
第 1 章 ASP.NET 3.5 与开发工具

从本章开始，读者将能够系统的学习 ASP.NET 3.5 技术，相对于 ASP.NET 2.0 而言，在 3.5 版本的 ASP.NET 中并没有太多的变化，而更多的变化则在于 C# 编程语言中。而作为微软主推的编程语言，ASP.NET 3.5 能够使用 C# 的最新特性进行高效的开发，本章从基础讲解什么是 ASP.NET，以及开发工具的使用。

1.1 什么是 ASP.NET

ASP.NET 是微软推出的 ASP 的下一代 Web 开发技术。ASP.NET 顾名思义是基于 .NET 平台而存在的，在了解 ASP.NET 之前就需要了解 .NET 技术，了解 .NET 平台的相关技术才能够深入的了解 ASP.NET 是如何运作的。

1.1.1 .NET 历史与展望

.NET 技术是微软近几年推出的主要技术，微软为 .NET 技术的推出可谓是不遗余力，在 .NET 平台下，微软有着极大的野心，.NET 技术的发展历程如下所示。

- ❑ 2000 年 6 月，微软公司总裁比尔·盖茨在“论坛 2000”的会议上向业内公布 .NET 平台并描绘了 .NET 的愿景。
- ❑ 2002 年 1 月，微软发布 .NET Framework 1.0 版本，以及 Visual Studio .NET 2002 进行 .NET Framework 1.0 应用程序的辅助开发。
- ❑ 2003 年 4 月，微软发布 .NET Framework 1.1 版本，以及针对 .NET Framework 1.1 版本的开发工具 Visual Studio 2003，.NET Framework 1.1 版本较之于 .NET Framework 1.0 而言有重大的改进。
- ❑ 2004 年 6 月，微软在 TechEd Europe 会议上发布 .NET Framework 2.0 beta 版本，以及 Visual Studio 2005 的 beta 版本，在 Visual Studio 2005 的 beta 版本中包含了多个精简版，以便不同的开发人员的需要。
- ❑ 2005 年 4 月，微软发布 Visual Studio 2005 的 beta 2 版本。
- ❑ 2005 年 11 月，微软发布 Visual Studio 2005 的正式版和 SQL Server 2005 的正式版。
- ❑ 2006 年 11 月，微软发布 .NET Framework 3.0 版本，在其中加入了一些新特性，以及语法特性，这些特性包括 Windows Workflow Foundation、Windows Communication Foundation、Windows CardSpace 和 Windows Presentation Foundation。
- ❑ 2007 年 11 月，微软发布 .NET Framework 3.5 版本，在其中加入了更多的新特性，包括 LINQ、AJAX 等，为下一代软件开发做出准备。
- ❑ 2008 年 11 月，微软向业界发布 .NET Framework 4.0 社区测试版，以及 Visual Studio 2010 社区测试版，标识着 .NET 4.0 的到来。

在 .NET 发展的 8 年时间里，.NET 技术在不断的改进。虽然在 2002 年微软发布了 .NET 技术的第一个版本，但是由于系统维护和系统学习的原因，.NET 技术当时并没有广泛的被开发人员和企业所接受。

而自从.NET 2.0 版本之后,越来越多的开发人员和企业已经能够接受.NET 技术带来的革新。

而随着计算机技术的发展,越来越高的要求和越来越多的需求让开发人员不断的进行新技术的学习,这里包括云计算和云存储等新概念。.NET 平台同样为最新的概念和软件开发理念做出准备,这其中就包括 3.0 中出现并不断完善的 Windows Workflow Foundation、Windows Communication Foundation、Windows CardSpace 和 Windows Presentation Foundation 等应用。

在最新的操作系统 Vista 中,微软集成了.NET 平台,使用.NET 技术进行软件开发能够无缝的将软件部署在操作系统中,在进行软件的升级和维护中,基于.NET 平台的软件也能够快速升级。微软的.NET 野心不仅于此,微软的.NET 平台还在为多核化、虚拟化、云计算做准备。随着时间的推移,.NET 平台已经逐渐完善,学习.NET 平台以及.NET 技术对开发人员而言能够在未来的计算机应用中起到促进作用。

1.1.2 ASP.NET 与 ASP

对于 ASP.NET 而言,开发人员不可避免的会将 ASP.NET 与 ASP 进行比较,因为 ASP.NET 可以算是 ASP 的下一个版本。但是 ASP.NET 却与 ASP 完全不同,可以说微软重新将 ASP 进行编写和组织形成 ASP.NET 技术。

在传统的 ASP 开发中,开发人员可以在页面中进行 ASP 代码的编写,当服务器请求相应的页面时,服务器会解析 ASP 代码进行页面呈现。ASP 具有轻巧等特点,但是随着互联网的发展,ASP 也越来越多的呈现出其不足之处,这些不足之处包括 ASP 代码无法和 HTML 代码很好的分离,这就造成了页面代码混乱、维护性低等情况。当 ASP 中出现错误或者需要进行功能的添加,就需要多大部分的页面进行更改,这样就降低了 ASP 程序的复用性和维护性。

而随着互联网的不断发展,基于 Web 的应用程序诞生,ASP 已经不能满足日益增长的需求,于是诞生了 ASP.NET。ASP.NET 虽然同 ASP 都包含“ASP”这个词,但是 ASP.NET 与 ASP 完全是不同的编程模型,对于有 ASP 经验的人可以在页面中进行代码编写,而对于 ASP.NET 而言,ASP 的经验基本上不适用于 ASP.NET 的开发。ASP.NET 使用了软件开发的思路进行 Web 应用程序的编写,ASP.NET 是面向对象的开发模型,使用 ASP.NET 能够提高代码的重用性,降低开发和维护的成本。

而对于 ASP 而言,同样不能够满足日益增长的互联网需求,随着计算机科学与技术的发展,互联网和本地客户端的界限越来越模糊。一个 Web 应用程序可能是基于本地应用程序,而本地应用程序也可能基于服务器的服务进行开发的,这就对 Web 应用程序提出了更高的要求,相比之下,基于.NET 平台的 ASP.NET 却能够适应和解决复杂的互联网需求。

从历史发展的角度而言,不得不说 ASP 已经是过时的技术,但是并不代表 ASP 不会被使用,现在还有很多 ASP 应用程序,在小型的应用中,ASP 依旧是低成本的最佳选择。

1.1.3 ASP.NET 开发工具

相对于 ASP 而言,ASP.NET 具有更加完善的开发工具。在传统的 ASP 开发中,可以使用 Dreamware、FrontPage 等工具进行页面开发。当时使用 Dreamware、FrontPage 等工具进行 ASP 应用程序开发时,其效率并不能提升,并且这些工具对 ASP 应用程序的开发和运行也不会带来性能提升。

相比之下,对于 ASP.NET 应用程序而言,微软开发了 Visual Studio 开发环境提供给开发人员进行高效的开发,开发人员还能够使用现有的 ASP.NET 控件进行高效的应用程序开发,这些控件包括日历控件、分页控件、数据源控件和数据绑定控件。开发人员能够在 Visual Studio 开发环境中拖动相应的控件到页面中实现复杂的应用程序编写。

Visual Studio 开发环境在人机交互的设计理念上更加完善，使用 Visual Studio 开发环境进行应用程序开发能够极大的提高开发效率，实现复杂的编程应用，如图 1-1 所示。

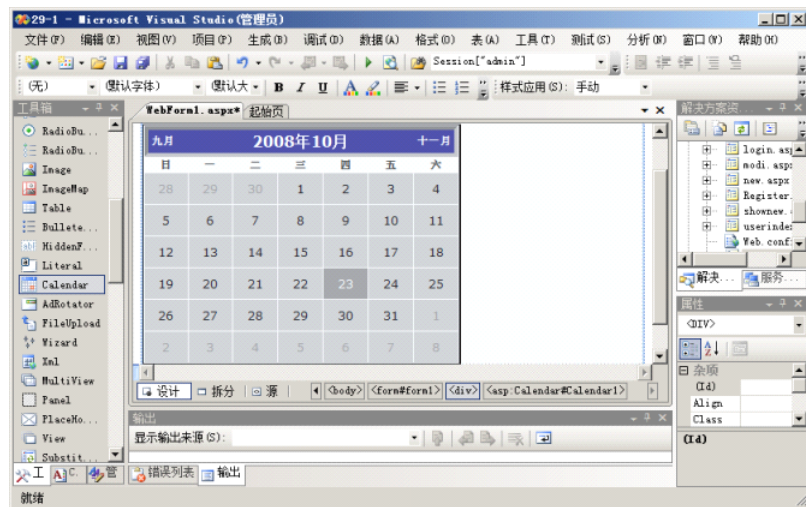


图 1-1 Visual Studio 开发环境

Visual Studio 开发环境为开发人员提供了诸多控件，使用这些控件能够实现在 ASP 中难以实现的复杂功能，极大的简化了开发人员的开发。如图 1-1 所示，在传统的 ASP 开发过程中需要实现日历控件是非常复杂和困难的，而在 ASP.NET 中，系统提供了日历控件用于日历的实现，开发人员只需要将日历控件拖动到页面中就能够实现日历效果。

使用 Visual Studio 开发环境进行 ASP.NET 应用程序开发还能够直接编译和运行 ASP.NET 应用程序。在使用 Dreamweaver、FrontPage 等工具进行页面开发时需要安装 IIS 进行 ASP.NET 应用程序运行，而 Visual Studio 提供了虚拟的服务器环境，用户可以像 C/C++ 应用程序编写一样在开发环境中进行应用程序的编译和运行。

1.1.4 ASP.NET 客户端

ASP.NET 应用程序是基于 Web 的应用程序，所以用户可以使用浏览器作为 ASP.NET 应用程序的客户端进行 ASP.NET 应用程序的访问。浏览器已经是操作系统中必备的常用工具，包括 IE 7、IE 8、Firefox、Opera 等常用浏览器都可以支持 ASP.NET 应用程序的访问和使用。对于 ASP.NET 应用程序而言，由于其客户端为浏览器，所以 ASP.NET 应用程序的客户端部署成本低，可以在服务器端进行更新而无需进入客户端进行客户端的更新。

1.1.5 ASP.NET 3.5 新增控件

在 ASP.NET 1.1 初期，开发人员抱怨微软自带的 ASP.NET 控件过少，无法满足日益增长的应用程序开发，而到了 ASP.NET 2.0 版本中，微软增加了数十种服务器控件用于应用程序的开发。这些服务器控件不仅在一定程度上实现的复杂的功能，还提升了应用程序的可维护性、可扩展性，同时这些服务器控件也提高了 ASP.NET 应用程序的代码的复用性。

在 ASP.NET 3.5 中，微软虽然没有像 ASP.NET 1.1 到 ASP.NET 2.0 一样增加数十种服务器控件，但是微软增加了 ListView 控件和 DataPager 控件两个颇受欢迎的服务器控件。使用 ListView 控件和

DataPager 控件能够快速地进行页面数据的呈现和布局，同时还能轻松的实现分页和数据更新等操作。

1. ListView 控件

ListView 控件是 ASP.NET 3.5 中新增的数据绑定控件。ListView 控件是介于 GridView 控件和 Repeater 之间的另一种数据绑定控件，相对于 GridView 来说，它有着更为丰富的布局手段，开发人员可以在 ListView 控件的模板内写任何 HTML 标记或者控件。

2. DataPage 控件

DataPager 控件通过实现 .NET 框架中 IPagableItemContainer 接口实现了控件的分页。在 ASP.NET 3.5 中，ListView 控件可以使用 DataPager 控件进行分页操作。

要在 ListView 中使用 DataPager 控件需要在 ListView 的 LayoutTemplate 模板中加入 DataPager 控件，DataPager 控件包括两种样式，一种是“上一页/下一页”样式，第二种是“数字”样式，方便了开发人员实现不同的分页效果。同时，用户不仅能够使用微软为开发人员提供的服务器控件，Visual Studio 2008 还能够让开发人员创建用户控件和自定义控件，以满足应用程序中越来越大的开发需求并提供了可扩展、可自定义控件。

1.1.6 ASP.NET 3.5 AJAX

在 Web 应用程序的开发中，越来越多的网站能够实现用户操作的无刷新效果。网站页面的无刷新效果能够提高用户体验、提高网站应用的操作性并能够降低服务器与客户端之间的通信次数。在 ASP.NET 3.5 中，Visual Studio 开发环境提供了 AJAX 应用环境，开发人员能够使用 Visual Studio 2008 进行 AJAX 应用程序和 AJAX 控件的创建，如图 1-2 所示。

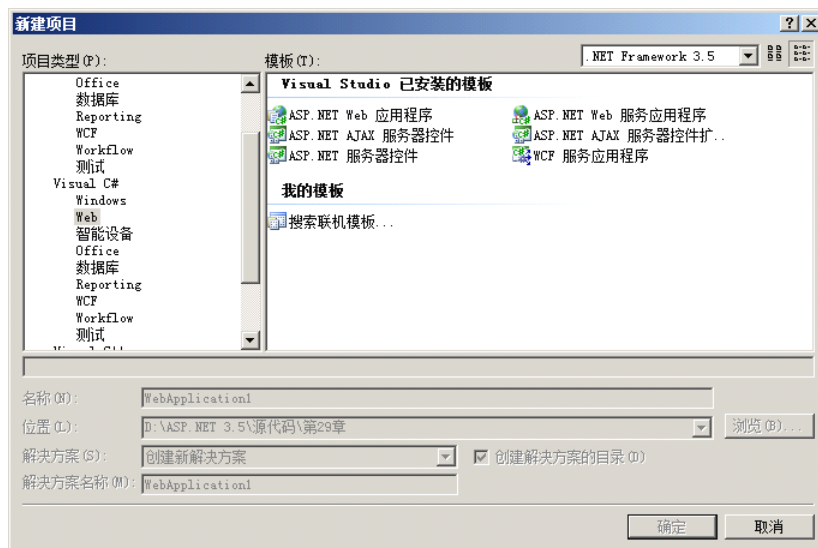


图 1-2 新增的 AJAX 服务器控件创建应用

用户可以创建 ASP.NET AJAX 服务器控件和服务扩展控件用于实现 ASP.NET AJAX 应用程序中所需使用的自定义控件。在 ASP.NET 3.5 中，Visual Studio 2008 还提供了默认的 AJAX 控件，这些控件包括脚本管理控件（ScriptManger）、脚本管理控件（ScriptMangerProxy）、时间控件（Timer）、更新区域控件（UpdatePanel）和更新进度控件（UpdateProgress）。使用 AJAX 控件能够同服务器控件一起使用从而实现服务器控件的无刷新。ASP.NET 3.5 为 AJAX 应用程序开发提供了原生环境，开发人员

使用 Visual Studio 2008 和默认的服务器控件就能够轻松的实现 AJAX 效果。

1.2 .NET 应用程序需框架

无论是 ASP.NET 应用程序还是 ASP.NET 应用程序中所提供的控件，甚至是 ASP.NET 支持的原生的 AJAX 应用程序都不能离开 .NET 应用程序框架的支持。.NET 应用程序框架作为 ASP.NET 以及其应用程序的基础而存在，若需要使用 ASP.NET 应用程序则必须使用 .NET 应用程序框架。

1.2.1 什么是 .NET 应用程序框架

.NET 框架是一个多语言组件开发和执行环境，无论开发人员使用的是 C# 作为编程语言还是使用 VB.NET 作为其开发语言都能够基于 .NET 应用程序框架而运行。.NET 应用程序框架主要包括三个部分，这三个部分分别为公共语言运行时、统一的编程类和活动服务器页面。

1. 公共语言运行时

公共语言运行时在组件的开发及运行过程中扮演着非常重要的角色。在经历了传统的面向过程开发，开发人员寻找更多的高效的方法进行应用程序开发，这其中的发展成为了面向对象的应用程序开发，在面向对象程序开发的过程中，衍生了组件开发。

在组件运行过程中，运行时负责管理内存分配、启动或删除线程和进程、实施安全性策略、同时满足当前组件对其它组件的需求。在多层开发和组件开发应用中，运行时负责管理组件与组件之间的功能的需求。

2. 统一的编程类

.NET 框架为开发人员提供了一个统一、面向对象、层次化、可扩展的类库集（API）。现今，C++ 开发人员使用的是 Microsoft 基类库，Java 开发人员使用的是 Windows 基类库，而 Visual Basic 用户使用的又是 Visual Basic API 集，在应用程序开发中，很难将应用程序进行平台的移植，当出现了不同版本的 Windows 时，就会造成移植困难。

注意：虽然 Windows 包括不同的版本，而这些版本的基本类库相同，但是不同版本的 Windows 同样会有不同的 API，例如 Windows 9x 系列和 Windows NT 系列。

而 .NET 框架就统一了微软当前的各种不同类型的框架，.NET 应用程序框架是一个系统级的框架，对现有的框架进行了封装，开发人员无需进行复杂的框架学习就能够轻松使用 .NET 应用程序框架进行应用程序开发。无论是使用 C# 编程语言还是 Visual Basic 编程语言都能够进行应用程序开发，不同的编程语言所调用的框架 API 都是来自 .NET 应用程序框架，所以这些应用程序之间就不存在框架差异的问题，在不同版本的 Windows 中也能够方便移植。

注意：.NET 框架能够安装到各个版本的 Windows 中，当有多个版本的 Windows 时，只需安装了 .NET 框架，任何 .NET 应用程序就能够在不同的 Windows 中运行而不需要额外的移植。

3. 活动服务器页面

.NET 框架还为 Web 开发人员提供了基础保障，ASP.NET 是使用 .NET 应用程序框架提供的编程类库构建而成的，它提供了 Web 应用程序模型，该模型由一组控件和一个基本结构组成，使用该模型让 ASP.NET Web 开发变得非常的容易。开发人员可以将特定的功能封装到控件中，然后通过控件的拖动进行应用程序的开发，这样不仅提高了应用程序开发的简便性，还极大的精简了应用程序代码，让代码

具更有复用性。

.NET 应用程序框架不仅能够安装到多个版本的 Windows 中，还能够安装其他智能设备中，这些设备包括智能手机、GPS 导航以及其他家用电器中。.NET 框架提供了精简版的应用程序框架，使用 .NET 应用程序框架能够开发容易移植到手机、导航器以及家用电器中的应用程序。Visual Studio 2008 还提供了智能电话应用程序开发的控件，实现了多应用、单平台的特点。

开发人员在使用 Visual Studio 2008 和 .NET 应用程序框架进行应用程序开发时，会发现无论是在原理上还是在控件的使用上，很多都是相通的，这样极大的简化了开发人员的学习过程，无论是 Windows 应用程序、Web 应用程序还是手机应用程序，都能够使用 .NET 框架进行开发。

1.2.2 公共语言运行时（CLR）

在前面的小结中可以看出，无论开发人员使用何种编程语言（如 C# 或 Visual Basic）都能够使用 .NET 应用程序框架进行应用程序的开发。那么何种原因使得开发人员使用任何 .NET 应用程序框架的支持的语言都能够使用 .NET 应用程序框架并实现相应的应用程序功能，这就要了解 .NET 中的公共语言运行库（CLR）。

公共语言运行时（Common Language Runtime，CLR）为托管代码提供各种服务，如跨语言集成、代码访问安全性、对象生存期管理、调试和分析支持。CLR 和 Java 虚拟机一样也是一个运行时环境，它负责资源管理（内存分配和垃圾收集），并保证应用和底层操作系统之间必要的分离。同时，为了提高 .NET 平台的可靠性，以及为了达到面向事务的电子商务应用所要求的稳定性和安全性级别，CLR 还要负责其他一些任务。

在公共语言运行时中运行的程序被称为托管程序。顾名思义，托管程序就是被公共语言运行时所托管的应用程序，公共语言运行时监视应用程序的运行并在一定程度上监视应用程序的运行。当开发人员进行应用程序开发和运行时，例如出现了数组越界等错误都会被公共语言运行库所监控和捕获。

当开发人员进行应用程序的编写时，编写完成的应用程序将会被翻译成一种中间语言，中间语言在公共语言运行时中被监控并被解释成为计算机语言，解释后的计算机语言能够被计算机所理解并执行相应的程序操作。在程序开发中，使用的编程语言如果在 CLR 监控下就被称为托管语言，而语言的执行不需要 CLR 的监控就不是托管语言，被称为非托管语言。在托管语言在解释时的效率没有非托管语言迅速，因为托管的语言首先需要被解释成计算机语言，这也造成了性能问题。

虽然如此，但是 CLR 所带来的性能问题越来越不足以成为问题，因为随着计算机硬件的发展，当代计算机已经能够适应和解决托管程序所带来的效率问题。

1.2.3 .NET Framework 类库

.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 类库包含了 .NET 应用程序开发中所需要的类和方法，开发人员可以使用 .NET Framework 类库提供的类和方法进行应用程序的开发。

.NET Framework 类库中的类和方法将 Windows 底层的 API 进行封装和重新设计，开发人员能够使用 .NET Framework 类库提供的类和方法方便的进行 Windows 应用程序开发，.NET Framework 还意图实现一个通用的编程环境。.NET Framework 想要实现的功能如下所示。

- ❑ 提供一个一致的面向对象的编程环境，无论这个代码是在本地执行还是在远程执行。
- ❑ 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以便于应用程序的部署和升级。

-
- ❑ 提供一个可提高代码执行安全性的代码执行环境，就算软件是来自第三方不可信任的开发商也能够提供可信赖的开发环境。
 - ❑ 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，.NET Framework 将应用程序甚至是 Web 应用相关类编译成 DLL 文件。
 - ❑ 使开发人员的经验在面对类型大不相同的应用程序时保持应用程序和数据的一致性，特别是使用面向服务开发和敏捷开发。
 - ❑ 提供一个可以确保基于 .NET Framework 的代码可与任何其他代码开发、集成、移植的可靠环境。

.NET Framework 类库用于实现基于 .NET Framework 的应用程序所需要的功能，例如实现音乐的播放和多线程开发等技术都可以使用 .NET Framework 现有的类库进行开发。.NET Framework 类库相比 MFC 具有较好的命名方法，开发人员能够轻易阅读和使用 .NET Framework 类库提供的类和方法。

无论是基于何种平台或设备的应用程序都可以使用 .NET Framework 类库提供的类和方法。无论是基于 Windows 的应用程序和基于 Web 的 ASP.NET 应用程序还是移动应用程序，都可以使用现有的 .NET Framework 中的类和方法进行开发。在开发过程中，.NET Framework 类库中对不同的设备和平台提供类和方法基本相同，开发人员不需要进行重复学习就能够进行不同设备的应用程序的开发。

1.3 安装 Visual Studio 2008

使用 .NET 框架进行应用程序开发的最好的工具莫过于 Visual Studio 2008，Visual Studio 系列产品被认为是世界上最好的开发环境之一。使用 Visual Studio 2008 能够快速构建 ASP.NET 应用程序并为 ASP.NET 应用程序提供所需要的类库、控件和智能提示等支持，本节会介绍如何安装 Visual Studio 2008 并介绍 Visual Studio 2008 中的窗口的使用 and 操作方法。

1.3.1 安装 Visual Studio 2008

在安装 Visual Studio 2008 之前，首先确保 IE 浏览器版本在 6.0 或更高，同时，可安装 Visual Studio 2008 开发环境的计算机配置要求如下所示。

- ❑ 支持的操作系统：Windows Server 2003; Windows Vista; Windows XP。
- ❑ 最低配置：1.6 GHz CPU，384 MB 内存，1024x768 显示分辨率，5400 RPM 硬盘。
- ❑ 建议配置：2.2 GHz 或更快的 CPU，1024 MB 或更大的内存，1280x1024 显示分辨率，7200 RPM 或更快的硬盘。
- ❑ 在 Windows Vista 上运行的配置要求：2.4 GHz CPU，768 MB 内存。

Visual Studio 2008 在硬件方面对计算机的配置要求如下所示。

- ❑ CPU：600MHz Pentium 处理器或 AMD 处理器或更高配置的 CPU。
- ❑ 内存：至少需要 128m 内存，推荐 256m 或更高。
- ❑ 硬盘：要求至少有 5G 空间进行应用程序的安装，推荐 10G 或更高。
- ❑ 显示器：推荐使用 800*600 分辨率或更高。

当开发计算机满足以上条件后就能够安装 Visual Studio 2008，安装 Visual Studio 2008 的过程非常简单。

(1) 单击 Visual Studio 2008 的光盘或 MSDN 版的 Visual Studio 2008 (90 天试用版) 中的 setup.exe 安装程序进入安装程序，如图 1-3 所示。

(2)进入 Visual Studio 2008 界面后,用户可以选择进行 Visual Studio 2008 的安装,单击【安装 Visual Studio 2008】按钮进行 Visual Studio 2008 的安装,如图 1-4 所示。



图 1-3 Visual Studio 2008 安装界面



图 1-4 加载安装组件

在进行 Visual Studio 2008 的安装前, Visual Studio 2008 安装程序首先会加载安装组件,这些组件为 Visual Studio 2008 的顺利安装提供了基础保障,安装程序在完成组件的加载前用户不能够进行安装步骤的选择。

(3)在安装组件加载完毕后,用户可以单击【下一步】按钮进行 Visual Studio 2008 的安装,用户将进行 Visual Studio 2008 的安装路径的选择,如图 1-5 所示。

当用户选择安装路径后就能够进行 Visual Studio 2008 的安装。用户在选择路径前,可以选择相应的安装功能,用户可以选择“默认值”、“完全”和“自定义”。选择“默认值”将会安装 Visual Studio 2008 提供的默认组件,选择“完全”将安装 Visual Studio 2008 的所有组件,而如果用户只需要安装几个组件,可以选择自定义进行组件的选择安装。

(4)选择后,单击【安装】按钮就能够进行 Visual Studio 2008 的安装,如图 1-6 所示。

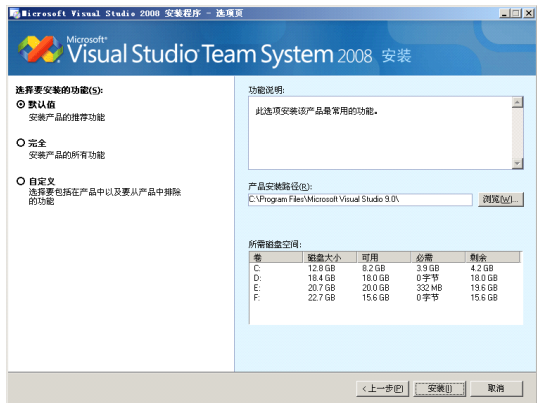


图 1-5 选择 Visual Studio 2008 安装路径

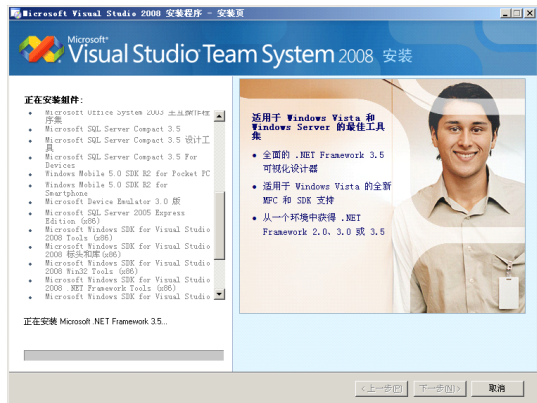


图 1-6 Visual Studio 2008 的安装

等待图 1-6 中的安装界面中左侧的安装列表的进度,当安装完毕后就会出现安装成功界面,说明已经在本地计算机中成功的安装了 Visual Studio 2008。

1.3.2 主窗口

在安装完成 Visual Studio 2008 后就能够进行 .NET 应用程序的开发，Visual Studio 2008 极大的提高了开发人员对 .NET 应用程序的开发效率，为了能够快速的进行 .NET 应用程序的开发，就需要熟悉 Visual Studio 2008 开发环境。当启动 Visual Studio 2008 后，就会呈现 Visual Studio 2008 主窗口，如图 1-7 所示。

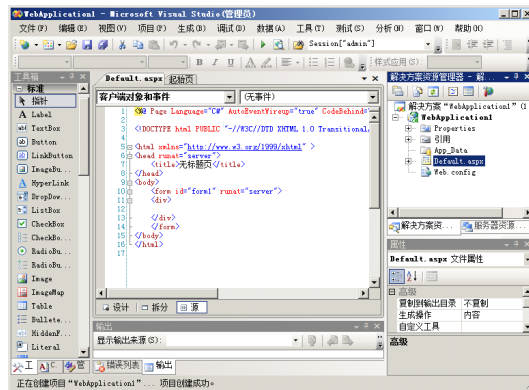


图 1-7 Visual Studio 2008 主界面

在图 1-7 中所示，Visual Studio 2008 主窗口包括其他多个窗口，最左侧的是工具箱，用于服务器控件的存放；中间是文档窗口，用于应用程序代码的编写和样式控制；中下方是错误列表窗口，用于呈现错误信息；右侧是资源管理器窗口和属性窗口，用于呈现解决方案，以及页面及控件的相应的属性。

1.3.3 文档窗口

文档窗口用于代码的编写和样式控制。当用户开发的是基于 Web 的 ASP.NET 应用程序时，文档窗口是以 Web 的形式呈现给用户，而代码视图则是以 HTML 代码的形式呈现给用户的，而如果用户开发的是基于 Windows 的应用程序，则文档窗口将会呈现应用程序的窗口或代码，如图 1-8、1-9 所示。

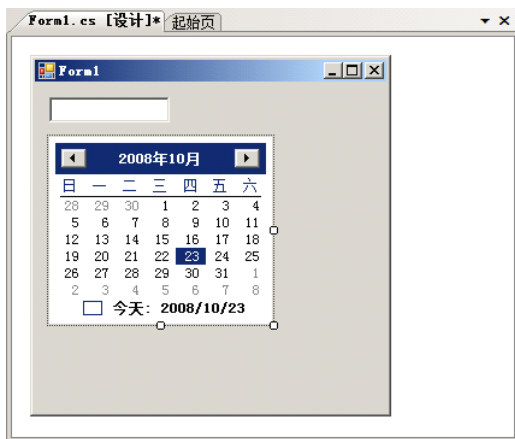


图 1-8 Windows 程序开发文档窗口

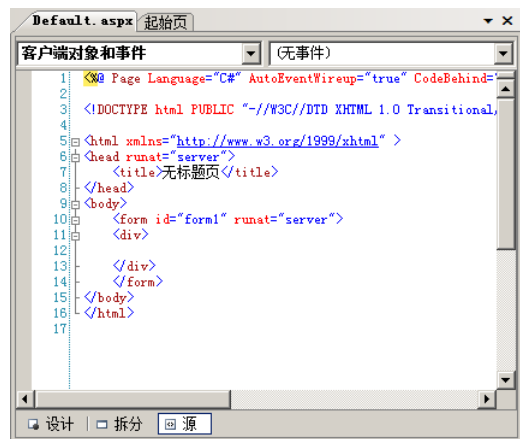


图 1-9 Web 程序开发文档窗口

当开发人员进行不同的应用程序开发时，文档窗口也会呈现为不同的样式以便开发人员进行应用程序开发。在 ASP.NET 应用程序中，其文档窗口包括三个部分，如图 1-10 所示。

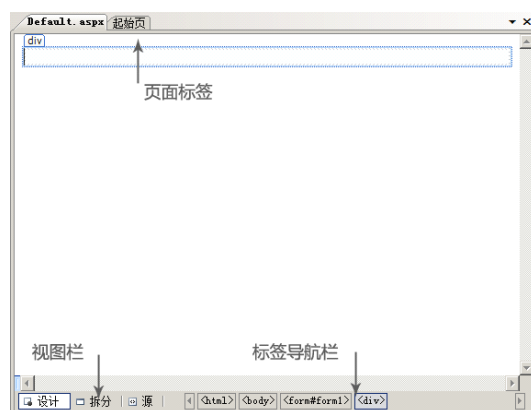


图 1-10 文档主窗口

正如图 1-10 所示，主文档窗口包括三个部分，开发人员可以通过使用这三个部分进行高效开发，这三个部分的功能如下所示。

- ❑ 页面标签：当进行多个页面进行开发时，会呈现多个页面标签，当开发人员需要进行不同页面的交替时可以通过页面标签进行页面替换。
- ❑ 视图栏：用户可以通过视图栏进行视图的切换，Visual Studio 2008 提供“设计”，“拆分”和“源代码”三种视图，开发人员可以选择不同的视图进行页面样式控制和代码的开发。
- ❑ 标签导航栏：标签导航栏能够进行不同的标签的选择，当用户需要选择页面代码中的<body>标签时，可以通过标签导航栏进行标签或标签内内容的选择。

开发人员可以灵活运用主文档窗口进行高效的应用程序开发，相比 Visual Studio 2005 而言，Visual Studio 2008 的视图栏窗口提供了拆分窗口，拆分窗口允许开发人员一边进行页面样式开发和代码编写。

注意：虽然 Visual Studio 2008 为开发人员提供了拆分窗口，但是只有在编写 Web 应用中文档主窗口才能够呈现拆分窗口。

1.3.4 工具箱

Visual Studio 2008 主窗口的左侧为开发人员提供了工具箱，工具箱中包含了 Visual Studio 2008 对.NET 应用程序所支持的控件。对于不同的应用程序开发而言，在工具箱中所呈现的工具也不同。工具箱是 Visual Studio 2008 中的基本窗口，开发人员可以使用工具箱中的控件进行应用程序开发，如图 1-11 和图 1-12 所示。

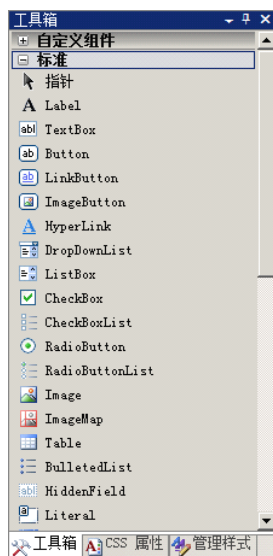


图 1-11 工具箱



图 1-12 选择类别

正如图 1-11 中所示，系统默认为开发人员提供了数十种服务器控件用于系统的开发，用户也可以添加工具箱选项卡进行自定义组件的存放。Visual Studio 2008 为开发人员提供了不同类别的服务器控件，这些控件被归为不同的类别，开发人员可以按照需求进行相应类别的控件的使用。开发人员还能够在工具箱中添加现有的控件。右击工具箱空白区域，在下拉菜单中选择【选择项】选项，系统会弹出窗口用于开发人员对自定义控件的添加，如图 1-13 所示。

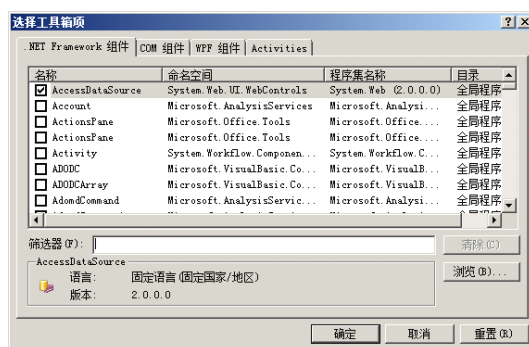


图 1-13 添加自定义组件

组件添加完毕后就能够在工具箱中显式，开发人员能够将自定义组件拖放在主窗口中进行应用程序中相应的功能的开发而无需通过复杂编程实现。

注意：开发人员能够在互联网上下载其他人已经开发好的自定义组件进行.NET 应用程序开发，这样就无需通过编程实现重复的功能。

1.3.5 解决方案管理器

在 Visual Studio 2008 的开发中，为了能够方便开发人员进行应用程序开发，在 Visual Studio 2008 主窗口的右侧会呈现一个解决方案管理器。开发人员能够在解决方案管理器中进行相应的文件的选择，双击后相应文件的代码就会呈现在主窗口，开发人员还能够单击解决方案管理器下方的服务器资源管理器窗口进行服务器资源的管理，服务器资源管理器还允许开发人员在 Visual Studio 2008 中进行表的创建

和修改。如图 1-14、1-15 所示。

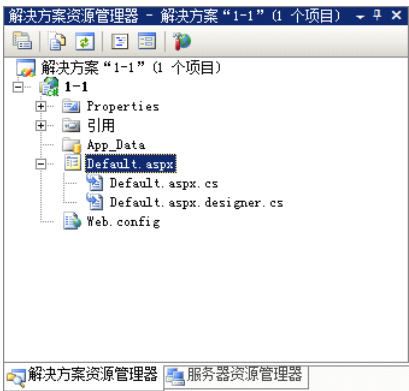


图 1-14 解决方案管理器

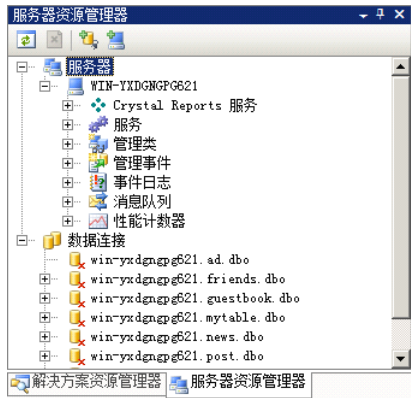


图 1-15 服务器资源管理器

解决方案管理器就是对解决方案进行管理，解决方案可以想象成是一个软件开发的整体方案，这个方案包括程序的管理、类库的管理和组件的管理。开发人员可以在解决方案管理器中双击文件进行相应的文件的编码工作，在解决方案管理器中也能够进行项目的添加和删除等操作，如图 1-16 所示。

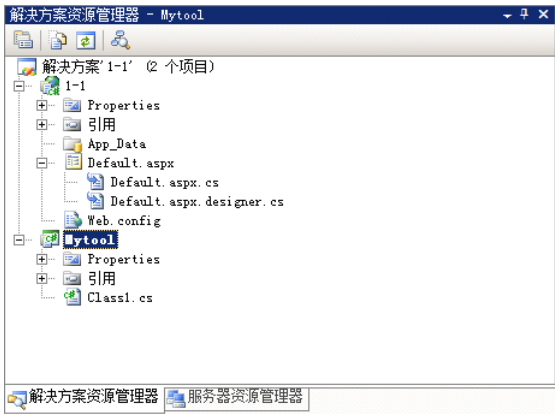


图 1-16 解决方案管理器

在应用程序开发中，通常需要进行不同的组件的开发，例如我开发用户界面，而我的一个同事进行后台开发，在开发中，如果将不同的模块分开开发或打开多个 Visual Studio 2008 进行开发是非常不方便。解决方案管理器就能够解决这个问题。将一个项目看成是一个“解决方案”，不同的项目之间都在一个解决方案中进行互相的协调和相互的调用。

注意：Visual Studio 2008 可能在默认情况下不会呈现解决方案管理器中的“解决方案‘1-1’这个标题”，开发人员可以在“工具”菜单栏的“选项”中的项目和解决方案中选择“总是显式解决方案”，如果没有项目和解决方案，则需要点击“显式所有设置”。

1.3.6 属性窗口

Visual Studio 2008 提供了非常多的控件，开发人员能够使用 Visual Studio 2008 提供的控件进行应用程序的开发。每个服务器控件都有自己的属性，通过配置不同的服务器控件的属性可以实现复杂的功能。服务器控件属性如图 1-17、1-18 所示。

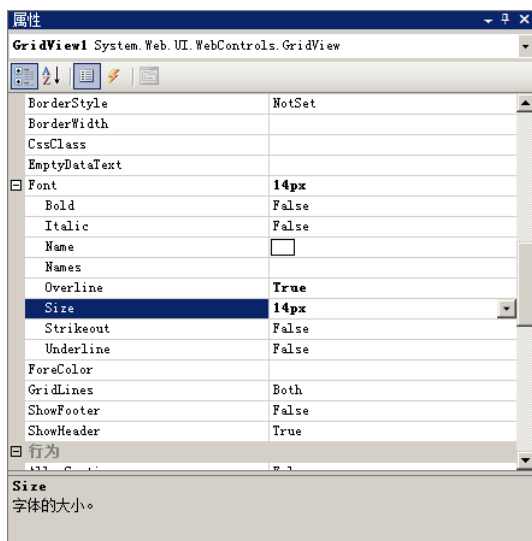


图 1-17 控件的样式属性

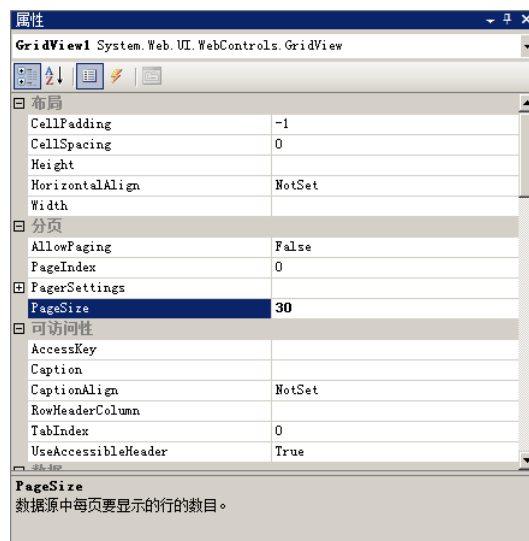


图 1-18 控件的数据属性

控件的属性配置中，可以为控件进行样式属性的配置，包括配置字体的大小、字体的颜色、字体的粗细、CSS 类等相关的控件所需要使用的样式属性，有些控件还需要进行数据属性的配置。这里使用了 GridView 控件进行数据呈现并将 PageSize 属性（分页属性）设置为 30，则如果数据条目数大于 30 则该控件会自动按照 30 条目进行分页，免除了复杂的分页编程。

1.3.7 错误列表窗口

在应用程序的开发中，通常会遇到错误，这些错误会在错误列表窗口中呈现，开发人员可以单击相应的错误进行错误的跳转。如果应用程序中出现编程错误或异常，系统会在错误列表窗口呈现，如图 1-19 所示。

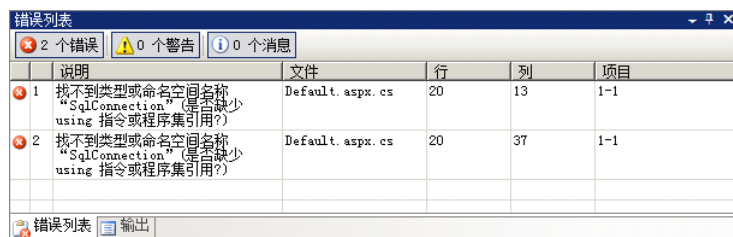


图 1-19 错误列表窗口

相对于传统的 ASP 应用程序编程而言，ASP 应用程序出现错误并不能良好的将异常反馈给开发人员。这在一方面是由于开发环境的原因，因为 Dreamware 等开发环境并不能原生的支持 ASP 应用程序的开发，另一方面也是由于 ASP 本身是解释型编程语言而无法进行良好的异常反馈。

对于 ASP.NET 应用程序而言，在应用程序运行前 Visual Studio 2008 会编译现有的应用程序并进行程序中错误的判断。如果 ASP.NET 应用程序出现错误，则 Visual Studio 2008 不会让应用程序运行起来，只有修正了所有的错误后才能够运行。

注意：Visual Studio 2008 的错误处理并不能将应用程序中的逻辑错误检测出来，例如 1 除以 0 的错误是不会被检测出来，错误处理通常情况下处理的是语法错误而不是逻辑错误。

在错误列表窗口中包含错误、警告和消息选项卡，这些选项卡中的错误的安全级别不尽相同。对于

错误选项卡中的错误信息，通常是语法上的错误，如果存在语法上的错误则不允许应用程序的运行，而对于警告和消息选项卡中信息安全级别较低，只是作为警告而存在，通常情况下不会危害应用程序的运行和使用。警告选项卡如图 1-20 所示。

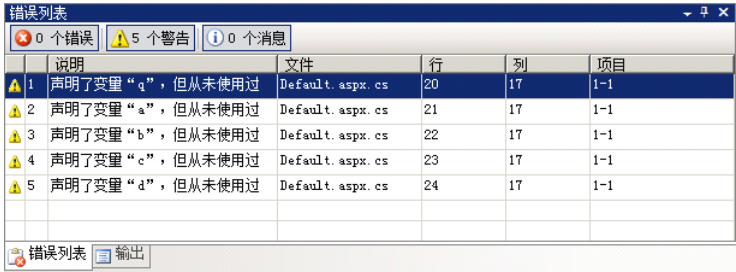


图 1-20 警告选项卡

在应用程序中如果出现了变量未使用或者在页面布局中出现了布局错误，都可能会在警告选项卡中出现警告信息。双击相应的警告信息会跳转到应用程序中相应的位置，方便开发人员对于错误的检查。

注意：虽然警告信息不会造成应用程序运行错误，但是可能存在潜在的风险，推荐开发人员修正所有的错误和警告中出现的错误信息。

1.4 安装 SQL Server 2005

Visual Studio 2008 和 SQL Server 2005 都是微软为开发人员提供的开发工具和数据库工具，所以微软将 Visual Studio 2008 和 SQL Server 2005 紧密的集成在一起，使用微软的 SQL Server 进行 .NET 应用程序数据开发能够提高 .NET 应用程序的数据存储效率。

(1) 打开 SQL Server 2005 安装盘，单击 SPLASH.HTA 文件进行安装，安装界面如图 1-21 所示。

(2) 进入 SQL Server 2005 安装界面后就能够选择相应的平台选择，开发人员可以为相应的开发平台选择安装环境，如图 1-22 所示。



图 1-21 SQL Server 2005 安装界面



图 1-22 选择安装平台

(3) 开发人员可以选择相应的平台进行安装，现在大部分的操作系统都是基于 X86 平台进行应用，而 X64 平台虽然少，但是却有长足的发展前景。选择相应的开发平台后就能够进行进入安装选择界面，如图 1-23 所示。

在安装选择界面中开发人员可以进行安装准备，安装准备包括检查硬件和软件要求、阅读发行说明

和安装 SQL Server 升级说明。在安装准备界面中的准备选项中开发人员可以检查自己所在的系统能否进行 SQL Server 2005 的安装，以及安装 SQL Server 2005 所需要遵守的协议。

(4) 在安装选择界面中需要选择【安装】连接可以进行 SQL Server 2005 应用程序的安装，可以选择【服务器组件、工具、联机丛书和示例】连接进行 SQL Server 2005 组件和应用程序的安装。单击【服务器组件、工具、联机丛书和示例】连接后如图 1-24 所示。



图 1-23 安装选择界面

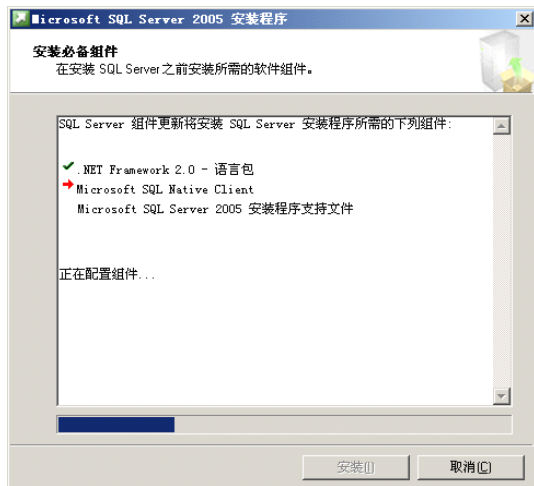


图 1-24 所示 检查安装组件

(5) 在安装 SQL Server 2005 之前首先需要安装 SQL Server 2005 所必备的组件，这些组件包括 .NET Framework 2.0 语言包，以及相应 SQL Server 2005 客户端组件，安装完成后就能够正式进入安装步骤，如图 1-25 所示。

SQL Server 2005 会进行应用程序的检查，检查包括系统的最低配置、IIS 功能要求、挂起的重新启动要求、ASP.NET 版本注册要求等等，这些要求系统会自行检查，如果 SQL Server 2005 安装程序提示安装成功则能够进行 SQL Server 2005 进一步的安装。

(6) 单击【下一步】按钮进行系统组件的安装，如图 1-26 所示。



图 1-25 系统配置检查



图 1-26 选择安装组件

(7) 选择相应的组件后单击【下一步】按钮就可以进行实例的选择，对于普通用户而言可以选择【默认实例】复选框进行 SQL Server 2005 的安装，如图 1-27 所示。

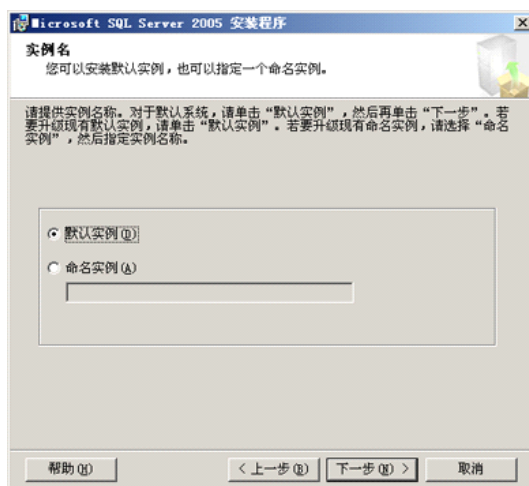


图 1-27 选择实例名称

(8) 在选择了【默认实例】复选框后就需要进行服务账户的配置, 如果用户需要使用域用户账户可以选择【使用域用户账户】选项进行域配置, 否则可以选择使用内置用户账户进行 SQL Server 2005 的安装并进行密码配置, 如图 1-28 和图 1-29 所示。



图 1-28 选择服务账户

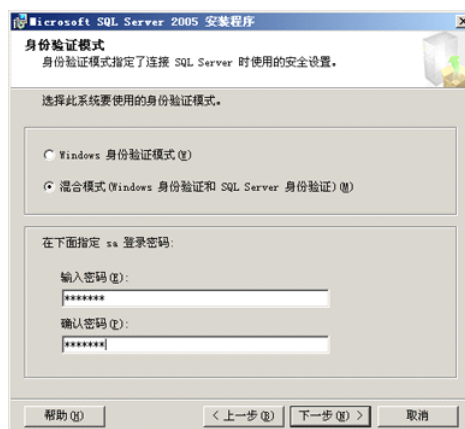


图 1-29 身份验证模式

(9) 单击【下一步】按钮进行身份验证模式选择, 开发人员可以选择“Windows 身份验证模式”和“混合模式”, 为了数据库服务器的安全, 推荐使用“混合模式”进行身份验证。

注意: 在有些操作系统上, 例如 Windows Server 2003 和 Windows Server 2008 操作系统, 可能需要强密码进行 SQL Server 2005 的安装。

(10) 在选择了身份验证模式后单击【下一步】按钮进行错误信息的配置和字符的配置, 普通用户可以直接单击【下一步】按钮进行默认配置直至安装程序安装完毕。

1.5 ASP.NET 应用程序基础

使用 Visual Studio 2008 和 SQL Server 2005 能够快速地进行应用程序的开发, 同时使用 Visual Studio 2008 和 SQL Server 2005 能够创建负载高的 ASP.NET 应用程序。通常情况下, Visual Studio 2008 负责 ASP.NET 应用程序的开发, 而 SQL Server 2005 负责应用的数据存储。

1.5.1 创建 ASP.NET 应用程序

使用 Visual Studio 2008 能够进行 ASP.NET 应用程序的开发，微软提供了数十种服务器控件能够快速地进行应用程序开发。

(1) 打开 Visual Studio 2008 应用程序后如图 1-30 所示。

(2) 打开 Visual Studio 2008 初始界面后，可以单击菜单栏上的【文件】按钮，选择【新建项目】按钮创建 ASP.NET 应用程序，如图 1-31 所示。

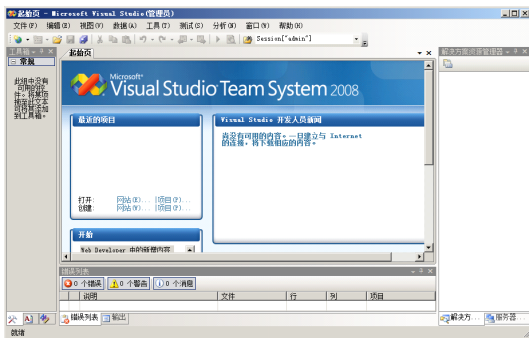


图 1-30 Visual Studio 2008 初始界面

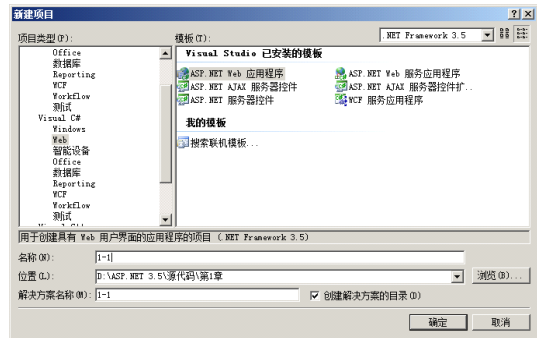


图 1-31 创建 ASP.NET Web 应用程序

(3) 选择【ASP.NET Web 应用程序】选项，单击确定就能够创建一个最基本的 ASP.NET Web 应用程序。创建完成后系统会创建 default.aspx、default.aspx.cs、default.aspx.designer.cs、以及 Web.config 等文件用于应用程序的开发。

1.5.2 运行 ASP.NET 应用程序

创建 ASP.NET 应用程序后就能够进行 ASP.NET 应用程序的开发，开发人员可以在【资源管理器】中添加相应的文件和项目进行 ASP.NET 应用程序和组件开发。Visual Studio 2008 提供了数十种服务器控件以便开发人员进行应用程序的开发。

在完成应用程序的开发后，可以运行应用程序，单击【调试】按钮或选择【启动调试】按钮就能够调试 ASP.NET 应用程序。调试应用程序的快捷键为【F5】，开发人员也可以单击【F5】进行应用程序的调试，调试前 Visual Studio 2008 会选择是否启用 Web.config 进行调试，默认选择使用即可，如图 1-32 所示。

选择“修改 Web.config 文件以启动调试”进行应用程序的运行。在 Visual Studio 2008 中包含虚拟服务器，所以开发人员可以无需安装 IIS 进行应用程序的调试。但是一旦进入调试状态，就无法在 Visual Studio 2008 中进行 cs 页面，以及类库等源代码的修改，如图 1-33 所示。

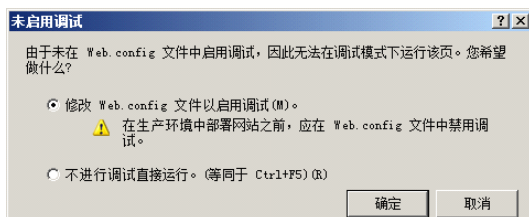


图 1-32 启用调试配置

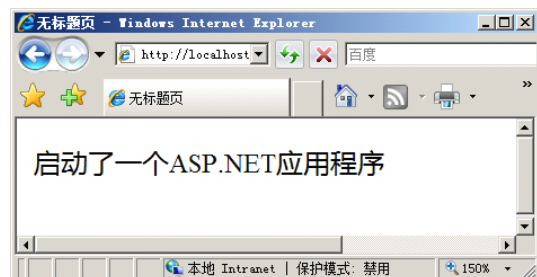


图 1-33 运行 ASP.NET 应用程序

注意：虽然 Visual Studio 2008 提供虚拟服务器，开发人员可以无需安装 IIS 进行应用程序调试，但是为了完好模拟 ASP.NET 网站应用程序，建议在发布网站前使用 IIS 进行调试。

1.5.3 编译 ASP.NET 应用程序

与传统的 ASP 应用程序开发不同的是，ASP.NET 应用程序能够将相应的代码编译成 DLL（动态链接库）文件，这样不仅能够提高 ASP.NET 应用程序的安全性，还能够提高 ASP.NET 应用程序的速度。在现有的项目中，打开相应的项目文件，其项目源代码都可以进行读取，如图 1-34 所示。

开发人员能够将源代码文件放置在服务器中进行运行，但是将源代码直接运行会产生潜在的风险，例如用户下载 Default.aspx 或其他页面进行源代码的查看，这样就有可能造成源代码的泄露和漏洞的发现，这样是非常不安全的。将 ASP.NET 应用程序代码编译成动态链接库能够提高安全性，就算非法用户下载了相应的页面也无法看到源代码。

单击项目然后右击【项目图标】，选择【发布】按钮发布 ASP.NET 应用程序，系统会弹出发布对话框用户应用程序的发布，如图 1-35 所示。

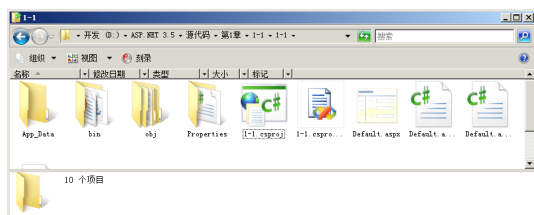


图 1-34 源代码文件



图 1-35 发布 Web

单击【发布】按钮后，Visual Studio 2008 就能够将网站编译并生成 ASP.NET 应用程序，如图 1-36 所示。编译后的 ASP.NET 应用程序没有 cs 源代码，因为编译后的文件会存放在 bin 目录下并编译成动态链接库文件，如图 1-37 所示。

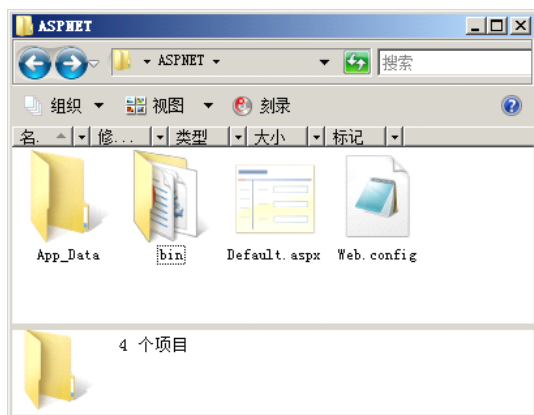


图 1-36 编译后的文件



图 1-37 动态链接库文件

正如图 1-36 所示，在项目文件夹中只包含 Default.aspx 页面而并没有包含 Default.aspx 页面的源代码 Default.aspx.cs 等文件，因为这些文件都被编译成为动态链接库文件。编译后的 ASP.NET 应用程序在

第一次应用时会有些慢，在运行后，每次对 ASP.NET 应用程序的请求都可以直接从 DLL 文件中请求，能够提高应用程序的运行速度。

1.6 小结

本章讲解了 ASP.NET 的基本概念，以及 .NET 框架的基本概念。这些概念在初学 ASP.NET 时会觉得非常的困难，但是这些概念会在今后的开发中逐渐清晰。虽然这些基本概念看上去没什么作用，但是在今后的 ASP.NET 应用开发中起着非常重要的作用，熟练掌握 ASP.NET 基本概念能够提高应用程序的适用性和健壮性。Visual Studio 2008 不仅提供了丰富的服务器控件还提供了属性、资源管理、错误列表窗口以便开发人员进行项目开发。本章还包括：

- ❑ .NET 历史与展望：包括 .NET 应用程序的过去和未来以及发展前景。
- ❑ ASP.NET 与 ASP：讲解了 ASP.NET 与 ASP 的不同之处。
- ❑ ASP.NET 开发工具：讲解了 ASP.NET 开发工具的基本知识。
- ❑ .NET 框架：讲解了 .NET 框架的基本知识。
- ❑ 公共语言运行时(CLR)：讲解了 .NET 框架的公共语言运行时。
- ❑ .NET Framework 类库：讲解了 .NET 框架的 .NET Framework 类库的基本知识。
- ❑ 安装 Visual Studio 2008：讲解了如何安装 Visual Studio 2008。
- ❑ 安装 SQL Server 2005：讲解了如何安装 SQL Server 2005。
- ❑ ASP.NET 应用程序基础：讲解了 ASP.NET 应用程序的安装，编译和运行。

本章着重讲解了 Visual Studio 2008 开发环境，以及如何安装 SQL Server 2005 以便于 ASP.NET 应用程序的数据存储。Visual Studio 2008 和 SQL Server 2005 的紧密集成能够提高 ASP.NET 应用程序的开发效率和运行效率。本章讲解了 ASP.NET 的基本知识，ASP.NET 使用的是 C# 语言进行开发的，了解 C# 编程语言是 ASP.NET 应用开发的第一步，下一章将会详细的讲解 C# 编程技术。