

 万水计算机技术实用大全系列

Visual C#

中文版

全面剖析

云巅工作室 编著

① 范例众多，内容完整，重点详尽！



中国水利水电出版社
www.waterpub.com.cn

万水计算机技术实用大全系列

Visual C#中文版全面剖析

云巅工作室 编著

中国水利水电出版社

内 容 提 要

本书全面介绍了微软最新推出的编程语言 Visual C#。全书共分 23 章。第 1 章简介 C# 的特点与 Visual Studio .NET 集成开发环境。第 2 章详细讲解 C# 的基础知识, 包括简单类型、运算符、变量等。第 3 章至第 5 章全面剖析了 C# 中的类及其方法成员。这是运用 C# 进行编程的基础。第 6 章和第 7 章讨论了 C# 中的流程控制功能与相关循环语句, 并简介了递归基本知识。第 8 章详细描述 C# 的数组及其应用。第 9 章介绍了 C# 并不提倡使用、但仍作了保留的指针。第 10 章至第 11 章介绍了 C# 面向编程的几个重要概念, 包括继承和多态。接下来的第 12 章至第 20 章分别介绍了 C# 的一些高级功能和结构, 如垃圾回收、名字空间、异常处理、委托和事件、线程处理、预处理器、XML 文档以及 I/O 知识。最后在第 21 章至第 22 章介绍了 Visual C# 的两个重要应用: 开发 Windows 应用程序与 ASP.NET。本书在第 23 章介绍了 Visual C# 应用程序的部署方法。

本书讲解全面、重点突出, 适合于学习 C# 的初中级读者。

图书在版编目 (CIP) 数据

Visual C# 中文版全面剖析 / 云巅工作室编著. —北京: 中国水利水电出版社, 2003

(万水计算机技术实用大全系列)

ISBN 7-5084-1438-1

I. V… II. 云… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 018055 号

书 名	Visual C#中文版全面剖析
作 者	云巅工作室 编著
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (发行部)、82562819 (万水)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京北医印刷厂
规 格	787×1092 毫米 16 开本 28.5 印张 639 千字
版 次	2003 年 4 月第一版 2003 年 4 月北京第一次印刷
印 数	0001—5000 册
定 价	45.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社发行部负责调换

版权所有·侵权必究

前 言

C#是微软最新推出的一种高级开发语言，是实施.NET 战略的重要工具之一。它在保留 C/C++强大功能的同时，摒弃了其中的各种不安全因素，而且还通过 Visual Studio .NET 集成开发环境为 C#提供了像 Visual Basic 一样的高效率。程序员可以采取两种方式开发、编译运行 C#程序：

- 利用.NET 框架 SDK（软件工具开发包）。.NET 框架 SDK 包含了编译、运行、测试 C#程序的所有资源。.NET 框架 SDK 可以免费从微软站点下载。由于它没有提供源代码编辑器，所以用户必须选择适合自己的文本编辑器（如 Windows 自带的记事本）来编辑源代码。本书的附录 C 提供了编译时可用的选项。
- 利用 Visual Studio .NET 集成开发环境。这是一个功能强大的集成开发环境。通过各种窗口、面板、菜单以及智能感知功能可以轻松完成 C#应用程序的编辑、编译、调试，甚至部署（由于 Visual Studio .NET 集成开发环境是多种语言的共用开发环境，所以并不只限于编译和调用 C#应用程序）。本书基于 Visual Studio .NET 集成开发环境讲解 Visual C#（简称为 C#）。

本书面向 C#的初中级读者，初级读者通过本书可以轻松步入 C#编程的殿堂，有一定经验的 C#程序员（或熟练的 C/C++程序员）可以从中学到一些高级编程技巧，如垃圾回收、编写事件驱动程序。同时，为了便于 C/C++程序员转向 C#编程，我们在讲解中进行了必要的语言间的语法比较，附录 A 和附录 B 还列表对比了各种语言间的关键字和运算符。

组织结构

全书共分为 23 章和 3 个附录，全面剖析 C#的编程技巧。对大部分知识点还提供了供读者实践时参考的实例。

第 1 章：C#与 Visual Studio .NET 概览。本章介绍了 C#的诞生背景，同时介绍了功能强大的 Visual Studio .NET 集成化开发环境。

第 2 章：C#程序设计初步。介绍 C#的关键字、运算符、语句、简单类型以及变量等基础知识。学习完本章，读者就会对 C#有一个基本的认识，并可以编写简单的 C#程序了。

第 3 章：类基础知识。类是 C#程序中相当重要的结构，掌握类是编写强大 C#程序的第一步。本章介绍类及其成员。

第 4 章：方法成员详解。本章着重讨论类的方法成员。方法是实现程序员目的的逻辑代码所在地。C#中类的方法为读者提供了解决各种问题的途径。

第 5 章：构建完整类。讲解了类的其他成员，如属性、索引器，还介绍了构造函数、运算符重载等重要知识。通过第 4、5 和 6 章的讲解，读者就可以轻松创建完整而有效的类。

第 6 章：流程控制。流程控制是任何语言不可缺少的语言元素。C#也不例外，而且

C#的控制语句比其他语言更安全，功能更强大。本章介绍了 C#的 if 语句、switch 语句、while 语句、do-while 语句以及 foreach 语句。

第 7 章：递归基础知识。本章介绍了递归的概念，并演示用递归解决两类典型的实际问题。

第 8 章：数组。数组是 C#中的重要数据结构。本章对一维数组、多维数组、矩形数组、交错数组的概念进行了详细介绍。通过大量实例演示了数组在实际编程中的应用。

第 9 章：告别指针。C#并没有完全抛弃总是在 C 或 C++中带来麻烦的指针。但必须在不安全上下文中使用指针。本章讨论了在不安全上下文中使用指针的方法。

第 10 章：继承与多态。继承和多态是面向对象编程的重要概念。本章讨论了 C#中的继承机制。

第 11 章：接口。本章介绍接口的定义、访问、实现以及转换。

第 12 章：结构类型与枚举类型。结构类型与枚举类型在有些情况下对于解决问题非常有用。本章在讲解结构、枚举类型基本概念的基础上，还与类进行了对比，并演示了它们的应用。

第 13 章：垃圾回收。这是 C#的高级功能之一。通过垃圾回收机制，程序员再也不必担心有限资源的耗尽而导致系统崩溃了。本章在对比 C++与 C#资源管理机制的基础上讲解垃圾回收器（GC）。

第 14 章：名字空间、编译单元和程序集。本章讨论名字空间、编译单元和程序集这三个相互关联的概念。

第 15 章：异常处理。C#提供了完备的异常处理机制。本章内容包括异常抛出、异常捕获以及编写自定义异常。

第 16 章：委托和事件。C#的委托相当于 C/C++中的函数指针，但委托更安全。本章介绍了委托的概念，更重要的是讲解了委托的重要应用——编写事件驱动程序。

第 17 章：线程处理。本章简介多线程的概念、C#的多线程创建与管理模式。

第 18 章：预处理器。本章讨论了 C#中的主要预处理器，包括 #if 与 #endif、#else、#elif、#define、#undef、#warning、#error、#line、#region 与 #endregion。

第 19 章：XML 文档及特性。编写详细的说明文档是任何成熟应用程序都必须具备的。本章讲解了根据源代码注释自动生成 XML 文档的方法。

第 20 章：I/O 基础。本章包括控制台 I/O 和文件 I/O。掌握本章所介绍的主要文件 I/O 类后，就可以自如地在程序中进行文件流的输入、输出控制了。

第 21 章：Windows 应用程序可视化设计。Visual Studio .NET 的特色之一就是可以快速构建 Windows 应用程序。本章通过实例演示了通过 Visual Studio .NET 提供的控件快速构建一个 Windows 应用程序框架。

第 22 章：用 C#开发 ASP.NET。本章简介了 ASP.NET，并讲解如何用 C#编写 ASP.NET 应用程序。

第 23 章：部署应用程序。程序的部署和发布是程序开发的最后、也是最重要的一环。本章介绍了 Visual Studio .NET 提供的用于部署应用程序的各种编辑器。最后演示了一个

应用程序的完整部署过程。

附录 A、附录 B 分别比较不同语言中的关键字和运算符。这部分内容无疑会加快其他语言程序员向 C#阵营的过渡。

附录 C: C#编译器选项。如果读者使用 .NET 框架 SDK 编译本书中的部分程序，就可以参考此表使用相应的编译命令。

本书基本涵盖了 C#常规编程的各个方面。全书内容重要突出，结构合理。

约定

- 本书中的所有代码都按下列样式排版：

```
using System;  
class Sum  
{
```

- 某些代码由于太长无法在一行中排版，所以采用了以下缩进方式：

```
Console.WriteLine ("此数组中值大于 1 的元素共有：  
{0}个",counter);
```

- 本书中可以直接编译运行的代码段均标上“程序清单”及相应编号。如：

程序清单 6-7 Sum1.cs

- 读者在学习过程中，还应特别注意本书中的“注意”、“提示”以及“技巧”语句，这些内容对于掌握 C#的编程技巧和避免编程中错误的出现大有裨益。

与我们联系

由于时间仓促，加上笔者水平有限，错误在所难免，欢迎广大读者提出宝贵意见 (webmaster@cloudcrown.net)。

作者

2002 年 12 月

目 录

前言

第 1 章 C#与 Visual Studio .NET 概览	1
1.1 C#简介.....	1
1.1.1 C#的诞生背景.....	2
1.1.2 C#的特点.....	3
1.2 Visual Studio .NET 简介.....	4
1.2.1 Visual Studio .NET 的特性.....	4
1.2.2 Visual Studio .NET 的安装.....	6
1.2.3 Visual Studio .NET 的集成开发环境.....	8
1.3 使用 Visual Studio .NET 创建 C#应用程序的大致过程.....	10
1.3.1 配置集成开发环境.....	10
1.3.2 创建新项目.....	11
1.3.3 编译与运行.....	12
1.4 小结.....	12
第 2 章 C#程序设计初步	13
2.1 第一个简单的 C#程序: Hello World!.....	13
2.1.1 按六步法编写 HelloWorld.cs.....	13
2.1.2 程序 HelloWorld.cs 源代码简短分析.....	15
2.2 面向对象编程方法简介.....	17
2.2.1 面向过程的先天缺陷与面向对象的优点.....	17
2.2.2 面向对象基础知识.....	18
2.3 Unicode.....	20
2.4 C#的标识符和关键字.....	21
2.5 简单类型和常量.....	23
2.5.1 数值类型.....	23
2.5.2 字面值.....	25
2.5.3 字符类型与字符字面值.....	26
2.5.4 布尔类型与布尔型字面值.....	27
2.6 字符串.....	28
2.6.1 字符串类型与常数.....	28
2.6.2 使用字符串.....	28
2.7 C#类型简介.....	30

2.7.1	类型划分	30
2.7.2	简单类型小结	32
2.7.3	装箱与拆箱	34
2.7.4	类型的优点	36
2.7.5	C#是一种强类型语言	36
2.8	C#的运算符和表达式	38
2.8.1	算术运算符	39
2.8.2	关系运算符	41
2.8.3	条件逻辑运算符	42
2.8.4	位运算符	43
2.8.5	赋值表达式	44
2.8.6	字符串连接	44
2.8.7	其他运算符和表达式	45
2.8.8	运算符的优先级与结合性	46
2.9	变量	48
2.9.1	变量定义	48
2.9.2	变量的赋值	49
2.9.3	变量的初始化	50
2.9.4	常量	51
2.10	语句	51
2.11	小结	52
第 3 章	类基础知识	53
3.1	类及其构成	53
3.1.1	类的定义	53
3.1.2	创建类的实例	53
3.1.3	类的继承和派生	54
3.1.4	类成员访问修饰符	54
3.2	类的数据成员	54
3.2.1	常量成员 (constant member)	55
3.2.2	变量成员	56
3.3	类的方法成员	61
3.4	类的属性成员	62
3.5	小结	66
第 4 章	方法成员详解	67
4.1	方法概述	67
4.2	使用方法	67
4.2.1	声明方法	67

4.2.2	方法返回值	68
4.2.3	调用方法	69
4.3	无参方法	69
4.4	有参方法	70
4.4.1	值参数	70
4.4.2	引用参数	71
4.4.3	输出参数	72
4.4.4	数组参数	72
4.5	方法重载	75
4.5.1	实现重载方法	76
4.5.2	调用重载方法	77
4.5.3	重载方法与隐式转换	78
4.6	静态方法	78
4.7	小结	82
第 5 章	构建完整类	83
5.1	实例构造函数与析构函数	83
5.1.1	创建和初始化类的实例	83
5.1.2	构造函数及其用途	86
5.1.3	使用实例构造函数	88
5.1.4	重载实例构造函数	93
5.1.5	实例析构函数	95
5.2	静态构造函数	97
5.3	只读成员	99
5.4	属性	100
5.4.1	属性访问函数	100
5.4.2	静态属性	102
5.5	索引器	104
5.5.1	定义索引器	104
5.5.2	调用索引器	105
5.5.3	索引器重载	107
5.6	运算符重载	107
5.6.1	用户自定义运算符重载	107
5.6.2	实现运算符重载	108
5.7	嵌套类	111
5.8	小结	112
第 6 章	流程控制	113
6.1	流程控制简介	113

6.2	if 分支语句.....	114
6.2.1	简单 if 语句.....	114
6.2.2	多分支 if-else 语句.....	116
6.2.3	嵌套 if 语句.....	118
6.3	switch 语句.....	119
6.3.1	switch 语句的基本结构.....	119
6.3.2	使用 switch 语句.....	120
6.4	while 循环语句.....	123
6.5	do-while 循环语句.....	125
6.6	for 循环语句.....	126
6.7	foreach 语句.....	129
6.8	跳转语句.....	131
6.8.1	break 语句.....	131
6.8.2	continue 语句.....	131
6.8.3	goto 语句.....	132
6.9	小结.....	133
第 7 章	递归基础知识.....	134
7.1	什么是递归.....	134
7.2	递归和迭代.....	135
7.3	递归应用.....	137
7.4	小结.....	139
第 8 章	数组.....	140
8.1	数组概述.....	140
8.2	一维数组.....	140
8.2.1	定义一维数组.....	140
8.2.2	初始化一维数组.....	142
8.2.3	访问一维数组.....	143
8.3	多维数组.....	144
8.3.1	定义二维数组.....	144
8.3.2	初始化二维数组.....	146
8.3.3	访问二维数组.....	147
8.4	交错数组.....	149
8.4.1	交错数组的定义.....	149
8.4.2	初始化交错数组.....	150
8.4.3	访问交错数组.....	151
8.5	数组与方法.....	153
8.5.1	数组元素作为方法中的参数.....	153

8.5.2	数组引用作为方法的参数	155
8.5.3	数组作为类中的实例变量	159
8.6	数组的排序	165
8.6.1	冒泡排序法	165
8.6.2	快速排序法	166
8.7	数组检索	167
8.7.1	数组顺序检索	167
8.7.2	数组二分检索	169
8.7.3	System.Array 的 IndexOf 方法检索	172
8.8	数组应用	174
8.8.1	矩阵元素求和	174
8.8.2	矩阵加法	176
8.8.3	矩阵乘法	179
8.9	小结	181
第 9 章	告别指针	183
9.1	不安全上下文与 unsafe 关键字	184
9.2	fixed 关键字	186
9.3	C#中的指针类型	188
9.4	指针运算	189
9.4.1	指针间接寻址	189
9.4.2	++/--以及+/-运算	190
9.4.3	stackalloc 关键字	191
9.5	指针类型转换	192
9.6	访问指针成员	192
9.7	小结	193
第 10 章	继承与多态	194
10.1	C#的继承机制	194
10.1.1	继承基础知识	194
10.1.2	访问与隐藏基类成员	200
10.1.3	多级继承	206
10.1.4	继承与访问修饰符	208
10.2	接口继承	213
10.3	控件继承	214
10.3.1	Windows 窗体继承	214
10.3.2	Control 类继承	216
10.3.3	UserControl 类继承	220
10.3.4	继承选择器对话框	223

10.4	多态性	224
10.4.1	什么是多态性	224
10.4.2	实现多态	225
10.4.3	虚方法	227
10.4.4	接口多态性	231
10.4.5	继承多态性	232
10.5	小结	233
第 11 章	接口	234
11.1	接口概述	234
11.1.1	接口	234
11.1.2	接口与组件	235
11.1.3	组件化程序设计	236
11.2	定义接口	236
11.3	定义接口成员	238
11.4	访问接口	240
11.4.1	对接口成员的访问	240
11.4.2	类对接口的实现	242
11.5	实现接口	244
11.5.1	显式实现接口成员	244
11.5.2	继承接口实现	247
11.5.3	重新实现接口	249
11.5.4	映射接口	251
11.6	接口转换	254
11.7	覆盖虚接口	260
11.8	小结	261
第 12 章	结构类型与枚举类型	262
12.1	结构类型	262
12.1.1	结构类型概述	262
12.1.2	结构声明	262
12.1.3	结构变量	264
12.2	结构和其他编程元素	267
12.2.1	结构和数组	267
12.2.2	结构和对象	268
12.2.3	结构和过程	272
12.2.4	结构中的结构	278
12.3	结构和类	280
12.4	枚举类型	282

12.4.1	声明枚举类型	282
12.4.2	枚举成员的赋值	284
12.4.3	枚举成员的访问	285
12.4.4	System.Enum 类	287
12.5	枚举类型应用	289
12.6	小结	294
第 13 章	垃圾回收	295
13.1	垃圾回收概述	295
13.2	自动动态内存管理机制	296
13.2.1	对象的废弃	296
13.2.2	GC 类	298
13.3	释放非内存有限资源	299
13.4	小结	303
第 14 章	名字空间、编译单元和程序集	304
14.1	名字空间	304
14.1.1	自定义名字空间	305
14.1.2	.NET 框架类库名字空间	306
14.1.3	使用名字空间的优点	308
14.2	using 指令	308
14.2.1	创建名字空间或类型的别名	308
14.2.2	导入名字空间中的类型	310
14.3	名字空间、编译单元与程序集	312
14.3.1	重用集合中的名字空间	312
14.3.2	生成多编译单元程序集	313
14.4	剖析程序集的内容	314
14.5	小结	316
第 15 章	异常处理	317
15.1	异常处理概览	317
15.2	.NET Framework 中的异常类	320
15.2.1	Exception 异常类	320
15.2.2	常用异常类	323
15.3	异常捕获	324
15.3.1	try 和 catch 块概述	324
15.3.2	使用 try-catch 捕获异常	328
15.3.3	嵌套 try 块	330
15.3.4	多重 catch 块	332
15.4	异常抛出	333

15.4.1	用 throw 显式抛出异常	333
15.4.2	使用 finally 语句抛出异常	335
15.5	编写自定义异常	336
15.6	小结	337
第 16 章	委托和事件	339
16.1	委托	339
16.1.1	声明委托	339
16.1.2	实例化委托	339
16.1.3	使用委托	340
16.1.4	组播委托	343
16.2	编写事件驱动程序	346
16.2.1	创建包含事件成员类	346
16.2.2	创建包含事件处理方法的类	347
16.3	小结	352
第 17 章	线程处理	354
17.1	多线程概述	354
17.2	创建多线程	355
17.2.1	创建新线程	355
17.2.2	暂停和继续线程	356
17.2.3	销毁线程	357
17.2.4	线程优先级	357
17.3	线程池	358
17.3.1	System.Threading 名字空间	358
17.3.2	ThreadPool 类	359
17.4	线程处理设计原则	361
17.5	小结	363
第 18 章	预处理器	364
18.1	预处理器概述	364
18.2	预处理器指令	365
18.2.1	#if 与 #endif	365
18.2.2	#define 与 #undef	365
18.2.3	#else	366
18.2.4	#elif	366
18.2.5	#warning	367
18.2.6	#error	368
18.2.7	#line	368
18.2.8	#region 与 #endregion	368

18.3 小结	369
第 19 章 XML 文档及特性	370
19.1 XML 文档	370
19.1.1 文档注释格式	370
19.1.2 文档注释标记	371
19.1.3 生成 XML 文档	373
19.2 特性 (attribute)	374
19.2.1 概述	375
19.2.2 自定义特性	377
19.3 小结	379
第 20 章 I/O 基础	381
20.1 控制台 I/O	381
20.1.1 控制台输入	381
20.1.2 控制台输出	383
20.1.3 格式化字符串	385
20.2 文件 I/O	388
20.2.1 文本文件和二进制文件	388
20.2.2 Stream 类	389
20.2.3 FileStream 类	390
20.2.4 BinaryReader 与 BinaryWriter 类	392
20.2.5 StreamReader 与 StreamWriter 类	392
20.2.6 File 类	393
20.2.7 Directory 类	394
20.2.8 Path 类	396
20.3 小结	397
第 21 章 Windows 应用程序可视化设计	398
21.1 Windows 应用程序简介	398
21.2 Visual Studio .NET 提供的控件	398
21.3 设置应用程序可视化界面	400
21.3.1 创建应用程序窗体	400
21.3.2 添加控件	401
21.3.3 控件选择与布局	404
21.4 为控件添加事件	404
21.5 小结	406
第 22 章 用 C# 开发 ASP .NET	407
22.1 ASP .NET 简介	407
22.1.1 ASP .NET 的特点	407

22.1.2	Web 窗体	407
22.1.3	编译运行 Web 应用程序	408
22.2	ASP.NET 服务器控件	409
22.2.1	简介	409
22.2.2	添加 ASP.NET 服务器控件	410
22.2.3	在 Web 窗体页中创建事件处理程序	411
22.3	创建 ASP.NET Web 应用程序示例	412
22.4	小结	413
第 23 章	部署应用程序	414
23.1	简介	414
23.1.1	Microsoft Windows 安装程序介绍	414
23.1.2	部署概念	414
23.2	部署应用程序	415
23.2.1	创建或添加部署项目	415
23.2.2	设置部署项目属性	416
23.2.3	向部署项目中添加项	416
23.2.4	添加和移除图标	417
23.2.5	从部署项目中排除项	418
23.3	部署编辑器	419
23.3.1	文件系统编辑器	419
23.3.2	注册表编辑器	419
23.3.3	文件类型编辑器	420
23.3.4	用户界面编辑器	421
23.3.5	自定义操作编辑器	422
23.3.6	启动条件编辑器	422
23.4	部署一个简单的 Windows 应用程序	423
23.5	小结	428
附录 A	不同语言中的关键字比较	429
附录 B	不同语言的运算符比较	435
附录 C	C#编译器选项	438

第 1 章 C#与 Visual Studio .NET 概览

比尔·盖茨在展望微软的技术发展时曾提出“未来 5 年，我们的目标就是超越今天各自为营的 Web 站点，把互联网建成一个可以相互交换组件的地方。”2000 年 6 月 22 日，.NET 的推出正是微软实现这一计划的关键一步，微软宣称“其意义不亚于从 DOS 转变为 Windows 的过程”。.NET 的内涵比较丰富，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。2002 年 2 月正式推出的 Visual Studio .NET 更是实现微软整个 .NET 战略的重型武器。尤其引人注目的是，在这“重型武器”之中还隐藏了另一种强大的武器——这就是最新诞生的 Visual C#（发音为“C sharp”）。(本书所讲授的大部分知识既适用于 .NET 框架 SDK 提供的 C#编译环境，也适用于 Visual Studio .NET 提供的 Visual C#编译环境。因此大部分情况下，我们只使用简称 C#，但在开发 Windows 应用程序时，则指 Visual C#。)

1.1 C#简介

要学习 C#，就不得不对 .NET 平台有所了解。这一平台不仅使现有的软件适用于 Web 时代的 PC，而且还能够满足呈强劲增长势头的许多新设备，如蜂窝电话以及个人数字助理（Personal Digital Assistant, PDA）等的需要。

请读者记住，.NET 的内涵比较广泛，但它首先是一个开发平台。.NET 统一了编程类库，提供了对下一代网络通信标准、可扩展标记语言（eXtensible Markup Language, XML）的完全支持，使得应用程序的开发变得更容易、更简单。

最重要的是，.NET 将改变互联网的行为方式，因为微软的目标是将软件变成服务。微软的 .NET 方案是对计算机和通信的一次革命，它是第一个实现“让信息无处不在”的平台，这一技术将完全改变 Web 和其他计算方面的体验。.NET 允许程序员、开发商和消费者充分享受这一技术。.NET 使得真正的分布式 Web 服务成为可能，这一服务将与广泛的辅助服务集成，通过相互间的协作帮助客户以当今 .com 时代梦寐以求的方式来运作。从广义上讲，也可将 .NET 视作一个运行库，因为 .NET 与微软的其他产品一样是与 Windows 平台紧密集成的，并且这种融合性与其他产品相比有过之而无不及。

.NET 所蕴含的基本思想是着重将个人 Web 站点或连接到互联网的设备转换成计算机、设备和服务群集，它们协同工作来提供更宽广、更丰富的解决方案。用户可以控制信息传递给它们的方式和时间。计算机、设备和服务可以相互协作提供更为丰富的服务，而不是只有用户能提供局部集成的孤立格局。厂商可以通过一种让客户无缝嵌入他们电子结构的方式来提供产品和服务。微软计划在软件中添加手写、语音识别等方面的功能，让人们可以与计算机进行更流畅的交流，并在此基础上继续扩展相关功能，增加对各种用户终端的支持能力。

微软的 .NET 使计算和通信比以前更简单、更容易。它将孵化出新一代的互联网服务，