



程序员书库

仇谷烽 澎洪洪 编著

Visual C#.NET网络编程



清华大学出版社

程序员书库

Visual C#.NET 网络编程

仇谷峰 彭洪洪 编著

清华大学出版社

北 京

内 容 简 介

本书结合大量实例,全面介绍了基于 Visual C#.NET 的网络编程,内容包括 Web Forms、ASP.NET、TCP/UDP 端口通信、远程访问、MSMQ 消息队列、电子邮件客户端系统、网络编程应用技巧与 Web Services 等。

本书所有的源代码可以通过 <http://www.tupwk.com.cn> 下载到,用户无须输入任何内容,即可使用本书中展示的所有技巧。

本书结构清晰,内容全面,由浅及深,注重实用,适合于初级和中级网络程序员、网络软件开发人员、网络管理员及广大计算机编程爱好者阅读。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933
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

Visual C#.NET 网络编程/仇谷烽,澎洪洪编著. —北京:清华大学出版社,2004.9
(程序员书库)
ISBN 7-302-09262-1

I. V… II. ①仇…②澎… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 084532 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
组稿编辑:胡伟卷
文稿编辑:刘金喜
封面设计:王 永
版式设计:康 博
印 装 者:三河市春园印刷有限公司
发 行 者:新华书店总店北京发行所
开 本:185×260 印 张:19.25 字 数:444 千字
版 次:2004 年 9 月第 1 版 2004 年 9 月第 1 次印刷
书 号:ISBN 7-302-09262-1/TP·6502
印 数:1~4000
定 价:30.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

前 言

C#是微软公司为其最新推出的.NET 平台量身定制的开发语言,综合了 C/C++和 Java 等语言的优点。网络编程一直是 Java 的天下,微软推出 C#的目的之一就是要与 Java 一争高下。C#.NET 在网络编程方面的功能相当强大。而本书的目的就是要介绍这一种新的语言怎样在基于.NET 平台的架构下,在网络编程领域中大展身手。

1. 本书内容

本书不是介绍怎样使用 C#语言,而是在假设读者已经基本了解并掌握 C#语言的基础上,着重讨论 C#怎样在网络编程上大展风采。

本书通过大量实例向读者展示 C#.NET 在网络编程方面的强大功能,内容包括 HTML、WWW 和 Internet 等基本概念;最普遍的 ASP.NET 编程;底层的 Socket 编程;比较高级的远程访问和 MSMQ(Microsoft Message Queue);电子邮件客户端系统;获取主机 IP 和 MAC 地址;Web 服务器及文件上传/下载等实用技巧;以及微软伴随其.NET 平台推出的 Web Services 编程。

2. 本书面向对象

本书适用的对象是已经对 C#语言有一定的了解,并想学习怎样利用 C#进行网络编程的读者。

当然由于 C#脱胎于 C/C++和 Java,因此如果读者是这几门语言中的任一种程序员,也能很快地适应 C#语言,并从本书中体会到 C#在网络编程方面的强大功能。

3. 本书结构

- 第1章介绍了微软的整个.NET 平台的概念、特点和架构。
- 第2章介绍了 Web 的基础知识,包括什么是 HTML、WWW 和 Internet 等基本概念和 Web 的工作原理。
- 第3章介绍 ASP.NET 网络编程的基础知识。
- 第4章介绍 TCP/UDP 协议的编程,同时也介绍了比 TCP 协议更为底层的 Socket 编程。
- 第5章介绍了微软公司在.NET 平台上引入的一个称为远程访问(Remoting)的平台,它采用了标准的 SOAP(Simple Object Access Protocol)协议作为消息封装协议,而用 HTTP/TCP 作为通信传输协议。
- 第6章介绍了一个简单的 MSMQ 消息中间件。
- 第7章介绍电子邮件客户端系统。本章利用 Socket 开发了基于 POP3 的邮件接收程序、基于 SMTP 的邮件发送程序和基于 ESMTP 的邮件发送程序。

- 第8章介绍了几个网络开发过程中常用的一些技术,如获取本机计算机名、IP地址、获取本机的MAC和远程计算机的MAC等,还开发了一个简单的Web服务器。
- 第9章详细讨论了Web Services的定义、架构和特点,并举例说明了怎样利用C#在.NET平台上开发部署Web Services。
- 第10章介绍一个大整数计算的Web Services。这一章给出了开发一个大整数基本运算的Web Services的完整过程,并讲述了怎样通过ASP.NET和普通Windows应用程序来利用这一Web Services。

4. 本书所用图标

 (演示程序): 表示该部分代码是一个演示程序,读者可用它来加深对本书相关内容的理解。

 (源代码): 表示相关内容的完整源代码在网上下载文件中的相对目录位置,具体下载地址为 <http://www.tupwk.com.cn/downpage/index.asp>。

本书的作者在北京网通公司从事网络编程多年,具有丰富的网络编程经验。另外,冉林仓、刘咏、刘伟、周松建、吴建文、刘旭、赵磊、于秋生、徐日强、张海霞等在本书的素材整理及排版录入方面做了很多工作,在此一并表示感谢。

尽管本书是作者精心构思、认真编写的成果,但是难免存在不足之处,希望读者提出宝贵意见。

作 者

2004年6月

目 录

第 1 章 .NET 平台	1
1.1 .NET 平台简介	1
1.1.1 什么是.NET	1
1.1.2 .NET 的功能	3
1.1.3 .NET 的核心组件	4
1.2 .NET 平台特点	4
1.3 .NET 平台架构	6
1.4 本章小结	7
第 2 章 C# Web 编程基础	8
2.1 Internet 发展简史	8
2.1.1 WWW 与 Internet	8
2.1.2 Web 的工作原理	9
2.1.3 超文本传输协议	10
2.2 Web 编程起步	11
2.2.1 HTML 页面	11
2.2.2 HTML 表单	15
2.2.3 静态和动态 Web 页面	19
2.2.4 Web 的演化过程	24
2.3 .NET Web 编程	25
2.3.1 ASP.NET	26
2.3.2 WebForms	26
2.4 本章小结	27
第 3 章 基于 ASP.NET 的网络编程	28
3.1 WebForms	29
3.1.1 创建第一个 Web 窗体	32
3.1.2 在 Web 窗体上添加控件	35
3.1.3 将控件与数据绑定	37
3.1.4 WebForms 的事件响应	40
3.2 服务器端控件	41
3.3 自定义控件	44
3.3.1 创建项目	46
3.3.2 将控件添加到工具箱	46
3.3.3 个性化控件	47
3.3.4 创建自定义设计器	48

3.3.5	测试控件	50
3.4	HTML 控件	50
3.5	ASP.NET 应用程序	54
3.5.1	什么是应用程序	54
3.5.2	配置 Web.config	58
3.5.3	编写 Global.asax	62
3.5.4	应用和会话	65
3.5.5	安全访问控制	70
3.6	本章小结	77
第 4 章	基于 TCP/UDP 协议的 Socket 编程	78
4.1	Internet 和 Socket 简介	78
4.1.1	TCP 协议简介	79
4.1.2	UDP 协议简介	80
4.1.3	端口简介	81
4.1.4	System.Net 名称空间	81
4.1.5	System.Net.Sockets 名称空间	82
4.2	基于 TCP 协议的网络会话实例	82
4.2.1	服务器端编程	83
4.2.2	客户端编程	87
4.2.3	编译运行	90
4.3	基于 TCP 的 P2P 文件共享应用	91
4.3.1	QRemoteFileStream 协议	92
4.3.2	QRemoteFileStreamServer 类	93
4.3.3	QRemoteFileStreamProxy 类	93
4.3.4	QFileSharingPeer 类	93
4.3.5	P2P-GUI 程序的编写和编译运行	96
4.4	基于 UDP 协议的网络会话实例	98
4.4.1	服务器端编程	99
4.4.2	客户端编程	103
4.4.3	编译运行	105
4.5	基于 UDP 广播的自动收报机的实现	106
4.5.1	服务器和客户端的公共类	106
4.5.2	服务器端编程	108
4.5.3	客户端编程	110
4.5.4	编译运行	114
4.6	基于 UDP 的客户—服务器聊天应用的实现	114
4.6.1	QTCPServerSession 类	115
4.6.2	QTCPServer 类	117

4.6.3 聊天协议	119
4.6.4 QChatServer 类	120
4.6.5 QChatClient 类	122
4.6.6 编译和运行程序	125
4.7 网络资料的访问	126
4.7.1 代理的概念	126
4.7.2 访问网络资源的客户端程序	127
4.8 本章小结	130
第 5 章 远程访问	131
5.1 远程访问概述	131
5.2 创建一个简单的远程访问客户—服务器程序	132
5.2.1 远程服务对象的创建	133
5.2.2 创建主应用	134
5.2.3 创建客户端应用	136
5.2.4 代码的理解	138
5.2.5 示例应用的改进	141
5.3 远程访问在企业内部网络中的应用	148
5.3.1 对象生命周期和持续时间	148
5.3.2 创建一个计数服务器	149
5.3.3 创建一个计数主应用	150
5.3.4 创建一个计数客户端应用	152
5.4 远程访问对象的版本问题	155
5.4.1 创建一个带版本的远程访问应用	156
5.4.2 创建一个带版本的远程访问主应用	157
5.4.3 创建一个带版本的远程访问客户端应用	159
5.5 本章小结	161
第 6 章 基于 MSMQ 的消息队列	162
6.1 MSMQ 概述	162
6.1.1 MSMQ 的架构	163
6.1.2 MSMQ 的安装	164
6.2 创建一个简单的 MSMQ 应用	164
6.2.1 一个简单的 MSMQ 应用	164
6.2.2 MSMQ 应用代码分析和一些高级技术	166
6.3 创建一个复杂的 MSMQ 应用	174
6.3.1 创建 MSMQDrawing 绘图类库	174
6.3.2 创建 DrawingSender 绘图发送工程	176
6.3.3 创建 DrawingReceiver 绘图接收工程	179

6.4	MSMQ 的异步调用	182
6.5	本章小结	184
第 7 章	电子邮件客户端系统	185
7.1	电子邮件的相关基础知识	185
7.2	POP3 邮件接收程序	186
7.2.1	POP3 协议简介	187
7.2.2	利用 POP3 协议的邮件接收程序	188
7.3	SMTP 邮件发送程序	195
7.3.1	SMTP 协议简介	195
7.3.2	利用 SMTP 协议发送邮件	206
7.4	ESMTP 邮件发送程序	210
7.5	本章小结	220
第 8 章	网络编程实用技巧示例	221
8.1	获得主机名和 IP 地址	221
8.1.1	DNS 类	221
8.1.2	读取计算机名和本机固定 IP 地址	222
8.1.3	读取计算机名和本机当前所有的 IP 地址	223
8.2	获取机器的 MAC 地址	225
8.2.1	获得本地机器的 MAC 地址	225
8.2.2	获得远程机器的 MAC 地址	226
8.3	简单 Web 服务器的开发	229
8.3.1	HTTP 协议的工作原理	229
8.3.2	简单 Web 服务器的开发	230
8.4	文件的下载和上传	240
8.4.1	文件的下载	240
8.4.2	文件的上传	242
8.5	本章小结	244
第 9 章	XML Web Services 编程	245
9.1	Web Services 的应用场合	245
9.2	Web Services 的体系结构	247
9.2.1	Web Services 的搜索引擎	248
9.2.2	Web Services 的发现	250
9.2.3	Web Services 可调用的方法	251
9.2.4	Web Services 客户端程序对方法的调用	251
9.3	Web Services 与 .NET	253
9.3.1	Web Services 的创建	254
9.3.2	Web Services 的客户端程序	255

9.4	一个简单的 ASP.NET Web Services	256
9.4.1	Web Services 的创建	256
9.4.2	Web Services 客户端程序	272
9.5	本章小结	277
第 10 章	一个大整数计算的 Web Services	278
10.1	大整数处理的功能设计	278
10.1.1	功能模块设计	278
10.1.2	关键技术	279
10.2	Web Services 的创建	280
10.2.1	大整数计算基本类库的创建	280
10.2.2	大整数计算 Web Services 的创建	282
10.3	客户端程序的创建	284
10.3.1	基于 Windows 的客户端程序	284
10.3.2	基于 ASP.NET 的网络客户端程序	289
10.4	程序测试	291
10.5	本章小结	294

第 1 章 .NET 平台

C#是伴随着微软公司的.NET 计划同时推出的,而且是为.NET 平台量身定制的开发语言。因此,在真正进入 C#网络编程前,最好先了解一下.NET 平台的基础知识。

本章要点

- .NET 平台简介
- .NET 平台特点
- .NET 平台架构

1.1 .NET 平台简介

2000 年 6 月 22 日微软公司正式推出了其下一代操作系统平台 Microsoft .NET(以下简称.NET)。这项计划使微软已有的软件在 Web 时代适用于传统的 PC,并满足目前呈强劲增长势头的新设备的需要,如蜂窝电话以及个人数字助理(Personal Digital Assistant, PDA)等。微软还计划通过创建新的工具来吸引软件开发人员和合作伙伴对 Microsoft.NET 的认同,并且开发出其他基于 Internet 的服务。本章将主要向读者介绍什么是.NET。

1.1.1 什么是.NET

从微软公司的角度来讲,因特网的革命就是要建设一个平台来创建、支持新一代的应用。因此,必须提供一套通用系统服务来支持这样的操作。

.NET 平台是一种新的计算平台,它简化了在高度分布式 Internet 环境中开发应用程序的过程。.NET 平台旨在实现下列目标:

- 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。
- 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。
- 提供一个保证代码(包括由未知的或不完全受信任的第三方创建的代码)安全执行的环境。
- 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。
- 使开发人员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)时保持方法一致。

- 按照工业标准生成所有通信, 以确保基于 .NET 框架的代码可与任何其他代码集成。

.NET 框架具有两个主要组件: 公共语言运行库和 .NET 平台类库。公共语言运行库是 .NET 平台的基础, 它可以被看作是一个在执行时管理代码的代理。公共语言运行库提供核心服务(如内存管理、线程管理和远程处理), 而且还强制实施严格的类型安全以及可确保安全性和可靠性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 平台的另一个主要组件是类库, 它是一个综合性的面向对象的可重用类型集合, 可以用来开发包含从传统的命令行或图形用户界面(GUI)应用程序到基于 ASP.NET 所提供的创新的应用程序(如 Web 窗体和 XML Web Services)在内的应用程序。

.NET 平台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 平台不但提供若干个运行库宿主, 而且还支持第三方运行库宿主的开发。

例如, ASP.NET 承载运行库以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 Web 窗体应用程序和 XML Web Services。

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, 但是它具有只有托管代码才能提供的重大改进(如不完全受信任的执行和安全的独立文件存储)。

图 1-1 显示了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。该插图还显示托管代码在更大的结构内运行的方式。



图 1-1 公共语言运行库与应用程序以及整个系统间的关系

简而言之, .NET 是一种面向网络、支持各种用户终端的开发平台。微软公司的宏伟目标是让 Microsoft .NET 彻底改变软件的开发方式、发行方式和使用方式, 并且不

只是针对微软公司一家,而是面向所有开发商与运营商。.NET 的核心内容之一就是搭建第三代因特网平台,这个网络平台将解决网站之间的协同合作问题,从而可以最大限度地获取信息。在.NET 平台上不同网站之间通过相关的协定联系在一起,网站之间自动交流、协同工作,从而提供最全面的服务。

1.1.2 .NET 的功能

某一天,出差到外地,在机场租借移动电话。在向该终端插入自己的 IC 卡后,自己的地址簿和计划簿被自动下载,随即移动电话就变成了你个人专用的 PDA。这就是.NET 为我们描绘的一个未来生活的场景。

人们的需求总是无法满足,正是人们这种无穷的需求,推动着技术的进步。21 世纪 Internet 将成为商业活动的主要场所,B2B、B2C 等电子商务的运作方式,即一对一营销的经营概念将把网络的服务功能提高到前所未有的程度。

微软公司在此时提出.NET 有其深远的战略考虑,主要体现在以下几个方面:

- 改革商务模型

微软公司打算改变只靠销售软件包的商务模型,今后将重心转移到可以在网络上使用的“服务”型商务。这样,首要的问题就是解决网络上用来开发并执行“服务”的平台,这就是 Microsoft .NET。

- 提高软件开发效率

使应用程序的发布更为容易,不用因为 DLL 版本不同而重新启动计算机以安装应用程序。

- 改进用户界面,并能支持多种用户终端

用户界面演进的结果体现在两个方面:一是完成传统的 PC 界面与基于 XML 的浏览器界面间的过渡;二是对自然语言和语音识别的支持,从而使用户与各种终端之间的沟通更加透明,真正达到网络互联的 3A(Anywhere、Anytime 和 Any device)。

今天,许多人仍觉得除了上网看新闻外,再没有什么有意义的事情可做。这是因为今天的互联网与旧时的大型计算机的工作模式还有许多相似之处,信息被存储在中央服务器内,而用户的所有操作都要依靠这些信息。在不同的网址之间相互传递信息,或者不同的网址之间相互合作以提供更广泛和更深层次的服务,还是一件十分困难的事。

现代人时常有一种困惑,感觉如今生活在技术与机器架构的丛林中,人们在努力地去适应机器,适应技术,而不是机器和技术适应人类。科技以人为本还只是一个美好的愿望。这是因为我们还不能将控制信息的权利交给那些需要信息的人们,.NET 的出现,意味着人们可以只用一种简单的界面就可以编写、浏览、编辑和分享信息,而且还可以得到功能强大的信息管理工具。由于要使用的所有文件都以符合网络协议的格式存在,所以所有的商业用户和个人用户都可以方便地查找和使用其中的信息,任何规模的公司都可以使用相同的工具与他们的供应商、商业伙伴和客户高效地沟通

和分享信息，这样就创造出了一种全新的协同工作模式。

总之.NET 战略是一场软件革命，主要原因有以下 3 点：

- .NET 对最终用户来说非常重要。因为计算机的功能将会得到大幅度提升，同时计算机的操作也会变得非常简单。用户将完全摆脱人为的硬件束缚，可以在因特网的多维时空自由访问、查看和使用自己的数据，而不用被束缚在便携式电脑的方寸空间中——可通过任何桌面系统、任何便携式电脑、任何移动电话或 PDA 进行访问，并可对其进行跨应用程序的集成。
- .NET 对开发人员来说也十分重要。因为它不但会改变开发人员开发应用程序的方式，而且使得开发人员能创建出全新的各种应用程序，大幅提高软件的开发效率。.NET 将保证完全消除当今计算技术中的所有缺陷，.NET 一定能实现用户从任何地点、任何设备都可访问其个人数据和应用程序的宏伟蓝图。
- .NET 把雇员客户和商务应用整合成一个协调的、能进行智能交互的整体。各公司无疑将是这场效率和生产力革命的最大受益者。.NET 承诺为人类创造一个消除任何鸿沟的商务世界。

1.1.3 .NET 的核心组件

.NET 的核心组件包括以下几点：

- 一组用于创建互联网操作系统的构建块。其中包括 Passport.NET 用于用户认证以及用于文件存储的服务，用户首选项日历管理，以及众多的其他任务。
- 构建和管理新一代服务的基本结构和工具。包括 Visual Studio.NET 企业服务器 .Net Framework 和 Windows .NET。
- 能够启用新型智能互联网设备的.NET 设备软件。
- .NET 用户体验。

1.2 .NET 平台特点

.NET 平台由公共语言运行时、基础类库和公共语言规范组成。基础类库展现了公共语言运行时的功能，这很类似于 Windows API(我们可以通过它来利用 Windows 操作系统的功能)。但基础类库比应用程序接口(Application Programming Interface, API)提供了更为高层的功能来方便代码的重用。

这一平台带来的好处，远不止提供了一个一致的 API。公共语言运行时和基础类库使所有的应用服务可通过公共的面向对象的编程模式得到利用。在传统的通过 DLL 或 COM 对象的调用方式访问操作系统功能的编程模式中，程序员需要做许多额外的枯燥工作，来保证程序的各个模块能无缝地衔接。而为了利用某些特性，程序员可能需要通过低级语言来完成，从而影响了设计和决策的效率。

新的平台极大地简化了编写 Windows DNA(Windows 分布式 Internet 应用结构)应用、几乎所有的 Win32 和 COM 工程。程序开发者不再需要深入了解与 Windows 或 COM 架构相关的 GUIDs、IUnknown()、AddRef()、Release()和 HRESULTS 等知识。.NET 平台不但隐藏了实现细节,且在新的平台上,这些概念已经消失了。新平台的另一改进是对错误的处理。旧的平台上,某些函数返回 Win32 错误代码,有的返回 HRESULTS,有的则处理异常。因此要求程序员编写不同的错误处理代码。新的平台全都用异常来处理错误,极大地简化了代码的编写、阅读和管理。

具体来说,新平台具有下述特点:

- 多语言开发

在.NET 平台中,微软采用了一种称为微软中间语言(MSIL)的技术来实现对多语言开发的支持。与以往直接编译成可执行文件不同,在新的平台上所有程序被编译成微软中间语言。因此,任何语言的编译器只要能支持微软中间语言,就能与.NET 平台相集成。目前已有 COBOL、Eiffle、Fortran、Perl、Python 及 Scheme 等已经声称支持.NET 平台,详细列表可在下列网址中查寻:

<http://msdn.microsoft.com/vstudio/partners/language/default.asp>

- 平台与处理器独立

微软中间语言独立于 CPU,是一种比传统机器语言层次更高的语言。应用一旦建立,可托管的.NET 应用可运行于任何支持.NET 公共语言运行时的操作系统中。因为在.NET 平台上的开发屏蔽了硬件和操作系统的差异。

- 自动化内存管理

内存泄漏可能是每个 C/C++程序员最头痛的事。不过在.NET 平台上,这一问题有望得到彻底解决。.NET 平台的托管对象引入了垃圾收集器的功能,该功能会定时检查被丢弃的内存,并进行相应的回收。从而使程序员可把复杂的内存管理交给平台自己来处理。

- 版本支持

DLL 版本的冲突可能是每个资深程序员都碰到过的问题。有人把它称为是 DLL 的“恶梦”。然而在新的平台上,已经不用再为此担心了。新的平台把应用组件分开了。采用一种称为装配器的技术,使应用只装载开发时编译测试用的组件。

- 支持开放的标准

在网络编程上,新平台支持一系列开放标准,如 XML、HTTP、SOAP、UDDI 及 WSDL 等。另外,作为新平台定制的语言 C#和核心技术公共语言架构(Common Language Infrastructure, CLI),微软也以前所未有的开放态度,公布了 C#语言和 CLI 规范,并提交给欧洲计算机制造协会(ECMA)申请为国际标准。2001 年 12 月 13 日,ECMA 批准 C#和 CLI 规范作为国际标准:

C#——ECMA-334

CLI——ECMA-335 (ECMA TR84)

- 容易部署

在旧的平台上,安装或卸载一个大型软件是一件很费劲的事,其中涉及注册表、

环境变量、安装目录等一系列事件。在新的平台上，微软采用了组件自描述技术，免去了以上这一系列麻烦。在新平台上，删除一个软件同删除一个文件一样容易。

● 分布式架构

现在的分布式计算环境与将来见到的将有很大的不同。新的平台中，微软把 Web Services 作为未来分布式计算的模式。Web Services 是基于网络的、分布式的模块化组件，它执行特定的任务，遵守具体的技术规范，这些规范使得 Web Services 能与其他兼容的组件进行互操作。

1.3 .NET 平台架构

.NET 平台的架构可用图 1-2 表示，整个平台由七大部分组成：支持公共语言规范的多种开发语言、公共语言规范、应用程序编程接口(包括 ASP.NET、Web Services、Windows GUI 和控制台程序等)、数据库编程接口(数据和 XML)、基础类库、公共语言运行时和一个集成开发环境 Visual Studio.NET。

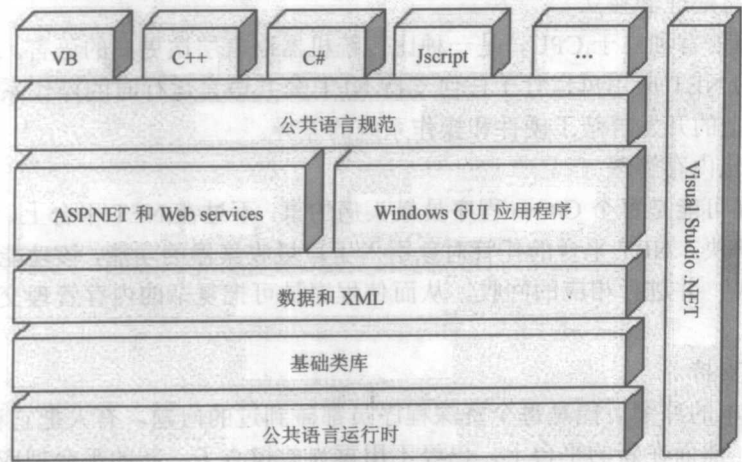


图 1-2 .NET 平台架构

对图 1-2 中各项简要说明如下。

- 多种开发语言：平台的最上层是一组支持公共语言规范的开发语言。
- 公共语言规范：最上层的多语言支持是通过公共语言规范来实现的。具体来说，微软提出了一种称为微软中间语言(MSIL)的概念，任何其他语言都要先编译成微软中间语言。这样任何开发语言只要能支持公共语言规范，就能集成到微软的.NET 平台上。
- 应用程序编程接口：在公共语言规范的下面是一系列应用程序编程接口。包括 ASP .NET、Web Services、Windows GUI 和控制台程序等模块。利用这些编程接口，开发人员可以快速开发出高质量的.NET 平台的应用程序。

- 数据库编程接口：在应用程序编程接口的下面，为数据库的管理和访问提供了一系列组件，使用它们可以大大简化数据库的编程工作。
 - 基础类库：在编程接口与公共语言运行时之间，是一个功能强大的基础类库。利用它，程序员可以方便、安全、可靠地访问.NET 平台提供的功能。
 - 公共语言运行时：最底层是公共语言运行时。从微软的设计意图来看，是利用它来实现跨平台操作的。新平台的程序编译过程类似 Java，程序先被编译成微软中间语言。但不会被编译成执行程序，它采用即时编译的方式(Just In Time, JIT)。具体来说，由当前操作系统平台上运行的公共语言运行时即时把微软中间语言编译成当前操作系统的机器码，从而实现.NET 的跨平台特性。
- .NET 上的程序开发到本地执行的整个过程可用图 1-3 表示。

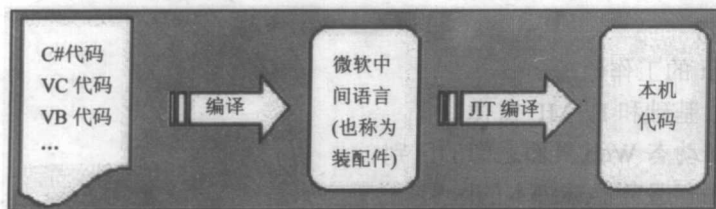


图 1-3 程序开发到本地执行的过程

- Visual Studio.NET：.NET 架构的最后部分是一个优秀的集成开发环境——Visual Studio.NET。在本书中我们将主要学习如何利用该开发环境开发网络应用程序。

1.4 本章小结

本章简要介绍了微软新推出的.NET 平台的概念、特点和架构。本章的主要目的是使读者对.NET 平台有一个宏观了解，而对.NET 平台在网络编程中的强大功能，读者必须在后续的章节中逐步学习，仔细体会。