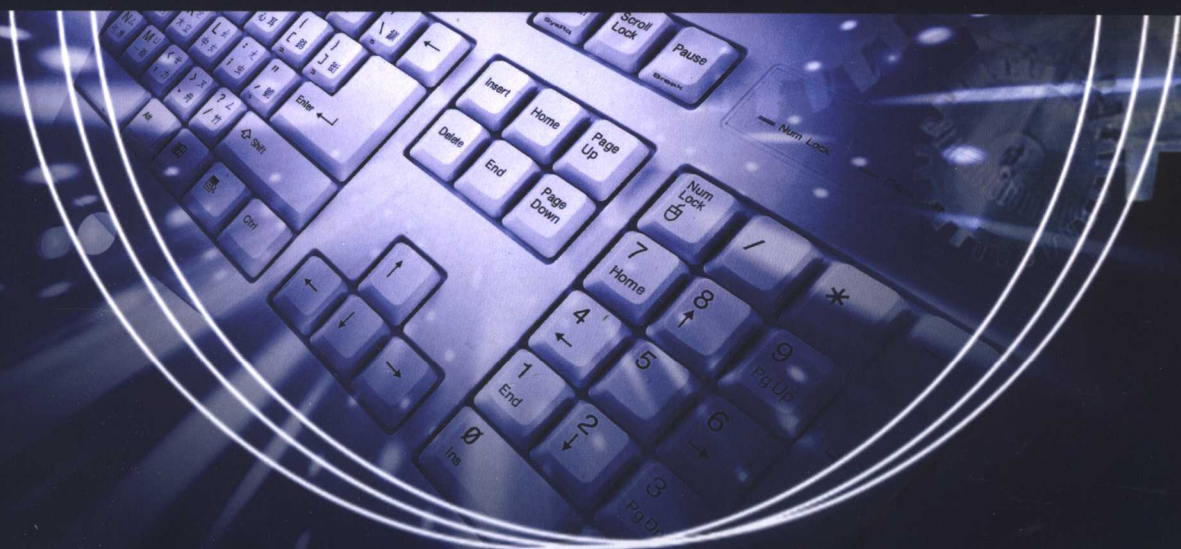




.NET企业级

汤涛等编著

应用程序开发教程



清华大学出版社

.NET 企业级应用程序开发教程

汤 涛 等 编 著

清 华 大 学 出 版 社

北 京

内 容 简 介

Visual Studio. NET 作为微软. NET 战略的核心开发工具,与 Visual Studio 6 相比,其开发效率具有显著性地提高,其开发框架和思路发生了根本性地改变。同时,Visual Studio. NET 又提供对 COM 组件的访问兼容性。。NET 全面引入对 XML 的支持是开发基于. NET 分布式应用的基石,XML. NET 支持在数据访问层面和对象传递层面上跨越 Internet 传递数据和对象。

本书基于 C# 语言深入阐述. NET 核心开发思路和. NET 应用程序运行的基本原理,详尽讲解 Windows 编程方法、网络访问和 Sockets 编程。还深入讨论了多线程应用程序开发、线程同步技术和线程池技术等. NET 深入编程技术。Web 服务和远程通信技术是. NET 下企业级三层结构应用开发的关键技术,本书讨论了 Web 服务和远程通信的基本原理和实现机制,与 COM+ 交互,. NET 企业服务技术和序列化技术,详细阐述基于. NET 的三层结构应用程序开发,这些技术都是实现. NET 企业级软件开发所必备的。企业应用部署向来是企业级应用开发的关键,本书详细阐述了. NET 应用部署的基本方法和原则。

本书结构清晰,讲解深入浅出,并通过丰富的实例加深读者对相关技术的把握。本书是从事微软 Visual C#. NET 开发和应用人员不可多得的参考书。同时适合广大计算机爱好者自学也可作为大专院校相关专业的教学参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

. NET 企业级应用程序开发教程/汤涛等编著. —北京:清华大学出版社,2005. 10
ISBN 7-302-11400-5

I. N… II. 汤… III. 计算机网络—程序设计—教材 IV. TP393

中国版本图书馆 CIP 数据核字(2005)第 079430 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:许振伍

文稿编辑:刘 霞

印 刷 者:北京市世界知识印刷厂

装 订 者:三河市金元装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:37.25 字数:922 千字

版 次:2005 年 10 月第 1 版 2005 年 10 月第 1 次印刷

书 号:ISBN 7-302-11400-5/TP·7490

印 数:1~3000

定 价:55.00 元

前 言

从 .NET 开发技术推出至今已有三年多的时间,国内的诸多软件企业特别是基于 Windows 平台的软件开发企业都在陆续向 .NET 开发上迁移。笔者作为中国 .NET 开发技术的先驱,长期从事软件项目开发、咨询和研究,特别是近几年来一直致力于中国软件企业开发架构和技术的推广与咨询工作,以及针对软件企业的高级程序员和项目经理的技术培训工作。在这期间,笔者把实际应用开发中遇到的问题、大多数程序员在 .NET 应用开发中可能遇到的陷阱以及针对一些 .NET 核心技术的理解等总结成文,这就是本书的由来。

Visual Studio .NET(简称 VS. NET)作为微软 .NET 战略的核心开发工具,与 Visual Studio 6 相比,其开发效率得到了显著提高,其开发框架和思路发生了根本性改变。同时,VS. NET 又提供对 COM 组件和 Win32 API 的访问兼容性,使得 VS. NET 成为微软平台上迄今为止最为强大的开发工具。其中,.NET 框架是 VS. NET 开发应用以及 .NET 应用程序运行的基础,它包含运行 .NET 应用程序所必需的虚拟机和基础类库。

本书深入阐述 .NET 核心开发思路和 .NET 应用程序运行的基本原理,详尽讲解 Windows 编程方法、网络访问和 Sockets 编程。本书还讨论了多线程应用程序开发、线程同步技术和线程池技术等 .NET 编程技术。Web 服务和远程通信技术是 .NET 下企业级三层结构应用开发的关键技术,本书深入讨论 Web 服务和远程通信的基本原理和实现机制,详细阐述基于 .NET 的三层结构应用程序开发,还讨论了 .NET 与 COM+ 的交互技术以及 .NET 企业服务技术,这些技术都是实现 .NET 企业级软件开发所必备的。企业应用部署一直是企业级应用开发的关键,本书详细阐述了 .NET 应用部署的基本方法和原则。

本书各章内容

第 1 章 .NET 概述,主要讨论 .NET 战略概要、.NET 框架组成部分及其本特征等内容,还讨论到了程序集(assembly)的概念。

第 2 章 基于 .NET 的 OOP,主要讨论了面向对象的基本概念。介绍了如何在 .NET 中定义类的基本结构,其中包含属性、方法、构造函数等,还讨论到了面向对象编程中的继承和多态如何在 .NET 下实现,也讨论了抽象类和接口的实现方法。

第 3 章 .NET 组件开发和部署,主要讨论了 .NET 中组件的开发和内存管理,以及应用程序的部署和 DLL 程序集的版本控制等技术。应用的命名空间主要为 System. ComponentModel。

第 4 章 Windows 窗体编程,主要讨论了建立多窗体应用、使用控件和窗体菜单、如何实现拖放操作,以及访问系统剪贴板等 Windows 编程的基本技术。应用的命名空间主要为 System. Windows. Forms。

第 5 章 数据处理技术,主要讨论了 ADO. NET 对象模型、在 .NET 基础类库中有关 ADO. NET 的命名空间和类,以及如何通过这些类来实现数据的增、删、改、查等基本操作,如何调用存储过程,实现单机事务处理和分布式事务处理等,还讨论了如何针对数据库存取

大对象的数据。最后,讨论了数据访问时的安全性问题。应用的命名空间主要为 System.Data 及其子命名空间。

第 6 章 数据流和文件处理,主要讨论与文件操作和访问相关的类、读写文本文件和二进制文件、对文件的异步读写操作、控制文件和目录的不同事件,以及操作独立存储文件等。应用的命名空间主要为 System.IO 及其子命名空间。

第 7 章 Internet 访问,主要讨论如何创建基于 TCP/IP 网络访问 Web 站点的客户端、如何异步访问网络资源、处理网络访问过程中遇到的错误、实现 DNS 域名解析,最后讨论了如何创建 TCP 服务器和客户端。应用的命名空间主要为 System.NET 及其子命名空间。

第 8 章 绘图和打印,介绍了 GDI+ 与 GDI 的区别,讨论了如何创建基本图形文件和复杂的图像、实现不同图像格式的转换,以及控制打印输出。应用的命名空间主要为 System.Drawing 及其子命名空间。

第 9 章 .NET 与非受管代码的交互,主要讨论了与 COM 组件的相互调用、如何调用 Win32 API、如何通过 WMI 访问操作系统,或是通过 .NET 框架中的一些诊断类来访问操作系统。应用的命名空间主要为 System.Runtime.InteropServices、System.Management 和 System.Diagnostics 及其子命名空间。

第 10 章 .NET 事件处理,主要讨论了委托(delegate)的基本概念、在 .NET 中事件委托的机制,以及在用户自定组件中如何产生并响应事件。应用的命名空间主要为 System 命名空间。

第 11 章 多线程和 Windows 服务,主要讨论了如何创建多线程应用程序、如何创建 Windows 后台服务、如何安装和卸载 Windows 服务以及在 Windows 服务中使用多线程来提高响应效率。应用的命名空间主要为 System.Threading 和 System.ServiceProcess。

第 12 章 ASP.NET 编程,主要讨论了 ASP.NET 与经典 ASP 的区别、使用 Web 控件,以及一些基本的 Web 编程技术,还讨论了如何开发 Web 控件和用户自定控件等内容。应用的命名空间主要为 System.Web 及其子命名空间。

第 13 章 ASP.NET 数据展现和报表技术,主要讨论了如何在 ASP.NET 页面中使用数据控件绑定来自于数据库的数据。如何使用 VS.NET 带的水晶报表和用户自定义报表。应用的命名空间主要为 System.Web 及其子命名空间。

第 14 章 ASP.NET 配置,主要讨论了 ASP.NET 的配置架构实现有效的错误处理和安全性控制等。应用的命名空间主要为 System.Web 及其子命名空间。

第 15 章 Web 服务和 Remoting,主要讨论并比较了 Web 服务和 Remoting 技术,介绍了如何开发 Web 服务、如何开发 Remoting 服务器端和客户端,以及如何实现远程组件的调用。应用的命名空间主要为 System.Web.Services、System.Runtime.Remoting 及其子命名空间。

第 16 章 XML.NET,主要讨论了 XML 基本概念和 XML 在 .NET 中的具体实现,以及如何操作 XML 数据等。应用的命名空间主要为 System.Xml 及其子命名空间。

本书使用的约定

- 技巧:用于更为有效地解决问题,或更快捷地实现编码的提示信息。
- 注意:通常是背景材料或是相关链接,用于加深读者对相应内容的理解。

- 小心：可能会遇到的程序代码问题、技术陷阱或是一不小心就会产生的错误。

本书面向的读者

本书是针对向 .NET 技术迁移的程序员编写的，帮助程序员更快地熟悉和掌握 .NET 编程技术。当然，对于已经开始使用 .NET 的程序员也会有帮助。

客户支持信息

我们尽可能地让本书在技术上更为准确，但错误是在所难免的，希望读者能告诉我们有本书的错误之处，我们的 E-mail 是：tonyacton@hotmail.com。

当然，如果读者在实际开发中遇到了问题，也可以给我们发 E-mail，我们将会提供相应的技术支持和咨询。

作 者

目 录

第 1 章 .NET 概述	1
1.1 .NET 简介	1
1.2 .NET 框架介绍	2
1.3 CLR 介绍	3
1.4 .NET 框架基本特征	4
1.5 程序集	6
1.6 小结	8
第 2 章 基于.NET 的面向对象编程技术	9
2.1 面向对象基本概念	9
2.2 属性和方法	11
2.3 继承和多态	17
2.4 接口和抽象类	21
2.5 方法的重载	29
2.6 运算符的重载	31
2.7 小结	38
第 3 章 组件开发和部署	40
3.1 组件开发概述	40
3.2 CLR 内存管理	41
3.3 组件内存管理	45
3.4 .NET 应用程序部署	51
3.5 程序集组件版本控制	57
3.6 DLL 程序集的迟签名技术	62
3.7 小结	64
第 4 章 Windows 窗体编程	65
4.1 如何建立和使用 Windows 窗体	65
4.1.1 创建简单的 Windows 窗体	66
4.1.2 设置窗体的属性	68
4.1.3 编写窗体事件处理代码	69
4.2 处理多窗体应用程序	74
4.3 使用控件	78
4.3.1 控件的格式处理	78

4.3.2 控件使用方法	79
4.4 使用窗体菜单	83
4.5 拖放操作的实现	87
4.6 剪贴板的访问	89
4.7 在窗体之间传递数据	90
4.8 小结	95
第 5 章 数据展现	96
5.1 ADO.NET 对象模型	96
5.2 ADO.NET 相关的类	98
5.3 实现数据的增删改查	100
5.3.1 建立数据库连接	100
5.3.2 检索数据	102
5.3.3 插入、修改和删除数据	105
5.4 使用存储过程	108
5.4.1 使用简单存储过程	108
5.4.2 使用带输入参数的存储过程	109
5.4.3 使用带输出参数的存储过程	110
5.5 存取大对象数据	112
5.6 事务处理	120
5.6.1 事务的基本特征	120
5.6.2 实现普通事务操作	121
5.6.3 分布式事务处理	127
5.6.4 并发控制和死锁问题	129
5.7 数据访问安全性控制	131
5.8 小结	132
第 6 章 数据流和文件处理	134
6.1 与流相关的类	134
6.2 文件操作相关的类	135
6.3 对文件系统的基本操作	136
6.3.1 创建文件和目录	136
6.3.2 删除和修改文件或目录	137
6.3.3 列举目录下的内容	137
6.4 读写文件	139
6.4.1 文本文件的读写	140
6.4.2 读写二进制文件	143
6.5.3 写字板应用程序	144
6.5 异步文件读写操作	150

6.6	响应文件和目录的更改事件	155
6.7	独立存储	159
6.8	小结	163
第 7 章	Internet 访问技术	164
7.1	Internet 访问概述	164
7.2	创建 Web 客户端	165
7.2.1	创建 Web 请求	165
7.2.2	发送请求数据	167
7.2.3	接收数据	171
7.2.4	使用 WebClient 类	175
7.2.5	异步访问网络资源	175
7.2.6	网络访问的错误处理	180
7.3	使用特定协议的 Web 客户端	182
7.4	Dns 类实现域名解析	189
7.5	建立 TCP 服务器和客户端	194
7.6	小结	204
第 8 章	绘图和打印	205
8.1	GDI+ 介绍	205
8.2	创建并显示图片文件	207
8.3	绘制基本图形	217
8.4	使用文本和字体绘制图片	226
8.5	绘制复杂图像	228
8.6	打印文本和图片	232
8.7	小结	242
第 9 章	.NET 与非受管代码的交互	244
9.1	COM 简介	244
9.2	与 COM 交互	245
9.2.1	COM 访问 .NET	246
9.2.2	.NET 访问 COM	249
9.3	调用 Win32 API	257
9.4	通过 WMI 访问操作系统	260
9.5	通过 .NET 的类访问操作系统	265
9.5.1	访问操作系统的进程信息和日志	266
9.5.2	访问系统注册表	271
9.5.3	访问活动目录	273
9.6	小结	276

第 10 章 .NET 事件处理	278
10.1 Windows 事件基础	278
10.2 委托	279
10.3 事件处理	281
10.4 小结	286
第 11 章 多线程和 Windows 服务	287
11.1 多线程	287
11.1.1 创建新线程	288
11.1.2 控制线程的运行状态	289
11.1.3 访问主线程	290
11.1.4 线程同步	291
11.2 创建 Window 服务	292
11.2.1 获取 Windows 服务的信息	293
11.2.2 控制 Windows 服务	296
11.2.3 编写 Windows 服务	299
11.3 安装和卸载 Windows 服务	302
11.4 调试 Windows 服务	306
11.5 小结	307
第 12 章 ASP.NET 编程	308
12.1 ASP.NET 与经典 ASP 的区别	308
12.1.1 语言支持	309
12.1.2 性能	309
12.1.3 代码和内容的分离	309
12.1.4 安全性	309
12.1.5 状态管理	309
12.1.6 服务器端表单和控件	310
12.1.7 开发调试	310
12.1.8 配置和部署	310
12.2 ASP.NET 基本技术	310
12.2.1 页面控件	316
12.2.2 Web 控件的使用	319
12.2.3 文件上载	326
12.2.4 发送 E-mail	329
12.2.5 动态加载控件	332
12.3 Web 用户控件	333
12.3.1 定义 Web 用户控件	333

12.3.2 使用 Web 用户控件	335
12.4 开发 Web 服务器控件	345
12.5 小结	351
第 13 章 ASP.NET 数据展现和报表技术	353
13.1 使用 Web 控件展现数据	353
13.1.1 使用 Repeater 控件	354
13.1.2 使用 DataList 控件	360
13.2 使用 DataGrid 控件实现数据更新	377
13.2.1 DataGrid 控件实现数据展现	377
13.2.2 DataGrid 控件实现数据更新	386
13.2.3 DataGrid 控件实现数据排序	393
13.3 服务器端缓存技术	395
13.3.1 用户界面缓存	397
13.3.2 数据缓存	407
13.4 水晶报表技术	412
13.4.1 拉模型实现数据展现	412
13.4.2 推模型实现数据展现	417
13.5 用户自定义报表	421
13.6 小结	429
第 14 章 ASP.NET 配置	431
14.1 配置概述	431
14.2 安全性配置	433
14.2.1 基于表单的身份验证	433
14.2.2 基于 Windows 账号的身份验证	442
14.3 客户端浏览器识别	445
14.4 编译选项配置	448
14.5 自定义错误配置	452
14.5.1 页面级用户自定义错误处理	453
14.5.2 应用程序级用户自定义错误处理	454
14.6 区域化信息配置	455
14.7 ASP.NET 进程配置	458
14.8 Session 状态管理配置	460
14.8.1 Cookie 非依赖的 Session	462
14.8.2 Session 数据存储区位置选择	463
14.9 跟踪配置	466
14.10 页面属性	470
14.11 小结	471

第 15 章 Web 服务和远程处理	473
15.1 Web 服务概要	473
15.2 创建 Web 服务	475
15.3 访问 Web 服务	480
15.4 Web 服务的异步调用	488
15.5 Remoting(远程处理)概要	496
15.6 实现远程处理服务端	498
15.7 部署远程组件	505
15.7.1 通过 IIS 部署远程处理组件	505
15.7.2 Windows 后台服务部署远程处理组件	506
15.8 访问远程组件	507
15.9 Web 服务与远程处理比较	517
15.10 小结	518
第 16 章 XML.NET	520
16.1 XML 基础	520
16.1.1 处理指令	521
16.1.2 元素	521
16.1.3 文档类型声明	522
16.1.4 命名空间	522
16.1.5 注释	523
16.1.6 CDATA 片断	523
16.2 XML 基本技术	524
16.3 XML.NET 介绍	532
16.4 使用 XML 架构(schema)定义数据结构	533
16.5 XSL 实现数据结构转换	538
16.6 读写 XML 数据	544
16.6.1 XmlReader 类	544
16.6.2 XmlDocument 类	545
16.6.3 XmlWriter 类	548
16.7 XML 与 ADO.NET 交互	550
16.8 小结	552
附录 A SMTP 和 POP3 协议(节选)	553
A.1 SMTP 协议(根据 RFC822 节选)	553
A.1.1 SMTP 命令	553
A.1.2 SMTP 规范	557
A.2 POP3 协议(根据 RFC1725 节选)	562

A. 2. 1 基本操作	562
A. 2. 2 授权状态	563
A. 2. 3 TRANSACTION 状态	565
A. 2. 4 UPDATE 状态	567
A. 2. 5 可选的 POP3 命令	568
A. 2. 6 POP3 会话的例子	570
A. 2. 6 POP3 会话的例子	571
 附录 B .NET 编码规则	 572
附录 C .NET 相关的资源	579

第 1 章 .NET 概述

本章将学习到如下内容:

- .NET 简介
- .NET 框架介绍
- CLR 介绍
- .NET 框架基本特征
- 程序集(Assembly)

.NET 无论是作为一种技术发展方向,还是一系列产品,都已经逐步渗透到软件行业的各个层面。只要是基于微软(Microsoft)的平台从事软件开发,无论是在开发路线选型还是在具体的技术实施上,都可能需要去了解和把握.NET 技术。

微软正在全球范围内以 .NET 新概念慢慢地改变着基于微软平台的软件开发思路,进而影响人们对于信息时代的思考方式。虽然这种改变和影响在目前看来似乎还有很长的路要走,但是毕竟.NET 已经逐步被广大厂商和程序员所接受和采用。

1.1 .NET 简介

微软的 .NET 计划是 2000 年微软公司在其 PDC(Professional Developer's Conference)大会上首次向外界推出的一个非常庞大的战略发展计划。在这个计划中,.NET 的目标是通过基于 Web 服务提供的分布式服务平台,能够在任何时候任何地点,把所有信息、设备、系统和人有机地连接起来,让人们可以随时随地进行通信和信息交互,而不用考虑是使用何种操作系统、设备或编程语言。Web 服务就是这些服务集成的核心,即.NET 的核心。作为第三代 Internet 平台,.NET 是一项非常庞大而宏伟的计划,其策略就是把 Internet 本身作为构建新一代操作系统和应用的基础,把 Internet 和操作系统进一步延伸,使得开发人员和用户可以摆脱设备硬件的束缚而轻松实现信息的连接和沟通。

.NET 作为一个服务平台,其主要组成部分从技术上可以划分为三层(见图 1.1)。首先是服务器层,主要是为 Web 服务提供支持的操作系统以及相关的微软 .NET 企业级服务器。支持 .NET 的操作系统目前几乎包括微软的大部分服务器操作系统,包括 Windows 2000/XP 以及最新发布的 Windows 2003 Server。而 .NET 企业级服务器则几乎包含微软提供的所有其他应用服务器产品,如 SQL Server 2000、Exchange Server 2000 和 BizTalk Server 等。这些服务器都被称为 .NET Enterprise Server。其含义是都可以用于为 Web 服务提供支持,或者是这些 .NET 企业级服务器将其组件或服务以 Web 服务的方式发布出去。

其次是基于服务端服务器支持所提供的 Web 服务。通过 Web 服务集成一系列不同类型的服务器产品提供的服务,以便以统一的格式为客户端提供信息和数据。这个统一的格

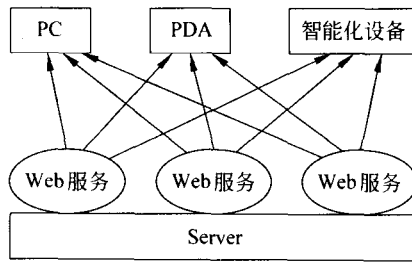


图 1.1 .NET 体系结构

式就是工业标准 XML。 .NET 平台提供的 Web 服务同样也是一个工业标准,因此它能在不同平台、不同开发语言下实现无缝地交互操作。通俗地说,就是可以将 Web 服务看成是跨 Internet、跨语言、跨平台的 API。这将从根本上改变我们设计、开发和使用计算设备的方式。以前我们看到更多的是基于局域网的企业级三层结构应用,或者是基于 Web 站点的、基于 HTML 的客户端与服务器的通信。现在,在采纳了 Web 服务之后,我们就将越来越多地看到松散耦合模型下的分布式计算的应用。但是,Web 服务的介入并不是将软件应用开发完全更新或替换,而是把一些旧系统集成到 .NET 体系中。这样的协作和互操作性将使我们的软件应用实现平滑的自然演变。

再次,是使用这些 Web 服务的客户端,包括任何桌面系统、笔记本电脑、移动电话、PDA 或其他智能化设备。这些客户端可以随时随地访问 Web 服务。这样用户就可以完全摆脱传统网络的限制,自由冲浪于多维的互联网中。比如,我们可以在网上存储个人健康资料的详细信息,同时控制对这些信息的访问权限,在全球范围内的任何医院只要具有智能化设备,医生就可以在你的授权下获取你的健康资料。

总之,微软的 .NET 作为第三代 Internet 平台,其计算模型正以不同的方式影响着企业、开发人员和用户,它扩展了设备、系统和用户之间的数据和信息的通信能力。

1.2 .NET 框架介绍

对于 .NET 架构来说,也许可以看成是 Internet 发展的又一个里程碑。在 .NET 应用程序体系结构中,微软为所有 .NET 应用程序的运行提供了一个全新的运行环境和基础类库,这就是所谓的 .NET 框架(.NET Framework,见图 1.2)。这个全新的运行环境即 CLR (Common Language Runtime,通用语言运行时)。所有 .NET 应用程序,包括 .NET 核心—— Web 服务都必须基于 .NET 框架才能正常运行,同时为了给第三方厂商和一些软件企业开发 Web 服务提供更方便、高效的开发工具,微软还推出了新一代开发工具 Visual Studio .NET。

Visual Studio .NET(有时简称为 VS .NET)并不是 Visual Studio 6 的简单升级。可以说 Visual Studio .NET 改变了开发应用程序的思路和方式。我们可以使用这些工具快速而轻松、高效地创建应用程序和 Web 服务。此外,还可以通过 Visual Studio .NET 将软件产品和服务转化为订购式服务,即软件租用,通过 Internet 来实现软件的销售和配送。

运用 Visual Studio .NET,大体上可以实现两大类 .NET 应用开发,一是经典的应用程

序,如 MIS、ERP 和 CRM 等企业级应用软件;二是新一代的分布式应用,包括集成 Web 服务的可租用软件。

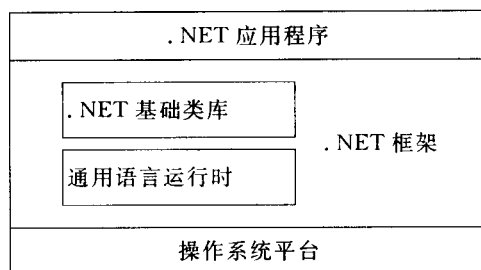


图 1.2 .NET 框架

Visual Studio .NET 基本自动实现了把应用程序转换为 Web 服务的过程。这样在开发一些分布式应用时就极大地减少了创建和部署方案的代价。将来的软件开发企业就可能集中提供 Web 服务,而最终用户就可以通过一些客户端随时随地去消费这些 Web 服务。

虽然目前大多数企业还是使用 Visual Studio .NET 开发一些经典的应用,比如 ERP 和 CRM 等。但是,很多软件开发企业正在或试图从不同的开发平台迁移到 .NET 平台。这不仅是着眼于未来的软件租用业务,而且是因为 .NET 框架提供了大量使用方便且功能强大的基础类库,使得开发应用的效率比 Visual Studio 6 或者是其他的一些开发平台的效率还要高。这样就给程序员提出了更高的目标,即要求程序员们全面掌握 .NET 开发思路 and 开发工具,如此才能充分运用 Visual Studio .NET 所提供的强大功能。

前面我们已经提及 .NET 框架中包含的两个最主要的组成部分:CLR(Common Language Runtime,通用语言运行时)和 .NET 基础类库(Base Class Library)。我们将在 1.3 节讨论 CLR,而基础类库中包含的类则是开发人员最需要关注的,因为这就是把握 .NET 应用程序开发的关键。这些类库将在本书大量讨论到。

1.3 CLR 介绍

简单地讲,CLR 就是 .NET 应用程序的虚拟机,它为所有的 .NET 应用程序提供运行时的支持。它提供的支持包括代码管理、线程支持和异常管理等。这是由 CLR 结构决定的。CLR 的结构如图 1.3 所示。

我们先来看一下 CLR 最底层的结构“类加载器”(Class Loader)。其作用就是加载 .NET 应用程序的代码。值得注意的是,.NET 应用程序的代码并不像传统的 COM 程序那样直接可以通过 Visual Studio .NET IDE(Integrated Development Environment,集成开发环境)调用语言编译器生成二进制代码,而是基于 .NET 微软推出中间代码(Microsoft Intermediate Language,缩写为 MSIL 或 IL),所有的项目经过编译后生成的都是 IL。包含这些 IL 的文件统称为程序集(assembly),而不是二进制代码。这些 IL 代码并不能直接运行,而是需要在运行时通过 CLR 中的二次编译器(即图 1.3 中的 MSIL to Native Compilers)将这些 IL 编译生成本机代码,然后再执行。其中类加载器负责把这些 IL 代码加载到内存中。

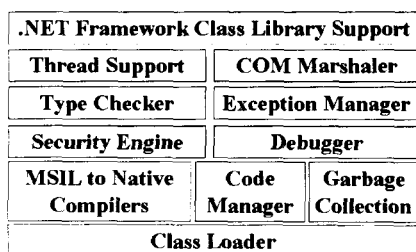


图 1.3 CLR 结构图

这个类加载器是一个智能化加载器,它能实现即用即加载。

CLR 中还包含异常管理器(Exception Manauager),用于在运行时处理 .NET 应用程序的错误或异常。类型检查器(Type Checker)则是检查不同对象类型之间的转换,以及数组的下标等。垃圾收集器(Garbage Collection)用于回收孤立对象资源。CLR 的这些结构都是为了提供更好的应用程序运行环境。

1.4 .NET 框架基本特征

.NET 框架具有非常强大的功能,也具有一系列显著特征。这些特征如下。

1. 统一编程模型

大家一定还记得在 Visual Studio 6 开发环境中,Visual C++ 6 具有一套 MFC 和 ATL 类库,对于开发人员来说,要使用 Visual C++ 6 开发程序就必须掌握这一套类库;而对于 Visual Basic 6 的开发人员而言,则需要掌握 Visual Basic 6 的一些函数库;同样对于 ASP 开发人员来说,了解一些服务器的内置对象(诸如 Response 对象)是必不可少的。总之,在 Visual Studio 6 开发中,不同的开发语言之间具有不同的类库,要使用这些不同的开发语言就必须首先掌握这些不同的类库。假设一个 ASP 开发人员需要转到 Visual C++ 或 Visual Basic 6 开发环境,那么他除了学习不同的语言语法外,还需要学习这些语言各自的类库。对于 .NET 来说,这一切都得到了根本性改变,因为所有语言都使用同样的编程基础类库,即 .NET 框架中包含的基础类库。这样就使得基于 .NET 的所有编程模型得到了统一。因此,我们将来要从一种 .NET 开发语言转到另一种 .NET 开发语言时,就不再需要像以前那样再次去学习那些不同语言对应的类库,只需要了解和掌握这些不同语言的语法就可以了。正是由于 .NET 框架统一了编程模型,消除了 .NET 语言之间在基础类库上的差异,因此,人们常把 .NET 框架称为语言中立的平台。

2. 多样的开发语言

基于 .NET 基础类库,Visual Studio .NET 提供了 Visual Basic .NET、Visual C++ .NET、Visual C# .NET、JScript .NET 和 J# .NET 五种开发语言。在这些不同的开发语言中,具有各自的一些特色,但其开发框架都是一致的,这就是 .NET 框架。

在这些语言中,Visual C# .NET 是一种新型的开发语言,该语言综合了 Visual Basic 6