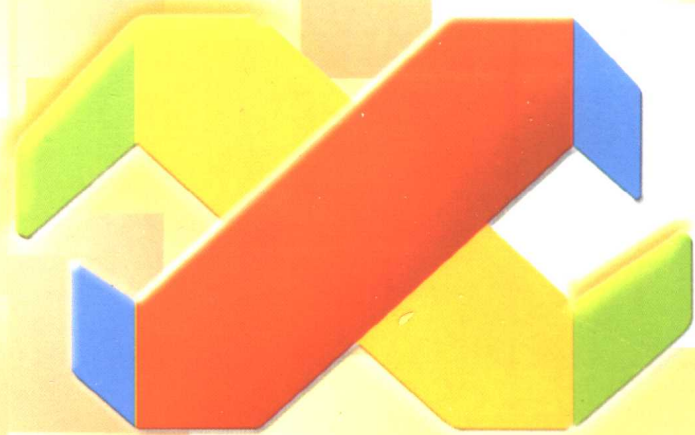




C# 系列丛书



深入浅出 C# 编程

北京希望电子出版社 总策划
张玉平 李长林 编 著



中国科学技术出版社
CHINA SCIENCE AND TECHNOLOGY PRESS



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn



C# 系列丛书



深入浅出 C# 编程

北京希望电子出版社 总策划
张玉平 李长林 编 著



中国科学技术出版社
CHINA SCIENCE AND TECHNOLOGY PRESS



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

图书在版编目 (CIP) 数据

深入浅出 C#编程/张玉平, 李长林编著. —北京: 中国科学技术出版社, 2004.6

ISBN 7-5046-3751-3

I. 深… II. ①张…②李… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 017597 号

书 名 : 深入浅出C#编程
文本著作者 : 张玉平 李长林
责任编辑 : 周 艳 沈葆华
出版、发行者 : 中国科学技术出版社 北京希望电子出版社
地 址 : 北京市海淀区中关村南大街 16 号 100081
北京市海淀区知春路甲 63 号卫星大厦三层 100080
网址: www.bhp.com.cn E-mail:lwm@bhp.com.cn zwb@bhp.com.cn
电话: 010-62520290, 62528991, 62630301, 62524940, 62521921,
62521724 (发行) 010-82675588-318,62532258,62562329 (门市)
010-82675588-501,82675588-201 (编辑部)
经 销 : 各地新华书店、软件连锁店
排 版 : 希望图书输出中心 李兴旺
印 刷 者 : 北京媛明印刷厂
开本 / 规格 : 787 毫米×1092 毫米 1/16 33.5 印张 782.5 千字
版次 / 印次 : 2004 年 6 月第 1 版 2004 年 6 月第 1 次印刷
印 数 : 0001~5000 册
书 号 : ISBN 7-5046-3751-3/TP·239
定 价 : 46.00 元



内 容 简 介

这是一本介绍 C# 编程技术的专著。

本书共由 15 章内容构成。主要内容如下：.NET 与 C#；C# 开发环境；C# 语言基础；面向对象的程序设计；C# 结构、文件处理；域、属性与索引器；接口；异常处理；C# 控件；图形开发；数据库；网络；多线程；应用配置和组件；代码调试和安全。

本书的特点是内容深入浅出，在各个知识点的后面都有一个例子，因此本书可读性和指导性较强，适合了解 C# 编程基本概念的初、中级软件开发者参考使用。

前 言

C#（读作“C-sharp”）作为 Microsoft 的下一代面向对象的语言产品，能够让开发人员在 .NET 平台上快速地建立大量的应用程序。按照微软给出的定义，C# 是一种源自于 C 和 C++ 的，简单的、现代的、面向对象的和类型安全的程序设计语言。C# 为程序员提供了开发飞速发展的 Web 应用程序所需的强大而灵活的功能。C# 和 Java 的核心与 C++ 比较有着相同的优势和局限，比起 C++，C# 将更容易被人们理解和接受，未来大量 .NET 平台的应用将由 C# 开发。

C# 将是未来开发企业级分布式应用程序的首选。当然这并不是说 C# 只适用于网络开发。例如，我们可以使用 C# 开发任何古典风格的 Windows 程序。不仅如此，不管是控制程序、图形程序、NT 服务程序，还是普通组件，乃至 Web 页面，都可以用 C# 开发出来。可以相信，在阅读完本书之后，不论您是一名 C 和 C++ 程序员，或者您一直在使用 VB 或 Delphi 进行编程，您都将欣赏 C# 为我们所带来的一切。

如果你学习过 Java 语言，会发现 C# 在很多方面也非常类型于 Java。Java 程序的执行以及 Java 语言的平台无关性，是建立在 Java 虚拟机 JVM 的基础上的，而 C# 语言则需要 .NET 框架的支持。我们所说的很多 C# 语言的新特点和功能，如类型安全、垃圾收集、版本控制、兼容性以及跨语言的异常处理和继承，这些都是依赖于底层的 .NET 框架来完成的。本书的第 1 章中介绍了微软 .NET 战略和 .NET 框架的背景知识，这对于更好的掌握 C# 编程会有很大的帮助。当然本书只是对 C# 学习的一个入门，要真正地掌握它在实际的 .NET 应用开发中去实践。

本书采用循序渐进的方式，帮助读者从入门到真正精通 C# 语言。在书中，我们并不是把 C# 仅仅作为一种简单的程序设计语言来对待，而是侧重于分析 C# 的语法现象以及它真正面向对象的特征。对于 C# 程序设计涉及到的一些系统理论和方法学，我们也简明扼要地进行了阐述。书中提供了大量的示例，通过这些实例，避免了空洞的语法规则罗列和简单的说教，通过它们，您将得心应手地开发属于自己的 C# 程序。

尽管 C# 是从 C 和 C++ 中发展起来的，但本书并不要求读者具有 C 或 C++ 语言的基础知识。无论您是其它程序设计语言的程序员，或者刚刚开始编程生涯，本书都将帮助您顺利进入 C# 的大门。对已经熟练掌握 C++ 程序设计的专业程序员来说，您将通过本书迅速过渡到新一代面向对象的语言——C#。

本书由张玉平执笔，参加本书编写的还有李长林、龚建、温尚清、丁满泉、黄司渭、张德强、陈博、陈发吉、李正希、周树成、胡方霞、史阳、张志、王东军、李立元、赖擎、周莹，在这里表示深深的感谢。由于时间比较仓促，书中难免存在一些不足之处，我们诚恳地期望广大读者批评指正。

目 录

第 1 章 .NET 与 C#.....	1	2.2.8 Window 菜单组.....	52
1.1 .NET 简介.....	1	2.2.9 .NET 开发环境中的工具条.....	53
1.1.1 什么是.NET.....	1	2.3 .NET 下开发程序的一般步骤.....	54
1.1.2 为什么要提出.NET.....	2	第 3 章 C#语言基础.....	56
1.1.3 .NET 能带来什么.....	3	3.1 绪论.....	56
1.1.4 .NET 的组成.....	4	3.1.1 自动内存管理.....	56
1.2 .NET 的技术核心.....	5	3.1.2 类型.....	57
1.2.1 XML 语言.....	5	3.1.3 预定义类型.....	59
1.2.2 Web 服务.....	6	3.1.4 数组类型.....	61
1.2.3 .NET 框架.....	9	3.1.5 统一系统类型.....	62
1.3 为什么需要 C#.....	15	3.1.6 语句.....	63
1.4 C#的主要特点.....	15	3.1.7 类.....	64
1.4.1 简单.....	16	3.1.8 结构类型.....	65
1.4.2 面向对象.....	17	3.1.9 接口.....	66
1.4.3 类型安全.....	18	3.1.10 代理(delegates).....	66
1.4.4 版本控制.....	18	3.1.11 枚举.....	66
1.4.5 现代.....	18	3.1.12 命名空间.....	66
1.4.6 灵活与兼容.....	18	3.1.13 属性.....	67
1.5 C#程序的编译和执行.....	18	3.1.14 索引器.....	68
1.5.1 即时编译器 JITer.....	21	3.1.15 事件.....	69
1.5.2 及时编译执行的效率.....	21	3.1.16 Attribute.....	70
第 2 章 C#开发环境.....	22	3.2 语法结构.....	71
2.1 Visual Studio .NET 集成开发		3.2.1 预处理.....	71
环境介绍.....	22	3.2.2 标识符.....	75
2.1.1 Visual Studio.NET 开发环境.....	22	3.2.3 关键字.....	75
2.1.2 Visual Studio .NET 集成		3.2.4 数据类型.....	76
开发环境的新内容.....	22	3.2.5 操作符和标点符号.....	79
2.1.3 Visual Studio .NET 常用窗口		3.3 基本概念.....	80
对象简介.....	26	3.3.1 声明.....	80
2.2 Visual Studio .NET 菜单介绍.....	37	3.3.2 成员.....	80
2.2.1 File 菜单组.....	37	3.3.3 成员可见性(member-access) ..	81
2.2.2 Edit 菜单组.....	40	3.3.4 存取控制.....	83
2.2.3 View 菜单组.....	42	3.4 类型.....	83
2.2.4 Project 菜单组.....	45	3.4.1 值类型.....	84
2.2.5 Build 菜单组.....	47	3.4.2 默认值.....	84
2.2.6 Debug 菜单组.....	48	3.4.3 简单类型.....	85
2.2.7 Tools 菜单组.....	49	3.4.4 引用型.....	87

3.5 变量.....	88	3.9.2 命名空间声明 (namespace-declaration)	138
3.5.1 变量的分类.....	89	3.9.3 Using 命令	139
3.5.2 变量默认值.....	90	3.9.4 命名空间的应用.....	140
3.5.3 变量明确赋值.....	90	3.10 类.....	144
3.5.4 变量引用 (variable-reference) ..	92	3.10.1 类声明.....	145
3.6 类型转换.....	92	3.10.2 类成员.....	147
3.6.1 隐式转换.....	92	3.10.3 常量(constant).....	150
3.6.2 显式转换.....	94	3.10.4 域.....	152
3.7 表达式.....	95	3.10.5 方法.....	157
3.7.1 表达式的分类.....	95	3.10.6 构造函数.....	171
3.7.2 运算符.....	96	3.10.7 析构函数.....	176
3.7.3 加括号的表达式.....	98	3.10.8 静态构造函数.....	176
3.7.4 表达式调用.....	98	3.11 结构.....	179
3.7.5 元素访问(element-access).....	99	3.11.1 结构声明.....	179
3.7.6 this 访问.....	101	3.11.2 结构成员.....	179
3.7.7 Base 访问.....	103	3.11.3 结构例子.....	180
3.7.8 递增和递减操作符.....	105	第4章 面向对象的程序设计.....	185
3.7.9 new 操作符.....	106	4.1 面向对象的基本概念.....	185
3.7.10 typeof 操作符.....	111	4.1.1 面向对象的技术的由来.....	185
3.7.11 sizeof 操作符.....	112	4.1.2 基本概念.....	186
3.7.12 checked 和 unchecked 操作符 ..	113	4.2 对象的模型技术.....	187
3.7.13 一元表达式.....	118	4.2.1 对象模型.....	187
3.7.14 二元表达式.....	119	4.2.2 动态模型.....	187
3.8 语句.....	123	4.2.3 功能模型.....	188
3.8.1 终点和可达.....	123	4.3 面向对象的分析.....	188
3.8.2 分程序.....	124	4.3.1 论域分析和应用分析.....	189
3.8.3 空语句 (empty-statement)	124	4.3.2 Coad 和 Yourdon 面向对象的 分析过程.....	189
3.8.4 标号语句 (labeled-statement) .	124	4.4 面向对象的设计.....	191
3.8.5 声明语句 (declaration-statement)	125	4.4.1 高层设计和底层设计.....	191
3.8.6 表达式语句 (expression-statement)	126	4.4.2 Coad 和 Yourdon 面向对象的 设计过程.....	192
3.8.7 条件语句 (selection-statement)	127	4.5 类的基础知识.....	193
3.8.8 循环语句 (iteration-statement) ..	131	4.5.1 类名与 class 类型.....	194
3.8.9 转至语句(Jump)	133	4.5.2 类名作用域.....	195
3.8.10 checked 和 unchecked 语句	136	4.5.3 类对象与类成员.....	195
3.8.11 Lock 语句	137	4.5.4 内部函数.....	196
3.9 命名空间.....	137	4.5.5 静态成员.....	196
3.9.1 编译单元.....	137	4.5.6 成员作用域.....	198

4.5.7 基类与派生类存取.....	199	5.4.5 写文件.....	241
4.5.8 虚基类.....	201	5.4.6 追加文件.....	243
4.6 构造函数.....	201	5.4.7 读文件.....	244
4.6.1 默认构造函数.....	202	5.5 随机文件.....	245
4.6.2 构造函数的重载.....	203	5.5.1 随机文件记录.....	245
4.6.3 构造函数的调用次序.....	203	5.5.2 打开随机文件.....	246
4.6.4 类的初始化.....	204	第6章 域、属性与索引器.....	248
4.7 析构函数.....	207	6.1 属性.....	248
4.7.1 析构函数的调用.....	207	6.1.1 为什么需要属性.....	248
4.7.2 atexit、#pragma exit 与 析构函数.....	207	6.1.2 属性的定义.....	252
4.7.3 exit 与析构函数.....	208	6.1.3 属性的读/写类型.....	253
4.7.4 abort 与析构函数.....	208	6.1.4 属性的底层实现.....	253
4.7.5 虚析构函数.....	208	6.1.5 属性的继承.....	256
4.8 类的方法.....	209	6.2 索引器.....	259
4.8.1 方法参数.....	210	6.2.1 索引器的定义.....	259
4.8.2 方法重载.....	212	6.2.2 索引器的底层实现.....	262
4.8.3 方法覆盖.....	214	6.2.3 多索引器例题.....	264
4.8.4 类的属性.....	216	6.3 域.....	265
4.9 索引.....	217	6.3.1 声明.....	266
4.10 事件处理.....	219	6.3.2 静态域和非静态域.....	266
4.11 使用修饰符.....	221	6.3.3 只读域.....	268
4.11.1 类修饰符.....	221	6.3.4 域的初始化.....	269
4.11.2 成员修饰符.....	222	6.4 复习题.....	270
4.11.3 存取修饰符.....	222	6.5 本章小结.....	271
第5章 C#结构、文件处理.....	225	第7章 接口.....	272
5.1 C#的结构及其用法.....	225	7.1 组件编程技术.....	272
5.1.1 结构的定义.....	225	7.1.1 应用程序的体系结构.....	272
5.1.2 结构的初始化.....	228	7.1.2 组件.....	273
5.1.3 嵌套结构.....	230	7.1.3 组件化程序设计.....	274
5.2 结构数组.....	232	7.2 接口定义.....	274
5.2.1 结构数组的声明.....	232	7.2.1 声明.....	274
5.2.2 结构数组的访问.....	234	7.2.2 接口的继承.....	275
5.2.3 建立结构数组.....	234	7.3 接口的成员.....	276
5.3 数组结构成员.....	235	7.3.1 接口成员的定义.....	276
5.4 顺序文件.....	237	7.3.2 对接口成员的访问.....	276
5.4.1 为什么要使用磁盘.....	237	7.3.3 接口成员的全权名.....	279
5.4.2 磁盘文件访问类型.....	237	7.4 接口的实现.....	280
5.4.3 顺序文件的操作.....	238	7.4.1 类对接口的实现.....	280
5.4.4 文件的打开和关闭.....	238	7.4.2 显式接口成员执行体.....	282
		7.4.3 接口映射.....	284

7.4.4	接口实现的继承机制.....	288
7.4.5	接口的重实现.....	290
7.5	抽象类与接口.....	292
7.6	本章小结.....	292
7.7	复习题.....	293
第8章	异常处理.....	295
8.1	什么是异常.....	295
8.2	异常的基本思想.....	296
8.3	C#的异常处理机制.....	296
8.3.1	C#与C++异常处理的区别.....	297
8.3.2	引发异常的条件.....	297
8.4	异常处理的实现.....	298
8.4.1	try-catch 组合.....	299
8.4.2	try-finally 组合.....	302
8.4.3	try-catch-finally 组合.....	305
8.5	异常处理规则.....	306
8.6	throw 语句.....	309
8.7	用户自定义异常.....	311
8.8	异常处理与跳转语句.....	314
8.9	本章小结.....	316
8.10	复习题.....	316
第9章	C#控件.....	319
9.1	Windows 下的编程.....	319
9.1.1	Windows 下编程要注意的 几个问题.....	319
9.1.2	开发一个 Windows 下的 应用程序.....	320
9.2	常用的 form 控件.....	327
9.2.1	控件的基本概念.....	327
9.2.2	从哪里得到控件.....	328
9.2.3	控件的使用方法简介.....	328
9.2.4	菜单、工具栏、状态栏、 和图标的使用.....	332
9.3	本章小结.....	335
第10章	图形开发.....	336
10.1	Visual csharp 画图基本概念.....	336
10.1.1	上下文设备, 句柄和 Graphics 对象.....	336
10.1.2	画一条直线.....	336
10.1.3	画图和填充.....	338

10.1.4	构造 Regions.....	339
10.1.5	Ellipses 和 Arcs.....	340
10.1.6	Drawing, Positioning 和 Cloning Images.....	340
10.1.7	Polygons.....	341
10.1.8	Pens, Lines 和 Rectangles 高级使用.....	342
10.1.9	Regions 高级使用.....	343
10.1.10	Paths.....	344
10.2	Graphics 类.....	345
10.2.1	Graphics 概述.....	345
10.2.2	Rectangle, Size 和 Point.....	345
10.2.3	Graphics 画图.....	346
10.3	Pen,Font,Brush,Bitmap 和 Color 简介.....	350
10.3.1	Brush.....	350
10.3.2	Bitmap.....	350
10.3.3	Font 和 Color.....	350
10.3.4	Pen.....	350
10.4	图形编程.....	351
10.4.1	创建应用程序框架.....	351
10.4.2	界面设计.....	352
10.4.3	代码设计.....	359
10.4.4	程序运行结果.....	364
第11章	数据库.....	366
11.1	数据库.....	366
11.1.1	数据库简介.....	366
11.1.2	数据库的体系结构.....	366
11.1.3	数据库的操作过程.....	367
11.2	数据库世界的世界语——SQL.....	368
11.2.1	数据定义语言 (DDL).....	369
11.2.2	数据控制语言 (DCL).....	373
11.2.3	数据处理语言 (DML).....	374
11.3	ADO 编程.....	382
11.3.1	基本的 ADO 编程模型.....	382
11.3.2	使用 ADO 的对象实现 ADO 的编程模型.....	383
11.4	ADO 数据库编程开发应用实例.....	397
11.4.1	设计应用程序.....	398
11.4.2	编译执行.....	420

11.5 本章小结	421	13.1.1 线程与进程	463
第 12 章 网络	423	13.1.2 线程是如何工作的	464
12.1 网络基本知识	423	13.1.3 线程的优点	464
12.1.1 网络发展的历史	423	13.2 工作线程	464
12.1.2 网络参考模型	424	13.2.1 Windows 的线程实现函数	465
12.1.3 Client / Server 编程模式	426	13.2.2 启动线程	465
12.2 VS.NET 的 Socket	427	13.3 多线程编程的常用类	468
12.2.1 建立 Socket	427	13.3.1 线程同步	468
12.2.2 连接一个 Socket	428	13.3.2 Thread 类	469
12.2.3 使用 Socket 类	429	13.3.3 Mutex 类	473
12.2.4 Socket 调用过程	431	13.3.4 AutoResetEvent 和 ManualResetEvent	474
12.3 Socket 类	432	13.3.5 管道 (Monitor)	475
12.3.1 Socket 构造函数	432	13.3.6 Interlocked	476
12.3.2 绑定地址 Bind ()	432	13.3.7 ThreadPool	476
12.3.3 建立连接 Connect ()	433	13.3.8 ReaderWriterLock 类	477
12.3.4 监听服务请求 Listen ()	433	13.4 多线程编程	477
12.3.5 发送数据 Send () 和 SendTo ()	434	13.4.1 界面设计	477
12.3.6 接收数据 Receive () 和 ReceiveFrom ()	436	13.4.2 设计几个线程函数	480
12.3.7 关闭 Close ()	438	13.4.3 增加 Mutex 类	482
12.3.8 Select ()	438	13.4.4 增加 Thread 类	482
12.4 网络上的数据表示	439	13.4.5 启动线程	483
12.5 Socket 编程实例	444	13.4.6 关闭线程	484
12.5.1 创建新项目	444	13.4.7 线程清除工作	484
12.5.2 添加命名空间	445	13.4.8 程序运行	485
12.5.3 设计几个辅助函数	445	13.5 本章小结	486
12.5.4 添加控件	449	13.6 复习题	486
12.5.5 接收和发送数据	451	第 14 章 应用配置和组件	487
12.5.6 退出程序	453	14.1 条件编译	487
12.5.7 执行程序	454	14.1.1 预处理器的用法	487
12.6 TCP/IP 网络应用	454	14.1.2 条件 (Conditional) 属性	490
12.6.1 获取本机名和 IP 地址	454	14.2 XML 中的文档注释	492
12.6.2 TCP 网络时间服务应用	455	14.2.1 描述一个成员	492
12.6.3 UDP 组播网络时间 服务应用	459	14.2.2 添加备注和列表	494
12.7 本章小结	461	14.2.3 提供例子	497
12.8 复习题	461	14.2.4 描述参数	499
第 13 章 多线程	463	14.2.5 描述属性	501
13.1 线程基本知识	463	14.2.6 编译文档	502
		14.3 代码版本化	503
		14.3.1 .NET 组件的版本号	503

14.3.2 私有.NET 组件.....	503	15.1.5 添加可执行程序到一个进程....	513
14.3.3 共享.NET 组件.....	504	15.1.6 修改当前变量.....	514
14.4 设计组件.....	504	15.1.7 处理异常事件.....	514
14.4.1 设计一个用 Visual C# 编写的组件.....	504	15.1.8 用 JIT 调试 C#程序.....	515
14.4.2 设计一个用 VB 写的组件.....	507	15.1.9 调试 C#组件.....	515
14.5 组件的调用.....	507	15.2 中间语言分解器的使用.....	515
14.6 本章小结.....	510	15.3 .NET 与 COM 互操作性.....	516
第 15 章 代码调试和安全	511	15.3.1 在 COM 中使用.NET 组件.....	516
15.1 C#代码调试任务.....	511	15.3.2 在.NET 中访问 COM 对象.....	521
15.1.1 创建一个调试版本.....	511	15.4 平台请求服务.....	523
15.1.2 选择可执行程序.....	512	15.5 C#支持的不安全代码.....	524
15.1.3 设置中断点的 4 种方法.....	512	15.6 代码访问安全机制.....	525
15.1.4 逐句测试 C#程序.....	513	15.6.1 代码类型安全的认定.....	526
		15.6.2 标准许可与身份许可.....	526

第 1 章 .NET 与 C#

C#读作 C sharp,它是微软公司针对其 Microsoft.NET 战略而开发的一种新的编程语言,微软公司于 2000 年 6 月正式发布 C#语言。微软对 C#语言的描述是:“C#是一种类型安全的、现代的、简单的、从 C 和 C++语言发展出来的、面向对象的编程语言,它牢牢根植于 C 和 C++语言之上,并且可以立即为广大的 C 和 C++程序员所熟悉。开发 C#的目的就是要使其同时具备 Visual Basic 语言的高生产率和 C++语言强大的控制能力。”

微软之所以提出 C#,是有着其深厚的战略背景的,即 Microsoft.NET 战略。C#只是.NET 战略中旗帜性的开发语言,C#不能够脱离.NET,因此我们有必要先了解一些有关.NET 的知识。

1.1 .NET 简介

1.1.1 什么是 .NET

2000 年 6 月,微软公司正式推出了“Microsoft.NET 下一代互联网软件和服务战略”,同年 9 月,微软公司的首席执行官史蒂夫·巴尔默(Steve Ballmer)来到中国,参加了.NET 在中国的新闻发布会,并就“下一代互联网”专题进行了演讲。同年 11 月,微软又在 COMDEX 计算机展上发布了 Visual Studio.NET beta1 测试版软件,并展示了其.NET 发展战略的框架体系和开发工具的相关特性,全面加速了微软以.NET 技术进军市场的步伐。那么到底什么是.NET 战略,微软为什么要在此时提出.NET 战略呢?

.NET 不是微软的一个新产品,而是微软设计的新一代互联网战略。在微软的“Microsoft.NET 让新一代因特网成为现实”的白皮书中,是这样定义.NET 的:“微软公司正在创造新一代的高级电脑软件,它革命性地综合了计算与通信技术,为开发商提供了工具,致力于网络和其他技术的变革。我们把这项创新称为 Microsoft.NET,它第一次使开发商、企业和消费者能够按照自己的意愿来运用科学技术。Microsoft.NET 将创造出真正的分布式网络服务,它能够整合并协调为数众多的辅助程序,为消费者提供独到的服务,而这正是今天的网络所梦寐以求的。Microsoft.NET 将推动新一代因特网的发展,它将真正地让人们可以在任何时间(an anytime)、任何地方(anywhere)、通过任何设备(any device)得到信息。”

这段描述似乎还是很难让人对.NET 有一个更具体的概念。那么到底什么是.NET? 微软公司的首席执行官巴尔默的描述简单扼要的表述了.NET 的特征:“Microsoft.NET 代表了一个集合、一个环境、一个可以作为平台支持下一代 Internet 的可编程结构。它是一个支持环境,与 Windows 一样,.NET 也是一个位于客户机上的用户环境和基础用户服务集合。”

.NET 首先是一个环境。这是一个理想化的未来互联网环境,微软的构想是一个“不再关注单个网站、单个设备与因特网相连的互联网环境,而是要让所有的计算机群、相关设备和服务商协同工作”的网络计算环境。简而言之,互联网提供的服务要能够完成更程度的自动化处理。未来的互联网,应该以一个整体服务的形式展现在最终用户面前,用户只需要知道自己想要什么,而不需要一步步地在网上搜索、操作来达到自己的目的。这是

一种理想，但的确确实是互联网的发展趋势所在。

.NET 谋求的是一种理想的互联网环境。而要搭建这样一种互联网环境，首先需要解决的问题是针对现有因特网的缺陷，来设计和创造一种下一代 Internet 结构。这种结构不是物理网络层次上的拓扑结构，而是面向软件和应用层次的一种有别于浏览器只能静态浏览的可编程 Internet 软件结构。因此.NET 把自己定位为可以作为平台支持下一代 Internet 的可编程结构。

.NET 的最终目的就是让用户在任何地方、任何时间，以及利用任何设备都能访问他们所需要的信息、文件和程序。而用户不需要知道这些东西存在什么地方，甚至连如何获得等具体细节都不知道。他们只需发出请求，然后只管接收就是了，而所有后台的复杂性是完全屏蔽起来的。所以对于企业的 IT 人员来说，他们也不需要管理复杂的平台以及各种分布应用之间的工作是如何协调的。

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环境。微软的宏伟目标就是让.NET 彻底改变软件的开发方式、发行方式、使用方式等，并且不止是针对微软一家，而是面向所有开发商和运营商。.NET 的核心内容就是要搭建第三代互联网平台，这个网络平台将解决网站之间的协作问题，从而最大限度地获取信息。

1.1.2 为什么要提出 .NET

微软为什么要提出.NET 战略呢？主要有以下两方面的原因：

- (1) 微软长期发展战略需要.NET。
- (2) 互联网的发展需要.NET。

在过去的 20 多年中，互联网技术与应用得到了巨大的发展并创造了一个又一个网络神话。尽管互联网的发展还存在着很多问题，“网络泡沫”的破灭让人们以更加谨慎和务实的态度审视网络发展的未来。但不能否认的事实是，网络已经改变了人们的生活，并且在将来更会进一步的融入到生活中的方方面面，彻底改变人们的生活方式。

互联网的发展蕴藏着巨大的商机，围绕着互联网商业和技术的竞争从来就没有停止过。1995 年，微软的主要竞争对手 SUN 公司正式推出了面向对象的开发语言 Java，并提出了跨平台、跨语言的概念，即 *write the code once and run it anywhere*，这一概念顺应了互联网的技术发展需要，使得 Java 语言逐渐成为了企业级应用系统开发的首选工具；并且越来越多的 C/C++ 程序员也开始从事 Java 的应用开发。据研究机构 Evans Data 公司最近公布的调查结果显示，Java 将在 2002 超过 C/C++ 成为全球软件开发人员的首选语言。同时 Java 语言所体现的先进思想，也为 SUN 在软件和互联网领域赢得了巨大的成功。虽然，微软随后也推出强大的开发工具 Visual J++，但由于它主要是运用在 Windows 平台的系统开发中，SUN 公司认为 Visual J++ 违反了 Java 的许可协议，即违反了 Java 的平台中立性，为此对微软提起了诉讼，使得微软处于非常被动的地位。

微软在互联网这一领域曾经落后过，但它一直在奋起直追，互联网所带来的巨大的商机和市场、激烈的业界竞争和不断的官司缠身，这一切都使得微软必须另辟蹊径提出一个崭新的战略和概念，重新树立自己在软件业的信心和地位，因此微软提出其进军互联网的 .NET 战略。微软.NET 是一项非常庞大的计划，也将是微软今后几年发展的战略核心。

当今的互联网技术虽已日趋成熟，但微软认为它仍然存在着巨大的改进空间。今天的



互联网在很大程度上还在模仿旧式的大型计算机的工作方式。尽管有着充足的带宽资源，大量的信息还是被“锁”在了中央数据库里。用户必须依靠网络服务器才能完成所有的上网操作，这酷似老式的分时复用系统。网站则看起来像是一个一个孤立的数字小岛 (digital island)，并不能真正的按照用户的指令在它们之间进行有意义的交流。今天的网络还只能通过单个的网站向单个的用户提供有限的服务，因为大多数的网页只能呈现 HTML 格式的数据“图片”，而非信息内容本身。浏览器在很多方面都只是一个被美化的“哑巴只读终端”，你可以用它来浏览信息，但是很难进行信息的编辑、分析和复制。

也正是这些互联网的不足，使得微软提出了 .NET 来解决这些问题。 .NET 的基本理念就是针对这些不足的，它不再关心单个的网站和与互联网连接的单个设备，而是要让所有的计算机群、相关设备和服务商协同工作，提供更加广泛和丰富的解决方案。人们将能够控制何种信息、在何时、以何种方式传送给自己。计算机群、相关设备和服务商能够相互协作，提高更加多样的服务，而不是像现在一样的一座座数字孤岛，由用户来提供整合。

.NET 将推动互联网的一次变革，HTML 格式的表述将被升格为 XML 格式的可编程的信息。XML 是由万维网联盟 (W3C) 制定的，并且得到业界广泛支持的标准，这个联盟也制定了网络浏览器的标准。XML 是将真实的数据从它的视觉表象中分离出来的一种方法。它是新一代互联网的一项关键技术，也是 .NET 的核心，它把信息“解放”出来，让信息可以被重新组织、编程和编辑；它同时也是一种新的思路，将数据信息以更加有用的形式传送给各种数字设备，它让网站之间相互协作，从而可以提供一系列的互动的新型网络服务。

1.1.3 .NET 能带来什么

.NET 战略将带来一场互联网和软件的革命，它将把计算和通信带入到一个丰富、合作和互动的环境中，远远胜过今天的单向网络。由新型高级软件支持的 .NET 将利用一系列的应用程序、服务程序和相关设备来创建一种个性化的数字体验，它能够不断地按照你本人、你的家人和公司的需要进行自动的调整。它意味着你将能够通过一整套新型软件构成的整体服务程序来管理自己在互联网时代的生活和工作。

对于消费者来说，.NET 意味着简单化的整体服务；统一的信息浏览、编辑和授权；整体的系统方案；随时随地的个性化；例如：对于你个人信息的任何修改，无论是通过个人电脑、便携机还是灵通卡，都可以及时和自动的通知到所有需要这些信息的地方。

对于知识工作者和企业，.NET 意味着统一的信息浏览、编辑和授权；丰富的同步传播；密切的移动通信联系；得力的信息管理和电子商务工具，在基于内部网和因特网的服务程序之间灵活地切换，为动态商务伙伴关系的时代提供支持。

对于独立的软件开发商，.NET 意味着他们将得到更多的机会，为互联网时代创造更多的新型高级服务，它们可以自动从本地或远程取得和利用所需信息，并通过任何语言和任何设备工作，无须为不同的工作环境重新编写程序。

伴随着 ASP 产业的兴起，软件正逐渐从产品形式向服务形式转化，即软件变服务 (Software as Services)，这是整个 IT 行业的大势所趋。在 .NET 中，最终的软件应用是以 Web 服务的形式出现并在 Internet 上发布的。微软的 .NET 战略意味着：微软公司以及在微软平台上的开发者将会制造服务，而不是制造软件。在未来几年之内，微软将陆续发布有关 .NET 的平台和工具，用于在因特网上开发 Web 服务。那时，工作在 .NET 上的用户、开发人员



和 IT 工作人员都不再购买软件、安装软件和维护软件。取而代之的是，他们将定制服务，软件会自动安装，所有的维护和升级也会通过互联网进行。史蒂夫·巴尔默在谈到软件服务时说：“今天的软件产品仅仅是一张光盘，用户购买软件，亲自安装、管理和维护。但是软件服务是来自因特网的服务，它替用户安装、更新和跟踪这些软件，并让它们和用户一同在不同的机器间漫游。它为用户存储自己的信息和参考资料。这些就是软件和软件服务各自不同的风格。”

总之，.NET 将使互联网能为用户提供更加完善、人性化的服务，为广大的开发人员提供更加高效、丰富的开发工具和手段。

1.1.4 .NET 的组成

.NET 主要包含以下的组成部分（如图 1-1 和表 1-1 所示）。

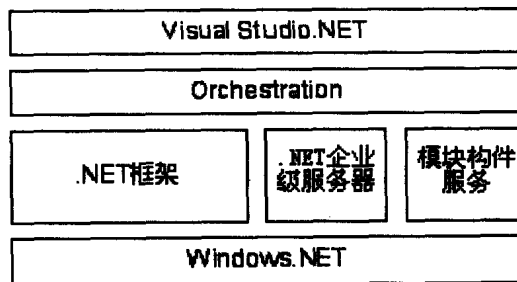


图 1-1 .NET 平台示意图

(1) .NET 平台：这一平台建立在 XML、SOAP、HTTP 和 UDDI 等因特网标准协议基础上，包含 .NET 基础结构和工具，为开发新型的互动协作软件提供了一个先进的体系结构模型。

(2) .NET 系列产品和服务：包括 Windows.NET，MSN.NET，Office.NET，Visual Studio.NET，bCentral.NET 等；

(3) 第三方 .NET 服务：众多微软的商务伙伴和第三方开发商将有机会制造基于 .NET 平台的企业软件和垂直型服务程序。

表 1-1 .NET 的组成部分

.NET 平台组件	说明
Windows.NET	是融入 .NET 技术的 Windows，它将紧密地整合了 .NET 的一系列核心构造模块，为数字媒体及应用间协同工作提供支持，是微软公司的下一代 Windows 桌面平台
.NET 框架	目的是便于开发商更容易地建立网络应用程序和 Web 服务，它的关键特色是提供了一个多语言组件开发和执行的环境
.NET 企业级服务器	在微软宣称的“第 3 代互联网”中，.NET 企业服务器是企业集成和管理所有基于 Web 的各种应用的基础，它提供企业未来开展电子商务的高可靠性、高性能、高可伸缩性以及高可管理性

续表

.NET 平台组件	说 明
模块构件服务	.NET 平台中的核心网络服务集合,它主要包括: Internet XML 通信、Internet XML 数据空间、Internet 动态更新、Internet 身份认证、Internet 目录服务和 Internet 即时信息传递等服务
Orchestration	一种基于 XML 的面向应用的软件集成和自动化处理技术。它的目标是尽量不受时间、组织、应用及个人的限制,最大程度和最好地把集成技术和自动处理技术接合起来,以便商业事务能够交互、动态、可靠地进行下去
Visual Studio.NET	Visual Studio.NET 是基于 XML 的编程工具和环境,它便于快速开发符合 .NET 体系的软件服务,使其在独立设备、企业数据中心和因特网之间的传送更加容易

1.2 .NET 的技术核心

.NET 有三大技术核心:

- XML 语言
- Web 服务
- .NET 框架

XML 是 W3C 组织制定的网络标准语言,相比 HTML 语言它更加规范,更容易扩展,是未来信息浏览和交换的统一标准,它的巨大潜能使得微软将其选择为 .NET 平台的基础。Web 服务是微软提出的重要概念,未来的许多软件和网站提供的服务项目都会作为 Web 服务的一部分,Web 服务可以被方便的调用和集成,软件智能代理也将得到极大的发展,Web 服务之间以 XML 格式进行交换。应用软件的开发将进入一个新的时代,作为主要软件开发工具提供商和 .NET 的核心部分,微软提供了新一代开发平台——.NET 框架。.NET 框架是开发 Web 服务的主要工具,它提供了一套统一的类库和通用语言运行环境,并创建了跨编程语言的通用 API,开发人员无论使用任何语言都可以开发 .NET 的应用服务。

1.2.1 XML 语言

Internet 提供了全球范围的网络互联与通信功能,Web 技术的发展更是一日千里,其丰富的信息资源给人们的学习生活带来了极大的便利。特别是 HTML(超文本标识语言)的产生,以其简单易学、灵活通用的特点,使人们发布信息、检索信息、交流信息都变得非常简单,因此,Web 成了最大的环球信息资源库。然而,HTML 是专门为描述主页的表现形式而设计的,它疏于对信息的寓意及其内部结构的描述,因此它无法适应于电子商务、电子出版、远程教育等基于互联网的更复杂、更多样化的要求。同时,HTML 的形式描述能力也是很不够的,它无法描述矢量图形、科技符号和一些特殊效果显示。更重要的是 HTML 是 SGML(Standard Generalized Markup Language,标准通用标记语言)的一个实例化的子集,其可扩展性差,用户根本不能自定义标记。这一切都成为了 Web 技术进一步发展的障碍。

为了弥补 HTML 的不足,W3C 组织(互联网联合组织)与 1998 年 2 月发布了 XML(eXtensible Markup Language,可扩展标记语言),它克服了 HTML 的诸多缺点,具有高度



的灵活性，因此业界一致认为它将成为网络信息交换的通用语言。XML 是 SGML 的一个简化子集，将 SGML 的丰富功能与 HTML 的易用性结合到 Web 的应用中。XML 以一种开放的自我描述的方式定义了数据结构，在描述数据内容的同时能够突出对结构的描述，从而体现出数据之间的关系，这样所组织的数据对于应用程序和人类都是友好的、可操作的。

XML 的优势之一是它允许各个组织、个人建立适合他们自己需要的标记集合，并且，这些标记可以迅速地投入使用，这一特征使得 XML 可以在电子商务、政府文档、司法、出版、CAD/CAM、保险机构、厂商和中介组织信息交换等领域中一展身手，根据不同的系统、厂商提供各具特色的独立解决方案。

XML 的最大优点在于它的数据存储格式不受显示格式的制约。一般来说，一篇文档包括 3 个要素：数据、结构以及显示方式。对于 HTML 来说，显示方式内嵌在数据中，这样在创建文本时，要时时考虑输出格式，如果因为需求不同而需要对同样的内容进行不同风格的显示时，要从头重新创建一个全新的文档，重复工作量过于繁重。再者 HTML 语言缺乏对数据结构的描述，对于应用程序理解文档内容，抽取语义信息都有诸多不便。XML 把文档的三要素独立开来，分别处理。首先把显示格式从数据内容中独立出来，保存在样式单 (Style Sheet) 文件中，这样如果需要改变文档的显示方式，只要修改样式单文件就足够了。另外对数据结构的详细描述，使得基于 XML 的应用程序可以在 XML 文件中准确高效地搜索相关的数据内容，忽略其他不相关的部分。另外，XML 的自我描述性质也使得许多复杂的数据关系得到良好的表现。XML 还有其他许多优点，比如它有利于不同系统之间的信息交流，完全可以充当网际语言，并有希望成为数据和文档交换的标准机制。

正是由于 XML 语言出色的数据表示能力，.NET 技术以 XML 作为数据的表示基础，使用 HTTP、TCP/IP 等标准的网络协议完成底层的传输，通过 SOAP 说明了 XML 在系统间交换信息的方法，通过 WSDL 等服务描述语言来描述和记录 Web 服务所产生和接收的消息，通过 UDDI 来登记和寻找 Web 服务。XML 与 Web 服务紧密的结合，形成了 XML Web 服务模型。

1.2.2 Web 服务

.NET 技术是围绕 Web 服务展开的。要理解 .NET，我们必须先接受 Web 服务这个概念，Web 服务已被作为一个标准提交给 W3C 组织。

1. 应用程序开发技术的革新

在过去，开发人员一直通过集成本地系统服务来构建应用程序。在这种模式下，开发人员可以访问丰富的开发资源并能严格控制应用程序的行为。现在，开发人员在很大程度上已经超越这种模式，致力于构建具有复杂结构的 n 层系统，这种系统能将网络中各处的众多的应用程序进行集成，并大大提升应用程序的价值。这样，开发人员便可集中精力挖掘软件独特的商业价值，而不必日夜为如何构建基本结构伤脑筋了。令人欣喜的局面将出现：软件投放市场的时间大大缩短、开发人员的编程效率明显提高，最为根本的是开发出质量上乘的软件。

我们正在进入一个崭新的计算时代，一个互联网时代，其核心技术是 XML。XML 创建出可供任何人从任何地方访问和使用的功能强大的应用程序。它极大地扩展了应用程序

