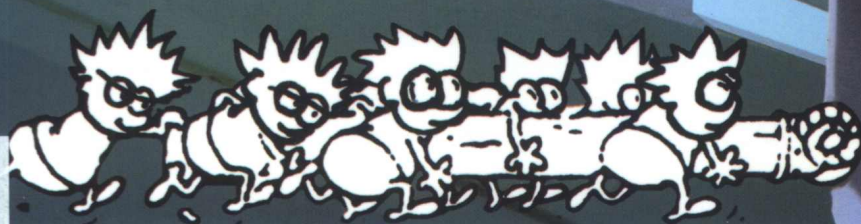


入门与提高

孙永强 陈宗斌 邱艳辉 编著

Visual C#.NET



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



23

TP312C
SF862

软件入门与提高丛书

Visual C#.NET 入门与提高

孙永强 陈宗斌 邸艳辉 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

为了开发.NET 应用程序, Microsoft 推出了 Visual Studio.NET 开发平台。在该版本中, 包含了 C# 这个全新的开发语言, 它是专门针对.NET 战略所开发的语言, 因此它能够充分利用.NET 平台的各种强大功能, 并且使用起来非常直观和方便。本书详细介绍了如何使用 C# 开发面向.NET 的应用程序, 包括.NET Framework 和 C# 的基本知识、Windows 窗体应用程序的开发、Web 窗体程序的开发、Web 服务的开发以及组件的开发, 内容由浅入深, 结构清晰, 论证充分。此外, 本书还提供了大量的例程, 使读者能够切实掌握 C# 各个方面的概念。

本书不要求读者具有某些编程语言的知识, 但是如果读者熟悉 C/C++ 语言的知识, 则可以更快地了解 C# 语言。另外, 在介绍 Web 应用程序的开发时, 要使用 HTML/DHTML 来开发网页, 所以要求读者熟悉这种网页制作语言。

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

Visual C#.NET 入门与提高/孙永强等编著. —北京: 清华大学出版社, 2002.8

软件入门与提高丛书

ISBN 7-302-05692-7

I. V... II. 孙... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 050057 号

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

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

责任编辑: 许振伍

印 刷 者: 世界知识印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 30.5 字数: 757 千字

版 次: 2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

书 号: ISBN 7-302-05692-7/TP·3354

印 数: 0001~5000

定 价: 41.00 元

《软件入门与提高丛书》特色提示

- ☑ 精选国内外著名软件公司的流行产品，以丰富的选题满足读者学用软件的广泛需求
- ☑ 以中文版软件为介绍的重中之重，为中国读者度身定制，从而便捷地掌握国际先进的软件技术
- ☑ 紧跟软件版本的更新，连续推出配套图书，使读者轻松自如地与世界软件潮流同步
- ☑ 明确定位，面向初、中级读者，由“入门”起步，侧重“提高”，愿新手老手都能成为行家里手
- ☑ 围绕用户实际使用之需取材谋篇，着重技术精华的剖析和操作技巧的指点，使读者深入理解软件的奥秘，举一反三
- ☑ 追求明晰精练的风格，用醒目的步骤提示和生动的屏幕画面使读者如临操作现场，轻轻松松地把软件用起来

丛书编委会

主 编	李振格		
编 委	焦金生	李幼哲	黄娟娟
	丁 岭	章忆文	许振伍
	吕建忠	应 勤	王景先

《软件入门与提高丛书》序

普通用户使用电脑最关键也最头疼的问题恐怕就是学用软件了。软件范围之广，版本更新之快，功能选项之多，体系膨胀之大，往往令人目不暇接，无从下手；而每每看到专业人士在电脑前如鱼得水，把软件玩得活灵活现，您一定又是惊羡不已。

“临渊羡鱼，不如退而结网”。道路只有一条：动手去用！选择您想用的软件 and 一本配套的好书，然后坐在电脑前面，开机、安装，按照书中的指示去用、去试，很快您就会发现您的电脑也有灵气了，您也能成为一名出色的舵手，自如地在软件海洋中航行了。

《软件入门与提高丛书》就是您畅游软件之海的导航器。它是一套包含了现今主要流行软件的使用指导书，能使您快速便捷地掌握软件的操作方法和编程技术，得心应手地解决实际问题。

让我们来看一下本丛书的特色吧！

■ 软件领域

本丛书精选的软件皆为国内外著名软件公司的知名产品，也是时下国内应用面最广的软件，同时也是各领域的佼佼者。目前本丛书所涉及的软件领域主要有操作平台、办公软件、编程工具、数据库软件、网络和 Internet 软件、多媒体和图形图像软件等。

■ 版本选择

本丛书对于软件版本的选择原则是：紧跟软件更新步伐，推出最新版本，充分保证图书的技术先进性；兼顾经典主流软件，给广受青睐、深入人心的传统产品以一席之地；对于兼有中西文版本的软件，采取中文版，以全力满足中国用户的需要。

■ 读者定位

本丛书明确定位于初、中级用户。不管您以前是否使用过本丛书所述的软件，这套书对您都非常合适。

本丛书名中的“入门”是指，对于每个软件的讲解都从必备的基础知识和基本操作开始，新用户无需参照其他书即可轻松入门；老用户亦可从中快速了解新版本的新特色和新功能，自如地踏上新的台阶。至于书名中的“提高”，则蕴涵了图书内容的重点所在。以我们的经验，当前软件的功能日趋复杂，不学到一定的深度和广度是难以在实际工作中应付自如的。因此本丛书在让读者快速入门之后，就以大量明晰的操作步骤和典型的应用实例，教会读者更丰富全面的软件技术和应用技巧，使读者真正对所学软件融会贯通、熟练在手。

■ 内容设计

本丛书的内容是在仔细分析用户使用软件的困惑和目前电脑图书市场现状的基础上确定的。简而言之，就是实用、明确和透彻。它既不是面面俱到的“用户手册”，也并非详解原理的“功能指南”，而是独具实效的操作和编程指导书。一切围绕用户的实际使用需要选

择内容,使读者在每个复杂的软件体系面前能“避虚就实”,直指目标;对于每个功能的讲解,则力求以明确的步骤指导和丰富的应用实例准确地指明如何去做,读者只要按书中的指示和方法做成、做会、做熟,再举一反三,就能扎扎实实地轻松过关。

风格特色

本丛书在风格上力求文字精练、图表丰富、脉络清晰、版式明快。另外,还特别设计了一些非常有特色的段落,以在正文之外为读者指点迷津。这些段落包括:

 **注意**——提醒操作中应注意的有关事项,避免错误的发生,让您少一些傻眼的时刻和求救的烦恼。

 **提示**——提示可以进一步参见的章节,以及有关某个内容的详细信息,使您可深可浅,收放自如。

 **技巧**——指点一些捷径,透露一些高招,让您事半功倍,技高一筹。

 **试一试**——精心设计各种操作练习,您只要照猫画虎,试上一试,就不仅能在您的电脑上展现出书中的美妙画面,还能了解书中未详述的其他实现方法和可能出现的其他操作结果。随处可见的“试一试”,让您边学边用,时有所得,常有所悟。

 **故障解析**——分析常见软硬件故障的原因,说明排除故障的方法,使用户能“有病自医”进而“久病成医”,积累诊断和排除的实战经验,最终成为高手。

经过紧张的策划、设计和创作,本套丛书已陆续面市,市场反应良好。许多书在两个月内迅速重印,本丛书自面世以来,已累计售出800多万册。大量的读者反馈卡和来信给我们提出了很多好的意见和建议,使我们受益匪浅。严谨、求实、高品味、高质量,一直是清华版图书的传统品质,也是我们在策划和创作中孜孜以求的目标。尽管倾心相注,精心而为,但错误和不足在所难免,恳请读者不吝赐教,我们定会全力改进。

本丛书在创作过程中得到了微软中国公司产品部的大力支持,对于他们在软件和技术资料的提供及有关目录的审定方面所给予的协助,我们表示衷心的感谢。

《软件入门与提高丛书》编委会

前言

1. Visual C#.NET简介

为适应未来社会的发展, Microsoft 推出了 .NET 战略, 以此作为下一代的网络应用基础。Microsoft 的所有产品包括操作系统、办公软件和开发工具等都将转移到 .NET 平台上来。 .NET 所包含的技术全面而先进, 必将对今后的软件开发和应用形式产生巨大的影响, 因此, 及时地了解与掌握 .NET 技术, 对于多数开发人员来说都是必需的。

为了促进 .NET 应用程序的开发, Microsoft 推出了 Visual Studio.NET 集成开发环境。这个集成环境中包含了许多强大的工具, 并且支持多种编程语言, 如 C#、Visual Basic.NET、C++托管扩展、JAVASCRIPT、DHTML 等, 其中 C#就是 Microsoft 为适应未来网络技术的发展而推出的新一代编程语言, 它是专为 .NET 战略所开发的语言。

虽然 Visual Studio.NET 中包含了多种编程语言, 并且所有语言开发出的 .NET 应用程序的效果也是一样的, 但是, C#本身作为一种功能强大的编程语言, 它从著名的 C++进化而来, 并综合了 C++与 Java 的优点, 是 .NET 的主推语言, 在 .NET 中起着不可或缺的作用, 所以本书以 C#语言为基础来介绍如何开发 .NET Framework 应用程序。

2. 本书阅读指南

本书详细介绍了如何使用 C#开发面向 .NET 的应用程序, 包括 .NET Framework 和 C#的基本知识、Windows 窗体应用程序的开发、Web 窗体程序的开发、Web 服务的开发以及组件的开发, 具体内容如下。

第 1 章介绍了 Visual Studio.NET 的新特性, 及安装 Visual Studio.NET 的方法。

第 2 章详细介绍了 Visual Studio.NET 集成开发环境的各个组成部分, 帮助用户尽快熟悉整个开发环境。

第 3 章通过创建第 1 个 C#程序, 向用户介绍了在 Visual Studio.NET 开发环境中创建、生成、运行程序的方法。

第 4 章向用户简单介绍了 C#语言的语法知识。在掌握了这些基本知识后, 用户就可以深入学习 Visual C#编程。

第 5 章介绍了创建 Windows 窗体、对话框, 及添加菜单和设计多文档界面的编程方法。通过学习本章, 用户将可以创建简单的 Windows 应用程序。

第 6 章详细介绍了常用的窗体控件, 以及设置控件属性及为控件添加事件的方法。

第 7 章介绍了几种常见的通用对话框, 以及如何显示通用对话框, 设置通用对话框属性及显示通用对话框结果的方法。

第 8 章介绍了如何使用 GDI+进行绘图操作, 以及在图形上显示文字的方法。

第 9 章介绍了基本的文件和目录的输入/输出操作。

第 10 章介绍了在 Windows 程序中实现打印和打印预览的方法。

第 11 章介绍了创建帮助及在应用程序中显示帮助的方法。

第 12 章介绍了进程和线程的基本概念，以及创建、启动和结束进程或线程的方法，同时还介绍了多个线程之间同步的知识。

第 13 章介绍了直接访问数据库的类和方法，主要包括数据库连接、数据命令以及如何处理数据命令的返回结果等知识。

第 14 章介绍了 ADO.NET 的基本概念，以及如何在 Windows 窗体中使用 ADO.NET 通过数据绑定机制来访问数据库的方法。

第 15 章介绍了有关组件的一些基本概念，以及使用 Visual C# 开发组件的方法。

第 16 章详细介绍了有关 Web 表单及 Web 表单页面中的事件处理和状态管理等知识，同时还介绍了在 Visual Studio.NET 开发环境中使用 ASP.NET，开发 Web 应用程序的方法。

第 17 章介绍了使用服务器控件的基本知识，然后依次分类介绍一些常用的服务器控件，并通过示例程序来实际地演示各种常用控件的用法。

3. 本书约定

本书以 Microsoft 最新推出的 Visual Studio.NET 为开发环境，来介绍 Visual C# 编程的知识。为便于阅读，本书作如下约定。

- 本书中出现的中文菜单和命令将用“【】”括起来，以示区分；而英文的菜单和命令直接写出，即省略“【】”。此外，为了语句更简洁易懂，本书中所有菜单和命令之间以竖线“|”分隔，例如单击 File 菜单再选择 Save As 命令，就用 File | Save As 来表示。
- 用“+”号连接的两个或三个键表示组合键，在操作时表示同时按下这两个或三个键。例如，Ctrl+V 是指在按下 Ctrl 键的同时，按下 V 字母键；Ctrl+Alt+F10 是指在按下 Ctrl 和 Alt 键的同时，按下功能键 F10。
- 在没有特殊指定时，单击、双击和拖动是指用鼠标左键单击、双击和拖动，右击是指用鼠标右键单击。

4. 网上资源

本书中示例的源代码，可到网站<http://www.tupwq.com>上下载。有关本系列丛书的最新消息，也可以从该网站获取。

目 录

第 1 章 概述.....1	
1.1 Visual Studio.NET 的新特性.....2	
1.1.1 .NET 框架.....2	
1.1.2 集成开发环境 (IDE) 的新特性.....2	
1.1.3 Visual Studio.NET 的其他新特性.....3	
1.2 C#简介.....4	
1.3 安装 Visual Studio.NET.....4	
1.3.1 系统需求.....5	
1.3.2 在 Windows 2000 Server 中安装 Visual Studio.NET5	
1.4 开发环境介绍.....11	
1.4.1 开始页面.....11	
1.4.2 解决方案资源管理器.....12	
1.4.3 代码编辑器.....19	
1.4.4 对象浏览器.....21	
1.4.5 属性窗口.....24	
1.4.6 命令窗口.....25	
1.4.7 任务列表.....26	
1.4.8 调试器.....27	
第 2 章 第 1 个 C#程序.....34	
2.1 创建程序.....35	
2.2 程序结构分析.....37	
2.2.1 Class1.cs 文件.....37	
2.2.2 AssemblyInfo.cs 文件.....38	
2.2.3 程序结构小结.....39	
2.3 生成并运行程序.....40	
第 3 章 C#语言基础.....42	
3.1 概述.....43	

3.2 C#的操作符..... 44	
3.3 C#的常用语句..... 46	
3.3.1 变量声明语句..... 46	
3.3.2 条件选择语句..... 47	
3.3.3 循环迭代语句..... 48	
3.3.4 方法调用语句..... 50	
3.3.5 异常处理语句..... 52	
3.3.6 非安全语句..... 55	
3.4 预处理器指令..... 57	
3.5 C#的类型系统..... 60	
3.5.1 命名空间..... 61	
3.5.2 结构类型..... 64	
3.5.3 枚举类型..... 66	
3.5.4 包装 (boxing) 和拆包 (unboxing) 67	
3.5.5 类..... 67	
3.5.6 代理 (delegate) 76	
3.5.7 接口 (interface) 77	
3.5.8 object 和 string 类型..... 80	
3.5.9 数组..... 80	
3.6 属性 (Attribute) 81	
第 4 章 第 1 个 Windows 应用程序..... 85	
4.1 概述..... 86	
4.2 创建窗体..... 87	
4.2.1 使用新建项目模板创建主窗体..... 87	
4.2.2 使用添加项目模板创建窗体..... 89	
4.3 设置窗体的属性..... 90	
4.4 处理窗体的事件..... 91	
4.5 对话框..... 93	

4.6 菜单.....	95	5.6.1 分割条控件.....	135
4.6.1 添加主菜单.....	95	5.6.2 分组框控件.....	135
4.6.2 添加快捷菜单.....	97	5.6.3 面板控件.....	135
4.6.3 示例程序.....	98	5.6.4 选项卡控件.....	136
4.7 多文档界面.....	101	5.6.5 示例程序.....	138
4.7.1 创建 MDI 主窗体.....	102	5.7 工具栏控件和状态栏控件.....	140
4.7.2 创建 MDI 子窗体.....	102	5.7.1 工具栏控件.....	141
4.7.3 操作子窗体.....	103	5.7.2 状态栏控件.....	142
4.7.4 示例程序.....	105	5.7.3 示例程序.....	144
第 5 章 控件.....	110	5.8 列表视图和树视图.....	146
5.1 向窗体中添加控件.....	111	5.8.1 列表视图.....	146
5.2 调整控件.....	112	5.8.2 树视图.....	150
5.2.1 对控件进行分层.....	113	5.8.3 示例程序.....	151
5.2.2 在窗体中定位控件.....	113	第 6 章 通用对话框.....	159
5.2.3 改变控件的大小.....	114	6.1 消息框.....	160
5.2.4 相对于窗体的边框固定 控件.....	114	6.2 文件对话框.....	161
5.2.5 设置控件的 Tab 键顺序.....	115	6.3 【字体】对话框.....	166
5.3 按钮类控件.....	115	6.4 【颜色】对话框.....	169
5.3.1 按钮控件.....	116	第 7 章 GDI+ 编程.....	173
5.3.2 单选按钮.....	116	7.1 Graphics 对象.....	174
5.3.3 复选框.....	116	7.2 Paint 事件.....	175
5.3.4 示例程序.....	116	7.3 常用的数据结构.....	176
5.4 标签控件、文本框控件、列表框 控件和组合框控件.....	120	7.4 示例程序.....	176
5.4.1 标签控件.....	120	7.5 颜色.....	179
5.4.2 文本框控件.....	122	7.6 字体.....	181
5.4.3 列表框控件.....	123	7.7 画笔.....	185
5.4.4 组合框控件.....	124	7.8 画刷.....	189
5.4.5 示例程序.....	125	7.9 显示文字.....	190
5.5 图片框控件、图像列表控件和 时钟控件.....	131	7.10 显示图像.....	191
5.5.1 图片框控件.....	131	第 8 章 基本的文件输入/输出操作.....	195
5.5.2 图像列表.....	132	8.1 Stream 类.....	196
5.5.3 时钟控件.....	133	8.2 FileStream 类.....	197
5.5.4 示例程序.....	133	8.3 用于读写数据的类.....	199
5.6 分割条控件、分组框控件、面板 控件和选项卡控件.....	134	8.3.1 BinaryReader 类.....	199
		8.3.2 BinaryWriter 类.....	199
		8.3.3 StreamReader 类.....	200
		8.3.4 StreamWriter 类.....	201

8.4 文件和目录类.....202	第 13 章 使用 ADO.NET 访问数据库..... 300
8.4.1 文件类.....202	13.1 Windows 窗体中的数据体系
8.4.2 目录类.....205	结构..... 301
8.5 Path 类.....207	13.2 数据适配器..... 302
8.6 示例程序.....209	13.2.1 数据适配器的属性和
第 9 章 打印和打印预览.....226	方法..... 302
9.1 PrintDocument 类.....227	13.2.2 创建数据适配器..... 305
9.1.1 设置类的属性.....227	13.3 数据集..... 314
9.1.2 处理打印事件.....229	13.3.1 创建数据集对象..... 315
9.2 PrintDialog 对话框.....232	13.3.2 创建非类型化数据集..... 317
9.3 打印预览.....233	13.4 DataTable 类..... 321
9.4 示例程序.....234	13.4.1 创建数据表..... 321
第 10 章 帮助系统.....243	13.4.2 定义表的结构..... 321
10.1 Help 类.....244	13.4.3 操作表中的数据..... 325
10.2 HelpProvider 类.....245	13.5 DataRelation 类..... 330
10.3 ToolTip 类.....246	13.6 使用视图..... 331
10.4 示例程序.....247	13.6.1 数据视图..... 332
第 11 章 进程与线程.....252	13.6.2 数据行视图..... 334
11.1 进程.....253	13.6.3 数据视图管理器..... 334
11.2 进程示例程序.....257	13.7 CurrencyManager 类和 Binding-
11.3 线程.....265	Context 类..... 335
11.3.1 创建并启动线程.....266	13.8 示例程序..... 336
11.3.2 控制线程的执行.....269	第 14 章 组件开发..... 348
11.3.3 线程之间的同步.....270	14.1 基本概念..... 349
第 12 章 访问数据库.....276	14.2 在组件中使用事件..... 351
12.1 数据库连接.....277	14.2.1 事件代理..... 351
12.1.1 SqlConnection 类.....277	14.2.2 在类中触发事件..... 352
12.1.2 OleDbConnection 类.....282	14.2.3 示例程序..... 353
12.2 数据命令.....283	14.3 开发 Windows 窗体控件..... 357
12.2.1 SqlCommand 类.....283	14.3.1 创建控件..... 358
12.2.2 OleDbCommand 类.....288	14.3.2 为控件添加属性、方法
12.3 SqlDataReader 类和 OleDbData-	和事件..... 362
Reader 类.....288	14.3.3 控件的绘制..... 364
12.4 示例程序.....291	14.3.4 为控件添加设计时支持..... 365
	14.3.5 示例程序..... 369
	第 15 章 开发 Web 应用程序..... 385
	15.1 概述..... 386

15.2 创建一个基本的 Web 应用程序	387	16.2.8 HyperLink 控件	433
15.3 Global.asax 和 Global.asax.cs 文件	393	16.2.9 DropDownList 控件	434
15.4 Web 表单	394	16.2.10 ListBox 控件	435
15.4.1 Web 表单的代码模型	395	16.2.11 CheckBox 和 Radio- Button 控件	436
15.4.2 Web 表单页面的处理 过程	396	16.2.12 CheckBoxList 和 Radio- ButtonList 控件	437
15.4.3 Web 表单页面中的事件 处理	397	16.2.13 Panel 控件	438
15.4.4 Web 表单状态管理	399	16.2.14 Table 控件	439
15.4.5 Page 类	401	16.2.15 示例程序	441
15.5 其他相关类	402	16.3 HTML 服务器控件	447
15.5.1 HttpRequest 类	403	16.3.1 HtmlGenericControl 控件 ..	449
15.5.2 HttpResponse 类	407	16.3.2 HtmlButton 控件	450
15.5.3 HttpBrowserCapabilities 类	409	16.3.3 HtmlInputText 控件	450
15.5.4 HttpServerUtility 类	410	16.3.4 HtmlTextArea 控件	451
15.6 示例程序	412	16.3.5 HtmlInputCheckBox 和 HtmlInputRadioButton 控件	451
第 16 章 使用 ASP.NET 服务器控件	421	16.3.6 HtmlInputHidden 控件	452
16.1 服务器控件的基本概念	422	16.3.7 HtmlInputFile 控件	452
16.1.1 System.Web.UI.Control 类	422	16.3.8 HtmlTable 控件	453
16.1.2 向 Web 页面中添加服务 器控件	424	16.3.9 HtmlImage 控件	454
16.2 常用的 Web 服务器控件	426	16.3.10 HtmlSelect 控件	454
16.2.1 AdRotator 控件	426	16.3.11 示例程序	455
16.2.2 Button 控件	429	16.4 验证控件	460
16.2.3 Label 控件	430	16.4.1 BaseValidator 类	461
16.2.4 TextBox 控件	430	16.4.2 必须字段验证控件	463
16.2.5 LinkButton 控件	431	16.4.3 比较验证控件	464
16.2.6 Image 控件	432	16.4.4 正则表达式验证控件	465
16.2.7 ImageButton 控件	432	16.4.5 自定义验证控件	465
		16.4.6 ValidationSummary 控件 ..	466
		16.4.7 示例程序	467

第 1 章

概 述

本章要点

为了配合 .NET 应用程序的开发, 促进 .NET 战略的发展, Microsoft 推出了新一代的开发工具: Visual Studio.NET。Visual Studio.NET 是快速建立企业级 Web 应用程序和高性能桌面应用程序的工具。Visual Studio.NET 完全支持 .NET 框架。

C# (读作 C Sharp) 是 Visual Studio.NET 中包含的一种全新的开发语言。它从 C/C++ 语言发展而来, 具有与 C/C++ 语言相似的语法, 继承和发扬了 C/C++ 语言的优点, 使用起来更加方便和直观。C# 语言是针对 .NET 框架平台开发的一种新语言, 它完全建立在 .NET 框架平台基础之上。

目前, 开发 C# 语言程序的最好工具就是 Microsoft 新推出的 Visual Studio.NET。它包含了许多强大的工具, 掌握好这些工具可以极大地促进程序的开发。本章将会介绍 Visual Studio.NET 的新特性、安装过程和集成开发环境。

本章内容包括:

- ▶ Visual Studio.NET 的新特性
- ▶ C# 简介
- ▶ 安装 Visual Studio.NET
- ▶ 集成开发环境

1.1 Visual Studio.NET 的新特性

相对于以前的版本更新,这一版本 Visual Studio.NET 的变化可以说是天翻地覆。首先, Visual Studio.NET 支持新的应用程序开发平台,即 Microsoft .NET 框架,并针对它做了彻底的更新;其次, Visual Studio.NET 把 Microsoft 的所有语言的开发环境统一起来,所有语言使用同一套工具,在同一个集成开发环境中进行开发,并且改进了环境中所包含的各种工具。下面,依次介绍 Visual Studio.NET 的各种更新和增强特性。

1.1.1 .NET 框架

.NET 框架是一个建立、配置和运行 Web 服务以及应用程序的多语言环境,是 Microsoft 的新一代 Web 应用程序开发平台。它包含以下 3 个主要部分。

- 公共语言运行时环境 (Common Language Runtime, CLR): 这个运行时环境在组件的运行期间和开发期间都起着重要的作用。在组件的运行期间,这个运行时环境管理内存分配、启动及结束线程和进程并且执行安全策略,同时还满足该组件对其他组件的所有依赖需求。在开发期间,这个运行时环境主要用来简化开发工作。
- 统一编程类库: .NET 框架提供一个统一的、面向对象的、层次化的及可扩展的类库集。Visual Studio.NET 中的所有编程语言,包括 C#、C/C++、VB 等以及各种脚本语言,都可以使用这个类库。C# 中的各种数据类型就是建立在这个类库之上。
- ASP.NET: ASP.NET 建立在 .NET 框架编程类库之上,提供了一个 Web 应用程序模型。使用这个模型可以方便迅速地开发出功能强大的 Web 应用程序。

实际上,在 .NET 框架中,所有的编程语言,从相对简单的 JavaScript 到复杂的 C++ 语言,一律是等同的,它们都将生成托管代码,并且可以“一次编写,随处运行”。.NET 框架的另一个强大功能是不同的语言之间可以进行交互,即一种语言可以使用另一种语言编写的组件,并可以从另一种语言编写的类派生新类或创建实例。语言互用性允许编程人员使用不同的编程语言编写各个组件,再将这些组件连接成一个大型的应用程序。更令人振奋的是,它们还可充分利用互联网和企业内部网的优势,访问整个网络上的 Web 服务。

1.1.2 集成开发环境 (IDE) 的新特性

这一版本的集成开发环境做了一些很大的更新,首先是统一了所有语言的开发环境,在这个统一的开发环境中可以进行 Visual Basic、Visual C++、C#、HTML、XML 等语言的开发;其次是新增了一些工具窗口,比如资源管理器窗口、属性窗口及工具箱窗口等;还有就是全新的窗口管理方式。在这一版本中,大量使用了标签化窗口,这种窗口可以极大

地提高屏幕利用率，当要切换窗口时，只需用鼠标单击窗口的标签就可以了；除了标签化之外，集成开发环境中的各种窗口还具有自动隐藏的特性。即当焦点离开窗口时，窗口就会自动地隐入主窗口的某一边框中，而当单击自动隐藏窗口的标签时，该窗口就会自动地从边框中滑出来。

Visual Studio.NET 在调试工具上也有所更新。Visual Studio.NET 在所有语言上使用单一的集成化调试器。这个调试器可以进行跨语言的调试，既可以调试为公共语言运行时环境编写的应用程序，也可以调试 Win32 本地应用程序，调试方式可以是本地的也可以是远程的。

Visual Studio.NET 中的配置工具提供许多支持 Microsoft Window Installer 技术和简化分布式配置的新特性。用户可以使用这个配置工具在设计期间进行部分应用程序的配置。其中合并模块功能允许用户创建共享组件的标准化安装。Visual Studio.NET 的图形化配置编辑器使得安装程序的创建更加容易。

1.1.3 Visual Studio.NET 的其他新特性

除了对集成开发环境所做的更新外，Visual Studio.NET 还新增了一些功能来支持新的应用程序开发方式，如下所述。

1. Web服务

Web 服务是基于网络的分布式应用程序的基本构造模块，这些分布式程序独立于平台，通常使用对象模板和多种语言创建。Web 服务建立在 HTTP 和 XML 之类的开放的 Internet 标准之上，并且形成了微软的可编程网络理论的基础。Web 服务可以通过 HTTP 接收请求和 XML 数据。Web 服务不特定于某一种组件技术和对象调用约定，所以它可以被任何语言、组件模型或操作系统访问。Web 服务是一种应用程序，它可以使用标准的互联网协议，将功能纲领性地体现在互联网和企业内部网上。可将 Web 服务视作 Web 上的组件编程。在 Visual Studio.NET 中，可以使用 Visual Basic、C# 或 ATL 服务器来创建和引用 Web 服务。

2. XML支持

可扩展标注语言（XML，Extensible Markup Language）提供了一个描述结构化数据的框架。它是国际标准，是 SGML 的一个子集，并且专门针对 Web 进行了优化。XML 是 .NET 战略的核心技术，它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的功能，并实现了软件的不间断传输。Visual Studio.NET 完全支持 XML，并且提供了简化 XML 和 XML 方案创建工作的 XML Designer。

除了以上介绍的新增功能外，Visual Studio.NET 中的示例程序和文档也进行了很大的更新。新版本提供了两个全新的企业级例程，DuwamishOnline 和 Fitch and Mather 7.0，这两个例程演示了如何使用 Visual Studio.NET 的各种技术和工具来开发企业级应用程序。此外，帮助文档的内容和使用方式都针对 .NET 的出现而进行了明显的调整。

1.2 C# 简介

C#语言从 C++语言发展而来，它的语法与 C++非常相似，并且非常简单，它摒弃了 C++语言中容易出错的指针、多重继承和模板等概念。C#是一种面向对象的、类型安全的语言，它充分利用了 .NET 框架的优点，如严格的类型安全、版本控制及无用单元收集 (garbage collect) 等，可以说，C#天生就是面向 .NET 框架的，它是开发 .NET 框架应用程序的最好语言。

如果读者熟悉 C++语言的话，应该知道 C++被设计为一种低级的与平台无关的面向对象的语言；而 C#则被设计为一种比较高级的面向组件的语言，它完全建立在 .NET 框架平台之上，它的所有类型都是从 .NET 框架提供的统一类库中派生的，并且所有的功能实现都依赖于这个类库。在 C++中可以进行一些比较低级的操作；而 C#就不能进行这些低级的操作。这些操作由 .NET 框架来完成，C#本身则集中于比较高级的操作。例如，在 C++中，用户可以控制对象的创建过程，甚至可以控制对象在内容中的布局：用户可以使用 `new` 操作符在栈中、堆中，甚至在内存的特定位置来创建对象。而在 .NET 的托管环境中，用户就不能进行这种级别的控制操作。当用户选择对象的类型时，该对象在何处创建就已经隐含地确定。简单类型 (`int`、`double` 和 `long int` 等) 总是在栈中进行创建 (除非这些类型位于其他的对象中)，而类则总是在堆中进行创建，并且用户无法获得它们的地址也不能改变它们的位置。用户不再能够控制对象的生存期，另外 C#中也没有析构函数，无用单元收集器将会在合适的时候回收对象的内存，但是回收时间是不确定的。

在 C#中没有多重继承并且没有模板类。C#中的简单类型仅仅是到底层 CLR 类型的直接映射。例如，C#的 `int` 对应于 `System.Int32`。C#中的类型不是由语言本身确定的，而是由公共类型系统确定的。另外，托管环境和 CLR 带来了许多切实的好处。除了无用单元收集功能和统一的类型系统之外，还有完全基于组件的语言，它完全支持版本化并提供可扩展的元数据，这些元数据可以在运行时通过映像服务来获得。后期绑定不再需要特别的支持，类型发现和后期绑定已经内建到语言中。在 C#中，枚举、属性、事件以及代理 (类型安全的函数指针) 都已内建到语言中。虽然所有面向 .NET 的语言都可以使用 .NET 框架，但是 C#是完全针对这个框架的，它可以充分使用这个框架所提供的类、接口和对象。

1.3 安装 Visual Studio.NET

Visual Studio.NET 本身的安装过程比较简单，用户只需进行几步简单的操作，安装向导就可以自动完成全部产品的安装。Visual Studio.NET 可以安装在 Windows 98/Me、Windows NT 4.0 和 Windows 2000 操作系统上。但是，在不同的操作系统环境中，用户所能开发的应用程序的种类和所需进行的设置都有所不同。例如，在 Windows 98/Me 操作系统中，用户就不能进行 Web 应用程序的开发。如果用户想要在 Windows 98/Me 环境中开发 Web 应用程序，则必须把开发机器连接到运行 Internet 信息服务器 (Internet Information

Server, IIS) 4.0 (或更新版本) 的服务器上, 并且需要对 IIS 服务器进行正确的配置。而在 Windows NT/2000 环境中则没有这些限制, 因为 Visual Studio.NET 支持 Windows NT/2000 中的 PWS (Peer Web Server) 和 IIS, 用户可以在同一台机器中进行 Web 应用程序的开发。另外, 在 Windows 98/Me 中也不能进行 Windows 服务应用程序的开发。

本节主要介绍在 Windows 2000 Server 下安装 Visual Studio.NET 的步骤。在其他操作系统中安装 Visual Studio.NET 的过程与此大致相同, 用户可以参照本节中的步骤进行安装。

1.3.1 系统需求

Visual Studio.NET 是一个比较庞大的系统软件, 它对机器的要求较高。下面给出了安装和运行 Visual Studio.NET 的系统需求。

- 处理器: 奔腾 II 以上级别的处理器, 速度最小为 450MHz, 建议使用奔腾 III 733MHz 或速度更快的处理器。
- 内存: 最小内存为 96MB, 建议使用 128MB 或更大的内存。
- 硬盘空间: 完全安装 Visual Studio.NET 大致需要 2GB 的硬盘空间, 再加上用户开发的项目文件占用的空间, 最少需要 3GB 以上的硬盘空间。
- 显示设备: 最小屏幕分辨率为 800×600, 并且使用 16 位颜色。但是, Visual Studio.NET 集成开发环境包含大量的窗口, 在 800×600 的分辨率下将会显得非常拥挤。建议使用 1024×768 以上的分辨率, 更大当然更好。
- 光驱: 在安装 Visual Studio.NET 时, 光驱是必需的。如果所需的所有组件都已安装在硬盘上, 则在开发过程中就可以不再使用光驱。
- 操作系统: Windows 98/Me、Windows NT 4.0 和 Windows 2000。建议使用 Windows 2000 或更新版本的操作系统。使用更新、更稳定的操作系统有助于应用程序的开发, 最起码不会因为死机丢失劳动成果, 更何况还有许多新特性可以使用。

上面的最小需求仅能使 Visual Studio.NET 得以运转。Visual Studio.NET 在运行过程中要使用大量的内存和处理器资源, 而且其众多的工具窗口在方便用户使用的同时也占据了宝贵的屏幕面积。所以要想流畅地开发应用程序, 最好还是满足建议需求。

1.3.2 在 Windows 2000 Server 中安装 Visual Studio.NET

在 Windows 2000 Server 中使用 Visual Studio.NET 有许多优点, 首先可以在同一台机器中建立起一个和现实使用环境基本相同的开发和测试环境, 这样, 用户就可以在一台机器中模拟客户/服务器环境或浏览器/服务器环境对开发的组件进行测试; 其次, 许多软件只有在服务器环境中才能具备全部功能, 比如数据库软件。下面详细介绍在 Windows 2000 Server 中安装 Visual Studio.NET 的步骤。

- (1) 检查是否正确安装了 IIS 4.0 或更新版本。IIS 是开发 Web 应用程序所必需的, 只有先安装了 IIS, 才能正确安装 Visual Studio.NET。



提示: 在默认情况下, 安装 Windows 2000 Server 的时候会安装 IIS。但用户有可能