

计算机类专业人才培养内涵建设项目系列教材

ASP.NET 项目开发实践教程

主编 陈品华 李鸿君



WUHAN UNIVERSITY PRESS

武汉大学出版社

计算机类专业人才培养内涵建设项目系列教材

ASP.NET 项目开发实践教程

主 编 陈品华 李鸿君

副主编 耿春喜 昌 鸣



WUHAN UNIVERSITY PRESS

武汉大学出版社

图书在版编目(CIP)数据

ASP.NET 项目开发实践教学/陈品华,李鸿君主编. —武汉:武汉大学出版社,2016.9

计算机类专业人才培养内涵建设项目系列教材

ISBN 978-7-307-18566-1

I. A… II. ①陈… ②李… III. 网页制作工具—程序设计—高等学校—教材 IV. TP393.092.2

中国版本图书馆 CIP 数据核字(2016)第 203422 号

责任编辑:张欣 曲生伟

责任校对:周卫思

装帧设计:张希玉

出版发行:武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件:whu_publish@163.com 网址:www.stmpress.cn)

印刷:虎彩印艺股份有限公司

开本:787×1092 1/16 印张:17.5 字数:443 千字

版次:2016 年 9 月第 1 版 2016 年 9 月第 1 次印刷

ISBN 978-7-307-18566-1 定价:44.00 元

版权所有,不得翻印;凡购买我社的图书,如有质量问题,请与当地图书销售部门联系调换。

计算机类专业人才培养内涵建设项目系列教材 编写委员会

主 任	刘新宇					
副主任	黄 群	张 健				
委 员	蔡 红	支永昌	田 春	吴建平		
	孔 令	朱景德	杨 柳			

前 言

.NET 平台的出现对于 Web 应用程序的开发具有十分重要的意义,它推动了下一代 Internet 的发展。作为 .NET 平台的一部分,ASP.NET 提供了一种新的编程模型和结构,能更快速、更高效地建立灵活、安全和稳定的 Web 应用程序。由于使用了众多的服务器控件和 .NET Framework 提供的 .NET 类,ASP.NET 使程序员可以只编写少量的代码便能够为 Web 应用程序创建丰富的功能。开发者可以使用任何与 .NET 兼容的语言来编写 ASP.NET 应用程序,本书采用的是 C#。

本书以通俗易懂的语言、丰富多彩的实例,详细介绍了如何利用 ASP.NET 技术进行 Web 程序的开发。本书具有“边做边学、工学结合”的鲜明特色。全书以实际项目为主线,以任务为载体,主要围绕 C# 编程、ASP.NET 控件、数据库开发等几个方面展开。

本书内容共分 5 个项目。

项目 1 重在讲解 ASP.NET 开发理论知识,通过 3 个任务讲述了 ASP.NET 和 .NET Framework 简介、C# 语言基础、Visual Studio,让读者能够在进行 ASP.NET 开发之前了解一些基础理论知识,并通过实例来进行知识点强化。

项目 2 重在讲解 ASP.NET 开发基础技术,通过 4 个任务讲述了 ASP.NET 4.0 基础知识、ASP.NET 服务器控件、ASP.NET 应用程序状态管理、异常处理和日志追踪等知识,让读者在实际训练过程中入门。

项目 3 重在讲解 ASP.NET 开发中的数据访问技术,通过 5 个任务讲述了 ADO.NET 数据访问技术、ASP.NET 数据绑定技术、数据绑定控件、缓存技术、LINQ 技术等知识,并通过任务训练来掌握数据访问技术。

项目 4 重在讲解 ASP.NET 应用程序安全知识,通过实例训练来掌握 ASP.NET 应用程序安全设置。

项目 5 介绍综合项目实例——“幼儿歌曲表演与设计”精品课程 CMS 系统,详细讲述了该系统的架构、设计思想、业务分析过程以及最后的代码实现。通过这部分的学习,读者可以掌握企业级项目的设计思路、开发流程和解决问题的方法,从而对 ASP.NET 编程有更深入的认识。

本书具有如下特点:

(1) 本书选取真实项目开展教学,以够用为原则,结合校企合作经验设置教学内容,同时有明确的职业岗位针对性,学习目标非常明确,可以按照“做中学、学中做”的教学方法来引导学生学习。



(2)引入 CDIO 工程教育模式,开展基于项目的教育和学习,学生按照项目的推进顺序进行学习,在完成具体项目的过程中学习技能,构建理论知识,发展职业能力。

(3)任务实施中步骤清晰,引导读者举一反三,帮助读者完成项目的同时构建理论知识,有助于提高读者分析和解决问题的能力。

(4)作为系列教程,上下册无论从结构上还是内容上都保持了一致,给计算机专业学生系统学习 Web 开发技术带来了很大的便利。

本书由陈品华、李鸿君担任主编,负责教材总体设计及统稿;耿春喜、昌鸣担任副主编;纪承周、吴成峰、王昊、赵强强、胡耀华、肖枫担任参编。其中,项目 1 由陈品华、耿春喜共同编写,项目 2 由昌鸣编写,项目 3、项目 4 由耿春喜编写,项目 5 由耿春喜、李鸿君编写。

本书在编写过程中得到了上海行健职业学院、上海市市北职业高级中学领导和老师的大力支持,在此表示感谢。

本书在编写过程中还参考了近年来出版的 ASP.NET 技术相关的教材及杂志,以及因特网上的相关资料,对于疏忽遗漏未列出来的参考教材和网址,一并致谢并表歉意。

本书是一种新的教学尝试,能否得到同行的认可,能否给学生带来新的感受,都要经过实践的检验。由于编者水平有限,书中疏漏、不足之处在所难免,敬请读者批评指正。

编 者

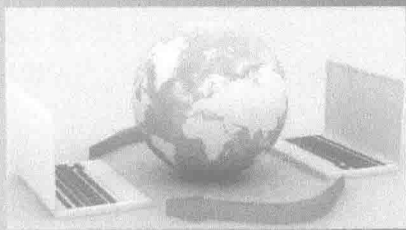
2016 年 6 月

目 录

项目 1 ASP.NET 开发理论	1
任务 1 ASP.NET 和 .NET Framework 简介	3
任务 2 C# 语言基础	11
任务 3 Visual Studio 2015 集成开发环境安装	19
项目 2 ASP.NET 开发基础	27
任务 1 ASP.NET 4.0 技术简介	29
任务 2 ASP.NET 服务器控件	36
任务 3 ASP.NET 应用程序状态管理	68
任务 4 异常处理和日志追踪	89
项目 3 数据访问技术	102
任务 1 ADO.NET 数据访问技术	104
任务 2 ASP.NET 数据绑定技术	126
任务 3 数据绑定控件	132
任务 4 缓存技术	153
任务 5 LINQ 技术	167
项目 4 成员和角色管理	180
任务 成员和角色管理技术	182
项目 5 项目开发	207
任务 1 需求分析	209
任务 2 系统架构	214
任务 3 产品实现	225
任务 4 产品部署	261
参考文献	271

项目 1

ASP.NET 开发理论



ASP.NET 技术是目前 Web 开发技术中主流技术之一,也是公司网络开发使用最多和应用人群最广的技术。本项目采用层层递进的方法,详细介绍了 ASP.NET 的新技术、新特色,包括 .NET Framework 和 Visual Studio 2015 集成化开发环境、C# 语言核心语法及用法等。

◆ 知识目标

- (1) 掌握 Visual Studio 2015 集成化开发环境的部署和操作。
- (2) 掌握 C# 编程语言基本语法。
- (3) 理解 .NET Framework 基础。
- (4) 理解 ASP.NET。
- (5) 了解 C# 4.0 语言特色。

◆ 能力目标

- (1) 能够熟练安装 Visual Studio 2015。
- (2) 能够熟练掌握 Visual Studio 2015 IDE 环境中的主要操作方法。
- (3) 能够熟练创建 ASP.NET Web 应用程序。
- (4) 能够使用 C# 编写简单事例。

◆ 引例描述

小胡在完成了《Web 前端项目开发实践教程》的学习和实践后,对 Web 前端开发技术有了较全面的认识,具备了开发能力,他对 Web 后台开发技术也产生了浓厚的兴趣,于是就去找经理请教。

小胡:经理,我已经完成了《Web 前端项目开发实践教程》的学习和实践,现在我想接触后端项目开发知识,不知道该从何入手。

经理:小胡,你接受新知识的能力确实挺强,要想学习后端项目开发知识,必须先想好学习什么知识,我推荐你学习 ASP.NET 项目开发,它入门快,操作方便。

小胡:经理,如果要学习 ASP.NET 项目开发,应该从何处入手呢?

经理:如果你下定决心要学习 ASP.NET 项目开发,就要先了解以下知识。

(1)ASP.NET 和 .NET Framework;

(2)C# 语言基础;

(3)Visual Studio 集成开发工具。





任务1 ASP.NET 和 .NET Framework 简介



任务陈述

本任务讲述 ASP.NET 和 .NET Framework 理论知识,同时学习 .NET Framework 相关知识。



知识储备

1.1 ASP.NET

ASP.NET 是一个统一的 Web 开发平台,包括使用尽可能少的代码生成企业级 Web 应用程序所必需的各种服务。ASP.NET 作为 .NET Framework 的一部分提供。当编写 ASP.NET 应用程序的代码时,可以访问 .NET Framework 中的类。可以使用与公共语言运行库 (CLR) 兼容的任何语言来编写应用程序的代码,这些语言包括 Microsoft Visual Basic、C#、JavaScript .NET 和 J#。使用这些语言,可以开发公共语言运行库、类型安全、继承等方面优点的 ASP.NET 应用程序。ASP.NET 包括:

- (1) 页和控件框架。
- (2) ASP.NET 编译器。
- (3) 安全基础结构。
- (4) 状态管理功能。
- (5) 应用程序配置。
- (6) 运行状况监视和性能功能。
- (7) 调试支持。
- (8) XML Web services 框架。
- (9) 可扩展的宿主环境和应用程序生命周期管理。
- (10) 可扩展的设计器环境。

ASP.NET 的原始设计构想,就是要让开发人员能够像使用 Visual Basic 开发工具那样,可以使用事件驱动式程序开发模式 (Event-driven Programming Model) 的方法来开发网页与应用程序,若要以 ASP 技术来做这件事,必须要使用大量的辅助信息,如查询字符串或窗体字段数据来识别与判断对象的来源、事件流向及调用的函数,等等,需要撰写的代码量



相当多,但 ASP.NET 很巧妙地利用窗体字段和 JavaScript 脚本把事件的传递模型隐藏起来了。

ASP.NET 的事件模型由数个 Hidden Field 组合而成,基于 HTTP 模型的限制,所有的网页程序在运行结果输出到用户端后,就会退出运行。为了维护在 ASP.NET 网页与控件的状态数据,因此在输出 ASP.NET 控件时,ASP.NET 会将部分状态数据储存到网页的 Hidden Field 中,这种类型的状态数据称为 ViewState(ID 为 _VIEWSTATE),在服务器端即会被解译出状态与事件数据。在大多数的内置 Web 控件中都有使用到这个机制,因此在使用大量 ASP.NET Web 控件的网页中,会有许多的 ViewState 存放在网页中并随着 HTTP 数据流输出到用户端。ViewState 在输出时,会被加密为一组乱码字符串,其密钥值定义在计算机中,并且每一个对象都会被序列化(serialize)成字符串(若是自定义对象要放到 ViewState 时,则要让它支持序列化),再输出到 _VIEWSTATE 字段中,在每次的网页来回时都会被传输,较大的 ViewState 会让网页膨胀,不利于快速的网络传输,不过 ASP.NET 本身有提供将 ViewState 关闭的功能,因此如果控件不需要保存状态时,可将它关闭以减少输出的数量。

为确保控件的事件能够被引发,让事件驱动能够被运行,需要在控件事件引发命令时引入参数,交由 JavaScript 脚本在用户端引发时,填入另一个 Hidden Field(ID 为 _EVENTTARGET 及 _EVENTARGUMENT),并且引发窗体送出指示(submit),传送到服务端后,服务端的 HttpApplication 中的工具函数会解析 _EVENTTARGET 和 _EVENTARGUMENT 字段中的信息,并且交由控件所实现的 RaisePostBackEvent()来引发事件,并由 .NET Framework 内部的事件处理器接手处理(调用控件设置的事件处理程序)。

1.2 .NET Framework

.NET Framework 简称 .NET,是微软为开发应用程序创建的一个富有革命性的新平台。在这个平台中,可以开发出运行在 Windows 上的几乎所有的应用程序。简单地说,.NET Framework 就是一个创建、部署和运行应用程序的多语言多平台环境,包含了一个庞大的代码库,各种 .NET 语言都可以共用这些代码库。

.NET Framework 功能非常强大,主要体现在以下几个方面。一、它提供了一个面向对象的编程环境,完全支持面向对象编程。二、它提高了软件的可复用性、可扩展性、可维护性、灵活性,这些特点读者在今后的学习中要慢慢体会。三、对 Web 应用的强大支持。如今是互联网时代,大量的网络应用程序发挥着越来越重要的作用。Web Service 是 .NET 非常重要的内容。比如登录 Hotmail 和 MSN 时都要使用 Hotmail 和 MSN 的账户,其实支持这个账户的应用就是一个 Web 服务(Passport)。Visual Studio 是一个世界级的开发工具,与 .NET Framework 配合,能够方便快捷地开发出多种 .NET 应用程序,还可以进行测试、版本控制、Team 开发和部署等。接下来,一起来了解 .NET Framework 中的重要组件,如图 1-1 所示。

从图 1-1 中可以看出,.NET 安装在 Windows 之上,支持如 C#、VB.NET、VC++、.NET 等开发语言,也就是所谓的跨语言开发。图 1-1 中的公共语言运行时通常写成 CLR(Common Language Runtime),CLR 是所有 .NET 应用程序的运行环境,是所有 .NET 应用程序都要使用的编程基础。

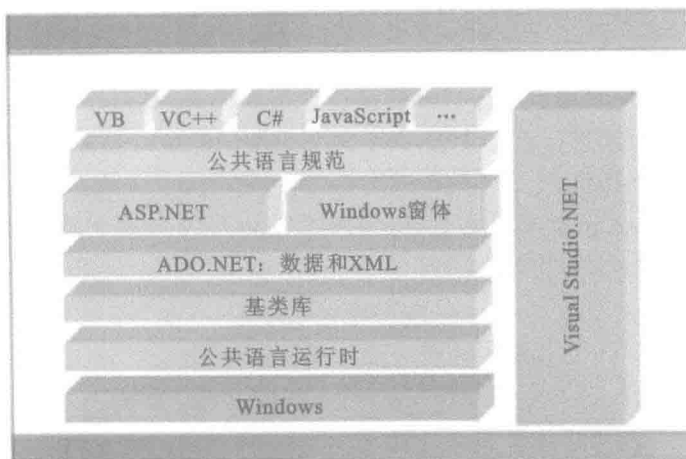


图 1-1 .NET Framework 结构中的组件

.NET Framework 是支持生成和运行下一代应用程序和 XML Web Services 的内部 Windows 组件。.NET Framework 旨在实现下列目标:

- ① 提供一个一致的面向对象的编程环境,而无论对象代码是在本地存储和执行,还是在本地执行但在 Internet 上分布,或者是在远程执行的。
- ② 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- ③ 提供一个可提高代码(包括由未知的或不完全受信任的第三方创建的代码)执行安全性的代码执行环境。
- ④ 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- ⑤ 使开发人员的经验在面对类型大不相同的应用程序(如基于 Windows 的应用程序和基于 Web 的应用程序)时保持一致。
- ⑥ 按照工业标准生成所有通信,以确保基于 .NET Framework 的代码可与任何其他代码集成。

.NET Framework 具有两个主要组件:公共语言运行库和 .NET Framework 类库。公共语言运行库是 .NET Framework 的基础。可以将运行库看作一个在执行时管理代码的代理,它提供内存管理、线程管理和远程处理等核心服务,并且强制实施严格的类型安全以及可提高安全可靠的其他形式的代码准确性。事实上,代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码,而不以运行库为目标的代码称为非托管代码。.NET Framework 类库是一个综合性的面向对象的可重用类型集合,可以使用它开发多种应用程序,这些应用程序包括传统的命令行或图形用户界面(GUI)应用程序,也包括基于 ASP.NET 所提供的创新应用程序(如 Web 窗体和 XML Web Services)。

.NET Framework 可由非托管组件承载,这些组件将公共语言运行库加载到它们的进程中并启动托管代码的执行,从而创建一个可以同时利用托管和非托管功能的软件环境。.NET Framework 不但提供若干个运行库宿主,而且支持第三方运行库宿主的开发。

例如,ASP.NET 承载运行库来为托管代码提供可伸缩的服务器端环境。ASP.NET 直接使用运行库以启用 ASP.NET 应用程序和 XML Web Services(本任务稍后将对这两者进行讨论)。



Internet Explorer 是承载运行库(以 MIME 类型扩展的形式)的非托管应用程序的一个示例。使用 Internet Explorer 承载运行库使用户能够在 HTML 文档中嵌入托管组件或 Windows 窗体控件。以这种方式承载运行库使得托管移动代码(类似于 Microsoft® ActiveX® 控件)成为可能,不过它需要进行重大改进(如不完全受信任的执行和独立的文件存储),而这种改进只有托管代码才能提供。

图 1-2 显示公共语言运行库和类库与应用程序之间以及与整个系统之间的关系,该图还显示托管代码如何在更大的结构内运行。

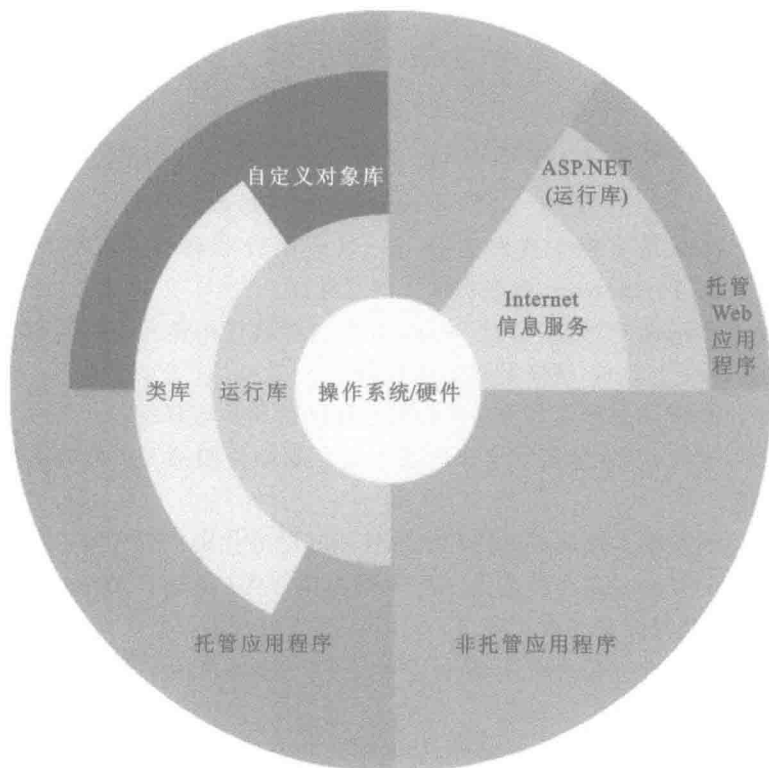


图 1-2 .NET Framework 环境

下面将更加详细地描述 .NET Framework 的主要组件和功能。

1. 公共语言运行库

公共语言运行库管理内存、线程执行、代码执行、代码安全验证、编译及其他系统服务。这些功能是在公共语言运行库上运行的托管代码所固有的。

公共语言运行库的安全性取决于包括托管组件的来源(如 Internet、企业网络或本地计算机)在内的一些因素,托管组件被赋予不同程度的信任。这意味着即使在同一应用程序中,托管组件既可能能够执行文件访问操作、注册表访问操作或其他须小心使用的功能,又可能不能够执行这些功能。

运行库强制实施代码访问安全。例如,用户可以相信嵌入网页中的可执行文件能够在屏幕上播放动画或歌曲,但不能访问他们的个人数据、文件系统或网络。这样,运行库的安全性功能就使通过 Internet 部署的合法软件能够具有特别丰富的功能。

运行库还通过实现称为通用类型系统(CTS)的严格类型验证和代码验证基础结构来加强代码的可靠性。CTS 确保所有托管代码都是可以自我描述的。各种 Microsoft 和第三方语言编译器生成符合 CTS 的托管代码。这意味着托管代码可在严格实施类型保真和类型安全的同时使用其他托管类型和实例。

此外,运行库的托管环境还解决了许多常见的软件问题。例如,运行库自动处理对象布局并管理对象的引用,在不再使用它们时将它们释放。这种自动内存管理避免了两个最常见的应用程序错误:内存泄漏和无效内存引用。

运行库还提高了开发人员的工作效率。例如,程序员可以用他们选择的开发语言编写应用程序,也能充分利用其他开发人员用其他语言编写的运行库、类库和组件。任何选择以运行库为目标的编译器供应商都可以这样做。以 .NET Framework 为目标的语言编译器使得用该语言编写的现有代码可以使用 .NET Framework 的功能,这大大减轻了现有应用程序在迁移过程中的工作负担。

尽管运行库是为未来的软件设计的,但是它也支持现在和以前的软件。托管和非托管代码之间的互操作性使开发人员能够继续使用所需的 COM 组件和 DLL。

运行库旨在增强性能。尽管公共语言运行库提供许多标准运行库服务,但是它从不解释托管代码。一种称为实时(JIT)编译的功能使所有托管代码能够以它在其上执行的系统的本机语言运行。同时,内存管理器排除了出现零碎内存的可能性,并增大了内存引用区域以进一步提高性能。

运行库可由高性能的服务器端应用程序[如 Microsoft® SQL Server™ 和 Internet 信息服务(IIS)]承载。此基础结构使用户在享受支持运行库宿主的行业最佳企业服务器的优越性能的同时,能够使用托管代码编写业务逻辑。

2. .NET Framework 类库

.NET Framework 类库是一个与公共语言运行库紧密集成的可重用类型集合。该类库是面向对象的,并提供用户自己的托管代码可从中导出功能的类型。这不但使 .NET Framework 类库易于使用,而且减少了学习 .NET Framework 的新功能所需要的时间。此外,第三方组件可与 .NET Framework 中的类无缝集成。

例如,.NET Framework 集合类实现一组可用于开发用户自己的集合类的接口。用户自己的集合类将与 .NET Framework 中的类无缝地混合。

正如用户对面向对象的类库所希望的那样,.NET Framework 类库能够完成一系列常见编程任务(包括诸如字符串管理、数据收集、数据库连接以及文件访问等任务)。除这些常见任务之外,类库还包括支持多种专用开发方案的类型。例如,可使用 .NET Framework 开发下列类型的应用程序和服务:

- (1)控制台应用程序。
- (2)Windows GUI 应用程序(Windows 窗体)。
- (3)ASP.NET 应用程序。
- (4)XML Web Services。
- (5)Windows 服务。

例如,Windows 窗体类是一组综合性的可重用类型,它大大简化了 Windows GUI 的开发。如果要编写 ASP.NET Web 窗体应用程序,可使用 Web 窗体类。



3. 客户端应用程序开发

客户端应用程序在基于 Windows 的编程中最接近传统风格的应用程序。这是用户在桌面上显示窗口或窗体能够执行任务的应用程序类型。客户端应用程序包括诸如字处理程序和电子表格等应用程序,还包括自定义的业务应用程序(如数据输入工具、报告工具等)。客户端应用程序通常使用窗口、菜单、按钮和其他 GUI 元素,并且它们可以访问本地资源(如文件系统)和外围设备(如打印机)。

另一种客户端应用程序是作为网页通过 Internet 部署的传统 ActiveX 控件(现在被托管 Windows 窗体控件所替代)。此应用程序非常类似于其他客户端应用程序:它在本机执行,可以访问本地资源,并包含图形元素。

过去,开发人员结合使用 C/C++ 与 Microsoft 基础类(MFC)或应用程序快速开发(RAD)环境(如 Microsoft® Visual Basic®)来创建此类应用程序。NET Framework 将这些现有产品的特点合并到了单个且一致的开发环境中,该环境大大简化了客户端应用程序的开发。

包含在 .NET Framework 中的 Windows 窗体类旨在用于 GUI 开发。用户可以轻松创建能够适应多变的商业需求所需的灵活性的命令窗口、按钮、菜单、工具栏和其他屏幕元素。

例如,.NET Framework 提供简单的属性以调整与窗体相关联的可视属性。某些情况下,基础操作系统不支持直接更改这些属性,而在这些情况下,.NET Framework 将自动重新创建窗体。这是 .NET Framework 集成开发人员接口从而使编码更简单、更一致的方法之一。

和 ActiveX 控件不同,Windows 窗体控件具有对用户计算机的不完全受信任的访问权限。这意味着二进制代码或在本机执行的代码可访问用户系统上的某些资源,如 GUI 元素和访问受限制的文件,但这些代码不能访问或危害其他资源。由于具有代码访问安全性,许多曾经需要安装在用户系统上的应用程序现在可以通过 Web 部署。用户的应用程序可以像网页部署时那样实现本地应用程序的功能。

4. 服务器应用程序开发

在托管领域中,服务器端应用程序是通过运行库宿主实现的,非托管应用程序承载公共语言运行库,后者使用户的自定义托管代码可以控制服务器的行为。此模型在获得主服务器的可伸缩性性能的同时提供给用户公共语言运行库和类库的所有功能。

图 1-3 显示的是在不同服务器环境中运行托管代码的基本网络架构。在应用程序逻辑通过托管代码执行时,服务器(如 IIS 和 SQL Server)可执行标准操作。

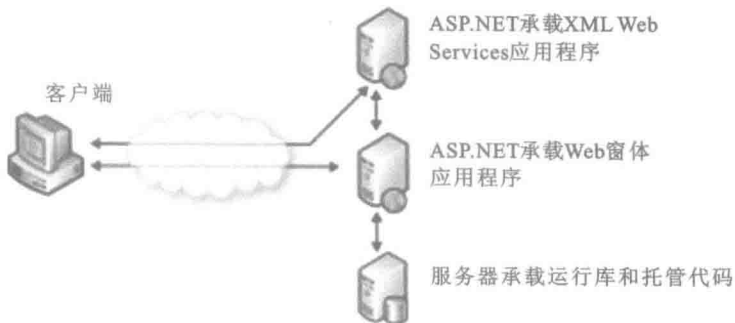


图 1-3 服务器端托管的基本网络架构

ASP.NET 是使开发人员能够使用 .NET Framework 开发基于 Web 的应用程序的宿主环境。但是,ASP.NET 不只是一个运行库宿主,它还是使用托管代码开发网站和通过 Internet 分布的对象的完整结构。Web 窗体和 XML Web Services 都将 IIS 和 ASP.NET 用作应用程序的发布机制,并且两者在 .NET Framework 中都具有支持类集合。

XML Web Services 作为基于 Web 技术的重要成果,是类似于常见网站的分布式服务器端应用程序组件。但是,与基于 Web 的应用程序不同,XML Web Services 组件不具有 UI 并且不以浏览器(如 Internet Explorer 和 Netscape Navigator)为目标。XML Web Services 由旨在供其他应用程序使用的可重用的软件组件组成,所谓的其他应用程序包括传统的客户端应用程序、基于 Web 的应用程序,甚至是其他 XML Web Services。因此,XML Web Services 技术正迅速地应用程序开发和部署推向高度分布式 Internet 环境。

使用过 ASP 技术的早期版本的用户能很快注意到 ASP.NET 和 Web 窗体提供的改进。例如,可以用支持 .NET Framework 的任何语言开发 Web 窗体页。此外,代码不再需要与 HTTP 文本共享同一个文件(尽管如果愿意,代码还可以继续这样做)。Web 窗体页用本机语言执行,这是因为与所有其他托管应用程序一样,它们能充分利用运行库。与此相对照,非托管 ASP 页始终被写成脚本并解释。ASP.NET 页比非托管 ASP 页更快、更实用,并且更易于开发,这是因为它们像所有托管应用程序一样与运行库进行交互。

.NET Framework 还提供类和工具的集合来帮助开发和使用 XML Web Services 应用程序。XML Web Services 是基于 SOAP(一种远程过程调用协议)、XML(一种可扩展的数据格式)和 WSDL(Web 服务描述语言)这些标准生成的。基于这些标准生成的 .NET Framework 的目的是提高与非 Microsoft 解决方案的互操作性。

例如,.NET Framework SDK 所包含的 Web 服务描述语言工具可以查询在 Web 上发布的 XML Web Services,分析它的 WSDL 描述,并产生 C# 或 Visual Basic 源代码,用户的应用程序可以使用这些代码而成为 XML Web Services 的客户端。这些源代码可以创建从类库中的类派生的类,这些类使用 SOAP 和 XML 分析处理所有基础通信。虽然用户可以使用类库来直接使用 XML Web Services,但 Web 服务描述语言工具和包含在 SDK 中的其他工具可以使用户更加方便地用 .NET Framework 进行开发。

如果用户开发和发布自己的 XML Web Services,.NET Framework 还提供了一组符合所有基础通信标准(如 SOAP、WSDL 和 XML)的类。使用这些类使用户能够将注意力集中在服务的逻辑上,而无须关注分布式软件开发所需要的通信基础结构。

最后,与托管环境中的 Web 窗体页相似,XML Web Services 将使用 IIS 的可伸缩通信以满足本机语言的速度运行。



任务实施

1. 启动安装包

双击鼠标运行 exe 应用程序,进入运行界面,如图 1-4 所示。

2. 接受许可条款

接受许可条款,点击“安装”按钮,如图 1-5 所示。

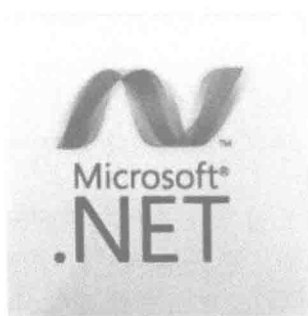


图 1-4 运行界面

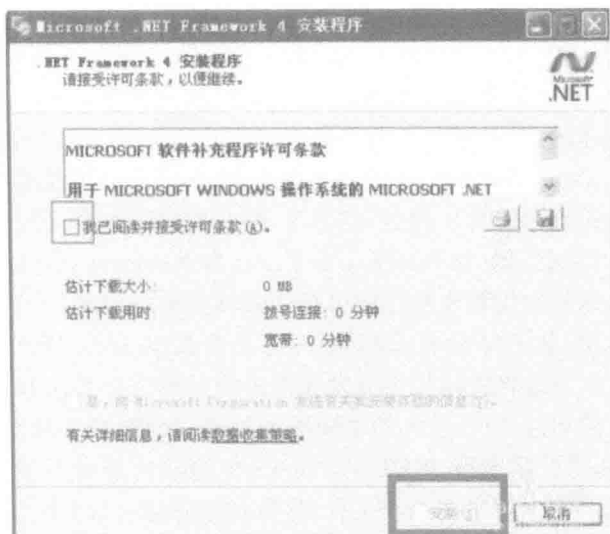


图 1-5 许可安装

3. 安装

安装提示双进度条,安装成功将会提示“已安装”,如图 1-6 所示。

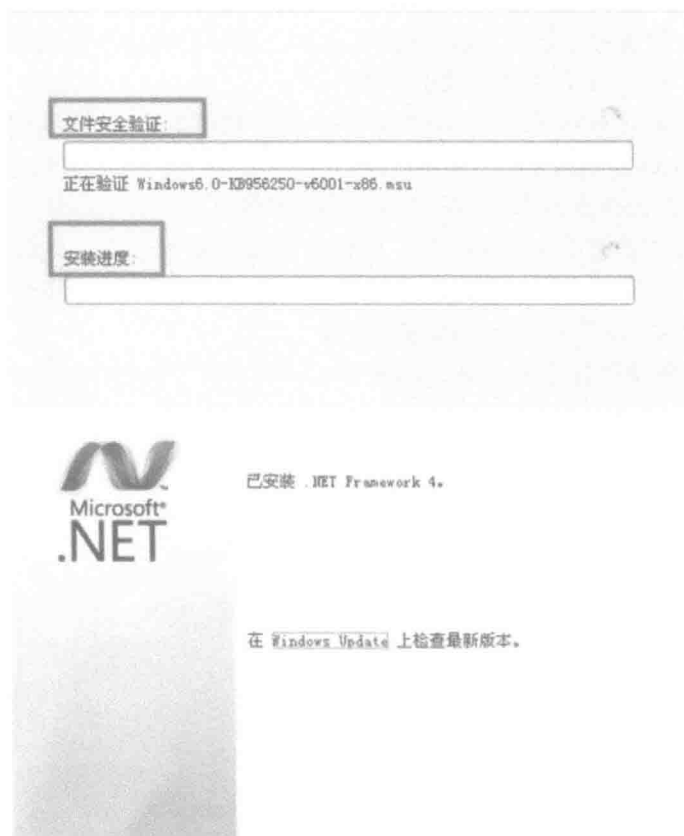


图 1-6 安装完毕