

Visual Basic .NET 程序设计

林联杉 李旭 顾青 编著

- 精选Visual Basic.NET内容，帮助读者全面了解Visual Basic.NET编程
- 全面的基础知识介绍，循序渐进的知识结构安排，引导读者渐入佳境
- 融入作者多年程序开发的心得，在介绍程序语言的基础上让读者掌握更清晰的编程思路
- 所学即所用，每章都附带应用实例，有利于读者进一步加深巩固
- 为了方便读者学习，书中范例源代码请到天勤网站：<http://www.tqbooks.net>“下载专区”下载



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

1200415772 = 4



1200415772

Visual Basic.NET 程序设计

林联杉 李旭 顾青 编著

TP312.13A

336



中国铁道出版社

2004·北京

(京)新登字 063 号

内 容 简 介

.NET 开发平台是时下最具实力的开发工具,其中最简单易学的语言就非 Visual Basic.NET 莫属了。.NET 平台主要包括两个方面的应用:应用程序开发和 Web 服务开发。本书主要从应用程序的设计开发着手,带领读者进入 Visual Basic.NET 的世界。

本书重在入门与提高,即使是对编程一片空白的初学者,也可以通过本书的学习达到一般编程开发人员的水平。本书主要涉及 Visual Basic.NET 整体可视化开发环境的使用、Visual Basic.NET 的语法基础、Visual Basic.NET 下 Windows 窗体及相应控件的使用等基础知识;然后通过综合的实例,一方面介绍面向对象编程思想、文件操作、数据库应用等知识,同时进一步巩固 Visual Basic.NET 的程序基础;最后本书还详细介绍了如何对编写完的代码进行编译和发布,以及如何代码开发中进行调试和异常处理,使读者对应用程序的开发有个完整的学习过程。

图书在版编目(CIP)数据

Visual Basic.NET 程序设计/林联杉,李旭,顾青编著. —北京:中国铁道出版社,2003.10

(热点编程系列丛书)

ISBN 7-113-05564-8

I. V… II. ①林…②李…③顾… III. BASIC 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 096181 号

书 名: Visual Basic .NET 程序设计

作 者: 林联杉 李 旭 顾 青

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓舟 魏 春

责任编辑: 苏 茜 王占清

封面设计: 孙天昭 王宏丽

印 刷: 北京鑫正大印刷有限公司

开 本: 787×1092 1/16 印张: 23.5 字数: 565 千

版 本: 2004 年 1 月第 1 版 2004 年 1 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN7-113-05564-8/TP·1058

定 价: 33.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前 言

曾经在某个论坛的 VB 版做过版主，那时总有人咨询：我想学习一门编程语言，不知道学哪种好？听说 VB 入门最简单，有没有什么比较合适的教材？

这其实是一个很普遍的问题，当然也很不好回答。因为现在的高级语言其实都比较接近自然语言，因此从语法角度看学习一般没问题。但对于初学者，如何能缩短编程开发的学习周期，又体会到编程的实质，才是最主要的问题。在 Visual Basic.NET 尚未出来之前，作者极力推荐 Visual Basic 6.0 作为程序开发的入门语言；在 Visual Basic.NET 出现之后，个人认为如果想迅速把握下一代程序开发工具的精髓，应当从 Visual Basic.NET 入手，同时有一本基础知识兼实例结合的书作为入门向导，就更容易登堂入室了。

Microsoft 曾预言 .NET 平台是下一代程序开发的规范。从 .NET 的 Beta 版出来到现在正式版的应用，逐步证明了这个实力，越来越多的程序开发者熟悉并喜欢上了 .NET 的风格，也写出了比以前更为高效、可靠、功能强大的代码。Visual Basic.NET 仍继承了 Visual Basic 6.0 好学易用的最大优点，同时结合了完全的面向对象编程方法，大大减轻程序语言对开发者的障碍，即使是从没接触过编程开发的用户，也能在较短的时间内利用它作出漂亮的应用程序。

☛ 本书适用对象

本书适合初、中级程序开发人员使用。对于初级编程人员，即刚刚开始接触编程的读者，本书提供了 Visual Basic.NET 的最基本知识的介绍，知识层次由低到高，由浅入深，每个章节是根据一般读者学习、实践应用的发展过程来安排的，有很强的针对性，同时附带诸多实例对知识内容进行讲解，有助于读者的理解和实践；对于中级编程人员，除了本书提供的范例有很大的参考性之外，对于一些基本的编程思想和思路，也进行了一定的剖析，对于提高编程能力、对程序设计的总体思考能力都是有很大帮助的。从总体上看，本书运用大量的图片和典型的实例，向您展示了 Visual Basic.NET 的强大功能和丰富的内容，手把手地教您学习 Visual Basic.NET。因此在不知不觉中，您的编程水平将有很大的提高。当您阅读完本书之后，就可以和其他高级编程人员一样，用最新的、最强大的工具，开发最流行的、功能完善的软件。

☛ 本书特点和主要内容

Visual Basic.NET 已经非常强大、全面，在这样一本书中要逐一介绍难免泛泛。.NET 平台主要包括两个方面的应用：应用程序开发和 Web 服务开发。本书主要从应用程序的设计开发着手，带领读者进入 Visual Basic.NET 的世界。之所以重点介绍应用程序的开发是因为 Web 服务程序也是在应用程序的基础上进一步发展的，但需要涉及更多的网络方面的知识，对一般读者可能会要求过高。本书具有以下特定：

(1) 从 Visual Basic.NET 最基本的概念、语法入手，通过窗体、控件的介绍使读者对应用程序的开发有基本的把握。接着向读者介绍面向对象的概念，使读者在前面的基础上进一步得到提高和升华，从而体会到应用程序开发的实质。从书的章节安排上来讲，适合初学者

学习——应用——再学习——提高的这样一个认识过程。

(2) 基础知识结合实例讲解。在各章知识的讲解过程中，都会有适当的实例来弥补读者对基础知识抽象理解的空白。通过对实例程序中的源代码进行详细学习，读者可以充分理解并掌握提出的概念，还可以举一反三作出更多的应用实例。

(3) 注重知识的综合。本书不但给出了某个知识点的实例，而且给出了将各方面知识点融合起来的综合实例。通过综合练习，让读者感受真实的编程项目的需求分析——设计——实现这一过程，进一步提高读者的实战能力。

本书重在入门与提高，主要涉及 Visual Basic.NET 整体可视化开发环境的使用、Visual Basic.NET 的语法基础、VB.NET 下 Windows 窗体及相应控件的使用等基础知识；然后通过若干综合的例子，一方面介绍面向对象编程思想、文件操作、数据库应用等知识，同时进一步巩固 VB.NET 的程序基础；最后本书还详细介绍了如何对编写完的代码进行编译和发布，以及如何在代码开发中进行调试和异常处理，使读者对应用程序的开发有个完整的学习过程。

☛ 学习所需的运行环境

有部分读者可能会担心，Visual Basic.NET 是比较新的开发工具，对计算机的软、硬件要求是不是会很高？根据作者的实际使用和体会，可以坦言这个顾虑是多余的。对于硬件的要求，相对现在普及的 PC 来说，无非就是 CPU 速度快，内存多，则运行速度快一些，反之则慢一些，安装和运行 .NET 基本都是没有问题的。以作者所见的较低的一台 PC 机的硬件配置为例：CPU 主频 300Hz，128MB 内存，20GB 硬盘，16MB 显存的 AGP 显卡，40 速光驱，顺利运行 Visual Basic.NET 没有任何问题，高于以上的硬件配置更是毋庸置疑了。对于软件的要求，基本上 .NET 平台要求在 Windows 2000 或更高级的（如 XP）操作系统上使用，当然如果只有 Windows 98 操作系统，勉强也是可以安装运行的，只是作者不推荐。在安装 Visual Basic.NET 之前，注意要先在操作系统上安装 IIS、.NET Framework 以及 IE6.0 等必备软件，具体一般在 .NET 的安装光盘上有说明，这里不再赘述。

☛ 本书作者

本书是由林联杉、李旭、顾青主笔，中国铁道出版社的很多朋友在程序调试、市场调查、书稿预读等方面都做了很多工作。魏春、秦绪好编辑提供了宝贵的建议，在此表示深深地感谢！另外，陈武、叶建、吕玉明、张燕、李旭、徐建刚、焦雷、赵亮、张晓冬、李小松、赵洪建、陈至立、王杰忠、李伟、郝谦、张志仁、李利等人在通读、校对和编排等方面做了大量的工作，在此一并表示感谢！由于水平和时间所限，本书难免有疏漏之处，敬请读者不吝指教。我们会在适当的时间进行修订与补充，并发布在天勤网站：<http://www.tqbooks.net> “图书修订”栏目中。

本书所有源代码可以到天勤网站：<http://www.tqbooks.net> “下载专区”中下载。

另外，如果读者在使用本书的过程中有什么问题可与作者直接联系：

E-MAIL: zero.23@263.net

编者

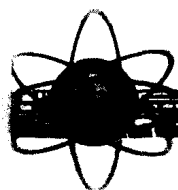
2003 年 11 月

目 录

第 1 章 Visual Basic.NET 的新特性.....	1
1-1 Visual Studio.NET 框架	2
1-1-1 .NET 框架的结构	2
1-1-2 .NET 框架的主要内容	3
1-1-3 .NET 框架的核心要素	5
1-1-4 .NET 的新特性	5
1-2 Visual Basic.NET 的新特性	7
1-2-1 概要	7
1-2-2 VB 获得了继承能力	7
1-2-3 一切都是对象	8
1-2-4 自由线程的危险	9
1-2-5 强大的 CLR	9
1-2-6 垃圾收集需要新体系结构	10
1-2-7 语法方面的变化	11
1-2-8 新的窗体和 IDE 面孔	12
1-2-9 创建编译的服务器端代码	13
1-2-10 Web 服务策略	13
本章小结	14
第 2 章 Visual Basic.NET 的开发环境	15
2-1 初试 IDE 启动界面	16
2-2 IDE 的各个组成部分	17
2-2-1 菜单栏	19
2-2-2 工具栏	21
2-2-3 工具箱	22
2-2-4 主工作区	23
2-2-5 解决方案资源管理器	26
2-2-6 属性窗口	27
2-2-7 动态帮助窗口	28
2-3 定制自己的 IDE 环境	29
2-4 我的第一个 .NET 应用程序	29
本章小结	30
第 3 章 Visual Basic.NET 的语法基础	31
3-1 语法基础	32
3-1-1 数据类型	32
3-1-2 数据类型转换	38

3-1-3	变量	40
3-1-4	常量	47
3-1-5	数组	48
3-1-6	集合	51
3-1-7	表达式	52
3-2	结构化控制语句	60
3-2-1	条件语句	61
3-2-2	循环语句	63
3-2-3	异常处理语句	68
3-2-4	控制流语句	70
3-3	过程与函数	72
3-3-1	内部函数	72
3-3-2	Sub 过程	81
3-3-3	Function 函数	82
3-3-4	参数传递	83
3-4	API 的应用	86
3-4-1	API 简介	86
3-4-2	API 的声明	87
3-4-3	API 的应用实例	89
	本章小结	91
第 4 章	可视化界面设计	93
4-1	Windows 窗体概述	94
4-1-1	窗体结构简介	94
4-1-2	窗体类简介	96
4-2	窗体属性、事件和方法	97
4-2-1	窗体的各种属性	98
4-2-2	窗体的事件	101
4-2-3	窗体的方法	104
4-3	窗体界面样式选择	107
4-3-1	单文档界面 (SDI)	107
4-3-2	多文档界面 (MDI)	108
4-3-3	资源管理器样式界面	110
4-4	特殊窗体的实例	111
4-4-1	启动界面的实现	111
4-4-2	特殊形状窗体的实现	113
	本章小结	115
第 5 章	常用控件的实例应用	117
5-1	控件概述	118
5-1-1	控件的分类	118

5-1-2	控件的添加.....	118
5-1-3	控件的排列.....	119
5-1-4	控件的通用属性.....	119
5-2	基本控件.....	122
5-2-1	Button 控件.....	122
5-2-2	Label 控件.....	123
5-2-3	TextBox 控件.....	123
5-2-4	PictureBox 控件.....	123
5-2-5	Timer 控件.....	123
5-2-6	基本控件综合实例.....	124
5-3	常用控件.....	128
5-3-1	CheckBox 和 RadioButton 控件.....	128
5-3-2	CheckedListBox 控件.....	129
5-3-3	ComboBox 控件.....	130
5-3-4	DateTimePicker 控件.....	131
5-3-5	GroupBox 控件.....	133
5-3-6	ImageList 控件.....	133
5-3-7	LinkLabel 控件.....	135
5-3-8	ListBox 控件.....	136
5-3-9	Dialog 控件.....	136
5-3-10	ProgressBar 控件.....	141
5-3-11	ScrollBar 控件.....	142
5-3-12	常用控件综合实例 1.....	144
5-3-13	常用控件综合实例 2.....	148
5-4	体现 Windows 风格的控件.....	151
5-4-1	MainMenu 控件.....	151
5-4-2	ToolBar 控件.....	153
5-4-3	ContextMenu 控件.....	155
5-4-4	NotifyIcon 控件.....	157
5-4-5	HelpProvider 控件.....	158
5-4-6	ListView 控件.....	160
5-4-7	StatusBar 控件.....	166
5-4-8	ToolTip 控件.....	168
5-4-9	TreeView 控件.....	170
	本章小结.....	177
第 6 章	面向对象的程序设计方法.....	179
6-1	面向对象的基本概念.....	180
6-1-1	对象与类.....	180
6-1-2	对象的组成.....	180



6-1-3	封装、继承与多态性.....	181
6-1-4	VB.NET 对面向对象支持能力.....	182
6-2	对象的生存周期.....	182
6-2-1	对象的创建.....	182
6-2-2	对象的使用.....	183
6-2-3	对象的释放.....	187
6-3	类的创建.....	188
6-3-1	类的声明.....	189
6-3-2	类的成员.....	190
6-3-3	类与命名空间.....	191
6-3-4	定义成员变量.....	193
6-3-5	定义成员属性.....	193
6-3-6	定义成员方法.....	197
6-3-7	定义成员事件.....	201
6-3-8	共享方法和共享成员.....	208
6-4	类的继承.....	211
6-4-1	继承的概念.....	211
6-4-2	基本继承.....	211
6-4-3	限制继承.....	214
6-4-4	交叉语言的继承.....	215
6-4-5	可视化继承.....	218
6-5	多态性.....	219
6-5-1	重复定义.....	221
6-5-2	多重定义.....	222
6-5-3	接口.....	225
6-6	模块与类.....	229
	本章小结.....	231
第7章	编写文件应用程序.....	233
7-1	文件及其读写性概述.....	234
7-1-1	文件及其访问类型.....	234
7-1-2	System.IO 命名空间.....	235
7-1-3	Stream 类.....	236
7-1-4	FileStream 类.....	237
7-1-5	BinaryReader 类.....	238
7-1-6	BinaryWriter 类.....	238
7-1-7	StreamReader 和 StreamWriter 类.....	239
7-2	文件类和目录类.....	240
7-2-1	File 文件类.....	240
7-2-2	Directory 目录类.....	241

7-2-3 FSO 模型	242
7-3 文件操作实例	247
7-3-1 功能模块分析	247
7-3-2 界面设计	247
7-3-3 代码实现	249
本章小结	260
第 8 章 编写数据库应用程序	261
8-1 数据库基础知识	262
8-1-1 数据库与数据模型简介	262
8-1-2 关系型数据库	264
8-2 创建简单的数据库	266
8-2-1 创建数据库文件	266
8-2-2 创建表及表之间的关系	273
8-3 SQL 基础	274
8-3-1 SQL 基本概念	274
8-3-2 SQL 基本语法	275
8-4 ADO.NET 技术概述	285
8-4-1 ADO.NET 简介	285
8-4-2 OleDbConnection 对象	287
8-4-3 OleDbCommand 对象	291
8-4-4 OleDbDataReader 对象	293
8-4-5 OleDbDataAdapter 对象	294
8-4-6 DataSet 对象	296
8-4-7 使用 Sql 对象访问 Sql 数据库	300
8-4-8 使用 DataGrid 控件	302
8-5 数据库应用程序实例	303
8-5-1 数据关系分析	303
8-5-2 功能模块划分	304
8-5-3 界面设计	304
8-5-4 代码实现	306
本章小结	320
第 9 章 应用程序的部署	321
9-1 部署应用程序的基本概念	322
9-1-1 编译可执行文件	322
9-1-2 程序组件和 DLL 陷阱	323
9-1-3 XCOPY 形式的应用程序部署	324
9-1-4 已有的应用程序部署方式	324
9-2 利用部署向导生成安装项目	326
9-2-1 使用部署向导生成安装项目	326

9-2-2	编译安装项目.....	329
9-2-3	发布并执行安装文件.....	336
9-3	利用部署工程生成安装文件.....	337
9-3-1	添加新项.....	337
9-3-2	为安装项目添加内容.....	337
9-3-3	添加指定文件夹和快捷方式.....	339
	本章小结.....	341
第 10 章	应用程序调试.....	343
10-1	规范化程序代码.....	344
10-1-1	注释代码.....	344
10-1-2	合适的命名规则.....	346
10-1-3	使用文本编辑器工具栏.....	347
10-1-4	相关 IDE 选项设置.....	348
10-2	使用调试工具.....	349
10-2-1	程序执行方式.....	349
10-2-2	Debug 工具栏.....	351
10-2-3	调试窗口.....	353
10-2-4	输出窗口和任务列表窗口.....	357
10-3	异常处理.....	359
10-3-1	错误和异常.....	359
10-3-2	非结构化的异常处理.....	360
10-3-3	结构化的异常处理.....	361
10-3-4	Try 语句实例.....	362
	本章小结.....	366

1

Chapter

Visual Basic.NET 的新特性

本章导读：

随着 Internet 的飞速发展，现有的开发平台和开发环境，不论是从开发技术上，还是从开发模式上越来越无法满足 Internet 时代需要的基于 Web 的应用程序和 Web 服务。需求即是商机，作为软件业界的老大，微软（Microsoft）早已抢先下手，于 2002 年 11 月 13 日正式宣布了 Visual Studio.NET Beta1 版的试用。新一代的开发平台——.NET Platform，提出了许多新概念，实现了诸多新技术的飞跃，其目的就是提供开发分布式网络应用程序的方便快捷的工具和技术，以加快网络应用的飞速发展，满足日益增长的网络服务的需要。这无疑又是给软件开发业带来了一场新的技术革命。

本章知识点：

- Visual Studio.NET 框架
- Visual Basic.NET 的新特性



1-1 Visual Studio.NET 框架

Visual Basic.NET 是 Visual Studio.NET 不可分割的一部分，所以我想有必要再对 Visual Studio.NET 作一番简介。Visual Studio.NET Platform 是一个软件开发平台，而不仅仅是简单的一个开发工具或一门计算机语言，其主要构成有：.NET 框架 (.NET Framework)、.NET Building Block Services、.NET 企业服务 (Enterprise Servers) 和 Microsoft Visual Studio.NET。至于什么是 .NET，有许多不同的解释，有人认为 .NET 是一种全新的下一代可视化开发环境；有人认为 .NET 是一种新的针对 Internet 时代的开发语言——C#；有人认为它是基于 XML (可扩展标记语言) 和 SOAP (简单对象访问协议) 的新型信息交换平台，是面向未来的企业级的开发平台；也有人认为 .NET 只不过是 Windows DNA 技术的演变。这些看法应该是——横看成岭侧成峰，远近高低各不同，都是立于不同的角度来评价定义 .NET。我想再详细的定义可能比尔·盖茨也不会满意，下面只能对 .NET 从内容及结构上给予一定的阐述，以帮助大家的理解。

1-1-1 .NET 框架的结构

.NET 框架又被称为 Web 服务引擎，它提取出微软组件对象模型 (COM) 的精华，将它们与松散耦合计算的精华有机地结合在一起，生成了强大、高效的 Web 组件系统：简化程序员的“管道”操作、深入地集成了安全性，引进了基于互联网的操作系统，极大地改善应用程序的可靠性和可扩展性。.NET 框架的各个组成部分如图 1-1 所示，其主要部分是这三大块：通用语言运行库 (Common Language Runtime, 简称 CLR)、具有多层次结构的统一的类库集合 (Framework Class Library) 和高级版“活动服务器页面” (ASP.NET 又名 ASP+)。剩余的两块，一个是用户界面 (User Interface)，用于开发基于 Windows 上的更强大的人机交互界面；另一个就是从老版本的 ADO (Active Data Object) 升级上来的功能更为强大的新一代数据访问技术 ADO.NET，它支持断线编程模式，同时还支持可扩展标识语言 XML (eXtensible Markup Language)。



图 1-1 .NET 框架组成结构

1-1-2 .NET 框架的主要内容

上文提到,.NET 框架的主要部分是通用语言运行库(CLR)、统一的编程类(Class Library)和 ASP.NET,这三部分是整体框架的基础。下面分别给予简单的介绍,以让读者在一堆的英文专业名词前能有一个概要的认识。

通用语言运行库 (CLR)

此名称不能准确反映它的全部功能。实际上,通用语言运行库在组件的开发过程中以及软件的运行过程中,都扮演着非常重要的角色,在整个.NET 框架中,从技术角度上看,最重要的概念莫过于通用语言运行库(Common Language Runtime)。在组件运行过程中,运行库负责管理内存分配、启动或取消线程和进程、实施安全性策略、同时满足当前组件对其他组件的需求。在开发阶段,运行库的作用有些变化:与现今的 COM 相比,运行库的自动化程度大为提高(比如可自动执行内存管理),因而开发人员的工作变得非常轻松。尤其是,映射功能将使代码编写量锐减,这些代码是开发人员在将业务逻辑转化成可复用的组件进行编程时所需的。

对编程语言而言,运行库这个概念并不新奇:实际上每种编程语言都有自己的运行库。Visual Basic 开发系统具有最为明显的运行库(名为 VBRUN), Visual C++跟 Visual FoxPro、JScript、SmallTalk、Perl、Python 和 Java 一样,有一个运行库 MSVCRT。NET 框架的关键作用是它提供了一个跨编程语言的统一的编程环境,这也是它能独树一帜的根本原因所在。

从技术角度出发,如果把系统服务看成.NET 框架的基础的话,那 CLR 可以看作.NET 框架的核心。对于软件开发人员而言,理解.NET 的关键之处就在于对 CLR 的理解。实际上,Windows 操作系统本身就可以认为是运行库和库的集合体。运行库和库一起为应用程序提供服务,在一定程度上极大地节省了时间,并且有利于代码的重用。比较 COM 的编程模式可以更好地理解 CLR 的编程模式,对于 COM 而言,它的编程思想是基于类型而不是面向文件的。在这一点上,CLR 也是采用这种办法。在最早的 Windows 编程中,当需要调用某 DLL 接口时,一般利用 LoadLibrary 函数,然后再调用 GetProcAddress 函数,随着 COM 的出现,CoCreateInstance 和 QueryInterface 函数改变了这一切。CLR 也是以类型为中心的。虽然在编程模式上 CLR 与 COM 一样,但在实现方式上是不一样的。CLR 克服了一些 COM 本身固有的弊端,例如类型信息格式不统一、私有类型信息不能接触等。首先,在 CLR 中组件的概念成为头等类公民。在 COM 中,表示组件的方式有很多,对象、类、动态链接库都可以表示组件;而在 CLR 之中组件的概念是以集合(Assembly)的形式出现的,对于每一种类型,COM 采用 128 比特的 UUID 进行定义,而在 CLR 中为了更好地确保惟一性,采用了 128 字节的公钥和在局部范围内保持惟一的类型名称来提供全球惟一表示。当一个客户端应用程序调用某集合时,在客户应用程序中存有一 64 位的公钥 Hash 值,从而能确保被调用的集合是正确的集合。第二,对于 CLR,只有一种元数据交换格式存在。在 COM 编程中,需要在 IDL 中定义类型信息,然后再利用一种具体的语言(C++或 Java)去实现。在 CLR 中开发人员可以在任何一种语言中定义并实现该类型。第三,元数据是完全可扩展的。任何语言都可以扩展 CLR 的类型信息。第四,在 COM 中存在两种类型系统, IUnknown 型和 Variant 型;在 CLR

中所有的类型都来自 System.Object。第五，在 CLR 中允许出现接口的多继承。总而言之，正如不了解 COM 技术就无法真正了解 Windows 一样，理解 CLR 对于了解 .NET 是非常重要的。其实，对于一般的程序员来讲，可以简单地认为 CLR 就是 COM 技术向前的巨大飞跃。

统一的编程类（.NET Framework Class Library）

.NET 框架中的类为开发人员提供了一个统一的、面向对象的、层次化的、可扩展的类库集（API）。每一种基础类都定义了 .NET 平台潜在的某些属性。属性相似的基础类被包含到同一命名空间中（Namespace）。现今，C++ 开发人员使用的是微软基础类库，Java 开发人员使用的是 Windows 基础类库，而 Visual Basic 用户使用的又是 Visual Basic API 集。简而言之，.NET 框架统一了微软当前各种不同的框架。这样，开发人员不再需要学习多种框架就能顺利编程。远不止于此的是，通过创建一个公共的跨编程语言的 API 集，.NET 框架可实现跨语言继承性、错误处理功能和调试功能。实际上，从 JScript 到 C++ 的所有编程语言，都是相互等同的，开发人员可以自由选择理想的编程语言。Microsoft 的口号，是实现应用程序跨平台的无缝移植。在 .NET 平台下，Microsoft 确实实现了其内部各种语言间的轻松互换。.NET 提出的一些新概念和技术，为将来的跨平台的移植提供了发展的基础，虽然目前为止，真正意义上的完全跨平台的应用程序尚未出现，即使 Java 也是在为实现这一目的而努力。

这样我们可以想象得到，在 .NET 开发平台下，其体系内的各种语言如 Visual Basic.NET，C#，C++，JScript 等，除了语法上的差异外，其调用的类库是相同的。这也是为什么当熟悉了 Visual Basic 6.0 的程序员，在一开始接触到 Visual Basic.NET 的时候，大大地惊讶其变化，甚至更多的人是把其中的两种语言 C# 和 Visual Basic.NET 视为孪生兄弟。具体在谈到 Visual Basic 的新特性时，读者还将看到更为详细的介绍。

ASP .NET(又称 ASP+)

熟悉 Visual Basic 的程序员，对 ASP 肯定也不陌生，ASP 的全称是 Active Server Page，称为“活动服务器页面”，相当于 Java 脚本（JavaScript）一样的一种语言，不过它运行于服务器端，一般嵌于 HTML 文件中。当用户连接服务器端的页面时，ASP 将动态地将服务器端的内容解释为 HTML 页面反馈回客户端。ASP 的运用，大大推动了 Internet 的发展。

ASP+ 是使用 .NET 框架提供的类库构建而成的，它提供了一个 Web 应用程序模型，该模型由一组控件和一个基本结构组成。有了它，Web 应用程序的构建变得非常容易。开发人员可以直接使用 ASP+ 控件集，该控件集封装了公共的、用于超文本标记语言（HTML）用户界面的各种小组件（诸如文本框、下拉菜单等）。实际上，这些控件运行在 Web 服务器上，它们将用户界面转换成 HTML 格式后再发送给浏览器。在服务器上，控件负责将面向对象的编程模型呈现给 Web 开发人员，这种编程模型能提供面向对象的编程技术拥有的丰富功能。ASP+ 还提供一些基本结构服务（诸如会话状态管理和进程循环），这些服务进一步减少了开发人员要编写的代码量，并使应用程序的可靠性得到了大幅度提高。ASP+ 还允许开发人员将软件作为一项服务进行传送。通过使用 ASP+ Web 服务功能，ASP+ 开发人员只需进行简单的业务逻辑编程，而由 ASP+ 基本结构负责通过 SOAP 传送服务。

尽管 ASP+ 还未正式发行，但它已在改进应用程序功能方面创造出令人难以置信的奇迹：

在现有基于 ASP 的应用程序性能基础上,性能优化了三倍之多,更为激动人心的是生产效率再度攀升。

1-1-3 .NET 框架的核心要素

.NET 框架有几个要素值得一提。首先是它的安全系统和配置系统。这两个系统协同工作,有力地遏止了运行不安全代码的可能性,并大幅度减少了号称“DLL Hell”的对应用程序进行配置时所面临的挑战。

安全系统是一个高度细化、基于事实的系统,它赋予开发人员和管理员多种代码处理权限(而不仅仅是“On”或“Off”)。将来,还会根据代码本身的核心要素来决定如何实施上述权限。

例如,当.NET 框架应用程序被下载到某一系统中时,它会申请一组权限(诸如对临时目录的写入权限)。运行库将收集有关应用程序的事实信息(诸如:它是从何处下载的、是否用了有效签名、甚至它访问系统的准确程度),并按管理策略决定是否允许应用程序运行。运行库甚至还可告之应用程序它无法授权申请的所有权限,并允许应用程序自行决定是否继续运行。

有这种安全系统作保障,许多应用程序配置问题便会迎刃而解。开发人员和管理员(最终是用户)所面临的最大挑战之一是版本的管理问题。如果在您新装了某个应用程序之后,一切都限于瘫痪状态,而在这之前系统一直运行得非常良好,那么最大的可能是新安装的应用程序重写了一些共享库,并极有可能修正了现有应用程序正使用的程序错误。这种情况出现的频率很高,以致人们将它称为:“DLL Hell”。

.NET 框架拥有的几项高级功能可以彻底消除“DLL Hell”现象。首先,它有一个非常强大的内部命名系统,能够有效地防止两个库因互相重名而被错当为对方的情况发生。除此之外,它还提供一项被称作“并行”配置的新功能。如果前例中新安装的应用程序确实重写了共享库,现有应用程序可对该库进行修复。等现有应用程序再次启动时,它会检查所有的共享文件。如果发现文件被更改,同时这些更改又是不兼容的,则它可以请求运行库提取一个它可以使用的版本。得益于强大的安全系统,运行库可以安全地执行该操作,这样应用程序就完成了本身的修复工作。

1-1-4 .NET 的新特性

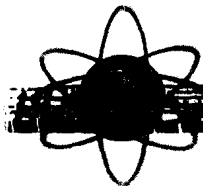
综上所述,.NET 是一种全新的技术,它包括了很多新特性。这里我们简单的归纳出一些比较重要的特性。

1. 一致的编程模式

在.NET 环境中,所有的应用程序都采用通用的面向对象编程模式,不再像 windows 环境中那样,既有 DLL 函数也有 COM 对象。

2. 简化了的编程模式

这也许是最令开发人员欢欣鼓舞的消息了,在.NET 环境下,由于 CLR 的作用,在进行编程时不再需要掌握 GUIDs、IUnknown、AddRef 等令人头疼的 COM 知识了。



3. 运行于多个平台

对于任何操作平台，只要支持.NET 运行时均可以运行.NET 应用程序。现在所有的 Windows 平台均可以实现这一点。在将来甚至可以运行在非 Windows 操作系统上。

4. 支持多语言的综合

按照 COM 的原理，代码重用是建立在二进制代码的级别上。在.NET 环境下，代码重用可以建立在源码的级别上的，也就是说，别人用 C#语言写的某个类可以直接在 C++这样的语言中使用。之所以.NET 有这样的巨大威力在于.NET 为所有的支持.NET 编程方式的语言提供了一整套通用的类型系统。

5. 自动资源管理

对于所有应用程序开发人员来说，最头疼的就是内存泄露问题。在.NET 环境下，这个问题得到彻底解决，自动资源管理功能已经加入到 CLR 之中。同时，由于资源回收功能的加入，使安全性在一定程度上得到了保障，诸如内存溢出攻击等将得到有效控制。

6. 一致的出错处理方式

相信所有的 Windows 开发程序员都对 Windows 环境下混乱的错误处理方式感到厌烦，Win32 错误代码、异常情况处理、Hresult 等。在.NET 环境下所有的程序都采用统一的错误处理方式——产生异常。

7. 安全性

正如我们已经知道的一样，.NET 的出现是为了迎合下一代 Internet 环境下的企业级计算，一般的访问控制已经不能满足它的要求，所以在安全性方面.NET 相对于 Windows 等其他系统而言，有了更深入的改进。从装载一个类开始，就进行确认检查；在访问代码和相应资源时，又实施代码访问安全措施。.NET 提供了一整套机制来判断角色和确认身份信息，并且能做到跨进程和机器从而确保所需的代码在远程没有受到破坏。.NET 的安全性也深深地嵌入到 CLR 结构中，以确保应用程序本身的安全。这些安全机制是对现有操作系统安全机制的一种质上的扩展，比较起来，.NET 在安全性上得到了进一步的加强。

8. XML 和 SOAP (Simple Object Access Protocol) 的引入

回忆一下过去分布式应用程序的设计，过去我们设计两层的应用程序，在此基础上出现了诸如 CORBA、IIOP、RMI 和 DCOM 这样的协议。人们已经熟悉了这样的分布式系统。但是分布式系统的弊端就是灵活性差，因为这种设计方式使得应用程序固定在服务器端。而 Internet 是个松散连接、非常分布的世界。原有的 Client/Server 结构已经过时，这样就提出了全新的编程模式，而 XML 和 SOAP 能使这种模式很好地工作。在.NET 中 XML 和 SOAP 已经深深地溶入其中，并成为非常重要的组成部分。

SOAP (Simple Object Access Protocol) 简单对象访问协议是在分散或分布式的环境中交换信息的简单的协议，是一个基于 XML 的协议。

9. 全新的编程语言 C#

随着.NET 的推出，微软也强力推出了一种新型的编程语言 C#。C#像 Visual Basic 一样简单，又像 C++一样强大。在有些人的眼里，C#是 Java 的复制品。这种说法似乎有一些道理，因为 C#太像 Java 了。但正确地说，C#绝不是 Java 的克隆，C#是微软在研究了 C、C++、Java、Modula2、SmallTalk 等大量语言的基础上推出的语言，比起 Java 来，C#的最大不同之处在于它更接近 C++，同时 C#也吸收了大量新的概念，例如 C#是面向组件的语言，C#能作到与