

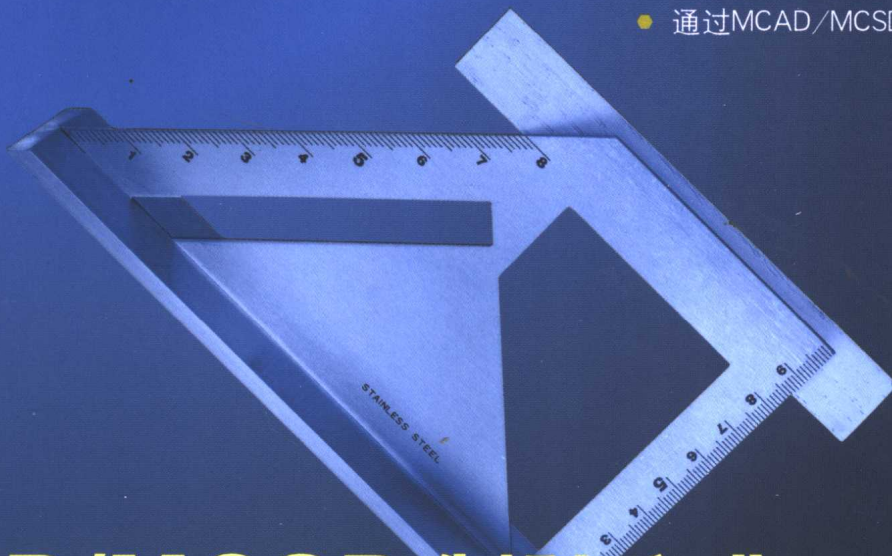
微软指定MCAD/MCSD教材

Microsoft Press

从入门到精通自学用书

考试70-310和70-320

- 单本全球销量超过200万册
- 应用程序开发人员最具价值的手册
- 通过MCAD/MCSD考试的可靠保障



MCAD/MCSD 制胜宝典

用Visual Basic .NET 和Visual C# .NET

开发XML Web 服务与服务器组件

[美] 微软公司 著
杨志娟 李朋朋 马云艳 张莉 等译



清华大学出版社

内 容 简 介

本书是 MCAD/MCSD 制胜宝典系列丛书之一,系统地介绍了 .NET 框架、创建和管理 Windows 服务、创建和使用服务组件、ADO.NET 数据库编程、XML Web 服务编程等内容。通过本书的学习,您不但可以学会如何使用 Visual Basic .NET 和 Visual C# .NET 开发 XML Web 服务与服务器组件,还能为参加 MCP 考试 70-310 和 70-320 做好准备。

本书非常适合准备从事 XML Web 服务开发的编程人员阅读和学习,更是准备 MCP 考试 70-310 和 70-320 时的必备辅导材料。

MCAD/MCSD 制胜宝典——用 Visual Basic .NET 和 Visual C# .NET 开发 XML Web 服务与服务器组件
MCAD/MCSD Self-Paced Training Kit: Developing XML Web Services and Server Components with Microsoft Visual Basic .NET and Microsoft Visual C# .NET (ISBN 0-7356-1586-1)
Microsoft Corporation

Copyright © 2003 by Microsoft Corporation.

Original English Language Edition Copyright © 2003 by Microsoft Corporation.
Published by arrangement with the original publisher, Microsoft Press,
a division of Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文版由 Microsoft Press 授权清华大学出版社出版。

北京市版权局著作权合同登记号 图字 01-2002-0245 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

MCAD/MCSD 制胜宝典——用 Visual Basic .NET 和 Visual C# .NET 开发 XML Web 服务与服务器组件/微软公司著;杨志娟等译.—北京:清华大学出版社,2003

书名原文:MCAD/MCSD Self-Paced Training Kit: Developing XML Web Services and Server Components with Microsoft Visual Basic .NET and Microsoft Visual C# .NET

(微软指定 MCAD/MCSD 教材)

ISBN 7-302-06822-4

I.M… II.①微…②杨… III.①电子计算机—工程技术人员—资格考核—自学参考资料②BASIC 语言—程序设计—工程技术人员—资格考核—自学参考资料③C 语言—程序设计—工程技术人员—资格考核—自学参考资料 IV.TP3

中国版本图书馆 CIP 数据核字(2003)第 050533 号

出版者:清华大学出版社

地 址:北京清华大学学研大厦

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

邮 编:100084

社总机:010-62770175

客户服务:010-62776969

译者:杨志娟 李朋朋 马云艳 张莉等

文稿编辑:赵飞虎 冯涛

封面设计:陈刘源

印刷者:北京市四季青印刷厂

发行者:新华书店总店北京发行所

开本:850×1168 1/16 印张:31 插页:1 字数:831千字

版次:2003年8月第1版 2003年8月第1次印刷

书号:ISBN 7-302-06822-4/TP·5069

印数:1~4000

定 价:52.00元

前 言

欢迎使用《MCAD/MCSD 制胜宝典——用 Visual Basic .NET 和 Visual C# .NET 开发 XML Web 服务与服务器组件》(考试 70-310 和 70-320)。`.NET` 是微软的一种新技术,它提供了新的应用程序框架,如 Web 服务、ASP.NET、Windows 窗体以及 ADO.NET 等。这项革命性的组件技术简化了组件的开发和部署。通过学习本书的课程及练习,您将能够开发 XML Web 服务和服务器组件。

本书还讲述了微软认证专家考试(Microsoft Certified Professional Exam)70-310 和 70-320 的考查目标。



注意 关于 MCAD(微软认证应用程序开发人员, Microsoft Certified Application Developer)以及 MCSD(微软认证解决方案开发人员, Microsoft Certified Solution Developer)的详细信息,请参考附录 D 部分的内容。

前言中“课前准备”这部分内容提供了重要的安装说明,介绍了完成本书学习的硬件和软件要求。还提供了安装和访问下载文件中内容的相关信息。在开始学习之前应仔细阅读这部分内容。

本书读者对象

本书读者对象包括需要设计、规划、实现及支持 XML Web 服务和服务器组件的软件开发人员,以及想要通过以下微软认证专家考试(Microsoft Certified Professional)的相关人员。

- 使用 Visual Basic .NET 和 .NET 框架开发 XML Web 服务以及服务器组件(MCP 考试 70-310)
- 使用 Visual C# .NET 和 .NET 框架开发 XML Web 服务以及服务器组件(MCP 考试 70-320)

先决条件

学习本书需要满足如下先决条件:

- 能够利用 Visual Studio .NET, 使用 Visual Basic .NET 或是 Visual C# .NET 编程语言创建 Windows 应用程序。
- 对于面向对象的编程概念有着基本的理解, 包括类、属性、方法以及事件。
- 熟悉基于 Windows、基于 Web 以及基于控制台的应用程序的开发及部署。
- 对 COM 编程有基本了解。
- 了解 Web 内容在 Internet 上的存储及访问方式。这也就要求读者能够解释 Web 服务器、Internet 协议以及 Web 客户端(如浏览器等)的作用。
- 熟悉 XML 以及相关技术, 如 XPath 和 XSD 等。

参考文献

下列参考材料对您的学习可能会有所帮助:

- Visual Studio .NET 在线帮助
- 微软开发人员网络(<http://msdn.microsoft.com>)
- XML 和 Web Service 的 W3C 标准(<http://www.w3c.org>)
- XML Web Service 开发者中心(<http://msdn.microsoft.com/webservices>)

本书特色

本书各章都分成一些小部分,可帮您从中获取更多教益。

- 每章都以“先决条件”部分开始,这部分为您提供学习本章前的先决条件。
- 各章还分为若干节。每节都包含了关于特定技能的参考及操作步骤等信息。
- “本章小结”总结了各章的主要内容。
- “实验”为读者提供了专业的练习,以便于您巩固各章节中所讲述的技能。其中的练习是按步骤进行的,这样就使得读者可以利用相关的技能,或者浏览相应章节中所介绍的应用程序的相应部分。

附录 A 中包含了各章的“本章复习”中的所有问题以及相应答案。

特殊段落

本书中包含以下几种特殊段落:



提示 对可能的结果或执行某个任务的其他方法进行说明。



要点 其中包含着对完成某个任务至关重要的信息。



注意 包含补充信息。



警告 说明了可能出现的数据损失。

体例约定

本书中使用了下列体例:

- 粗体 为输入的内容。
- 斜体 语法中的斜体表示网址以及占位符。

- **大写字母** 对于文件或文件夹的名称来说,除直接输入外,都会以大写字母的形式出现。在这里除非有特别说明,否则在对话框或命令提示符中输入文件名时,均可以使用小写形式。
- **文件扩展名** 当扩展名不与文件名在一起而单独出现时,是以小写形式表示的。
- **缩写** 缩写全部以大写字母表示。
- **等宽字体** 书中的示例代码、屏幕文本实例或者需要在命令提示符或初始化文件中输入的项。
- **方括号[]** 在语法中,括起可选项。例如,命令语法中的[filename]表示您可以选择在输入命令时同时输入文件名。在输入时,只需输入方括号中的内容即可,不输入方括号本身。
- **大括号{ }** 在语法中,括起必须的项。输入时,只需输入括号中的信息,不输入大括号本身。

各章综述

本书包含注释、实践型练习以及各章问题解答,从而能够帮助读者学习如何使用 Visual Studio .NET 开发 XML Web 服务及服务器组件。一般来说,您最好从头至尾、循序渐进地学习本书内容。但是您也可以选择适合自己的学习方式,只学习感兴趣的内容(相关信息,参见下部分内容“寻找您的最佳起点”)。如果您选择了自己的学习方式,那么应该仔细阅读各章开始的“先决条件”部分。有些操作步骤需要前面章节的预备工作,那么就需要仔细参阅相关章节。

本书分为以下几大部分及章节:

- **“前言”** 本部分介绍了本书概况,还介绍了本书的各个部分。为充分利用本书,读者应仔细阅读这部分内容,并且制定相应的学习计划。
- **第1章“.NET 框架”** 介绍了.NET 框架的基础知识,这是开发高效分布式 Internet 应用程序的基础。这一章讨论了公共语言运行库在.NET 框架中的作用,托管程序执行过程和非托管程序执行过程之间的差别,还有程序集在应用程序的部署和配置中所起的作用。另外,本章还描述了如何使用.NET 框架所提供的工具来创建、部署以及管理应用程序。
- **第2章“创建和管理 Windows 服务”** 解释了 Windows 服务编程模型,介绍了在 Visual Studio .NET 中创建 Windows 服务的相关内容,以及控制 Windows 服务的机制。通过本章的学习,您将了解到如何在默认事件日志中添加条目、创建自定义事件日志,以及如何在自定义事件日志中记录信息。还讲解了如何动态地配置服务,如何手动地通过编程来管理服务。
- **第3章“创建和使用服务组件”** 介绍了用于访问 COM+服务(如自动事务管理、对象池以及实时激活等)的服务组件。通过本章的学习,您将了解 COM+编程模型的基础知识,如何创建和注册服务组件,如何充分利用 COM+服务以及使用 Component Services 工具管理服务组件。
- **第4章“创建和使用.NET 远程处理对象”** 介绍了在无需了解一个分布式应用程序开发中所包含的协议或编码及解码机制的条件下,如何使用.NET 框架来建立对象之间的通信。通过本章的学习,您将了解如何使用.NET 框架所提供的类来创建、配置及保护.NET 远程处

理对象。

- 第 5 章“ADO.NET 数据库编程” 本章讲解了如何使用 .NET 框架的 ADO(ActiveX 数据对象, ActiveX Data Object)。在这一章中, 您将学会如何使用 .NET 数据提供程序, 以及怎样创建和使用 ADO.NET 的 DataSet。另外, 还将学习如何使用 DataReader 和 DataAdapter 对象来管理和控制数据。
- 第 6 章“访问和操作 XML 数据” XML 作为 .NET 框架中的应用程序之间进行数据交换的核心技术, 具有十分重要的地位, 本章就介绍了 XML 的重要性。本章讨论了 XML DOM(文档对象模型, Document Object Model), 介绍了如何在应用程序中访问 XML 数据。您将学会如何从 XML 文档中读写数据, 对 XML 文档执行查询, 以及如何使用 XML 架构验证 XML 文档。另外, 还介绍了如何使用 XML 文件中的数据填充 DataSet, 如何将 DataSet 中的数据写入 XML 文件。
- 第 7 章“创建与使用 XML Web 服务” 介绍了如何建立可缩放、松耦合的、不依赖平台的应用程序。XML Web 服务使得完全不同的应用程序可通过标准协议——如 HTTP、XML、XSD、SOAP 以及 WSDL(Web 服务描述语言, Web Services Description Language)——互换消息。
- 第 8 章“XML Web 服务高级编程” 介绍了如何使用属性来控制 Web 方法的特征。另外, 还介绍了如何创建和使用 SOAP 扩展、如何创建异步 Web 方法等。
- 第 9 章“测试和调试 XML Web 服务” 介绍了如何调试 XML Web 服务和服务组件。本章还介绍了如何使用跟踪技术, 以及如何进行代码检测。
- 第 10 章“部署 XML Web 服务和 Windows 服务” 说明了 .NET 框架所提供的部署选项。通过本章的学习, 您将学会创建能安装 .NET 应用程序的安装程序。另外, 还将介绍实现版本管理和并行部署的相关内容。
- 附录 A “问题与答案” 列出了本书各章的问题、各问题所在位置及参考答案。
- 附录 B “COM 互操作性” 提供了关于如何在托管应用程序中调用非托管 COM 对象。您将学会从 COM 类库中导出 COM 类型, 并创建互用的程序集。
- 附录 C “考试重点索引” 列出了重点测试技能在本书中的位置。
- 附录 D “微软认证专家计划” 包括了有关微软认证专家计划的详细信息。
- 术语表 列出了书中术语和概念的定义。

寻找您的最佳起点

由于本书为自学辅导书, 所以您可以跳过一些章节, 以后再去阅读。请参考下表找到您的最佳起点。

如果您……	学习顺序
准备参加 MCP 考试 70-310 或 70-320	阅读“课前准备”部分内容。然后以任意顺序详细学习书中全部章节
欲查看关于该考试的特定主题的信息	利用附录 C 中所介绍的内容

课 前 准 备

本书为自学书，书中包含实践型练习，能帮助您掌握 XML Web 服务及服务器组件的开发。



警告 书中的部分实验可能要求您对服务器做出某些更改，如果这台服务器连接着一个大型网络，那么可能会导致一些令人不快的结果。在尝试这些实验时，最好与网络管理员进行协商。

硬件要求

每台计算机都必须符合如下最低配置。所有的硬件都应该兼容 Windows XP 或 Windows 2000。

- Pentium II 类处理器，450 MHZ
- 160 MB 物理内存，推荐使用 256 MB 物理内存
- 光驱或 DVD 驱动器，推荐使用 12 速以上
- 安装驱动器上至少 3.5 GB 空余空间，其中包含系统驱动器 500 MB 空间
- SVGA(800*600)或更高解析度的 256 色显示器
- 微软鼠标或兼容指针设备

软件要求

要想完成本书中的实验，那么如下软件是必须的：

- Windows 2000 或 Windows XP Professional



注意 如果您的计算机上未安装 IIS，那么可打开【控制面板】中的【添加或删除程序】，然后选择【添加/删除 Windows 组件】。从组件列表中选择【Internet 信息服务(IIS)】，然后根据屏幕指示进行操作，完成安装即可。

- Visual Studio .NET 专业版或 Visual Studio .NET 企业开发版。推荐使用专业版，但企业开发版也可以使用。
- SQL Server 2000。可从 <http://microsoft.com/sql/default.asp> 下载 SQL Server 2000 的免费试用版本。

支 持 信 息

为了降低书的成本，减轻读者负担，对于因内容很少而不值得单独配盘的图书，我们将其范例代码或练习文件放在我们的网站上，供读者下载。敬请访问以下网址：<http://www.wenyuan.com.cn>，查找本书的有关链接。

如果您对本书或配书文件有任何建议、意见或想法，请通过以下电子邮件与清华大学出版社计算机应用编辑二室客户服务部取得联系：

service@wenyuan.com.cn

或致函：

北京 100084-157 信箱

读者服务部

邮编：100084

亦可致电：010-62792098-220。

请注意，上述地址并不提供软件产品的支持。

目 录

前言	vii	1.9.2 练习 2: 使用 Ildasm 查看 MSIL 代码	28
第 1 章 .NET 框架	1	1.9.3 练习 3: 创建多文件程序集	29
1.1 .NET 框架概述	1	1.10 本章复习	31
1.1.1 了解 .NET	2	第 2 章 创建和管理 Windows 服务	32
1.1.2 .NET 框架及其在 .NET 中的作用	2	2.1 了解 Windows 服务	32
1.1.3 .NET 框架体系结构	4	2.1.1 Windows 服务概述	33
1.2 公共语言运行库	5	2.1.2 服务应用程序和其他 Visual Studio .NET 应用程序的区别	34
1.2.1 公共语言运行库的体系结构	5	2.1.3 Windows 服务应用程序的 编程模型	34
1.2.2 公共语言运行库组件	6	2.2 创建 Windows 服务	36
1.3 托管执行过程	7	2.2.1 创建 Windows 服务应用程序	36
1.3.1 托管执行过程的定义	7	2.2.2 更改 Windows 服务应用程序 的默认属性	38
1.3.2 自动内存管理	8	2.2.3 向服务应用程序添加功能	40
1.4 程序集和全局程序集高速缓存	14	2.3 在 Windows 服务应用程序中处理事件 和登录信息	44
1.4.1 程序集概述	14	2.3.1 处理 Windows 服务的事件	45
1.4.2 命名空间简介	14	2.3.2 在系统事件日志中记录信息	45
1.4.3 程序集的类型	15	2.3.3 创建自定义事件日志	47
1.4.4 程序集标识	16	2.3.4 使用性能计数器	49
1.4.5 程序集的部署	16	2.4 添加安装程序、指定安全上下文以及 安装和卸载 Windows 服务	51
1.5 配置和安全性	18	2.4.1 了解安装程序的作用	52
1.6 应用程序域和运行库宿主	20	2.4.2 添加安装程序	52
1.6.1 进程和应用程序域	20	2.4.3 指定服务应用程序的 安全上下文	57
1.6.2 运行库宿主	21	2.4.4 安装和卸载 Windows 服务	57
1.7 .NET 框架工具简介	21	2.5 管理 Windows 服务	59
1.7.1 程序集链接器	22	2.5.1 使用 SCM	59
1.7.2 使用 ilasm.exe 工具	23		
1.7.3 使用 Ildasm.exe 工具	23		
1.7.4 使用 Caspol.exe 工具	24		
1.7.5 Mscorcfg.msc	25		
1.8 本章小结	26		
1.9 实验: 创建程序集和查看 MSIL	26		
1.9.1 练习 1: 创建单文件程序集	26		

2.5.2 使用 ServiceController 类.....	61	3.5.4 编程方式实现组件安全	125
2.6 配置和调试 Windows 服务	68	3.5.5 安全调用上下文信息	125
2.6.1 配置 Windows 服务.....	68	3.5.6 COM+安全服务和.NET 框架 间的互操作性.....	127
2.6.2 调试 Windows 服务.....	70	3.6 本章小结.....	128
2.7 本章小结	72	3.7 实验: 创建、配置和管理服务组件	128
2.8 实验: 创建和管理 Windows 服务.....	72	3.7.1 练习 1: 创建服务组件	128
2.8.1 练习 1: 创建和安装 Windows 服务	72	3.7.2 练习 2: 创建客户端应用程序	135
2.8.2 练习 2: 创建客户端应用程序	77	3.7.3 练习 3: 创建 Accounts 表	140
2.8.3 练习 3: 管理 Windows 服务.....	81	3.7.4 练习 4: 使用 Component Services 工具配置服务组件.....	142
2.9 本章复习	90	3.8 本章复习.....	142
第 3 章 创建和使用服务组件	91	第 4 章 创建和使用 .NET 远程处理对象	144
3.1 COM+编程概述	91	4.1 理解 .NET 远程处理机制	144
3.1.1 COM+的演化	92	4.1.1 .NET 远程处理机制概述	145
3.1.2 COM+的组件服务	94	4.1.2 跨远程处理边界访问对象	145
3.2 了解、创建和注册服务组件.....	96	4.1.3 .NET 远程处理体系结构	146
3.2.1 开发服务组件.....	96	4.1.4 可远程处理和不可远程 处理的对象.....	147
3.2.2 注册服务组件.....	100	4.2 实现服务器激活和客户端激活对象	148
3.3 利用 COM+服务	106	4.2.1 理解远程对象激活	148
3.3.1 启动 COM+事务	106	4.2.2 使用生存期租约	153
3.3.2 启用 JIT 激活	108	4.2.3 发布范围.....	154
3.3.3 使用松耦合事件服务.....	109	4.3 使用信道跨应用程序域传输消息	155
3.3.4 使用对象池服务.....	111	4.3.1 理解信道.....	155
3.3.5 使用排队组件服务.....	112	4.3.2 为远程通信选择信道	157
3.3.6 使用对象构造.....	112	4.3.3 HTTP 信道.....	158
3.4 使用 Component Services 工具管理 服务组件	112	4.3.4 TCP 信道.....	160
3.4.1 Component Services 工具.....	113	4.3.5 接收器和接收链.....	162
3.4.2 使用 Component Services 工具 配置激活	116	4.4 事件和委托的实现.....	163
3.4.3 使用 Component Services 工具 配置事务	117	4.4.1 理解事件和委托	164
3.4.4 使用 Component Services 工具 配置安全	118	4.4.2 在远程处理应用程序中实现事件 和委托.....	165
3.5 实现服务组件的安全.....	123	4.5 实现异步方法.....	167
3.5.1 实现服务组件的安全.....	123	4.5.1 异步方法.....	168
3.5.2 基于角色的安全.....	123	4.5.2 在远程处理应用程序中实现 异步方法.....	168
3.5.3 应用程序中的组件级别 安全检查	124	4.6 配置和保护 .NET 远程处理对象	169
		4.6.1 远程对象的配置	170

4.6.2 保护.NET 远程处理对象	171	5.4 本章小结.....	210
4.7 本章小结	172	5.5 实验: 创建和使用数据访问组件	210
4.8 实验: 创建和使用.NET 远程处理对象	172	5.5.1 练习 1: 创建航空数据库	210
4.8.1 练习 1: 创建 ChatCoordinator.dll 库	173	5.5.2 练习 2: 创建数据访问组件	214
4.8.2 练习 2: 为配置 ChatCoordinator 对象创建配置文件.....	175	5.5.3 练习 3: 创建一个远程数据 访问组件.....	218
4.8.3 练习 3: 创建一个控制台程序来 激活 ChatCoordinator 对象.....	176	5.5.4 练习 4: 创建一个 Windows 服务 来控制远程数据访问组件	221
4.8.4 练习 4: 创建一个聊天客户端 程序	177	5.5.5 练习 5: 创建一个远程客户端 应用程序.....	223
4.8.5 练习 5: 创建一个配置文件来 配置 ChatClient 对象	179	5.6 本章复习.....	227
4.8.6 练习 6: 运行聊天程序.....	180	第 6 章 访问和操作 XML 数据.....	228
4.9 本章复习	181	6.1 了解 XML 文档对象模型	228
第 5 章 ADO.NET 数据库编程.....	183	6.1.1 XML 文档对象模型概述	229
5.1 理解 ADO.NET	183	6.1.2 XML 文档对象模型	229
5.1.1 ADO.NET 概述	184	6.1.3 创建 XML 语法分析程序	233
5.1.2 ADO.NET 体系结构	185	6.2 使用 XmlReader 和 XmlWriter	235
5.1.3 ADO.NET 的特性	186	6.2.1 XmlReader 概述.....	236
5.1.4 ADO 和 ADO.NET 之间 的区别	187	6.2.2 使用 XmlTextReader 读取 XML.....	236
5.2 理解.NET 数据提供程序.....	188	6.2.3 XmlWriter 概述	239
5.2.1 .NET 数据提供程序概述	189	6.2.4 使用 XmlTextWriter 写入 XML.....	239
5.2.2 OleDbConnection 类.....	189	6.3 使用 XPathNavigator.....	240
5.2.3 SqlConnection 类.....	190	6.3.1 了解 XPathNavigator 类	241
5.2.4 OleDbCommand 类.....	190	6.3.2 执行 XPath 查询.....	241
5.2.5 使用数据适配器.....	194	6.4 了解 XML Schema 对象模型	244
5.2.6 DataReader 类.....	197	6.5 确认 XML 文档.....	247
5.2.7 使用 Data Adapter Configuration Wizard 创建数据访问组件.....	198	6.5.1 了解 XML 确认	247
5.3 使用 DataSet.....	202	6.5.2 使用 XmlValidatingReader 确认 XML 文档.....	248
5.3.1 理解 ADO.NET DataSet.....	202	6.5.3 使用 XmlValidatingReader 读取 XML 片段.....	252
5.3.2 创建 ADO.NET DataSet.....	203	6.6 使用 XML 和 DataSets.....	253
5.3.3 操纵 ADO.NET DataSet.....	205	6.6.1 XML 和 DataSet 概述.....	254
5.3.4 创建数据视图.....	207	6.6.2 使用 XML 文件和 DataSet.....	254
5.3.5 处理 ADO.NET DataSet 事件.....	208	6.6.3 用 XML 数据填充 DataSet.....	257
5.3.6 创建一个类型化 ADO.NET DataSet.....	209	6.7 本章小结.....	258
		6.8 实验: 访问和操作 XML 数据	258
		6.8.1 练习 1: 创建产品目录	259

6.8.2 练习 2: 创建产品目录 Windows 应用程序	260	8.2.2 在 XML Web 服务中进行 异步编程	337
6.8.3 练习 3: 运行产品目录 应用程序	277	8.3 使用 SOAP 扩展	340
6.9 本章复习	279	8.3.1 SOAP 扩展概述	340
第 7 章 创建与使用 XML Web 服务	280	8.3.2 生成 SOAP 扩展	341
7.1 了解 XML Web 服务	280	8.4 配置和保护 Web 服务	345
7.1.1 XML Web 服务简介	281	8.4.1 Web.config 文件的结构	345
7.1.2 XML Web 服务基础结构	281	8.4.2 使用 Web.config 文件中的 元素配置 Web 服务	346
7.1.3 在客户端与 XML Web 服务 之间进行通信	283	8.4.3 为 Web 服务配置安全性	347
7.2 创建 XML Web 服务	283	8.5 本章小结	348
7.3 部署和发现 XML Web 服务	299	8.6 实验: 实现事务处理、异步方法和 SOAP 扩展	349
7.3.1 发布 XML Web Service	299	8.6.1 练习 1: 创建 LabTest 数据库和 Authors 表	349
7.3.2 Web 服务发布的项	301	8.6.2 练习 2: 实现 WebMethod 特征的 事务属性	351
7.3.3 了解 XML Web 服务发现机制	301	8.6.3 练习 3: 在 XML Web 服务中 实现 SOAP 扩展	352
7.3.4 为 Web 服务配置发现信息	302	8.6.4 练习 4: 创建 XML Web 服务 客户程序	360
7.4 使用 XML Web 服务	303	8.6.5 练习 5: 异步调用 Web 方法	364
7.4.1 使用 XML Web 服务	304	8.7 本章复习	376
7.4.2 使用 Web 服务公开的方法	305	第 9 章 测试和调试 XML Web 服务	377
7.5 本章小结	315	9.1 设计单元测试计划	377
7.6 实验: 创建和使用 XML Web 服务	316	9.1.1 测试过程概述	378
7.6.1 练习 1: 创建 XML Web 服务	316	9.1.2 组织测试计划	378
7.6.2 练习 2: 创建 Windows 客户端 应用程序	323	9.1.3 基于要求的测试	379
7.7 本章复习	328	9.1.4 制定测试计划	379
第 8 章 XML Web 服务高级编程	329	9.1.5 制定测试计划的方法	379
8.1 用特性控制 Web 方法的特征	329	9.2 Visual Studio .NET 调试工具概述	380
8.1.1 Web 方法的属性	330	9.2.1 Visual Studio .NET 调试 过程简介	380
8.1.2 缓冲 Web 方法的响应	330	9.2.2 使用 DbgClr 工具调试 Visual Studio .NET 应用程序	380
8.1.3 缓存 Web 方法的结果	331	9.2.3 使用 CorDbg 工具调试 .NET 应用程序	385
8.1.4 为 Web 方法提供说明	331		
8.1.5 为 Web 方法启用会话状态	332		
8.1.6 用别名标识重载的 Web 方法	332		
8.1.7 使 Web 方法作为事务 的根对象	333		
8.2 创建异步方法	334		
8.2.1 异步编程概述	334		

9.2.4 在 Visual Studio .NET 中调试 XML Web 服务	386	10.2.2 部署 Windows 服务	415
9.3 代码插入	392	10.2.3 部署 XML Web 服务	422
9.3.1 代码插入概述	392	10.2.4 使用 Merge Module 项目 部署组件	426
9.3.2 利用跟踪功能插入代码	392	10.2.5 使用 Cabinet 项目部署 .NET 组件和应用程序	430
9.3.3 异常处理机制	395	10.3 注册及定位组件和程序集	431
9.4 创建及测试多区域性附属程序集和 测试数据	399	10.3.1 注册组件和程序集	432
9.4.1 附属程序集概述	400	10.3.2 定位程序集	433
9.4.2 创建附属程序集	400	10.4 实现版本管理和并行部署	435
9.4.3 用强名称编译附属程序集	402	10.4.1 在程序集中实现版本管理	435
9.4.4 安装附属程序集	402	10.4.2 在全局程序集缓存中部署一个 程序集的多个版本	437
9.4.5 从资源文件中提取数据	402	10.5 本章小结	440
9.5 本章小结	403	10.6 实验: 创建 Setup 项目	440
9.6 实验: 跟踪和调试	403	10.6.1 练习 1: 创建 Setup 项目	440
9.6.1 练习 1: 实现跟踪功能	404	10.6.2 练习 2: 用 Web Setup 项目部署 Web 服务	441
9.6.2 练习 2: 调试已部署的 XML Web 服务	407	10.7 本章复习	442
9.7 本章复习	408	附录 A 问题与答案	444
第 10 章 部署 XML Web 服务和 Windows 服务	410	附录 B COM 互操作性	461
10.1 理解 .NET 部署特性	410	附录 C 考试重点索引	465
10.1.1 理解 .NET 部署	411	附录 D 微软认证专家计划	468
10.1.2 打包和分发选项	412	术语表	472
10.1.3 部署方法	413		
10.2 .NET 部署选项和安装程序	414		
10.2.1 .NET 部署项目概述	414		

第 1 章 .NET 框架

1.1 .NET 框架概述.....	1
1.2 公共语言运行库.....	5
1.3 托管执行过程.....	7
1.4 程序集和全局程序集高速缓存.....	14
1.5 配置和安全性.....	18
1.6 应用程序域和运行库宿主.....	20
1.7 .NET 框架工具简介.....	21
1.8 本章小结.....	26
1.9 实验：创建程序集和查看 MSIL.....	26
1.10 本章复习.....	31

本章概要

在这一章中，将介绍.NET 框架的基础知识。.NET 框架是开发高效分布式 Internet 应用程序的基础结构。另外，还将介绍公共语言运行库在.NET 框架中的作用，并且区分托管和非托管程序执行之间的差异，以及在应用程序的部署和配置中如何使用程序集的相关内容。通过本章的学习，您还会了解.NET 框架怎样提供了一种具有高度安全性和容错性的执行环境。最后，还将讲解如何使用.NET 框架提供的工具来创建、部署及管理应用程序。

先决条件

要完成本章的学习，您必须：

- 在计算机上安装了 Visual Studio .NET
- 熟悉基于 Windows 的、基于 Web 的以及基于控制台的应用程序的开发和部署
- 能够创建、编译和执行 Visual Basic .NET 和 Visual C# .NET 应用程序

书中的每一章都以一个实验作为结尾，实验中包含一系列练习，帮助您巩固该章所学的概念和技能。下载文件中包含这些应用程序的完整版本，可以将其与您的练习结果进行比较，也可以用作解决问题时的参考。配书文件的\Solution 文件夹中包含完成各实验所需的代码，因而无需手动输入这些代码。

1.1 .NET 框架概述

.NET 这一软件使您可以为不同的环境和设备开发应用程序。例如，可以为具有高度分布式环境

(如 Internet)创建 XML(eXtensible Markup Language)Web 服务和 Web 应用程序。也可以创建传统的基于 Windows 的应用程序、服务器组件以及能在任何设备上(如个人电脑、手持设备等)运行的应用程序。另外, .NET 使得不同的应用程序和设备之间能够进行无缝数据交换。.NET 框架是为使用 .NET 创建的应用程序提供执行引擎和运行时服务的基础结构。

本节学习目标

- 了解 .NET
- 了解 .NET 框架以及它在 .NET 中的作用
- 了解 .NET 框架的体系结构

估计学习时间: 25 分钟

1.1.1 了解 .NET

.NET 提供了开发工具、运行时环境、服务器基础结构和智能软件, 它们允许为多种平台和设备生成应用程序。.NET 还通过使用诸如 HTTP(超文本传输协议, Hypertext Transfer Protocol)、XML 以及 SOAP(简单对象访问协议, Simple Object Access Protocol)等标准来集成各种应用程序和设备。.NET 克服了软件行业的一个最大的难题: 在以不同语言、为不同环境编写的应用程序之间交换数据。.NET 允许不同的应用程序通过 XML Web 服务交换数据。另外, .NET 提供了远程处理结构, 这一技术使运行在不同的进程中的应用程序(无论是否在同一计算机上)能够使用二进制或 HTTP 协议交换数据。

.NET 提供的工具及操作系统包括:

- **智能客户端软件** 此软件允许客户端、PC 或移动设备利用 XML Web 服务从任何位置和设备访问数据。
- **.NET Server 基础结构** 提供了用于部署 .NET 应用程序的具有高度安全性和可伸缩性的平台。.NET Server 基础结构包括 Windows 2000 Server、Windows .NET Server 以及 .NET Enterprise Server。
- **XML Web 服务** .NET 环境下应用程序集成的核心。允许以不同语言、在不同平台上编写的 Internet 和 Intranet 应用程序可使用标准协议(如 HTTP、XML 和 SOAP 等)交换数据。关于 XML Web 服务的详细内容, 请参见第 7 章和第 8 章。
- **Visual Studio .NET 和 .NET 框架** Visual Studio .NET 和 .NET 框架提供了创建、托管及使用 XML Web 服务的完整解决方案。Visual Studio .NET 支持多种编程环境和语言, 并提供了对所需工具的单点访问, 这使它成为最为高产的工具之一。

图 1.1 所示是 .NET 的组件。在该图中, XML Web 服务由允许客户端应用程序与其他客户端或服务器应用程序交换数据的核心组件构成。通过 XML Web 服务, 服务器应用程序之间也能够进行数据交换。另外, 运行在任何设备上的应用程序都能与其他运行在其他任何设备上的应用程序交换数据。

1.1.2 .NET 框架及其在 .NET 中的作用

.NET 框架是使用 .NET 创建应用程序的基础结构。.NET 框架提供了一致的面向对象编程模型, 可以用来创建各种类型的应用程序。创建不同的应用程序(如基于 Windows 的应用程序和 XML Web

服务)的方法是相同的。要想创建.NET 应用程序,应该创建一个类,然后通过类的属性、事件和方法定义应用程序的功能。即使是对于 Web 应用程序,控制 Web 页行为的代码也是封装在类中的。另外,类支持面向对象的属性,如继承、封装和多态等。因此,类是.NET 环境下编程的基础。可以用.NET 框架所支持的任何语言来创建类。以一种语言编写的类可为用另一种语言编写的类所重用。还可以跨语言继承类,因为.NET 框架允许语言的互操作性,从而支持跨语言继承。

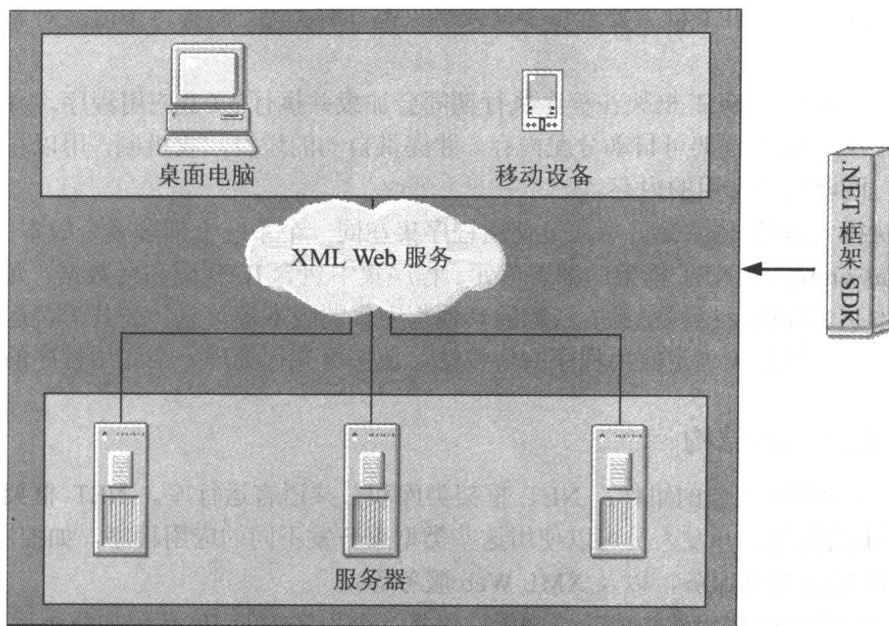


图 1.1 .NET 组件

ECMA(欧洲计算机制造商协会, European Computer Manufacturers Association)标准中定义了公共语言规范(CLS, Common Language Specification)。CLS 中包含有关语言互操作性的规则。用一种符合 CLS 的语言编写的代码可与用另一种符合 CLS 的语言编写的代码进行互操作,这是因为以符合 CLS 的语言编写的代码都会被编译为中间语言(IL, Intermediate Language)代码。运行时引擎会执行 IL 代码。这就确保了符合 CLS 的语言之间的互操作性。

.NET 框架提供了 4 种符合 CLS 的语言: Visual Basic .NET、Visual C# .NET、Visual C++ .NET 和 Visual J# .NET。2001 年 12 月 13 日, ECMA 批准 Visual C# 为国际通用标准。这些语言的编译器可生成一种中间代码,称为微软中间语言(MSIL, Microsoft Intermediate Language), MSIL 使得用 .NET 语言编写的程序可进行互操作。因此,在 .NET 框架中,可以使用任何语言创建应用程序,这些应用程序可与用其他语言编写的应用程序进行互操作。

ECMA 标准——CLI(公共语言基础结构, Common Language Infrastructure)——定义了执行 IL 代码所需的基础结构所遵循规范。CLI 提供了通用类型系统(CTS, Common Type System)和服务,如类型安全和托管代码执行等。 .NET 框架在每个 CLI 规范中都提供了基础结构和服务,其中包括:

- **公共语言运行库** 包括 CLI, 并为 .NET 应用程序提供了运行环境。所有的 .NET 语言编译器都将源代码编译为 MSIL 代码, 当运行应用程序时, 公共语言运行库就会加载并执行这些代码。
- **通用类型系统** 提供在使用不同语言开发应用程序时所必需的数据类型、值和对象类型。

所有 .NET 语言都遵循 CTS。这也就意味着 Visual Basic .NET 中的字符串与 Visual C# .NET 或 Visual C++ .NET 中的字符串是相同的。所有 .NET 语言都可以访问同一个类库。另外，所有语言的功能都同样强大，在 .NET 中没有什么高级语言，Visual Basic .NET 与 Visual C# .NET 或 Visual C++ .NET 一样强大。

- **类型安全** .NET 框架可确保在一个值或对象上所执行的操作仅对该值或对象发生作用。为了实现这一点，.NET 框架要求每个值或对象都具有类型，而且每个值或对象的引用也都有一个类型。
- **托管代码执行** .NET 框架在程序执行期间会加载并执行 .NET 应用程序，并管理对象的状态。另外，.NET 框架可自动分配内存，并提供自动的垃圾回收机制，用以在对象不再被需要时，回收其所占用的内存。
- **并行执行** .NET 框架允许您通过使用程序集在同一个系统上部署多个版本的应用程序。程序集(assembly)是 .NET 框架的部署单元。程序集中包含 IL 代码和元数据。元数据包含程序集版本、名称以及该程序集所依赖的其他程序集的版本等信息。公共语言运行库使用元数据中的版本信息来确定应用程序的依赖性，使您可并行处理一个应用程序的多个版本。

1.1.3 .NET 框架体系结构

.NET 框架包含两个主要的组件：.NET 框架类库和公共语言运行库。.NET 框架类库提供了所有 .NET 语言通用的类型。开发人员可以使用这些类型来开发不同的应用程序，如控制台应用程序、Windows 服务和 Web 窗体服务，以及 XML Web 服务等。

公共语言运行库由一些组件构成，这些组件可将一个程序中的 IL 代码加载到运行库中、将 IL 代码编译为本机代码、执行并管理代码、增强安全性和类型安全，并提供线程支持及其他一些有用的服务。

图 1.2 给出了 .NET 框架的组件。

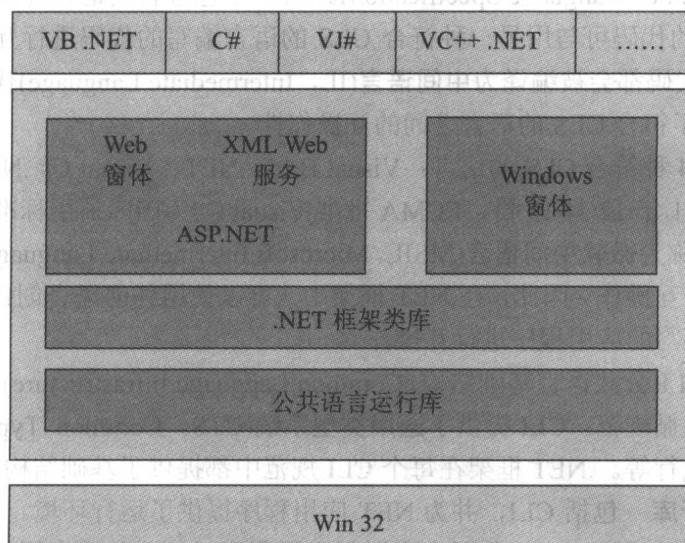


图 1.2 .NET 框架的组件

运行在公共语言运行库中的代码叫做托管代码，而那些不在公共语言运行库中运行的代码就叫