-۶۶۲۱۱۷۱۵

فهرست مطالب

مقدمه	۱۹
چه کسی باید این کتاب را بخواند	۱۹
در این کتاب چه می‌توان یافت	۲۰
فصل اول / مقدمه‌ای بر چارچوب NET برای برنامه‌نویس Visual Basic	۳۳
برنامه‌های کاربردی که شما می‌توانید ایجاد کنید	۳۴
برنامه‌های کاربردی هوشمند Windows	۳۵
برنامه‌های کاربردی وب	۳۶
سرویس‌دهنده‌های وب	۳۶
سرویس‌های Windows	۳۶
برنامه‌های Console	۳۷
کتابخانه‌های کلاس	۳۷
برنامه‌های کاربردی برای ابزارهای هوشمند	۳۸
پروژه‌های نصب و ارائه	۳۸
یک محیط مجتمع برنامه‌سازی واقعی	۳۸
NET چیست؟	۴۰
Common Language Runtime و زبان میانی	۴۰
سیستم مشترک نوع داده (Common Type System)	۴۵
شیء‌ها و مقدارها	۴۶
کلاس‌ها، رویه‌ها، ویژگی‌ها، فیلدها، رویدادها	۴۸
کتابخانه کلاس چارچوب NET	۵۲
فضاهای نام	۵۳
گردایه‌ها و شبه داده‌ها	۵۷
دسترسی به داده‌ها : ADO.NET	۵۸

۶۱.....	برنامه‌های کاربردی Windows
۶۱.....	تفاوت‌ها و شباهت‌ها بین Visual Basic 6 و Visual Basic 2005
۶۴.....	جنبه‌های مهم جدید در Visual Basic 2005
۶۵.....	برنامه‌های کاربردی وب
۶۵.....	شباهت‌ها و تفاوت‌ها بین ASP, ASP.NET 1.0, و ASP.NET 2.0
۶۸.....	جنبه‌های مهم و جدید ASP.NET
۶۹.....	پیشرفت‌های Visual Studio 2005 و چارچوب .NET 2.0
۶۹.....	پیشرفت‌های Visual Basic
۷۰.....	پیشرفت‌های IDE
۷۰.....	Visual Web Developer جدید
۷۱.....	ارائه Click Once
۷۲.....	نتیجه‌گیری
۷۳.....	فصل دوم/ پیشرفت‌های زبان در Visual Basic 2005
۷۴.....	کاربرد : Generic ها
۷۴.....	مفاهیم جدید
۷۵.....	Collection های Visual Basic 6.0
۷۶.....	مجموعه‌های چارچوب .NET
۷۷.....	بررسی عمیق
۷۷.....	استفاده از نوع‌های Generic
۸۰.....	ایجاد نوع‌ها و رویه‌های Generic
۸۰.....	رویه‌های Generic
۸۲.....	نوع‌های Generic
۸۲.....	نتیجه‌گیری
۸۳.....	کاربرد: شیء MY

۸۲.....	مفاهیم جدید
۸۵.....	بررسی عمیق
۸۵.....	My.Application
۸۸.....	سناریوهایی برای استفاده از My.Application هنگام ایجاد برنامه‌ها
۸۹.....	My.Computer
۹۱.....	سناریوهای استفاده از My.Computer هنگام ساختن برنامه‌ها
۹۳.....	My.User
۹۴.....	سناریوهایی برای استفاده از My.User هنگام ساختن برنامه‌ها
۹۴.....	My.Forms و My.WebServices
۹۵.....	نتیجه‌گیری
۹۶.....	کاربرد: عملگرها
۹۶.....	مفاهیم مهم
۹۷.....	بررسی عمیق
۹۷.....	عملگرهای تعریف شده
۹۸.....	عملگر Is Not
۹۹.....	عملگرهای Is True ، Is False و Not
۱۰۰.....	عملگر CType
۱۰۱.....	نتیجه‌گیری
۱۰۱.....	کاربرد: کلمات کلیدی
۱۰۲.....	مفاهیم جدید
۱۰۲.....	کلمه کلیدی Global
۱۰۲.....	کلمه کلیدی Using
۱۰۳.....	کلمه کلیدی Continue
۱۰۴.....	بررسی عمیق

۱۰۶.....	استفاده از کلمه کلیدی Using
۱۰۷.....	استفاده از کلمه کلیدی Continue
۱۰۸.....	نتیجه‌گیری
۱۰۹.....	فصل سوم/ محیط مجتمع برنامه سازی در Visual Studio 2005
۱۱۰.....	کاربرد: پیشرفت‌های IDE
۱۱۰.....	تنظیمات مربوط به بیکربندی
۱۱۱.....	Options Dialog
۱۱۲.....	کادر محاوره‌ای Import/Export Settings
۱۱۶.....	سیستم مجتمع Help و Community
۱۱۶.....	آنچه بود و آنچه حالا هست
۱۱۸.....	استفاده از محتویات آنلاین Help
۱۱۹.....	پیشرفت‌های پروژه‌ها
۱۲۱.....	ProjectDesigner
۱۲۲.....	بخش Application
۱۲۴.....	مدیریت اطلاعات مربوط به گردایه
۱۲۶.....	بخش Signing
۱۲۷.....	بخش References
۱۲۷.....	بخش Compile
۱۲۹.....	بخش Resources
۱۳۰.....	مجتمع کردن منابع با چارچوب .NET
۱۳۱.....	پنجره Settings
۱۳۱.....	سطوح کارایی
۱۳۲.....	ترکیب تنظیمات با چارچوب .NET
۱۳۳.....	پنجره‌های ابزار جدید و ارتقاء یافته

۱۳۳.....	Code Editor (ویراستار کد).....
۱۳۴.....	Data Preview Dialog.....
۱۳۵.....	Document OutLine.....
۱۳۶.....	پیشرفت‌های مربوط به Docking پنجره‌ها.....
۱۳۷.....	Class Designer.....
۱۳۸.....	نوشتن کد صحیح.....
۱۳۸.....	پیشرفت‌های اعمال شده در IntelliSense.....
۱۳۹.....	Syntax Error Assistance.....
۱۴۰.....	تعیین مقدار عبارت در زمان طراحی.....
۱۴۰.....	Task List و Error List.....
۱۴۰.....	Code Snippets.....
۱۴۱.....	نامگذاری مجدد نمادین.....
۱۴۲.....	پیشرفت‌های اعمال شده در خطایاب (Debugger).....
۱۴۳.....	Data Tips.....
۱۴۵.....	Visualizerها.....
۱۴۶.....	Exception Assistant.....
۱۴۶.....	سفارشی کردن نمایش داده‌ها.....
۱۴۶.....	Override کردن رویه ToString متعلق به نوع‌های ایجاد شده به وسیله کاربر.....
۱۴۶.....	استفاده از شاخص‌های نمایش خطایاب.....
۱۴۸.....	بررسی عمیق.....
۱۴۸.....	نتیجه‌گیری.....
۱۳۹.....	فصل چهارم / ساختن برنامه‌های داده محور.....
۱۵۱.....	کاربرد : Table Adapter ها و کلاس DataConnector.....
۱۵۱.....	مفاهیم جدید.....

۱۵۲.....	System.Data.DataSet
۱۵۴.....	بازآوری داده‌ها از یک پایگاه داده
۱۵۵.....	بررسی عمیق
۱۶۰.....	DataConnector
۱۶۱.....	Table Adapter
۱۶۲.....	DataNavigator
۱۶۴.....	DataSet Designer (طراح DataSet)
۱۶۸.....	نتیجه‌گیری
۱۶۹.....	کاربرد: تگ‌های هوشمند برای داده‌ها
۱۶۹.....	مفاهیم جدید
۱۶۹.....	بررسی عمیق
۱۶۹.....	فرم‌های Master-Details
۱۷۴.....	نتیجه‌گیری
۱۷۵.....	کاربرد: فرم‌های Master-Details و درخواست‌های پارامتردار
۱۷۵.....	مفاهیم جدید
۱۷۵.....	بررسی عمیق
۱۷۶.....	رابط کاربر برنامه
۱۸۲.....	Data Navigator
۱۸۴.....	نتیجه‌گیری
۱۸۴.....	کاربرد: Object Data Source
۱۸۴.....	مفاهیم جدید
۱۸۵.....	بررسی عمیق
۱۸۸.....	پیوند شیء‌ها - روش مشکل
۱۹۰.....	یک راه آسان‌تر برای پیوند شیء‌ها

۱۹۳.....	نتیجه‌گیری
۱۹۵.....	فصل پنجم / ایجاد رابطه‌ای کاربر
۱۹۶.....	کاربرد: کنترل‌های جدید فرم‌های Windows
۱۹۶.....	مفاهیم جدید
۱۹۶.....	کنترل Tool Strip
۱۹۸.....	کنترل‌های FlowLayoutPanel و TableLayoutPanel
۱۹۸.....	کنترل Web Browser
۱۹۹.....	کنترل ActiveDocumentHost
۱۹۹.....	بررسی عمیق
۱۹۹.....	ایجاد رابطه‌های Office
۲۰۰.....	تبادل با محتویات وب
۲۰۲.....	کار با محتوای سند
۲۰۳.....	نتیجه‌گیری
۲۰۳.....	کاربرد: تغییرات اعمال شده در کنترل‌های موجود Windows Forms
۲۰۳.....	مفاهیم جدید
۲۰۴.....	حاشیه‌ها (Margins) و فضای خالی درون کنترل‌ها (Padding)
۲۰۵.....	کامل کردن خودکار (Auto-Completion)
۲۰۶.....	پیکربندی (Configuration)
۲۰۷.....	تنظیمات پیکربندی
۲۰۸.....	کلاس‌های پوششی (Wrapper)
۲۰۹.....	پشتیبانی IDE
۲۱۰.....	بررسی عمیق
۲۱۰.....	استفاده از Padding
۲۱۱.....	استفاده از خاصیت کامل کردن خودکار

۲۱۲.....	استفاده از سیستم بیکربندی جدید
۲۱۳.....	نتیجه‌گیری
۲۱۵.....	فصل ششم/ ساختن برنامه‌های کاربردی وب
۲۱۷.....	کاربرد : وب سایت‌های بدون کد (Zero-Code)
۲۱۷.....	مفاهیم جدید
۲۱۸.....	زبان‌های توصیفی
۲۱۹.....	پیوند با داده‌ها
۲۲۰.....	بررسی عمیق
۲۲۲.....	پیوند دو طرفه با داده‌ها
۲۲۶.....	پیرایش یک سایت Zero-Code
۲۲۸.....	نتیجه‌گیری
۲۲۸.....	کاربرد : عضویت، پروفایل، و Role ها
۲۲۸.....	مفاهیم جدید
۲۲۹.....	عضویت
۲۳۰.....	پروفایل کاربر
۲۳۰.....	امنیت بر پایه نقش
۲۳۱.....	بررسی عمیق
۲۳۱.....	عضویت (Membership)
۲۳۹.....	نتیجه‌گیری
۲۳۹.....	کاربرد : کنترل‌های جدید وب
۲۳۹.....	مفاهیم جدید
۲۴۱.....	کنترل‌های نقشه سایت (Site Map)
۲۴۲.....	کنترل Substitution (Post-Cache Substitution)
۲۴۲.....	بررسی عمیق

۲۴۳.....	کنترل‌های امنیتی
۲۵۰.....	LoginStatus کنترل
۲۵۲.....	ChangePassword کنترل
۲۵۲.....	نقشه‌های سایت (Site Maps)
۲۵۴.....	جانشینی بعد از Cache (Post-Cache Substitution)
۲۵۶.....	نتیجه‌گیری
۲۵۷.....	کاربرد: صفحات Master و Theme ها
۲۵۷.....	مفاهیم جدید
۲۵۸.....	صفحات Master
۲۵۹.....	Themes
۲۶۰.....	بررسی عمیق
۲۶۰.....	ایجاد یک صفحه Master
۲۶۲.....	استفاده از یک صفحه Master
۲۶۶.....	استفاده از Theme ها
۲۶۷.....	ایجاد یک theme سفارشی
۲۶۸.....	نتیجه‌گیری
۲۶۹.....	فصل هفتم/ پیشرفت‌های چارچوب NET. مایکروسافت
۲۷۰.....	کاربرد: پیشرفت‌های کتابخانه پایه کلاس‌ها
۲۷۰.....	مفاهیم جدید
۲۷۰.....	پیشرفت‌های مربوط به نوع داده
۲۷۲.....	مجموعه‌های Generic
۲۷۳.....	منابع نوع‌دار
۲۷۳.....	Exception ها و Tracing
۲۷۴.....	بررسی عمیق

۲۷۴.....	تعیین درستی و تبدیل داده‌های رشته‌ای (String)
۲۷۶.....	استفاده از مجموعه‌های generic
۲۷۶.....	نظارت بر سرعت اجرای برنامه
۲۷۷.....	گرفتن رد رویدادهای برنامه
۲۷۹.....	استفاده از منابع نوع‌دار
۲۷۹.....	نتیجه‌گیری
۲۷۹.....	کاربرد : پیشرفت‌های رابط Console
۲۸۰.....	مفاهیم جدید
۲۸۰.....	دستکاری رابط کاربر
۲۸۰.....	دریافت متن
۲۸۱.....	دستکاری Buffer
۲۸۱.....	بررسی عمیق
۲۸۲.....	دستکاری پنجره Console
۲۸۲.....	گرفتن اطلاعات مربوط به کلیدها
۲۸۳.....	نتیجه‌گیری
۲۸۴.....	کاربرد : پیشرفت‌های مربوط به امنیت
۲۸۴.....	مفاهیم جدید
۲۸۵.....	StrongNameIdentityPermission
۲۸۶.....	مجوز هویت GAC
۲۸۶.....	بررسی عمیق
۲۸۷.....	اعمال GACIdentityPermission
۲۸۸.....	اعمال StrongNameIdentityPermission
۲۸۹.....	سایر موارد اضافه شده یا ارتقاء یافته
۲۸۹.....	نتیجه‌گیری

۲۹۱	فصل هشتم / ارائه نهایی برنامه‌ها
۲۹۲	کاربرد : به روز کردن برنامه‌ها توسط کاربر با Click Once
۲۹۲	مفاهیم جدید
۲۹۳	مقدمه
۲۹۴	امنیت دسترسی به کد (Code Access Security)
۲۹۵	مزایای Click Once
۲۹۶	منتشر کردن یک برنامه Click Once
۲۹۸	نصب و اجرای یک برنامه Click Once
۲۹۹	به روز کردن یک برنامه Click Once
۳۰۲	پیکربندی ویژگی‌های Click Once
۳۰۲	مکان انتشار
۳۰۳	مد نصب
۳۰۴	گزینه‌های انتشار
۳۰۵	زبان انتشار
۳۰۵	نام برنامه در منو Start
۳۰۵	صفحه Publish
۳۰۶	تعیین نیازمندی‌های مربوط به روزآمد کردن برنامه
۳۰۷	پیکربندی گزینه‌های خودکار به روزشدن
۳۰۸	تنظیم کردن بازه‌های زمانی برای چک کردن به روزشدگی
۳۰۹	پهنای باند خود را در نظر داشته باشید
۳۰۹	پیش نیازهای برنامه
۳۱۱	فایل‌های برنامه
۳۱۳	امنیت Click Once
۳۱۳	برنامه‌های Click Once در یک Sandbox اجرا می‌شوند

۳۱۴.....	Click Once برای Code Access Security
۳۱۷.....	Permission Calculator
۳۱۷.....	اشکال‌زدایی در یک منطقه امنیتی (Security Zone)
۳۱۸.....	ارتقای مجوزها
۳۱۹.....	مدل ارتقاء یافته برای سیاست امنیتی
۳۲۰.....	ضمانت اصالت کد
۳۲۰.....	پیشرفت‌های مربوط به Windows Installer
۳۲۱.....	پیشرفت‌های Windows Installer 3.0
۳۲۱.....	ترکیب با Visual Studio 2005
۳۲۲.....	ساختن یک بسته نصب Windows Installer
۳۲۴.....	نصب فایل‌ها، پوشه‌ها، و میانبرها
۳۳۰.....	بررسی عمیق
۳۳۰.....	فراهم کردن دسترسی از UI
۳۳۱.....	افزودن کد به Check for Updates
۳۳۲.....	نتیجه‌گیری

مقدمه

هدف این نسخه از Visual Studio مایکروسافت بالا بردن بهره‌وری برنامه‌نویس بوده است. صرف‌نظر از نوع برنامه کاربردی یا زبان برنامه‌نویسی که شما انتخاب می‌کنید، مایکروسافت تلاش‌های خود را روی ساختن یک محیط برنامه‌سازی مجتمع^۱ (IDE)، کتابخانه کلاس‌ها^۲ و مجموعه‌ای از زبان‌های برنامه‌نویسی متمرکز کرده، که به وسیله آنها ایجاد برنامه‌های کاربردی تا آنجا که ممکن است آسان شود.

برما، نویسندگان این کتاب روشن است که مایکروسافت تمام جنبه‌های برنامه‌نویسی شامل نوشتن خود کد، رفع اشکال^۳ و ارائه نهایی^۴ آن را در نظر گرفته و واقعاً روش‌هایی که می‌توان انجام این کارها را بهبود بخشید را مورد بررسی قرار داده است. پیشرفت‌ها گسترده هستند و شامل تغییراتی در خود زبان Visual Basic.NET مایکروسافت، بهینه‌سازی‌هایی در IDE، ساده‌سازی دسترسی به داده‌ها و نمایش آنها، تغییراتی که بهره‌وری برنامه‌نویس را هنگام ایجاد رابط‌های کاربر^۵ حرفه‌ای افزایش می‌دهد، افزودن تعداد قابل توجهی کلاس‌های جدید به کتابخانه کلاس‌ها برای کاهش مقدار کدی که شما باید بنویسید و افزایش قدرت و سادگی ارائه نهایی می‌شود.

این کتاب به طور عمقی به هریک از این موضوع‌ها می‌پردازد. علاوه بر این، درحالی که این کتاب بر ساختن برنامه کاربردی desktop تمرکز می‌کند، بیشتر برنامه‌نویسان حداقل کمی برنامه‌نویسی وب هم انجام می‌دهند. به این دلیل، فصل ۶ «ساختن برنامه‌های کاربردی وب» بعضی تغییرات مهم که در ASP.NET اعمال شده را پوشش می‌دهد.

در این مقدمه به‌طور مختصر به این عنوان‌ها می‌پردازیم و به شما نشان می‌دهیم برای یافتن اطلاعات عمیق‌تر به سراغ کدام فصل از کتاب بروید. شما می‌توانید این کتاب را از ابتدا تا انتها مطالعه کنید یا فقط فصل‌هایی که به آنها علاقه دارید را مطالعه کنید.

چه کسی باید این کتاب را بخواند

این کتاب برای برنامه‌نویسان Visual Basic است. اگر شما یک برنامه‌نویس Visual Basic 6.0 هستید، باید از فصل اول «مقدمه‌ای بر چارچوب NET. مایکروسافت برای برنامه‌نویس Visual Basic» که شما را برای برنامه‌نویسی در چارچوب NET آماده می‌کند. بقیه کتاب به معرفی پیشرفت‌های انجام شده در Visual Basic می‌پردازد، که سبب بالا رفتن خلاقیت شما در ایجاد برنامه‌های کاربردی می‌شود. اگر شما

¹. Integrated Development Environment

². class library

³. debugging

⁴. Deployment

⁵. user interfaces

در حال حاضر یک برنامه‌نویس .NET هستید، می‌توانید از فصل ۱ چشم‌پوشی کنید و مستقیماً به سراغ فصل ۲ «پیشرفت‌های زبان در Visual Basic 2005» بروید.

در این کتاب چه می‌توان یافت

این کتاب شامل هشت فصل است. فصل اول «مقدمه‌ای به چارچوب .NET، مایکروسافت برای برنامه‌نویس Visual Basic» بیش از همه برای برنامه‌نویسانی که با Visual Basic 6.0 کار می‌کنند مفید است. در این فصل مقدمه‌ای از برنامه‌نویسی برای چارچوب .NET، ارائه می‌شود.

این بخش هر فصل را توضیح می‌دهد و در مورد هر عنوان اطلاعات مقدماتی به شما می‌دهد.

مهم: برای اجرا کردن برنامه‌های نمونه در این کتاب، باید دو نرم‌افزار را نصب کرده باشید.



Visual Studio 2005 و یکی از نسخه‌های Enterprise, Developer, Desktop یا SQL server 2005 Desktop Engine. SQL Server 2000 باید به همراه پایگاه داده نمونه Nothwind نصب شوند. کدهای نمونه را می‌توانید در CD همراه کتاب پیدا کنید.

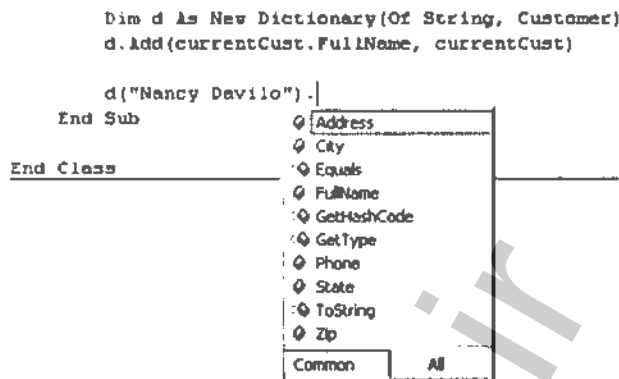
فصل ۲ «پیشرفت‌های زبان در Visual Basic 2005»

این فصل به معرفی generic ها، شی My و عملگرهای جدید می‌پردازد.

Generic ها: احتمالاً مهمترین پیشرفت در چارچوب .NET معرفی generic ها بوده است. به زبان ساده، generic ها به شما امکان می‌دهند تا مجموعه‌هایی را ایجاد کنید که مخصوص نوع اشیایی هستند که در آن مجموعه‌ها ذخیره می‌کنید.

```
Dim Customer Dictionary As New Dictionary(Of String, Customer)
Customer Dictionary.Add(Current Cust.Full Name, Current Cust
```

در این مثال Customer Dictionary فقط می‌تواند اعضای از نوع Customer را ذخیره کند و عامل جستجو به‌طور خاص یک مقدار از نوع String است. چون مجموعه می‌داند که چه چیزی را ذخیره می‌کند، (Customeres به طور خاص و نه به طور کلی Objects)، می‌تواند از قابلیت IntelliSense مایکروسافت بهره‌برداری کامل کند، همان‌طور در شکل زیر نشان داده شده است.



شکل م-۱: قابلیت IntelliSense برای مجموعه‌های generic

شیء MY^۱

یک پیشرفت مهم دیگر در Visual Basic.NET ویژگی است که «شیء MY» نامیده می‌شود. این ویژگی سبب دسترسی آسان و سریع به عملگرهای قدرتمند می‌شود. برای مثال دستور My.Application.Log.WriteEntry به شما امکان می‌دهد اطلاعات را در یک فایل متنی، یک فایل لوگ رویداد^۲ یا هر جای دیگری بنویسید. دستور My.Computer.Audio.Play به شما امکان می‌دهد یک فایل wav را پخش کنید. همچنین شیء My به شما اشیاء پیش‌فرضی برای نوع‌هایی که بخشی از پروژه شما هستند را می‌دهد. برای مثال، شما می‌توانید با استفاده از دستور My.Forms.MainForm.show یک فرم را نمایش دهید بدون اینکه مجبور باشید ابتدا یک شیء از آن را ایجاد کنید.

عملگرهای جدید

همچنین Visual Basic.NET شامل تعدادی کلمه کلیدی^۳ جدید می‌باشد. IsNot مقایسه با Nothing را آسان می‌کند. در گذشته، شما اغلب مجبور بودید بنویسید If Not x Is Nothing Then حالا می‌توان آن را به صورت If x Is Not Nothing Then نوشت. همچنین Visual Basic.NET از تعریف عملگرها، generic ها و ویژگی‌های برنامه‌نویسی پیشرفته دیگری پشتیبانی می‌کند.

برای اینکه بیشتر در مورد این پیشرفت‌ها بدانید، فصل ۲ «پیشرفت‌های زبان در Visual Basic 2005» را ببینید.

^۱. My Object

^۲. event log

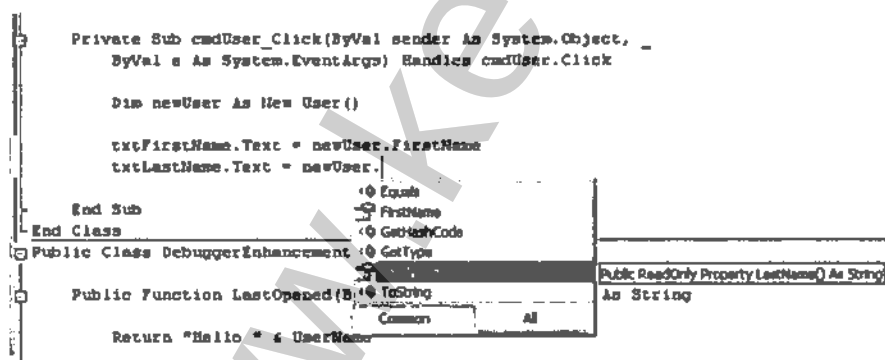
^۳. keyword

فصل سوم «محیط مجتمع برنامه‌سازی Visual Studio 2005»

برنامه‌سازی با Visual Basic همواره روی بهره‌وری تکیه داشته است و ابزار کلیدی برای فراهم آوردن این بهره‌وری Visual Studio می‌باشد. با Visual Studio 2005، بهره‌وری به میزان وسیعی پیشرفت کرده است. حالا دسترسی به محتویات help بهتر و آسانتر شده، محیط برنامه‌سازی آسانتر سفارشی می‌شود، کار کدنویسی و رفع اشکال^۱ با بهره‌وری بیشتری همراه است و شامل معرفی دوباره یک ویژگی نام ویرایش - و - ادامه^۲ می‌شود.

پیشرفت‌ها در کدنویسی و رفع اشکال

وقتی با محیط مجتمع برنامه‌سازی جدید (IDE) کار می‌کنید، مشاهده می‌کنید که تلاش زیادی برای افزودن یک زمینه به کدنویسی شما انجام شده است. یعنی، در بسیاری حالت‌ها، نیازی ندارید تا برای بدست آوردن اطلاعاتی که به آن نیاز دارید به یک پنجره دیگر در IDE بروید. در سایر وضعیت‌ها، IDE نوعی پالایش هوشمند اطلاعات را انجام می‌دهد به طوری که آنچه شما به احتمال فراوان به آن نیاز دارید در جلو و وسط قرار می‌گیرد و گزینه‌هایی که به ندرت مورد استفاده قرار می‌گیرند، جلوی دست و پا را نمی‌گیرند. این طرز کار از همان بار اولی که شما به یک ویژگی از یک شیء دسترسی پیدا می‌کنید آشکار می‌باشد، همان‌طور که در شکل زیر نشان داده شده است.



شکل ۲- نمایش مرحله به مرحله اعضای یک شیء

همان‌طور که می‌توانید ببینید، Visual Basic 2005 می‌تواند تمام اعضای یک شیء را نمایش دهد، یا فهرست را به صورتی پالایش کند که فقط اعضای که به صورت متداول مورد استفاده قرار می‌گیرند نمایش داده شوند. برای مثال یافتن ویژگی `ConnectionString` به این صورت آسان می‌شود.

همچنین Visual Basic 2005، خاصیت آشنای کامل کردن کد Visual Basic را ارتقا داده است. قبلاً شما اعلان ابتدایی `Function` را تایپ می‌کردید و سپس IDE به صورت خودکار یک `End Function` برای شما

^۱. debugging

^۲. Edit- and- continue

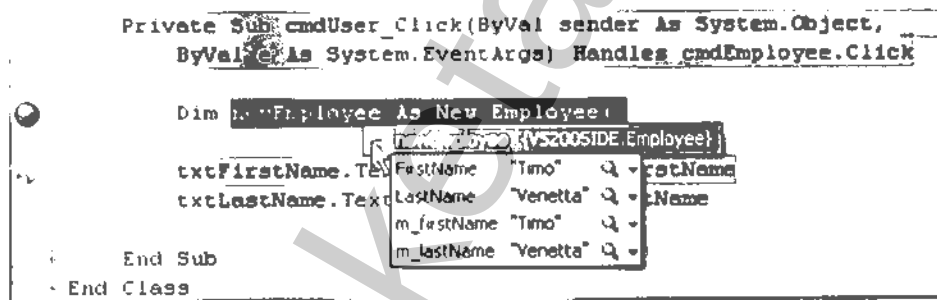
قرار می‌داد. حالا این خاصیت به چیزهایی مانند اعلان ویژگی‌ها^۱ هم توسعه یافته است. با تایپ کردن Public Property User Name as String و فشار دادن کلید Enter کد زیر به صورت خودکار ایجاد می‌شود.

```
Public Property UserName() As String
    Get
```

```
End Get
    Set (ByVal value As String)
```

```
End Set
End Property
```

همچنین Visual Basic 2005 در زمان رفع اشکال، اطلاعات را در نوک انگشتان شما قرار می‌دهد و به شما امکان می‌دهد بدون رفتن به سراغ پنجره Watch، نوع‌های ساده و پیچیده را به طور دقیق مورد بررسی قرار دهید. شکل زیر نشان می‌دهد که چگونه یک نوع پیچیده مستقیماً در پنجره ویرایش کد نمایش داده می‌شود.



شکل م-۳ نمایش ویژگی‌های یک نوع پیچیده، در محل

یکی از بزرگترین پیشرفت‌ها در بهره‌وری که بوسیله Visual Basic 6 ارائه شده، ویرایش - و- ادامه^۲ بود. این قابلیت یکی از عناصر مهم در رفع اشکال محسوب می‌شد و متأسفانه در نسخه‌های 2002 و Visual Studio.NET 2003 در دسترس نبود. ویرایش - و- ادامه برگشته است و به شما امکان می‌دهد تا بدون نیاز به متوقف کردن و شروع دوباره، ضمن اجرای برنامه اشکالات را رفع کنید.

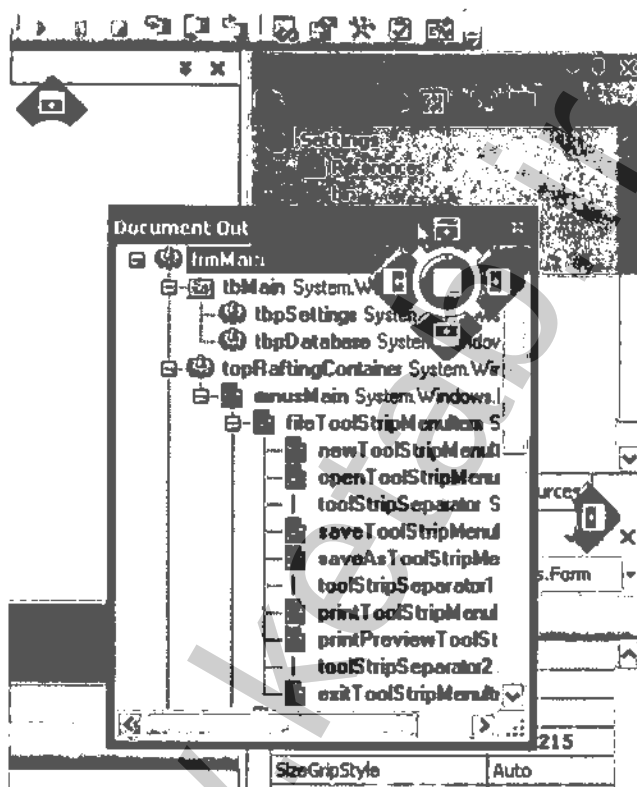
سفارشی کردن IDE و پیشرفت‌های سیستم Help

برنامه سازان Visual Basic ارزش زیادی را برای قابلیت‌های سفارش کردن محیط برنامه‌سازی و قرار دادن پنجره‌ها در جاهایی که بیشتر برای آنها مفید است، قائلند. Visual Studio همواره این قابلیت را فراهم آورده است، اما در Visual Basic 2005 این قابلیت به صورت چشمگیری پیشرفت کرد. که چندین ناحیه مختلف را دربرمی‌گیرد. اول اینکه Visual Basic 2005 مشکل «نمی‌دانم این پنجره به کجا متصل

^۱ Properties

^۲ Edit- and- Continue

می‌شود» را حذف کرده است. این مشکل باعث می‌شود پنجره‌ها آن‌طور که شما فکر می‌کردید در جای خود قرار نگیرند. در Visual Basic 2005، به کمک docking guides عمل قرار دادن پنجره‌ها در جای مورد نظر در یک لحظه انجام می‌شود، همان‌طور که در شکل زیر نشان داده شده است.



شکل م-۴ استفاده از docking guides

شما به سادگی پنجره را با پیکان مناسب جابجا می‌کنید تا روی، بالای، در سمت چپ یا راست یا پایین یک پنجره دیگر قرار گیرد همچنین سفارشی کردن IDE را می‌توان منتقل^۱ کرد یا آن را وارد کرد^۲. علاوه بر این سیستم Help بطور عمده‌ای ارتقا یافته است، و شما حالا تجربه بهتری از فشار دادن «F1» بدست می‌آورید، و می‌توانید به آخرین help های آنلاین دسترسی پیدا کنید. سیستم help حتی به یک جامعه از وبسایت‌ها متصل می‌شود و دسترسی به سایت‌هایی مانند Code Guru، Net247 و ASP.NET را درست در داخل IDE آسان می‌کند. برای اطلاعات بیشتر در مورد پیشرفت‌های، مهم انجام شده در Visual Basic 2005، می‌توانید مستقیماً به فصل ۲ «محیط مجتمع برنامه‌سازی Visual Basic 2005» مراجعه کنید.

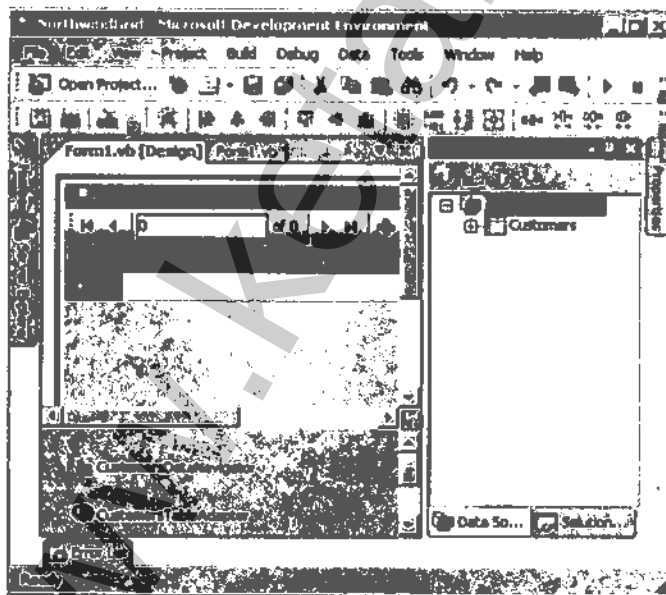
^۱. Export

^۲. Import

فصل ۴ «ساختن برنامه‌های کاربردی داده محور»^۱

Visual Basic 2005 به همراه خود مجموعه‌ای از ابزارهای قدرتمند برای ساختن برنامه‌های کاربردی داده محور را می‌آورد. اول، تعدادی کلاس جدید به ADO.NET 1.0 اضافه شده‌اند، که شامل یک کلاس Data Adapter می‌شود که کار فراخواندن یک رویه ذخیره شده را به آسانی فراخواندن یک تابع می‌کند. کادر ابزار^۲ هم حالا شامل یک کنترل شبکه‌ای^۳ جدید است که به آسانی سفارشی و کنترل می‌شود. همچنین IDE، شامل یک پنجره جدید Data Sources می‌شود که به شما کمک می‌کند این اشیاء را مدیریت کنید و در زمان طراحی رابط کاربر خودتان را ایجاد کنید.

پنجره Data Sources برای ایجاد Data Adapter های نوع دار^۴ و DataSet ها به کار می‌رود که شما می‌توانید از آنها در یک برنامه کاربردی استفاده کنید. همینکه به یک پایگاه داده متصل شدید و جدول‌ها و رویه‌هایی که می‌خواهید را انتخاب کردید، به آسانی می‌توانید اشیاء را از پنجره Data Sources به صفحه طراحی بکشید تا رابط کاربر ایجاد شود. همانطور که در شکل زیر نشان داده شده است.



شکل م-۵ ساختن رابط کاربر با استفاده از پنجره Data Sources

شما می‌توانید به صورت پیش‌فرض این نکته را ببینید، وقتی جدول Customers روی سطح design قرار می‌گیرد، با استفاده از کنترل جدید Grid View به این سطح اضافه می‌شود. همچنین می‌توانید کاری کنید

^۱. data-centric

^۲. Toolbox

^۳. grid

^۴. typed

که این جدول به صورت فیلدهای جداگانه به سطح design اضافه شود، و می‌توانید نوع کنترل به کار رفته توسط هر فیلد را تعیین کنید.

همچنین می‌توانید با استفاده از DataSet Designer متدها را به رویه‌های ذخیره شده^۱ تصویر کنید.^۲ با این کار فراخواندن یک رویه ذخیره شده به آسانی فراخواندن کد زیر می‌شود:

```
ds = MyDataSet.GetByCustomerID(customerID)
```

برای بدست آوردن اطلاعات بیشتر در مورد استفاده از داده‌ها در Visual Basic 2005، فصل ۴ «ساختن برنامه‌های کاربردی داده محور» را ببینید.

فصل ۵ «ایجاد رابط‌های کاربری»^۳

برنامه‌سازان Visual Basic هیچگاه مشکلی در ساختن سریع رابط‌های کاربری نداشته‌اند. اگر یک حیطه وجود داشته که Visual Basic همیشه در آن درخشان است، آن وجود جعبه ابزار و طراح فرم^۴ بوده است. با این حال همیشه یک چیز ورد زبان برنامه‌سازان Visual Basic بوده است و آن این است که «کنترل‌هایی را که مایکروسافت در ساختن برنامه‌های کاربردی خودش به کار می‌برد را در اختیار ما قرار دهید». و به طور خاص برنامه نویسان همیشه خواهان عناصر رابط کاربری مانند نوار ابزارهای شناور^۵ بوده‌اند که در محصولاتی مانند Office مایکروسافت دیده می‌شوند.

کنترل‌های جدید

شما از دانستن این موضوع خوشحال خواهید شد که این نسخه از Visual Studio فقط روی یک جعبه ابزار پر از چیزهای مختلف متمرکز نشده، بلکه این نسخه از نرم‌افزار واقعاً به دنبال آسان کردن فرایند ایجاد رابط‌های سفارشی می‌باشد. اولین کنترلی که در فصل ۵ مورد بررسی قرار می‌گیرد ToolStrip است که در شکل زیر نشان داده شده است.

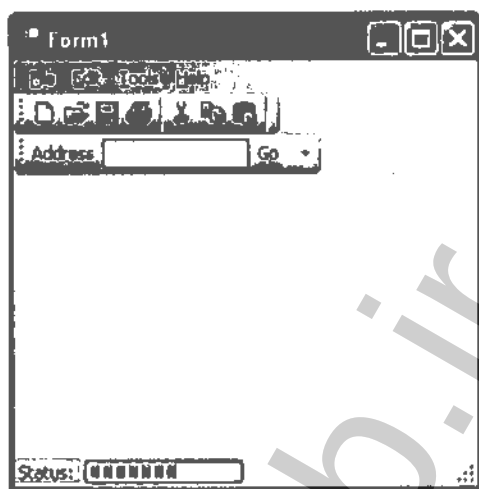
¹. Stored- Procedure

². Map

³. User interfaces

⁴. form designer

⁵. rafting



شکل ۶-۱ MenuStrip، Status Strip و ToolStrip

می‌توانید مشاهده کنید که یک فرم ساده با دو کنترل ToolStrip ایجاد شده است. این کار در چند ثانیه انجام شده است، اما در عمل چند چیز وجود دارند که باید در مورد آنها بحث شود. اول اینکه، شما می‌توانید ببینید که کنترل‌های ToolStrip از خاصیت شناور بودن پشتیبانی می‌کنند یعنی کاربر می‌تواند کنترل‌های ToolStrip را به این طرف و آن طرف بکشد و در زمان اجرای برنامه آنها را مرتب کند. دوم اینکه، ToolStrip بالایی حاوی نمادهای استانداردی است که شما با آنها آشنایی دارید. اینها با یک کلیک ساده اضافه شده‌اند. اگر به یک نماد نیازی نداشتید، به سادگی آن را انتخاب کرده و کلید Delete را فشار دهید. همچنین ToolStrip از چیزهایی بیش از فقط دکمه‌ها پشتیبانی می‌کند و می‌تواند شامل کادرهای متنی^۱، فهرست‌های کرکره‌ای^۲، برچسب‌ها^۳ و چیزهایی دیگر باشد در آخر اینکه نوار ابزارها دقیقاً شبیه به آن چیزی هستند که در Office مایکروسافت می‌بینیم. به این ترتیب کار ساختن برنامه‌های کاربردی با ظاهر حرفه‌ای که کاربران به سرعت استفاده از آنها را بیاموزید را ساده می‌کند.

نوار منو^۴ و نوار وضعیت^۵ بر پایه ساختار ToolStrip ساخته می‌شوند و به شما امکان می‌دهند تا گستره وسیع‌تری از کنترل‌ها و قابلیت‌ها را در نوارهای منو و وضعیت داشته باشید. در جعبه ابزار، شما یک پوشش^۶ کامل NET. حول کنترل ActiveX مرورگر وب را پیدا می‌کنید، که کار افزودن عملکرد وب به برنامه‌های کاربردی شما را آسان می‌کند. از این کنترل می‌توان برای نمایش help بر پایه وب یا بردن کاربر به یک صفحه وب برای خرید یک محصول استفاده کرد.

^۱. Text boxes

^۲. Drop Down Lists

^۳. Labels

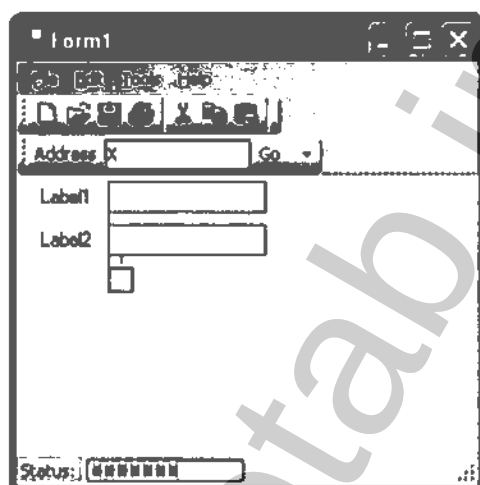
^۴. Menu bar

^۵. Status bar

^۶. Wrapper

تغییرات در کنترل‌های موجود

یکی از اولین چیزهایی که موقع ساختن یک فرم متوجه آن می‌شوید Snapline ها هستند، که کار ردیف کردن کنترل‌ها در زمان ساختن رابط کاربر را آسان می‌کنند، همان‌طور که در شکل زیر نشان داده شده است.



شکل م-۷ استفاده از Snapline ها برای ردیف کردن کنترل‌ها.

وقتی کنترل‌ها را روی سطح design به این طرف و آن طرف می‌کشیم، به جای دیدن نقطه چین‌های همیشگی، Snapline ها ظاهر می‌شوند و کار ردیف کردن کنترل‌ها را آسان می‌کنند. Snapline ها به شما کمک می‌کنند تا لبه‌های کنترل‌ها را هم‌خط کنید و همچنین یک فاصله خالی مناسب را بین کنترل‌ها ایجاد می‌کنند. این فاصله خالی توسط یک ویژگی^۱ جدید برای کنترل‌ها به نام Margin ایجاد می‌شود.

همچنین بسیاری از کنترل‌ها شامل ویژگی‌های جدید «کامل کردن خودکار»^۲ هستند که به کنترل‌ها امکان می‌دهد به صورت خودکار نام فایل‌ها، URL ها، فهرست‌های سفارشی یا سایر منابع را مهیا کنند.

برای اینکه بیشتر در مورد ساختن رابط‌های کاربر حرفه‌ای، و ساختار جدید برنامه‌ها یاد بگیرید فصل ۵ «ایجاد رابط‌های کاربر را ببینید»

فصل ششم، «ساختن برنامه‌های کاربردی وب»

نسخه‌های ۲۰۰۲ و Visual Studio ۲۰۰۳ انقلابی را در برنامه‌سازی وب ایجاد کردند، و سبب شدند برنامه‌سازان وب نسبت به آنچه در ASP کلاسیک امکان‌پذیر بود، قدرت خلاقیت بسیار بیشتری داشته باشند. هدف تیم سازنده این نرم‌افزار این بوده که حداقل ۷۰ درصد از مقدار کدنویسی برای ساختن یک

^۱ Property

^۲ Auto Completion

سایت وب را کاهش دهد. با در نظر گرفتن تمام جنبه‌ها، وقتی Visual Basic 2005 به بازار آمد، تیم سازنده آن مشاهده کردند که به هدف خود رسیده‌اند. حالا حتی بعضی از سایت‌ها را می‌توان بدون حتی یک خط کدنویسی هم ایجاد کرد.

با نسخه‌های 1.0 و ASP.NET 1.1، شما می‌توانید به صورت تصویری (Visual) رابط کاربر را ایجاد کنید. اما برای ارتباط با پایگاه داده، باز آوری داده‌ها و پیوند دادن رابط کاربر (UI) با داده‌ها نیاز به کدنویسی دارید. در ASP.NET 2.0، مؤلفه‌هایی^۱ وجود دارند که منبع داده‌ها^۲ را به صورت یک جزء مستقل دربرمی‌گیرند. این به شما امکان می‌دهد که به روشنی بتوانید مثلاً با یک منبع داده‌های SQL Server ارتباط برقرار کنید. عمل پیوند کاملاً پیچیده است. شما نیاز دارید که تغییراتی را در داده‌ها ایجاد کرده و این تغییرات را در پایگاه داده هم اعمال کنید. این ویژگی جدید مانند کنترل‌های «زمان طراحی» که در گذشته در دسترس بودند، نمایشی نیستند. کنترل‌ها به صورتی تغییر کرده‌اند که ظاهر و محتوای UI کاملاً سفارشی شونده باشد، و شما می‌توانید با نوشتن کد با استفاده از ابزارهای تصویری (Visual)، با این کنترل‌ها ارتباط برقرار کنید.

همچنین میکروسافت سناریوهای کلیدی را شناسایی کرده و زیر ساخت‌های این سناریوها را در چارچوب .NET گنجانده است. برای مثال، ASP.NET 2.0 مفهوم عضویت^۳ و شخصی شدن^۴ را به صورت درون ساخته^۵ دارد. عضویت یک مجموعه کارایی‌ها است که کنترل‌های متداول برای چیزهایی مانند Login، تغییر رمز عبور و مانند اینها را فراهم می‌کند. همچنین ساختاری را برای ذخیره کردن و بازآوری اطلاعات از یک پایگاه داده را نیز مهیا می‌کند. و در آخر، عضویت به شما API^۶هایی را می‌دهد که بوسیله آنها می‌توانید از طریق برنامه‌نویسی با کاربر ارتباط برقرار کنید.

همچنین ASP.NET به همراه خود تعدادی کنترل جدید را می‌آورد که ساختن سناریوهای کلیدی را آسان می‌کنند. برای مثال، کنترل SiteMapPath قابلیت درون ساخته bread crumb را فراهم می‌کند، و Site Map Data Source ایجاد ساختار راهبری سایت را آسان می‌کند. Master Pages به شما امکان می‌دهند تا ناحیه‌های مشترک صفحه را در یکجا تعریف کنید، بطوریکه فقط نواحی اختصاص داده شده به مطالب صفحه تغییر کنند. Caching ارتقا پیدا کرده و حالا داده‌ها بر یک پایه «برای کاربر» در Cache ذخیره می‌شوند.

برای اطلاعات بیشتر در مورد ASP.NET 2.0، فصل ۶ «ساختن برنامه‌های کاربردی وب» را ببینید.

^۱. Components

^۲. Data Source

^۳. Membership

^۴. Personalization

^۵. Built-in

^۶. Application Programming Interface

فصل هفتم «پیشرفت‌های چارچوب NET، مایکروسافت»

پیشرفت‌ها منحصر به محیط برنامه‌سازی نمی‌شوند. چارچوب زیر بنایی هم دستخوش پیشرفت‌های مهمی شده است. genericها که یک پیشرفت مهم و یک ویژگی قدرتمند زبان هستند در نسخه 2.0 چارچوب NET، معرفی شده‌اند و در فصل دوم شرح داده می‌شوند. این فصل پیشرفت‌های مربوط به کتابخانه کلاس‌های اصلی، رابط Console و امنیت را دربرمی‌گیرد.

چارچوب NET، حاوی تعداد بیشماری کلاس‌های جدید است که هر چه بیشتر مقدار کدی که باید بنویسید را کاهش می‌دهد و افزودن قابلیت‌ها به برنامه‌تان را آسان‌تر می‌کند.

StopWatch قابلیت‌های انعطاف بیشتری را در چگونگی گرفتن رد داده‌ها مربوط به رویدادها فراهم می‌کند. پیشرفت‌های مهمی در فضای نام^۱ System.Net فراهم آمده. حالا یک پوشش^۲ کاملاً مدیریت شده برای Internet Explorer وجود دارد. System.XML پیشرفت‌های مهمی کرده، هم از نظر قابلیت‌ها و هم از نظر کیفیت اجرا و حالا شامل پشتیبانی برای Xquery می‌شود.

برای اطلاعات بیشتر در مورد این کلاس و پیشرفت‌های انجام شده در Console و امنیت^۳، فصل ۷ «پیشرفت‌های چارچوب NET، مایکروسافت» را ببینید.

فصل هشتم، «ارائه نهایی^۴ برنامه‌های کاربردی»

در Visual Basic 6.0 ارائه یک برنامه کاربردی عموماً شامل ساختن یک بسته نصب کننده^۵ برای Windows می‌شد. این فرایند ضروری بود چون Visual Basic 6.0 بر پایه COM عمل می‌کرد و برنامه‌های کاربردی به تعدادی DLL نیاز داشتند که قبل از اجرای برنامه باید ثبت^۶ می‌شدند. برنامه نصب کننده، همچنین میانبرهای منو Start را ایجاد می‌کرد، پایگاه‌های داده را نصب می‌کرد و وظایف ضروری دیگر را انجام می‌داد.

نسخه‌های 1.0، 1.1 چارچوب NET، درب رابه روی گزینه‌های بیشتری برای نصب باز کرد. اول، نسخه 1.0 از آنچه اغلب ارائه Xcopy نامیده می‌شود پشتیبانی می‌کرد. به این معنی که برنامه‌هایی کاربردی ساده را می‌توان روی دیسک سخت کپی و اجرا کرد. این کار به این خاطر انجام می‌شود که DLLهای NET، نیازی به ثبت شدن ندارند. همچنین نسخه 1.0 از آن چیزی که به عنوان ارائه بدون تماس^۷ یا

^۱. Name Space

^۲. Wrapper

^۳. Security

^۴. Deploying

^۵. Installer Package

^۶. Register

^۷. No- touch deployment

href-exe شناخته می‌شود. پشتیبانی می‌کرد. در این روش ارائه یک بخش قابل اجرای یک برنامه روی یک سرور وب قرار می‌گیرد. وقتی با مرورگر به URL برنامه می‌روید، برنامه بارگذاری^۱ و اجرا می‌شود. نتیجه یک برنامه کاربردی Windows است که به سهولت یک برنامه کاربردی وب ارائه می‌شود. در .NET حتی اگر نیاز به ایجاد یک بسته نصب کننده برای Windows داشته باشید، فرایند انجام آن کار به میزان زیادی آسان شده است، چون دیگر نیازی به نگرانی در مورد ثبت DLL ها وجود ندارد.

روش ارائه بدون تماس از طرف بسیاری یک روش عالی برای برنامه‌های کاربردی اینترنت معرفی شد. بوسیله یک ورودی ساده شما همیشه می‌توانید مطمئن باشید که کاربران آخرین نسخه برنامه را اجرا می‌کنند. در Visual Studio 2005 میکروسافت به طور عمده‌ای قابلیت پخش برنامه‌ها از طریق وب را ارتقا داده است. اول اینکه، حالا برنامه‌ها می‌توانند به صورت کامل بارگذاری و نصب شوند این قابلیت میانبرهای منوی Start مناسب را ایجاد می‌کند و (مهمتر از همه) و به برنامه‌ها امکان می‌دهد، حتی زمانی که به اینترنت وصل نیستید هم به طور کامل اجرا شوند.

همچنین برنامه‌های کاربردی Click Once شما را ملزم نمی‌کنند که برای نصب آنها حتماً مدیر سیستم باشید (بسته‌های نصب برای Windows چنین الزامی را دارند).

در آخر اینکه Click Once به شما امکان می‌دهد، نرم‌افزارهای دیگر را هم به عنوان بخشی از عمل نصب در نظر بگیرید. برای مثال، یک برنامه کاربردی Click Once می‌تواند MDAC یا حتی چارچوب .NET را در صورت نیاز به عنوان بخشی از نصب برنامه نصب کند. برای اطلاعات بیشتر در مورد Click Once، فصل ۸ «ارائه نهایی برنامه‌های کاربردی» را ببینید.

^۱. Download