

Visual Basic.NET 程序设计入门

李万红 梁 靓 等 编著

清华大学出版社

(京) 新登字 158 号

内 容 简 介

本书从 Visual Studio.NET 这个崭新且功能强大的开发平台开始, 全面展示了 Visual Basic.NET 的集成开发环境及其蕴含的诸多程序设计功能和程序设计理念。在此基础上, 以具体的程序设计实例介绍了基本的程序设计步骤、方法和技巧。全书共分为 11 章, 内容分别涉及到 .NET 框架的基本内容、Visual Studio.NET 的集成开发环境、程序设计的基础知识、语言和语法的新特性、面向对象的编程方法、创建程序的步骤和方法、Windows 窗体和菜单设计、各种对话框设计、使用 ADO.NET 进行数据访问、使用新的 Web 技术、程序调试和错误处理等。

本书结构清晰、内容翔实, 可以作为初中级用户学习 Visual Basic.NET 的参考书, 也可以作为大中专院校的学习教材。

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。

书 名: Visual Basic.NET 程序设计入门

作 者: 李万红 梁 靓 等 编著

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮政编码: 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 丁岭

印 刷 者: 北京市密云胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 21.25 字数: 515 千字

版 次: 2002 年 6 月第 1 版 2002 年 6 月第 1 次印刷

书 号: ISBN 7-302-05510-6/TP·3239

印 数: 0001~5000

定 价: 28.00 元

前 言

为在未来的计算机发展过程中处于有利位置，Microsoft 积极推出了自己的.NET 策略，它将 Internet 本身作为构建新一代操作系统和应用程序的基础，大大延伸了当前的网络 and 系统设计思想。这样，开发人员将能够彻底地摆脱硬件设备和非兼容环境的束缚，轻松地实现各种环境的网络互连。

作为.NET 框架中的最重要的开发语言之一，Visual Basic.NET 重新对 Visual Basic 6.0 进行了全方位的打造，大大提高了程序设计的方便性和功能性，克服了原有蹩脚的资源访问能力，使开发人员高效、快捷地开发功能强大的分布式应用程序成为可能。

经过改进的 Visual Basic.NET 被赋予了过去需要使用 C++ 这样的语言才具有的访问某些系统资源的能力。更重要的是，Visual Basic.NET 增加了对象的继承性，成为了真正的面向对象编程语言。在 Visual Basic.NET 中，还有一个重要的语言新特性是垃圾收集，它由公共语言运行时（Common Language Runtime）环境管理，提供了更好的内存管理能力。通用类型系统允许更强大的互操作性，也有助于 Visual Basic.NET 功能性和灵活性的提高。

对于 Visual Basic 开发人员来说，不仅可以拥有 Visual Basic.NET 的全新功能，而且还可以在通用语言开发环境下，借用其他.NET 开发语言和兼容开发语言来为自己的应用程序开发提供方便和实现 Visual Basic.NET 无法单独完成的功能。可以说，Visual Basic.NET 的出现，必将使 Visual Basic 开发人员如虎添翼。

如果还没有学习过 Visual Basic .NET 编程语言，那么现在是你了解它的时候了。本书的目标是：就 Visual Basic .NET 的程序设计方法和新的微软.NET 平台，为读者提供一个全面而深入的多维视角，最终为读者建立起有关 Visual Basic .NET 程序设计的思想，并掌握其中的一些重要方法。

由于 Visual Basic.NET 的开发环境比较新颖、复杂，因此特邀请多名有关专家和人员参与本书的编写工作。除了封面署名外，还有胡辰浩、薛焱、贾丽华、王琳等参与了本书的编写。鉴于 Visual Basic.NET 的变动比较大和作者的水平有限，疏漏之处在所难免，敬请读者指正。

作 者

2001 年 11 月

目 录

第 1 章 Visual Basic.NET 简介	1
1.1 了解 .NET	1
1.1.1 何谓.NET	1
1.1.2 .NET 的战略构想	2
1.1.3 .NET 对计算机用户的意义	3
1.1.4 了解“软件作为服务”的意义	4
1.2 深入了解 Visual Studio.NET	5
1.2.1 Visual Studio.NET 概述	5
1.2.2 Visual Studio.NET 的设计目标	6
1.3 Visual Basic .NET 语言的新特性	8
1.3.1 新增的面向对象编程特性	8
1.3.2 其他的现代化语言特性	10
1.4 安装 Visual Studio.NET	12
1.4.1 Visual Studio.NET 的安装需求	12
1.4.2 安装前的注意事项	12
1.4.3 Visual Basic.NET 的安装过程	13
1.5 启动与配置 Visual Studio.NET	17
1.6 关闭 Visual Studio.NET	23
第 2 章 创建第一个 Visual Basic.NET 程序	25
2.1 初步了解 Visual Basic.NET 的编程环境	25
2.1.1 窗口简介	25
2.1.2 语法说明	28
2.1.3 程序的生成与保存	29
2.2 创建第一个 Visual Basic.NET 程序	30
2.2.1 编程步骤	31
2.2.2 创建用户界面	31
2.2.3 设置窗体与控件对象的属性	33
2.2.4 编写代码	34
第 3 章 Visual Studio.NET 的集成开发环境	38
3.1 Visual Studio.NET 的起始页	38
3.1.1 环境配置文件	39

3.1.2	设置帮助选项	40
3.1.3	起始页中的其他超级链接页面	42
3.2	解决方案的项目组成	43
3.3	配置管理器与解决方案配置	43
3.4	自定义 Visual Basic.NET 的工具栏	45
3.5	Visual Basic.NET 中的菜单结构	46
3.5.1	“文件”菜单	46
3.5.2	“编辑”菜单	47
3.5.3	“视图”菜单	48
3.5.4	“项目”菜单	49
3.5.5	“生成”菜单	49
3.5.6	“调试”菜单	49
3.5.7	“格式”菜单	50
3.5.8	“工具”菜单	50
3.5.9	“窗口”菜单	51
3.5.10	“帮助”菜单	52
3.6	Visual Studio.NET 集成开发环境详解	52
3.6.1	选项卡式的窗口	52
3.6.2	调试程序过程中常用的窗口	60
3.6.3	浮动窗口	63
3.6.4	窗口的自动隐藏特性	64
3.6.5	Web 窗体与网页编辑器	64
3.7	Visual Studio 中的宏	68
3.8	XML、XSD Schemas 和 XSLT File 的可视编辑器	68
3.9	跨项目和进程的调试	69
第 4 章	Visual Basic.NET 程序设计基础	70
4.1	Visual Basic.NET 中的数据类型、常量、变量、数组定义	70
4.1.1	数据类型	70
4.1.2	常量	71
4.1.3	变量	72
4.1.4	数据类型转换函数	75
4.1.5	数组	76
4.1.6	自定义数据类型	79
4.2	Visual Basic.NET 的运算符	79
4.2.1	算术运算符	80
4.2.2	连接运算符	80

4.2.3 比较运算符	80
4.2.4 逻辑运算符	81
4.2.5 运算符的优先顺序	82
4.3 判断与循环	82
4.3.1 If...Then	82
4.3.2 If ... Then ... Else	83
4.3.3 Select Case	84
4.3.4 Do...Loop 循环	85
4.3.5 For Next 循环	87
4.3.6 While ... End While	88
4.3.7 循环的退出	88
4.4 过程与函数	89
4.4.1 事件过程	89
4.4.2 通用过程结构	90
4.4.3 函数结构	92
4.4.4 过程和函数的退出	92
第 5 章 Visual Basic.NET 语言 and 语法的新特性	94
5.1 名称空间	94
5.1.1 何谓名称空间	94
5.1.2 局部和全局名称空间	99
5.1.3 名称空间在 Visual Studio.NET 中的应用	100
5.1.4 创建名称空间	101
5.1.5 Visual Basic 中常用的名称空间	103
5.2 数据类型、变量、数组和集合的变化	103
5.2.1 数据类型的变化	103
5.2.2 变量声明的改进	109
5.2.3 数组的变化	111
5.2.4 集合的改变	112
5.3 运算符的变化	116
5.3.1 算术运算符的改进	116
5.3.2 逻辑运算符和位级运算符	117
5.4 Visual Basic 语句和语法在 .NET 中的变化	117
5.4.1 Debug.Print	117
5.4.2 If...Then 语句的变化	118
5.4.3 On...GoTo 和 On...GoSub 语句的变化	118
5.4.4 Set 和 Let 语句的取消	119

5.4.5	Option 语句.....	120
5.4.6	使用保留字作为过程名.....	122
5.4.7	Property 例程的变化.....	122
5.4.8	过程语法的改变.....	124
5.5	新的自动重新实例化.....	127
5.6	Try...catch...Finally 结构.....	128
5.7	事件处理方式的改变.....	133
5.8	失效的 Visual Basic 语法.....	135
第 6 章	面向对象的程序设计方法.....	138
6.1	何谓面向对象语言.....	138
6.1.1	面向对象语言应具备的四个特性.....	138
6.1.2	Visual Basic.NET 中全新的继承性.....	140
6.2	理解面向对象编程的基本概念.....	141
6.2.1	对象.....	141
6.2.2	属性.....	142
6.2.3	方法.....	143
6.2.4	事件.....	144
6.3	Visual Basic.NET 中对象处理的改进.....	145
6.3.1	对象声明和实例化.....	145
6.3.2	取消引用对象.....	146
6.3.3	早期和晚期捆绑.....	147
6.4	面向对象的实现.....	149
6.4.1	类和对象的关系.....	149
6.4.2	创建类.....	150
6.4.3	对象的生命周期.....	154
6.4.4	继承.....	158
6.4.5	共享和类成员.....	164
6.4.6	界面.....	167
6.5	清除内存中失效的对象.....	170
6.5.1	垃圾收集.....	171
6.5.2	使用 Finalize 方法释放资源.....	176
6.5.3	使用 Dispose 方法释放资源.....	176
第 7 章	Windows 窗体及菜单设计.....	183
7.1	在 Visual Basic.NET 程序项目中创建窗体.....	183
7.2	认识 Windows 窗体.....	185
7.2.1	Windows 窗体包含的属性.....	185

7.2.2	常用的窗体事件.....	188
7.2.3	控制窗体的显示.....	191
7.3	多窗体设计.....	192
7.4	设计 Windows 菜单.....	195
7.4.1	菜单的基本概念.....	196
7.4.2	设计菜单与设置菜单属性.....	197
7.4.3	为菜单命令添加事件.....	200
7.4.4	设计弹出式菜单.....	200
第 8 章	为应用程序设计对话框.....	203
8.1	新的对话框设计方法.....	203
8.1.1	ShowDialog 的使用.....	203
8.1.2	DialogResult 的变化.....	204
8.1.3	新的从属窗体设计方法.....	205
8.1.4	新的按钮激活设置方式.....	206
8.2	创建标准对话框.....	207
8.2.1	OpenFileDialog 控件.....	207
8.2.2	SaveFileDialog 控件.....	210
8.2.3	ColorDialog 控件.....	211
8.2.4	FontDialog 控件.....	212
8.2.5	PrintDialog 控件.....	213
8.2.6	PrintPreviewDialog 控件.....	214
8.3	创建自定义对话框.....	215
8.3.1	理解自定义对话框.....	215
8.3.2	创建一个自定义对话框.....	216
8.4	创建消息对话框.....	219
8.4.1	认识消息对话框.....	219
8.4.2	MsgBox 函数.....	219
8.4.3	MessageBox 类.....	222
8.4.4	消息对话框综合示例.....	223
8.5	创建输入对话框.....	225
8.5.1	认识输入对话框.....	225
8.5.2	输入对话框应用示例.....	226
第 9 章	使用新的 Web 技术.....	228
9.1	ASP.NET 与 Web 窗体.....	228
9.1.1	ASP.NET 的新功能.....	228
9.1.2	认识 Web 窗体.....	231

9.1.3 Web 窗体应用实例	236
9.2 服务器控件	238
9.2.1 HTML 服务器控件	238
9.2.2 Web 窗体控件	240
9.3 Web 页面验证和传递	241
9.3.1 Web 页面的验证	241
9.3.2 Web 页面的传递	244
9.4 使用 Web 服务	245
9.4.1 认识 Web 服务	245
9.4.2 SOAP 协议	246
9.4.3 创建 Web 服务	247
9.4.4 引用 Web 服务	251
9.5 使用 XML	253
9.5.1 认识 XML	254
9.5.2 Visual Basic.NET 对 XML 的支持	255
9.5.3 创建 XML 文档	256
9.5.4 创建 Schema 语法规则	258
9.5.5 创建 XSLT 样式表	261
第 10 章 使用 ADO.NET 对数据进行访问	265
10.1 认识 ADO.NET	265
10.1.1 ADO 的发展	265
10.1.2 ADO.NET 的新内容	266
10.2 基本数据访问对象	268
10.2.1 基本数据访问对象与管理支持程序	268
10.2.2 Connection 对象	269
10.2.3 Command 对象	271
10.2.4 DataAdapter 对象	274
10.2.5 DataReader 对象	276
10.3 DataSet 的基本内容	278
10.3.1 认识 DataSet	278
10.3.2 DataTable 对象	280
10.3.3 关系与约束	281
10.3.4 DataSet 与 DataReader 的选择	282
10.4 DataSet 的创建和使用	283
10.4.1 手工创建 DataSet 对象	283
10.4.2 利用数据库系统创建 DataSet 对象	287

10.4.3	浏览 DataSet 对象中的数据	290
10.4.4	使用 DataSet 对象中的数据	291
10.5	DataSet 与 XML	291
10.5.1	将数据从 DataSet 对象写入到 XML 文件	292
10.5.2	将数据从一个 XML 文件读入 DataSet 对象	293
10.6	数据的排序和筛选	294
10.6.1	DataTable 对象中数据的排序和筛选	294
10.6.2	使用 DataView 对象	297
第 11 章	出错处理及程序调试	300
11.1	程序中的错误类型	300
11.1.1	语法及编译错误	301
11.1.2	运行时期的错误	305
11.1.3	逻辑错误	306
11.2	Visual Basic.NET 中常用的调试工具	307
11.2.1	Visual Basic.NET 中的工作模式	307
11.2.2	“调试”菜单	308
11.2.3	调试工具栏	309
11.3	常用的调试方法与技巧	310
11.3.1	在中断模式下进行程序调试	310
11.3.2	运行程序的特定部分	313
11.3.3	使用调试窗口	316
11.3.4	使用 Try...Catch...Finally 语句处理错误和异常	321
11.4	联机错误处理与集中化错误处理	325
11.4.1	联机错误处理	325
11.4.2	集中错误处理	327

第1章 Visual Basic.NET简介

Visual Basic(Visual Basic)的发展从来就没有停止过，新版的 Visual Basic.NET 增加了许多新的功能，这些功能 Visual Basic 编程人员以前连想都不敢想，它们的强大使得你可以利用 Visual Basic 完成许多以前根本无法实现的功能。同时，这些变化也给开发人员带来了一些小小的麻烦，要完全掌握它们还需要一段时间学习。与目前所使用的开发工具相比，Visual Basic.NET 有许多变化，无论你是否喜欢，它都将影响到我们的未来。如果从 Visual Basic3 到 Visual Basic4 是一个飞跃的话，那么这次升级更像是换了一个新的平台，而不仅仅是一个新版的 Visual Basic。所遇到的问题很类似于从 DOS 转向 Windows 时所遇到的问题。

本章将介绍 Visual Basic.NET 的语言特性、安装过程和语言的改进，主要包括：

- .NET 的概念与构想
- 了解 Visual Studio.NET 的程序设计思想
- Visual Basic.NET 的新特性
- 如何安装 Visual Studio.NET
- 启动与退出 Visual Studio.NET

1.1 了解 .NET

微软公司提出的.NET 概念，正从各个方面渗入到我们的生活中。它将产生的作用，诚如一位业内名家所描述的：“请忘掉你认为你所知道的，.NET 将改变一切！”既然如此，作为已经或将要使用 Visual Basic.NET 开发程序的程序员来说，快速领会这个新概念的含义就显得非常必要了。本节将介绍有关.NET 概念方面的内容。

1.1.1 何谓.NET

.NET 究竟是什么，微软给它的定义是：.NET 代表了一个集合、一个环境、一个编程的基本结构，作为一个平台来支持下一代的互联网。.NET 也是一个用户环境，是一组基本的用户服务，可以作用于客户端、服务器，以及任何地方，并与该编程模式具有很好的一致性且有新的创意。因此，这不仅是一个用户体验，而且是开发人员体验的集合，这就是对.NET 的概念性的描述。

而从设计角度上看, Microsoft.NET 平台和 Windows 平台较为相似。通过该平台, 用户可以创建出更多的应用软件。Windows 让人们能接触到数以千计的第三方应用软件, 并促使个人计算机产业发生了巨大的变革。同样地, .NET 能够让遍布全球的网站产生更有价值的服务, 还能让用户以非常简便的方式从多个网站上获取信息。

综上所述, 我们不难得到.NET 在概念层次上的准确定义: .NET 是一个可以支持下一代互联网服务和运营所需要的平台。这个平台将包含微软新一代的操作系统 Windows XP 和新的办公软件 Office XP, 包含大量的互联网服务软件, 包含对各种设备(PDA、机顶盒、信息家电等)的支持, 包含微软提供给开发人员的最新应用软件开发套件 Visual Studio.NET 等。在这个平台上, 用户可以得到完整的接入服务, 开发者可以在开发方面得到有力的支持, 以创建各式各样的应用软件。这个平台可以简单地用图 1-1 来表示。也就是说, 微软将未来 IT 发展的一切都以一个名词来概括, 这个名词就是.NET。

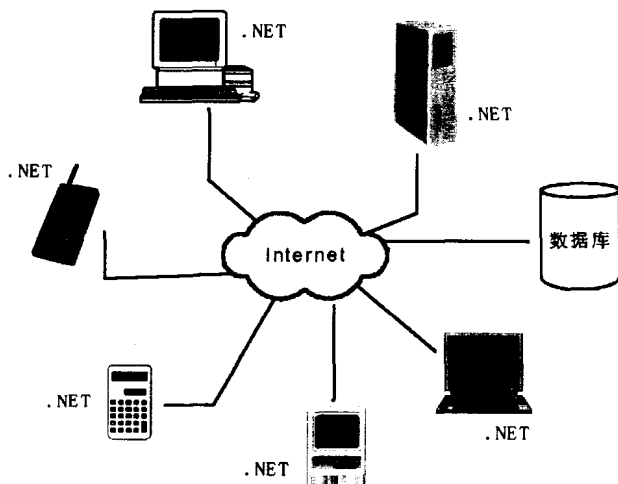


图 1-1 .NET 作为支持互联网的平台将所有类型的设备连接起来

1.1.2 .NET 的战略构想

微软声明, .NET 的核心内容就是要搭建第三代互联网平台, 同时建立这个平台上的运营规则。这个网络平台将解决网站之间的协同合作, 从而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, 网站之间可以自动交流、协同工作, 以提供最全面的服务。

微软认识到服务走向互联网是 IT 产业发展和世界经济发展的必然趋势, 所以.NET 的核心目标就是对其现有的计算模式和软件开发方式进行有针对性的调整, 使之符合第三代互联网运营的需要, 并利用其现有的科技开发实力, 同诸如 IBM 这样的厂商修改和定义网络通信协议, 建立未来在.NET 平台上的运作规则, 搭建第三代互联网平台。

微软认为.COM 是信息海洋中一个一个的数字孤岛,而将这些数字孤岛连接起来,打破不同的操作平台、不同的网络设施以及业界遗留下来的藩篱,从而实现互联网统一的解决方案就是.NET。该方案的示意图如图 1-2 所示。可以看出,微软在基本统一了 PC 操作平台之后,不但想向互联网进军,而且还要通过进一步的革新和发展来统一互联网。

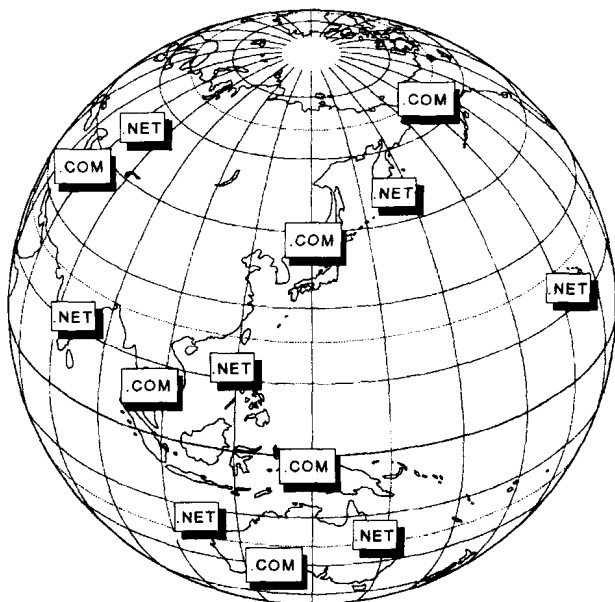


图 1-2 .NET 将.COM “数字孤岛”联系起来

1.1.3 .NET 对计算机用户的意义

从极少数专业人士才能接触到的大型计算机到装有 DOS 的 PC,使用电脑的人数已经大大增加,其原因是 DOS 操作系统的更简单易用。与此同时,计算机使用人数的增加也刺激了 PC 工业的发展,进而对 PC 操作系统的发展产生强大的推动力。从 Windows 开始装备 PC 的那一天起,使用电脑的人数呈几何级数增长,原因是 Windows 操作系统不仅提供了比 DOS 更为强大的功能,而且还提供了更为友好的人机交互界面,使人们能够比以往任何时候都可以方便地使用计算机,并将计算机看成是辅助自己完成繁琐工作的得力助手。但是,因为受限于当今软件科技的发展水平,距离“科技以人为本”、“科技服务于人”的理想标准,还有相当长的一段路要走。

.NET 对最终用户来说非常重要。以前人们必须具备相当的技术功底才能让计算机很好地为他们服务。很多计算机概念难以理解,以至于最终用户在掌握计算机使用技能时总是需要花费大量的时间和精力。但是,随着.NET 平台的诞生与发展,在.NET 网络平台上,人们可以使用键盘、鼠标、手写、语音、语言等各种方式进行人机交流。这将极大地降低人们使用网络的难度,从而为构建全真虚拟网上社会打下基础。.NET 平台提供的强大功能

可使用户轻松进行互联网链接，轻松完成那些在当今看来十分费时、费力的事务。用户可以自由访问、查看、使用他们的数据。

随着宽带技术的发展和普及以及新一代互联网技术的发展，越来越多的非 PC 设备和信息终端在提供网络服务方面扮演越来越重要的角色。例如，传统的电视将附以机顶盒，人们可以与电视节目进行交互，这将极大地丰富人们的生活。

.NET 的出现，意味着人们只用一种简单而统一的界面就可以编写、浏览、编辑和分享信息。由于 .NET 世界中的“世界语”是 XML，而作为信息交换载体的所有文件都以 XML 格式存在，所以不同规模的公司都可以使用相同的信息交换形式与供应商、伙伴和客户分享信息，这样，一种全新的协同工作模式就随着 .NET 的应用应运而生。

1.1.4 了解“软件作为服务”的意义

.NET 的设计目标之一就是让“软件作为服务”。但是对于这句话，我们应当怎样理解呢？在人们日常生活中，离不开各式各样的服务。比如，用户在当地 ISP 申请了网络使用账户，那么当网络设备连接妥当之后，就可以享受美妙的因特网服务；当观众在所属地区申请了有线电视服务，并按手续缴纳了服务费用后，就可以在闲暇之余，享受精彩的电视节目。二者都是客户通过对某些服务进行“订阅”，并按时缴纳服务费用后，得以享受相关服务的例子。其实对于“软件作为服务”的最朴素的认识与上面的例子无异，在网络上开通的各种在线杂志、金融信息等服务，就是我们身边形式最简单的软件服务。

所谓“软件作为服务”其实就是一种提供软件服务的机制。最终用户可以根据自己的需要来“订阅”应用软件的功能，并且这种“订阅”是有选择性的，他们可以在一个庞大的应用程序中选择他们所需要的那一部分。这种“订阅”机制不同于传统应用程序，它可以满足最终用户对于自有配置应用的需求。

举个例子，像 Word 这样的字处理软件，虽然功能强大，但是仍然有很多用户对其感到不满，因为 Word 中包含了太多他们根本就不需要的功能。用户都向往着能够对软件进行自由配置，因为只有这样，才可以在得到足够的功能的同时，还不必为没用的软件功能白白付费。

随着 .NET 应用的普及，软件最终可以实现在互联网上的动态安装——用户仅仅安装感兴趣的或当前工作需要的部分，或者干脆整个程序都放在网络上，这很像现在很多人使用的基于 Web 的电子邮件系统，只要连接到因特网，不论使用的是哪一台电脑或这台上网的电脑物理位置在何处，都可以自由地浏览、编辑自己的邮件。这种方式还有助于拉近软件开发者和最终用户的关系，因为一方面，用户可以从频繁的升级和补丁中获得软件的新特性，而另外一方面，软件开发者也可更好地理解用户的需求，进而开发出更受消费者欢迎的产品。

最终用户可以从 .NET 的许多方面享受到动态安装和订阅服务带给他们的好处。为了实现这些令人向往的功能，Assembly 作为 .NET 技术的一个核心概念被引入，使用 Assembly，

应用程序可以通过一个清单(manifest)来维护应用程序的组件信息。当然这不仅是一个专属因特网的新特性，它同时也是解决 DLL 文件兼容问题的基础。

“软件作为服务”的另外一个方面，就是在网络上能够公开可编程接口，并通过这种方式来共享站点的服务。目前，Web 网站的开发模式决定了站点彼此之间是相互隔离的孤岛。举个例子，如果某个 Web 站点开发者要为其站点提供信用卡验证功能的支持，那么他要么自己开发软件，要么从别人那里购买软件模块，而 .NET 采用的方式将简化他们的工作，因为相关的服务就放在网络上共享的地方，随时为需要它们的人们准备着。开发者可以直接引用这些接口来完善自己应用程序的功能。这个允许在 Web 站点之间公开可编程接口的技术叫做 Web Service，它通过将标准的 XML 语言和 SOAP(简单对象访问协议)作为信息描述和交换的手段，使“软件作为服务”不再是梦想。

1.2 深入了解 Visual Studio.NET

在安装与使用 Visual Studio.NET 之前，用户首先应对微软的 Visual Studio.NET 开发平台有一个大致的认识 and 了解，以便以后充分利用其中的功能为自己的程序开发服务。

1.2.1 Visual Studio.NET 概述

微软终于推出了 .NET 计划最重要的一环——Visual Studio.NET 开发语言。在完全继承了全世界 300 万开发人员使用习惯的基础上，Visual Basic.NET 基于 .NET 计划的网络开发功能得以大大加强。微软再一次希望用最简单的方式将自己的 .NET 理念通过 Visual Studio.NET 去影响全世界的微软开发人员。

也许成为微软程序员最大的好处是可以用最简单的方式写出最好的程序，大多数人不必牢记数量庞大的 API 函数，也可以放弃前辈程序员对系统底层的足够研究，甚至也不必对算法和数据结构有太多的了解，因为微软已经用 Visual Studio.NET 把所有基于 .NET 计划的复杂理念变得非常简单。

这一次的变化是巨大的，微软的设计构想并不是使 Visual Basic.NET 在 Visual Basic 6 的基础上增加一些新特性，微软重新设计了该产品，使其用来设计像 Web 和企业级多层系统那样的分布式应用程序更加容易。Visual Basic.NET 的新特性包括两个新的窗体包——Windows 窗体(Windows Forms)和 Web 窗体(Web Forms)；一个新版的 ADO 增加了访问离线数据源功能；改进了基本语言，去掉了以往版本遗留的一些关键字，增强了类型安全，并且应高级开发者需求揭示了函数方法实现的低级结构(类似于 Delphi)。另外 ASP.NET 作为 Visual Studio.NET 的一部分也有了很大的改进和变化。尽管以前版本的 ASP 开发十分方便，但还有许多不如人意的地方，并且只能使用 Visual Basic Script 或 Java Script 这两种脚本语言来开发，现在 ASP+ 从 Visual Basic Script 升级到 Visual Basic，而且还可以使用先

锋语言 C#；而最令 Web 程序员开心的是，再也不必去使用微软提供的蹩脚的调试器了，现在 ASP+使用 VS.NET 统一的调试器。同时 ASP+做了若干的小改动：ADO 可以直接与 DataGrid 元件相连，以前的 asp.dll，现在升级到统一的 .NET Framework 类库。ASP+还封装了通用 HTML 接口的各种试件和针对 HTML 不足而设计的 Web 控件。

微软这样做的目的是因为网络的发展，它试图统一所有的网络开发语言。比如 Visual Basic 和 Visual C++，尽管它们有各自不同的集成开发环境(IDE)、调试器、函数类库 API，许多语法设定也有很大的不同，比如 Visual Basic 允许可变变量的存在，允许数组自定义下标。而 Visual C++不行，在 Visual Studio.NET 中这些都不被允许了。Visual Basic 6 并不是一个面向对象的编程语言，而微软在 Visual Studio.NET 系列中为 Visual Basic 赋予了同 Visual C++一样的面向对象特性，继承、封装、多态、重载等一应俱全。Visual Basic、Visual C++现在使用统一的集成开发环境，统一的调试器，统一的面向对象、异步、层次结构的可扩展类库(即 .NET Framework 类库)，这一切基于公共语言运行时(CLR)。现在程序员可以实现跨语言继承、纠错处理以及程序调试，而且 .NET 并不只是统一了微软本身的编程语言，包括其他 17 种语言都可以在 .NET 环境下运行。这意味着在 .NET 环境下网络上存在已久的 PERL 这样的语言也会被微软收编在自己的体系内，也意味着程序开发人员不但可以用不同的语言来开发组件控件，自己开发的类还可以继承其他语言编写的类，只要使用了交叉语言继承特性。调试器也是统一的，这意味着程序开发人员也能充分利用网络上存在的各种语言代码，懒惰的程序员们只需把需要的多语言代码串在一起，调试出多语言混合编写的应用程序给老板就可以了，而不需要考虑它们是否能在本地或远程机器上运行。总之，不管使用什么编程语言，微软 .NET Framework 都将提供一套丰富的统一的 Windows API 和关于 Internet 的 API。

1.2.2 Visual Studio.NET 的设计目标

Visual Studio.NET 在原有语言工具的基础上作了大量的创新和升级，将许多变革特性加入其中。在这些新特性中，最引人注目的 Visual Studio.NET 是构筑在 .NET 架构(.NET Framework)之上的，因此拥有 .NET 框架的所有关键特性，并可快速开发出符合 .NET 架构的应用程序。

时至今日，尽管大多数的应用程序仍然基于客户机/服务器模式，但是企业对基于 Web、以浏览器为前端的、多层次结构应用程序的需求的增长是最快的。开发者必须将他们的注意力转移到分布式应用程序开发上来。Visual Studio.NET 正是为了迎合这种需要来进行设计和开发的，它提供给开发者开发强大 Web 应用的高效、快速的手段。Visual Studio.NET 是微软下一代针对 Web 与企业级应用开发的开发套件，它迎合了下一代软件科技在应用程序开发方面的变革，可以大幅度地简化分布式 Web 应用和服务的开发。Visual Studio.NET 的设计目标包括以下几个方面。

- 增强开发者的生产效率

决定一个软件项目成功的重要因素之一就是开发效率。企业要求项目开发必须加快进度、缩短开发周期,以尽快将业务系统投入使用来适应商业竞争的需要。Visual Studio.NET 为专业开发者提供了完整的解决方案,其中包括一个统一和高度集成化的开发环境(IDE),可以让开发者在这个环境中,完成包括编写代码、除错、测试、发布等所有环节的工作,故大大增加了开发者的工作效率。

- 强大的 Windows 和 Web 开发工具

开发者要面临开发多种类型应用。Visual Studio.NET 将为开发者提供最为详尽的工具来开发应用广泛的应用程序。使用 Visual Studio.NET,开发者可以使用一种相似的编程手段来开发各种用户接口,包括浏览器、移动设备以及 Windows 客户程序。

- 改善开发语言的协作

过去,开发者在选择开发语言和开发平台上面临着比较痛苦的抉择:为了获得软件运行的高效率和可靠性,开发者往往选择功能强大的 B 语言工具(B 语言是贝尔实验室开发的 C 语言的前身),但是因为 B 语言功能过于强大、特性繁多,让开发人员不但掌握起来颇费神思,而且完全非 RAD(随机存取)式的开发方式也使开发者在企业级开发中感到开发难度很大;而微软提供的 RAD 产品——Visual Basic,虽然学习难度小、易于掌握,开发速度也很快,但是却不能胜任关键任务的开发,所以在过去,开发者在同一个项目中使用多种开发工具,或者在一个开发团队中必须将多方面开发人员整合在一起的现象非常常见。由于前者对开发人员的素质要求较高,而后者由于人员之间的配合与协调又会产生许多不可预期的开销,所以为了彻底解决语言工具协同工作的问题,微软公司作了大量创造性工作:一方面创建了一种现代的开发语言——C#,另一方面,还创建了能够实现多语言无缝集成的通用语言框架——.NET 框架。前者,作为一种现代语言,博采众家之长,将 Visual Basic 的简便易用和 B 语言的强大特性集于一身,而后者,作为一个公共的运行和语言支持平台,让所有的语言(包含第三方开发语言)在这个平台上不再有地位差异。

- 简化基于服务的开发

分布式应用程序开发中最为复杂的部分大概就是服务端组件的开发了。服务端组件封装了项目当中的业务逻辑,担负着处理客户端服务请求、平衡访问负载、链接数据库等重要任务,在多层架构中扮演着核心角色。Visual Studio.NET 为了简化基于服务的开发,提供给开发者大量 RAD 手段来简化服务端组件的开发。

- 简化创建和使用 Web 服务

在下一代的互联网中,Web 站点之间的合作机制将大幅度加强,而不像过去,Web 服务都被设计和开发成相互隔离的孤岛。Web 站点一方面可以通过对外公开可编程接口为其他站点和开发者提供服务;另一方面,开发者也能够把互联网上已存 Web 服务直接整合到