

BOOKS FOR PROFESSIONALS BY PROFESSIONALS™

The Ultimate VB.NET and ASP.NET Code Book

支持
VS.NET 2002 & VS.NET 2003

VB.NET 和 ASP.NET 编程手册

- ❖ 提供了大量可以直接运行的代码例程，以及.NET 领域最新的编程技术
- ❖ 揭露缩短程序开发周期的秘诀，内容涉及 Windows 应用程序、Web 应用程序、数据库、Web 服务、智能设备项目和移动 Web 项目等
- ❖ 涵盖有关 C# 编程语言的必备知识

(美) Karl Moore 著
高升 李耀华 译



清华大学出版社

VB.NET 和 ASP.NET 编程手册

(美) Karl Moore 著
高升 李耀华 译

清华大学出版社

北 京

内 容 简 介

本书从实用角度出发,对如何使用 VB.NET 和 ASP.NET 开发应用程序做了系统、全面的介绍。全书共分 9 章,每章都先列出基本的知识点,然后再深入探讨开发人员在编程时遇到的常见问题,并给出了简明、快捷的解决方案。主要内容包括升级 VB6,创建大型 Windows 应用程序,快速建立 Web 站点,数据处理,Web 服务,ASP.NET 技术揭密,创建可以在掌上电脑、手机、微波炉上运行的应用程序,讨论正则表达式,以及为什么需要了解 C#等,并对 C#的基础知识做了简明扼要的概述。

本书既适合经验丰富的 VB.NET 和 ASP.NET 开发人员,也适合 .NET 初学者,具有很高的学习参考价值。

EISBN: 1-59059-106-2

The Ultimate VB.NET and ASP.NET Code Book

Karl Moore

Original English language edition published by Apress L. P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA. Copyright ©2003 by Apress L.P. Simplified Chinese-Language edition copyright ©2004 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 出版公司授权清华大学出版社出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2003-8761

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

VB.NET 和 ASP.NET 编程手册/(美)摩尔(Moore,K.)著;高升,李耀华译.—北京:清华大学出版社,2004

书名原文: The Ultimate VB.NET and ASP.NET Code Book

ISBN 7-302-08102-6

I. V… II. ①摩…②高…③李… III. ①BASIC 语言—程序设计②主页制作—程序设计

IV. ①TP312②TP393.092

中国版本图书馆 CIP 数据核字(2004)第 011175 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客 户 服 务: 010-62776969

组稿编辑: 曹康

文稿编辑: 崔伟

封面设计: 康博

版式设计: 康博

印 刷 者: 国防工业出版社印刷厂

装 订 者: 三河市李旗庄少明装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 20.5 字数: 425 千字

版 次: 2004 年 3 月第 1 版 2004 年 3 月第 1 次印刷

书 号: ISBN 7-302-08102-6/TP·5857

印 数: 1~4000

定 价: 38.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

前 言

“我既不是特别聪明，也不是很有天赋，我只是非常非常勤奋而已。”

—— 爱因斯坦

目前市面上的大多数.NET 图书似乎没有提供真正的代码。读者购买一本厚达 1500 页的.NET 图书，由于书中没有讲述有关程序设计方面的内容，读者在学习一个月后，仍然对如何把最基本的程序组合在一起一无所知。

而实际上，这些书的作者对程序设计方面也知之不多。

我认为，这些作者在研讨会上都能指出程序在理论上的设计方式。但在展示其代码时，又都不能提供真正的代码。

本书就不是这样。它并不很厚，但包含了大量的内容，既有许多有用的代码，又有.NET 编程提示，这些提示是我在过去三年内开发并总结出来的。

这里要澄清一下：这些代码并不是更新后的 Visual Basic 6 代码：它们都是为 VB.NET 和 ASP.NET 专门创建的，并通过了测试。这些代码都是非常有用的，我不想说出在编著本书的过程中，编写和放弃了多少代码。

本书只留下了最好的代码，而读者将拥有这些代码。

在本书中，读者将学习如何创建 XP 样式的界面，如何编写可以通过 Web 自动更新的小型 Windows 应用程序，理解 ASP.NET 的许多技术秘密，了解如何利用 Microsoft 代码节省开发时间，如何把 DataGrid 应用于 Windows 应用程序和 Web，如何在应用程序中添加 Google、Amazon 和 TerraServer 等强大的搜索功能。

本书还将介绍如何创建可以在 PDA、手机、微波炉等机器上运行的快速程序，讨论一种隐含的.NET 语言，论述为什么需要了解 C#，并对 C#的基础知识进行简单的介绍。

无论读者是一个编程高手，还是一个.NET 初学者，本书都颇具参考价值，我个人对有机会编写本书感到非常荣幸。

多谢读者与我一起分享本书。

如何使用本书

本书有两个主要用途。第一，由于本书采用了 Coke 模式，因此可以用作.NET 参考书。最好把它放在计算机旁边，这样还可以使洁净的书桌不会蒙上灰尘。在几个月后，

这本书看起来会比较破旧，但至少它会使读者有兴趣使用.NET。

第二，读者可以根据自己的喜好从前向后阅读本书，也可以从后向前阅读。如果读者是.NET新手(特别是从 Visual Basic 6 升级而来)，最好从“升级 VB6”一章开始，该章概述了.NET 的基础知识，并创建了一个基本的 Windows 应用程序和一个简单的 ASP.NET Web 站点，最后讨论了核心语言的一些变化。

如果读者以前使用过 VB.NET 或 ASP.NET，但还没有掌握所有的专业知识和技术，就可以直接从“创建大型 Windows 应用程序”跳到“隐含的.NET 语言”一章，本书的详细目录有助于读者找到对自己最有用的章节。

另外，如果读者不想输入本书中的许多代码，可以从www.apress.com上下载所有的源代码，以及在本书出版后新添加的代码。但要把本书放在手边，因为在下载代码时需要先回答几个验证身份问题。

尽情享受本书和 VB 编程的乐趣吧！

Karl Moore, 2003.7

英格兰 约克郡

目 录

第 1 章 升级 VB6	1
1.1 .NET 概述.....	1
1.2 探讨 Visual Studio .NET	2
1.3 创建 Web 应用程序	7
1.4 代码的变化	9
1.4.1 理解 Framework	9
1.4.2 整数的升级.....	10
1.4.3 字符串.....	11
1.4.4 Variant 和 Currency 数据类型.....	11
1.4.5 数组的变化.....	11
1.4.6 新添加的运算符.....	12
1.4.7 声明属性.....	12
1.4.8 用户定义的类型.....	13
1.4.9 作用域的变化.....	13
1.4.10 按值传递是默认的.....	13
1.4.11 不再使用 Set 语句	13
1.4.12 错误处理的变化.....	14
1.4.13 Class 关键字.....	15
1.5 本章小结	15
第 2 章 创建大型 Windows 应用程序	16
2.1 基础知识	17
2.2 开发秘诀	20
2.2.1 使窗体透明.....	21
2.2.2 谁偷走了工具提示.....	21
2.2.3 改变对象的大小.....	21
2.2.4 拆分面板、设置资源管理器样式的五个步骤.....	22
2.2.5 用 ErrorProvider 突出显示错误	22
2.2.6 学习 LinkLabel 控件.....	23
2.2.7 定制 MonthCalendar 控件	24
2.2.8 创建只能输入数字的文本框.....	24

2.2.9	显示 GIF 动画, 但不使用 Browser 控件	25
2.2.10	改变 Tab 键顺序的两种方法	25
2.2.11	使用系统任务栏	26
2.2.12	节省时间: 为组合框添加自动完成功能	27
2.2.13	命令行参数的功能	29
2.2.14	如何重置窗体	29
2.2.15	如何将光标置于按钮上	31
2.2.16	简便快捷地捕获屏幕	32
2.2.17	性能极佳的浏览工具栏	33
2.2.18	利用 Windows XP 主题的七个步骤	34
2.2.19	检查以前实例的 .NET 方式	35
2.2.20	把 RTF 转换为 HTML	36
2.2.21	从 Windows 资源管理器中拖放	40
2.2.22	对话框: 用户单击了什么	41
2.2.23	文本打印类	42
2.2.24	.PrintForm 的秘密重生	45
2.2.25	可视化继承	47
2.2.26	查看窗口、性能计数器等	47
2.3	使用 obfuscator 保护代码	49
2.3.1	创建瘦客户程序	50
2.3.2	下载全部应用程序更新的简单方法	51
第 3 章	快速建立 Web 站点	53
3.1	基础知识	53
3.2	开发秘诀	58
3.2.1	用户界面	58
3.2.2	编码技巧	58
3.2.3	优化、错误和其他技巧	59
3.3	用户界面	59
3.3.1	验证控件的五个步骤	60
3.3.2	显示 Web 警告消息: 方法 1	61
3.3.3	显示 Web 警告消息: 方法 2	63
3.3.4	创建默认的 Enter 按钮	65
3.3.5	鲜为人知的 SmartNavigation 属性	65
3.3.6	用户控件揭秘	66
3.3.7	Panel 控件非常有用	67

3.3.8	上下移动列表中的项目	68
3.3.9	在 Web 应用程序中改变控件的尺寸	69
3.3.10	禁止用户右击鼠标	69
3.3.11	创建可滚动的宏窗口	69
3.3.12	为什么要学习 CSS	70
3.4	编码技巧	71
3.4.1	在代码中修改页面标题的三个步骤	71
3.4.2	自动填充电子邮件属性值	72
3.4.3	在 ASP.NET 中发送邮件	73
3.4.4	创建用户友好的 URL	74
3.4.5	在页面中添加实时 HTML	75
3.4.6	上载文件的秘诀	76
3.4.7	在数据库中存储上载的文件	77
3.4.8	使用上载的图像	79
3.4.9	动态创建图像	81
3.4.10	生成缩略图	82
3.4.11	ASP.NET 身份验证的五个步骤	84
3.4.12	不使用 Web.config 的 Forms 身份验证	86
3.4.13	只验证站点的部分页面	86
3.4.14	存储设置的最佳位置	87
3.4.15	利用 Microsoft 的论坛代码	88
3.4.16	集成 PayPal 的即时付费通告功能	89
3.5	优化、错误和其他提示	90
3.5.1	Web 应用程序中的子文件夹	90
3.5.2	选择除 c:\inetpub\wwwroot\之外的目录	90
3.5.3	用高速缓存创建快速的 ASP.NET 应用程序	91
3.5.4	成功调试的九个步骤	92
3.5.5	对客户隐藏错误代码	93
3.5.6	不要使用 404 错误: 定制自己的“Page Not Found”错误	94
3.5.7	Server.Transfer 和 Response.Redirect	96
3.5.8	对较长的过程使用.IsClientConnected	97
3.5.9	用元标记阻止客户机高速缓存	97
3.5.10	上载大于 4MB 的文件	98
3.5.11	在 Session_End 不工作时如何操作	98
3.5.12	监视 Web 主机: 在 .NET 中检查其正常运行的时间	99

3.5.13	对 Web 应用程序进行压力测试	100
3.5.14	迁移 Web 服务器的两个方法	102
3.5.15	把文件放在 ASP.NET 主机的什么地方	103
3.5.16	在安装 Visual Studio .NET 之后安装 IIS	104
第 4 章	处理数据	105
4.1	基础知识	105
4.2	开发秘诀	110
4.2.1	快速生成 GUID	111
4.2.2	生成自己的连接字符串创建器	111
4.2.3	查找最近添加的 ID 编号	112
4.2.4	使用 SQL 的技巧	113
4.2.5	把多个表返回到 DataSet 中	114
4.2.6	检查 SQL Server 是否可用	116
4.2.7	快速建立可编辑的 Windows 栅格的七个步骤	117
4.2.8	快速建立可编辑的 Web 栅格的九个步骤	119
4.2.9	在 Web 栅格中使用 HTML	125
4.2.10	在 Web 栅格中使用超链接	125
4.2.11	日期、货币、百分比：格式化 Web 栅格的数据	126
4.2.12	对 Web 栅格进行彩色编码	127
4.2.13	确认 Web 删除的技术	129
4.2.14	采用 Hotmail 样式一次选择 Web 窗体中的多个数据项	130
4.2.15	在 Web 栅格中单击并选择	131
4.2.16	在 Web 栅格中使用下拉框	133
4.2.17	快速、个性化的 Web 数据绑定	140
4.2.18	利用 Web 栅格进行快速而简单的数据分页	141
4.2.19	利用 Web 栅格快速排序	143
4.2.20	把 Web 栅格导出到 Excel 中的简单方法	144
4.2.21	从 Excel 电子表格中返回 DataSet	146
4.2.22	利用 Microsoft 编写代码：无官方支持的 SQL Server 工具	148
第 5 章	Web 服务揭密	149
5.1	基础知识	149
5.2	开发秘诀	151
5.2.1	公开数据库信息的简便方法	152
5.2.2	在公开 Web 服务之前需要做的五项工作	154

5.2.3	用简单的高速缓存提高性能	156
5.2.4	在线翻译	156
5.2.5	在程序中添加 Google 搜索功能	157
5.2.6	查询 Amazon.com Web 服务	159
5.2.7	利用 TerraServer 在应用程序中查看各种信息	163
5.2.8	异步访问：在后台调用 Web 服务	166
5.2.9	Web 服务、代理服务器、连接已关闭	167
5.2.10	改变 Web 服务的 URL，而无需重新编译	167
5.2.11	在何处查找最好的 Web 服务	168
第 6 章	从微波炉到掌上电脑：特殊的项目类型	169
6.1	插入 Windows：创建自己的 Windows 服务	169
6.2	安装服务	171
6.3	创建基本的控制台应用程序	172
6.4	从手机到微波炉：利用 MIT 创建应用程序	174
6.5	创建移动 Web 应用程序	176
6.6	编写支持移动设备的代码	177
6.7	测试移动 Web 应用程序	179
6.8	便携式计算：为 PDA 创建应用程序	179
6.9	为 Compact Framework 构建应用程序	180
6.10	部署应用程序	182
6.11	其他参考资料	183
第 7 章	.NET 的更多秘密	185
7.1	开发秘诀	185
7.2	使用 Internet	187
7.2.1	创建自己的 Web 浏览器	188
7.2.2	提取 Web 页面的 HTML	189
7.2.3	利用超时提取 HTML	190
7.2.4	解析 Web 页面中的链接和图像	191
7.2.5	把 HTML 转换为文本	193
7.2.6	把数据传送给 Web 的方法	194
7.2.7	给收藏夹添加 Web 快捷方式	195
7.2.8	获取 IP 地址	197
7.2.9	Internet 连接可用吗	197
7.3	处理文件和文件夹	198

7.3.1	用于读写文件的两个函数	198
7.3.2	移动和删除文件，以及查看文件属性等方法	200
7.3.3	检查两个文件是否相同	200
7.3.4	处理临时文件的技巧	202
7.3.5	目录操作	203
7.3.6	监视目录的变化	204
7.3.7	文件的大小	205
7.3.8	获取指定文件夹的路径	206
7.3.9	处理文件扩展名的程序	206
7.3.10	获取磁盘的系列号	208
7.3.11	App.Path 在 .NET 中的替代物	208
7.3.12	INI 文件在 .NET 中的用法	209
7.4	日期、数字和字符串	212
7.4.1	它是一个整数吗	212
7.4.2	检查日期的 .NET 智能方式	213
7.4.3	在应用程序中使用序数	214
7.4.4	随机数的工作方式	216
7.4.5	确定一个月有多少天	217
7.4.6	对日期、月份、年份进行加减运算	217
7.4.7	计算下一个工作日	218
7.4.8	确定是否闰年	218
7.4.9	确定季度	219
7.4.10	计算两个日期之间相隔几年	220
7.4.11	把字符串转换为正确的大小写格式	221
7.4.12	把文本存储到剪贴板上	222
7.4.13	自动生成容易记忆的密码	222
7.4.14	用 12 行代码加密数据	224
7.4.15	实现 MD5 加密方法	225
7.4.16	把字符串转换为颜色类型	227
7.4.17	把组合框与枚举值绑定起来	227
7.5	图形和字体	229
7.5.1	设计自己的图标	229
7.5.2	使用字体的基本要求	230
7.5.3	图形格式之间的转换	230
7.5.4	图形的旋转和翻转	231

7.5.5	用 Windows 窗体绘图	232
7.5.6	在代码中添加渐变背景	234
7.5.7	启动自己的屏幕保护程序	235
7.6	使用注册表和事件日志	236
7.6.1	读写注册表	236
7.6.2	把消息放在事件日志中	238
7.7	分布式计算	239
7.7.1	XML 简介	239
7.7.2	用 COM+ 进行基本事务处理的六个步骤	247
7.7.3	使用 MSMQ	251
7.7.4	选择 Web 服务还是远程技术	254
7.8	Visual Studio 开发环境	255
7.8.1	编写 Developer TODO: 列表	255
7.8.2	在工具栏中存储常用的代码	256
7.8.3	用文件夹组织项目	256
7.8.4	使用命令窗口	257
7.8.5	确定是否在 IDE 中运行	258
7.8.6	通过录制宏来节省时间	258
7.8.7	使用 VS .NET 命令提示	258
7.8.8	升级、COM 和 API	259
第 8 章	隐含的 .NET 语言	260
8.1	工作原理	260
8.2	Regex 库	261
8.2.1	只检查一个数字	262
8.2.2	匹配实数	263
8.2.3	匹配字母和数字, 但不匹配空格和句点	263
8.2.4	检查 24 小时格式的时间	264
8.2.5	识别有效的日期	264
8.2.6	检查文件的路径和扩展名	264
8.2.7	检查重复的单词	265
8.2.8	获取首字母大写的单词	266
8.2.9	匹配字符串中的数字	267
8.2.10	姓名检查器	267
8.2.11	过滤恶意图语的表达式	268
8.2.12	检查电子邮件地址的有效性	268

8.2.13	检查 Web 站点地址的有效性	269
8.2.14	Internet URL 匹配器: FTP、HTTP、HTTPS	269
8.2.15	检查域的有效性	271
8.2.16	IP 地址检查器	271
8.2.17	从 HTML 中提取链接和图像	272
8.2.18	检查 HTML 颜色代码	272
8.2.19	检查信用卡的有效性	272
8.2.20	强制密码格式的表达式	273
8.2.21	用表达式定义自己的 HTML: 定制标记	273
8.2.22	ISBN 检查器	276
8.2.23	检查 GUID	276
8.2.24	检查美国邮政编码	277
8.2.25	检查美国社会安全号码	277
8.2.26	检查美国电话号码	278
8.2.27	检查美国的州名	278
8.2.28	匹配英国的邮政编码	279
8.2.29	检查英国的国家安全号码	279
8.2.30	检查英国的电话号码	280
8.2.31	转换美国和英国的日期	280
8.2.32	法国、德国和日本的表达式	281
8.2.33	“松散”表达式的修正方案	281
第 9 章	C#转换指南	283
9.1	把 C#转换为 VB.NET	283
9.2	转换列表	284
9.2.1	注释: 在 C#中使用//	285
9.2.2	去掉行末的分号	286
9.2.3	数据类型: int、bool、float 和 DateTime	286
9.2.4	函数: public bool writeEventLog(string entry)	286
9.2.5	方法: public void activateAlarm()	287
9.2.6	变量: string strText;	287
9.2.7	字符串内容: \n 和 @	288
9.2.8	对象: myObject = new myClass();	288
9.2.9	作用域: public、private、internal 和 static	289
9.2.10	参数: ref 和 out 关键字	289
9.2.11	算术运算: x++;	290

9.2.12	If-Then 语句: <code>if (x > y) { ... } else { ... }</code>	290
9.2.13	比较: <code>==</code> 、 <code>!=</code> 、 <code>&</code> 和 <code> </code>	291
9.2.14	Select Case: <code>switch {x} { ... }</code>	291
9.2.15	循环语句: <code>for (x=1; x<=10; x+=1) { ... }</code>	292
9.2.16	错误: <code>try { ... } catch { ... } finally { ... }</code>	294
9.2.17	<code>this</code> 关键字	294
9.2.18	事件: <code>obj.event += new class.delegate(handler);</code>	295
9.2.19	类: 属性	295
9.2.20	类: 构造器和终止器	296
9.2.21	类接口: <code>myInterface myObject2 =(myInterface)myObject1;</code>	297
9.2.22	类继承关键字: <code>base, virtual, abstract, sealed</code>	298
9.3	转换技巧	299
9.4	其他的 C#参考资料	299
附录	301

第1章 升级VB6

有关 VB.NET 的许多文章都许诺，VB.NET 可以大幅度减少程序的开发时间，还在促销广告中展示，用户一旦开始使用 VB.NET，就可以拥有迷人的硅谷生活方式。在最近的 PDC 促销活动中，还会赠送一件印有“憎恨 Linux，热爱 .NET”字样的 T 恤衫。

读者可能听说过 .NET，知道它是 Visual Basic 的一个新版本。但就是这么一个 VB 新版本，就可以在目前的新系统中提升程序的运行效率吗？没错，下面就介绍这个新版本。

首先介绍 .NET 的概况以及 Visual Basic 的适用场合。接着讨论 .NET 的新开发环境，在创建一个简单的应用程序时说明它与 VB6 的区别。

本章还将学习新的项目类型，并说明它们在本书的哪些地方作了进一步的讨论。最后，用大量的篇幅解释如何修改代码，包括核心数据类型的变化和新关键字。

女士们，先生们，请系好安全带，飞往 .NET 的航班就要起飞了。首先是一段有点颠簸的旅程……

1.1 .NET 概述

.NET 是一个程序，还是一个 Web 站点，抑或是一个不识字的康沃尔郡人的钓鱼设备？

都不是，那么，.NET 究竟是什么？

按照官方的解释，.NET “是一个 XML Web 服务平台，是允许把软件创建为服务的一种工作方式。”它是基于分布式计算的一个 Microsoft 版本，无论采用什么操作系统、设备或编程语言，都可以通过 Internet 获取信息。

根据软件巨头 Microsoft 的说法，这个 .NET 计划包含五个核心部分，但这五个核心部分还没有完全实现：

- 开发产品
- 服务器产品(如 Windows 2000/2003 和 SQL Server)
- 基础服务(如 Passport 和 Alert)
- 设备(如掌上电脑)
- 配套软件(如 MSN 和 Office .NET)

对于开发人员来说，.NET 指的是上述列表中的第一项——开发产品，它把两个重

要的软件组合在一起：.NET Framework 和 Visual Studio .NET。

免费的 .NET Framework 是整个 .NET 理念的核心，也是分布式编程的核心。这个程序基于操作系统，主要用于运行 .NET 应用程序（一般的 Windows 应用程序或 Web 站点）。它给程序分配内存，在其“基类”中提供了一个功能库。它可以使用 XML 与远程计算机通信，完成用户不想插手的所有内部工作，它抽象、隐藏了程序的复杂性，而且工作得很好。但是，.NET Framework 实际上只是 Windows 的一个升级。

提示：

可以把 .NET Framework 看作程序的智能运行库。如果没有安装它，就不能运行 .NET 应用程序。为了使 .NET Framework 随 Windows 2003 之后的版本一起发布，Microsoft 必须使后续的修订版本均具有向后兼容性（当前版本是 1.1）。偶尔会有人谈论，有公司正在开发运行在其他系统上的 .NET Framework 版本，目前 Microsoft 可以乐观一点：现在的 .NET Framework 只能用于 Windows 系统。

接着谈谈 Visual Studio .NET，它是 Visual Studio 的下一个版本。目前已发布了两个 Visual Studio .NET 版本：2002 版和 2003 版（Codename “Everett”）。这两个版本都包含 Visual C# .NET、Visual C++ .NET 和 Visual Basic .NET (VB.NET) 语言。

在这些语言中，C# 是一种全新的语言。据传 .NET 是比尔·盖茨钟爱的一个产品，其语法非常类似于 VB.NET（详见第 9 章）。之后是 C++ .NET，该语言是对原来的 C++ 语言进行扩展后，绑进 .NET Framework 中的。Everett 用户还会发现 J#，它是 Microsoft 为 Java 开发人员打开 .NET 大门的一把钥匙。最后是 VB.NET，这是 Visual Basic 的新版本，对我们来说是最重要的产品。

提示：

ASP.NET 适用于所有的场合吗？ASP.NET 不是一种语言，它只是一个涵盖 Internet 项目的术语，是指 .NET Framework 中的一组类库。在 VB.NET、C# 或其他 .NET 语言中可以使用这些类。本书的后面将阐述 ASP.NET。

在开始使用 VB.NET 之前，需要在机器上安装 .NET Framework 和 Visual Studio .NET。如果还没有安装它们，可以参阅附录 I。安装好后，就可以享用新的开发环境了。

1.2 探讨 Visual Studio .NET

如果还没有安装和配置 Visual Studio .NET，可以参考附录 I，按照其中的指示进行安装和配置。完成后，就可以使用它了。

(1) 选择 Start | Programs | Microsoft Visual Studio .NET | Microsoft Visual Studio .NET, 启动 Visual Studio .NET。

此时会看到一个彩色的起始页面, 它替代了 Visual Basic 6 中平淡无奇的 New 或 Open 弹出对话框。该页面的左边还有在线链接, 使这个页面变成一个可定制的开发入口。

(2) 单击 New Project 按钮。

这个对话框类似于 VB6 中的欢迎对话框(如图 1-1 所示)。它允许您选择要创建的项目类型。但是, 这个对话框中的选项与 VB6 中的大不相同。

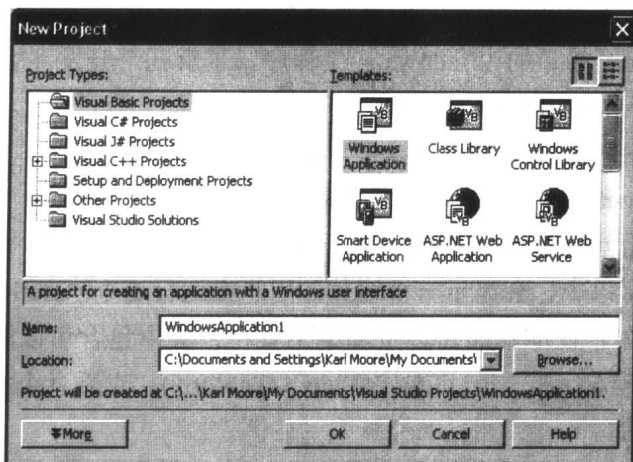


图 1-1 创建新的 Windows 应用程序项目

首先, 可以用三四种默认语言中的任一种来创建项目。本书当然只选择 Visual Basic 语言来创建, 但即使这样, 可用的项目也令人眼花缭乱。下面就一一解释。

可以创建 Windows 应用程序, 它相当于旧版本中的标准 EXE 程序。桌面应用程序永远都不会消亡(这些应用程序详见第 2 章)。接着就是类库了, COM 在 .NET 中已经销声匿迹了, 对应的类型是 ActiveX DLL 项目。之后是 Windows 控件库, 它允许您建立自己的控件, 这是一种 ActiveX 控件项目。

再往后是 ASP.NET Web 应用程序, 它把 ASP 的所有 Web 功能和 Visual Basic 的拖放功能及编码功能组合在一起, 允许用户快速创建功能全面的 Web 站点(详见第 3 章)。

列表中的下一项是 ASP.NET Web 服务, 这类项目允许把代码方法或函数通过网络公布于众。之后, 其他应用程序就可以调用这些代码块, 得到需要的结果。如果读者以前使用过 VB6 中的简单对象访问协议(SOAP), 这就是 .NET 的一个内部实现方式(详见第 5 章)。

下一项是 Web 控件库, 通过它可以构建在 Web 页面上使用的控件。这些控件建立在 HTML 的基础之上, 与 ActiveX 没有任何关系。列表中的下一项是控制台应用程序, 它允许创建类似于 DOS 的控制台程序, 而这种 DOS 控制台程序在 VB6 中是很难创建