

.NET 开发系列丛书

# VB .NET

## 程序设计



光盘含本书范例  
及对应程序代码

康际科技 著  
杨剑峰 常晓波 改编



.NET 开发系列丛书

# VB .NET

# 程序设计

康际科技 著  
杨剑峰 常晓波 改编

中国电力出版社

05  
10  
02

~42

## 内 容 提 要

Microsoft 为 VB .NET 新增了许多功能,使 VB .NET 成为完全面向对象的编程语言。VB .NET 的一个核心概念是“公用语言运行环境”,借助这个概念可以实现真正的跨平台开发。本书主要内容包括:VS .NET、VB .NET 的面向对象语法、VB .NET Windows 应用程序、VB .NET Web 应用程序、ADO .NET 与数据控件等。本书还通过编写大量应用程序范例,深入讲述了 VB .NET 的功能与技巧。

本书适合对 VB .NET 开发环境感兴趣的程序开发人员及广大电脑爱好者阅读。

## 图书在版编目(CIP)数据

VB .NET 程序设计/康际科技著;杨剑峰,常晓波改编.  
—北京:中国电力出版社,2002.6  
ISBN 7-5083-1070-5

I. V... II. ①康...②杨... III. BASIC 语言—程序设计  
IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 032893 号

著作权合同登记号:图字 01-2002-1214 号

## 版 权 声 明

本书为台湾金禾资讯股份有限公司独家授权的中文简体字版本。本书专有出版权属中国电力出版社所有。在没有得到本书原出版者和本书出版者书面许可时,任何单位和个人不得擅自摘抄、复制本书的一部分或全部,以任何方式(包括资料和出版物)进行传播。本书原版版权属金禾资讯股份有限公司。版权所有,侵权必究。

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://www.infopower.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

\*

2003 年 1 月第一版 2003 年 1 月北京第一次印刷

787 毫米×1092 毫米 16 开本 26 印张 631 千字

定价 39.80 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题,我社发行部负责退换)

# 序

总算大功告成了！

在改编的过程中，发现 VS.NET 确实是一个非常好的开发环境。以前的开发非常受限于语言：不仅在功能上，更在管理上。试想，即使每个开发人员都使用一种功能“最”为强大的语言（如果有）来开发，作主管的去哪里找足够的人员呢？有了 VS.NET，一切都不同了。VS.NET 的一个核心概念是“公用语言运行环境”，借助这个概念，可以实现真正的跨平台开发。这样，不仅对独立开发程序的个人用户引用现成的组件带来了很大的方便，更为团队的开发在技术和管理上带来一场革命。

VB.NET 是 VS.NET 中的一门核心语言，其功能强大，而又简单易用。本书由浅入深地介绍了如何在 VS.NET 环境中开发 VB.NET 应用程序，既适合于准备学习 VB 的人员，也适合于希望从 VB 6.0 快速迁移到 VB.NET 环境的人员。

改编过程中，所有范例都做了调试，纠正了原书中一些错误。这样读者可以把精力集中在范例本身的实现上，更高效地进入 VB.NET 的领域！

改编者

007

# 原 序

随着 Internet 的蓬勃发展，编程技术的进步也一日千里。您或许会抱怨：不久前才刚把 VB 6.0 提高到一个阶段，现在 Visual Basic.NET 又来了，是不是又要重新开始另一段学习之旅？答案是“否”。如果您已经熟悉了 VB 6.0 的开发，那学习 Visual Basic.NET 并没有您想像得那么困难。而如果您刚刚准备或者刚刚开始学习 VB 6.0，那么这正是时候，请直接跳到 Visual Basic.NET 领域吧！为什么这样说？如果您已经熟悉 VB 6.0 的开发，那祝贺您，您已经具备了 Visual Basic.NET 的基本功了，学习和掌握 Visual Basic.NET 对您来说，只要再迈出一小步就可以获得高效开发动态网站的技巧。而如果您正准备或者刚刚开始学习 Visual Basic，则直接跳到 Visual Basic.NET 可以节省很多摸索时间，因为前人已经将研究改进 Visual Basic 的大树种好了，您只要在树荫下轻松地学习 Visual Basic.NET，即可站在前人的肩膀上往前迈进！

康际科技

# 导 读

## 0.1 哪些人应该阅读本书？

### 0.1.1 一般的程序设计人员

若您是程序设计人员，不管您会使用哪一种程序语言，例如：VB/Basic、C/C++、dBase/FoxPro、Delphi、Power Builder 等，只要有简单的逻辑观念都可以通过本书学习到 Visual Basic.NET 的基础。

### 0.1.2 具有网页撰写经验的程序设计人员

如果您曾经撰写过动态网页程序，不论您是使用 IDC、Java、ASP 等动态网页开发工具，本书都将是您踏入 ASP.NET 的最佳选择。

### 0.1.3 对撰写动态网页非常精通的程序设计人员

若您对撰写网页非常精通，具有利用 COM、COM+来撰写动态网页的经验，则本书更能为您提供快速的对 ASP.NET 的认识，并且学习到 ASP.NET 的精神。

### 0.1.4 网站设计、规划系统分析师和规划人员

若您是网站设计、规划的系统分析师或是规划人员，若您不了解 ASP.NET 能为您提供哪些强有力的服务，相信您很难发挥 ASP.NET 的优势，因而也就无法高效率地开发您所想要的内容。通过本书您可以了解 ASP.NET 带给您的所有好处。

### 0.1.5 已经阅读过其他 ASP.NET 书籍的人

您或许已经阅读过其他 ASP.NET 书籍，甚至正在使用 ASP.NET 进行网站内容的开发，您可以通过本书的内容，发现 ASP.NET 还有其他的功能您尚未了解，本书详尽的说明可以让您对 ASP.NET 有耳目一新的感觉。

## 0.2 本书涵盖内容及建议阅读方式

为了让您快速领略及学习到 Visual Basic.NET 如何协助您高效率地建置动态网站，本书将用 7 章，带领您循序渐进地掌握 Visual Basic.NET。

## **第 1 章: .NET**

在本章中您可以了解到“.NET”是什么?为什么要使用 Visual Basic.NET? Visual Basic.NET 的特色是什么?这可以让您对 Visual Basic.NET 有一番了解后,进而让您更加发挥 Visual Basic.NET 的威力。

## **第 2 章: Visual Studio.NET**

当您已经了解 Visual Basic.NET 后,准备要大展身手之前,准备一些必要的工具,以便于快速学习及体验 Visual Basic.NET 的威力与所带来的快感。Visual Studio.NET 是开发 Microsoft .NET Framework 基础框架下的应用程序的最佳开发工具。本章以详实的说明,教您如何搭建起一个学习环境,所以本章是您不可不读的章节,“工欲善其事,必先利其器”,不是吗?

## **第 3 章: 面向对象的程序设计**

Visual Basic.NET 已经完全是面向对象的程序设计。本章将介绍面向对象程序设计的概念,包括继承、多态、覆盖、重载等设计观念。让您对面向对象有进一步的认识。

## **第 4 章: Visual Basic.NET 的面向对象语法**

进一步介绍 Visual Basic.NET 的基础语法及数据类型的用法,及面向对象的程序撰写方式及多线程的概念等。

## **第 5 章: Visual Basic.NET Windows 应用程序**

介绍 .NET Framework Windows 应用程序的制作方式。如何利用 Visual Studio.NET 快速地开发 Windows 应用程序项目。

## **第 6 章: Visual Basic.NET Web 应用程序**

.NET Framework 的 Web 应用程序就是 ASP.NET。本章将详细介绍 Web Form 的控件如何使用,以及 Web 应用程序的开发制作。

## **第 7 章: ADO.NET 与数据控件**

介绍 ADO.NET 的组件是如何与数据库连结,并详细说明 Web 控件及 Windows 窗体控件是如何与 ADO.NET 结合,显示数据库内的数据。

## **第 8 章: Visual Basic.NET 范例**

本章通过一个自制迷你计算器的范例,让读者加深对面向对象程序设计的认识,从而让读者尽快进入面向对象程序设计的领域。本章还通过一个范例介绍了在 Visual Basic.NET 中制作 Crystal Report 报表的方法。

## **0.3 本书体例**

为了方便您阅读本书,现将本书所用体例说明如下:

- 当内容中提到屏幕上的按钮时，会使用【】括住，并且里面的文字会使用粗体字表示，例如：【确定】、【取消】。

- 当内容中提到屏幕上的功能时，会使用【】括住，但里面的文字会用一般字体表示，例如：【文件】、【编辑】。

- 当提到鼠标单击过程时，会使用下面的表示方式，有些单击时是使用鼠标左键，而有些单击是使用鼠标右键，这些都遵循 Windows 惯例，本书不再对此进行说明，除非是使用到鼠标其他键的功能，才会特别指出。

例如：【开始】→【程序】→【管理工具】→【事件查看器】。

这表示请您使用鼠标单击任务栏上的【开始】按钮，当显示菜单时，把鼠标指针移动到【程序】，当显示下一级菜单时，再将鼠标指针移动到【管理工具】，显示下一级菜单后，单击【事件查看器】。

- 当内容中需要您在某些输入窗口中通过键盘键入一些文字、命令等，会用斜体字表示。例如：ASP.NET。

- 当内容中要请您注意的地方，会采用下面的方式表示：



此处会给出您应该注意的内容。

- 在章节中有时会提到一些小技巧，会用以下的方式说明。这是提高您的技能的重要信息，千万不要错过。



此处会给出一些技巧，请您细细品味一下。



# 目 录

序

原 序

导 读

|                                           |     |
|-------------------------------------------|-----|
| 第 1 章 .NET .....                          | 1   |
| 1.1 .NET 与 .NET Framework 简介 .....        | 1   |
| 1.2 ASP.NET 说明 .....                      | 6   |
| 第 2 章 Visual Studio .NET .....            | 10  |
| 2.1 安装前的准备 .....                          | 10  |
| 2.2 安装 Visual Studio .NET .....           | 10  |
| 2.3 Visual Studio .NET 简介 .....           | 15  |
| 第 3 章 面向对象的程序设计 .....                     | 28  |
| 3.1 面向对象与面向过程 .....                       | 28  |
| 3.2 对象与类 .....                            | 31  |
| 第 4 章 Visual Basic.NET 的面向对象语法 .....      | 37  |
| 4.1 Visual Basic.NET 基本语法 .....           | 37  |
| 4.2 Visual Studio.NET 的控制台应用程序项目类型 .....  | 55  |
| 4.3 面向对象程序语法 .....                        | 62  |
| 4.4 总结 .....                              | 72  |
| 第 5 章 Visual Basic.NET Windows 应用程序 ..... | 73  |
| 5.1 Windows 应用程序 .....                    | 73  |
| 5.2 Windows 窗体及组件 .....                   | 81  |
| 5.3 .NET Framework SDK Document .....     | 143 |
| 第 6 章 Visual Basic .NET Web 应用程序 .....    | 145 |
| 6.1 Web 控件 .....                          | 145 |
| 6.2 数据验证控件 .....                          | 228 |
| 6.3 总结 .....                              | 244 |
| 第 7 章 ADO.NET 与数据控件 .....                 | 245 |

|                                        |            |
|----------------------------------------|------------|
| 7.1 ADO.NET .....                      | 245        |
| 7.2 Web 控件数据控件 .....                   | 274        |
| 7.3 Windows 窗体数据控件 .....               | 317        |
| 7.4 总结 .....                           | 344        |
| <b>第 8 章 Visual Basic.NET 范例 .....</b> | <b>345</b> |
| 8.1 自制迷你计算器 .....                      | 345        |
| 8.2 简单的 CrystalReport 报表制作 .....       | 396        |

# 第 1 章 .NET

在介绍“Visual Basic.NET”之前，首先让您初步了解一下什么是“.NET”。因为 Visual Basic.NET 是 .NET 框架中的一环，所以当您对 .NET 有一个完整的概念之后，就更容易了解 Visual Basic.NET 在整个框架中扮演何种角色，因而也就更容易发挥 Visual Basic.NET 的威力。

## 1.1 .NET 与 .NET Framework 简介

2000 年 6 月，微软公司在美国召开了一个专业开发人员研讨会，会上透漏了有关微软在 Windows 及互连网络方面下一代产品的信息，微软对将来涉及生活和工作的方方面面都提供了完整的解决方案，这个方案就是“.NET”（您可以浏览中国微软网站上关于 .NET 的内容：<http://www.microsoft.com/china/net/default.asp>）。

.NET 基本上是框架在 Web 服务策略下，微软目前正通过完整的 .NET 框架平台（包含各项组件）来为 Web Services 提供稳固的基础开发框架。.NET 框架大致上可以分为下面几个部分：

### 1.1.1 .NET Framework（这是开发程序的核心部分）

Framework 提供了建立及部署 Web 服务的环境，这包含了通用语言运行时（Common Language Runtime）、多层次的统一对象类库（unified class libraries），以及 ASP.NET 等部分。

### 1.1.2 .NET Product

所有微软主要旗舰产品，从 Windows 到 Office 都被集成在 .NET 框架下，产品都通过 XML 来描述数据，并利用 SOAP 作为产品之间沟通的协议，这也就是 .NET 集成的框架。

这样说明好像太笼统了，但如果您只想了解什么是 .NET 的话，上述的说明可以让您大概知道：将所有产品通过 XML（extensible Markup Language）来描述数据，并通过 SOAP（Simple Object Access Protocol）在产品之间交换数据，进而集成产品的新框架，这就是 .NET 的基本精神。

到此，若您不想继续深入了解 .NET 框架，您可以略过下面 .NET 框架的说明，直接跳到下一节。不过多花十分钟还是非常值得的，因为下面所提到的概念，绝对可以让您对 .NET 了解得更多些。

首先，.NET Framework 包含的范围非常广，从系统层次到应用程序开发层次都包含在其中，.NET Framework 提供许多工具用来集成各个层次。其次 .NET Framework 整个框架都是为了让您快速地、更容易地开发互连网络上的应用程序而设计的。所以 .NET Framework

提供了很多组件以便您更高效地开发应用程序，也就是说，您不用一行一行地编写程序，而只需要引用组件或将组件稍加修改就可以符合您的需要，设计出您所需要的程序。例如，以往在网页上制作类似客户端/服务器应用程序中的 Grid，必须花费很多时间对每个元素的控制做调整或设计。现在方便多了，因为 .NET Framework 提供了现成的组件供您使用，从而可以快速地设计出非常美观的网页。

请先参考图 1-1，图中是 .NET Framework 的简易框架示意图，从图中可以看到，.NET Framework 分作几个大的层次，每一个层次分别具有不同的作用（当然每一层次还可以进一步细分下去）。下面就针对每一个层次的功能作一个说明（本书的重点放在 ASP.NET，因此，若您想要更深入地了解 .NET Framework，请您参考网址：<http://msdn.microsoft.com>）。

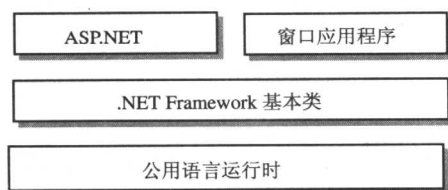
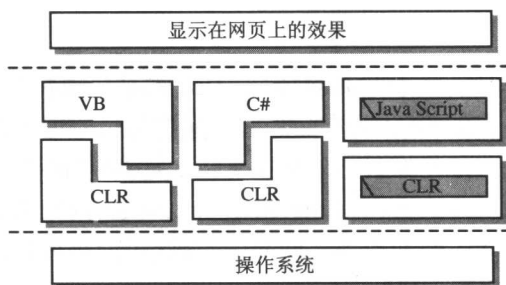


图 1-1

### 1.1.3 公用语言运行时（Common Language Runtime, CLR）

CLR 是 .NET Framework 的核心部分。一般来说，运行时间（Runtime）是提供程序执行的环境，所以 CLR 也是提供 .NET 程序编译（Compile）后的执行环境。或许您会有疑问，在以前编写 VB 或 C 的程序时，也是要编译和链接（Link）后才能成为可执行文件，然后再将这个可执行文件与动态链接库文件一起放在执行程序端才能执行，那 .NET 程序既然需要编译，那需不需要链接呢？这个问题非常好（笔者在刚开始学习 ASP.NET 时也有同样的疑问）！微软提供的 CLR 环境可以让使用不同程序语言开发的程序在同一个平台上共同使用（请参考图 1-2）。



不管您使用何种语言来开发 .NET 应用程序，通过 CLR 都可以达到同样的效果。也就是说，每种程序设计语言各有千秋，您可以用 VB 完成一部分程序的编写，用 C# 完成另外一部分程序的编写，而通过 CLR 将不同部分融合在一起，达到您想要的目的。

图 1-2

也就是说，只要可以编译成 MSIL（Microsoft Intermediate Language）程序语言，都可

以作为制作 ASP.NET 网页的程序语言。这就告诉您：只要将程序进行编译即可以提供执行，而根本就不需要链接成可执行文件。那 MSIL 是什么？请先参考图 1-3 您大概就会了解 MSIL 是什么了。

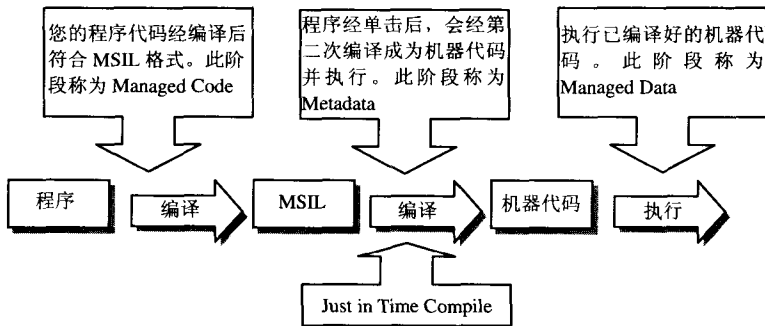


图 1-3

由图 1-3 可以看到，MSIL 是介于程序代码与机器代码之间的语言（所以您并不用编写 MSIL 程序），当您程序编译成 MSIL 代码后，只要用户单击程序，就会快速地进一步编译并执行。

看到这里您一定会想到：何不直接将程序编译成机器代码，然后再执行呢？为什么要如此大费周折地绕了一圈？（没错！笔者之前也有同样的疑问。）其实，MSIL 程序代码非常接近机器代码，从 MSIL 程序代码再编译成机器代码并执行是非常快的。不同类型的 CPU 可以有不同的 JIT 编译器，如此便可以将同样的 MSIL 程序代码在不同类型 CPU 的机器上执行，请参考图 1-4。

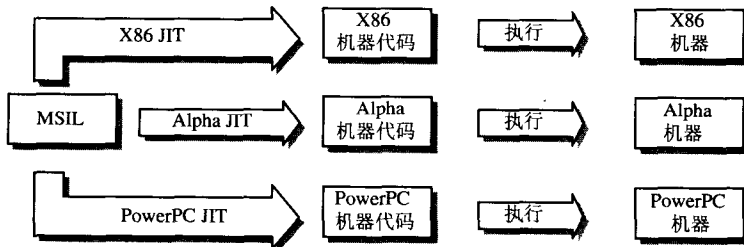


图 1-4

从图 1-4 中可以了解到，利用这样的执行框架，编写一种程序代码便可以跨平台执行程序，这就是为什么运用 MSIL 就可以实现跨平台了。虽然可以跨平台执行同样的程序是一个很大的好处，但是会不会因此而减缓执行的效率？当然会，不过要看是从哪一个角度来考虑了。若是 ASP.NET 网页首次执行，那速度自然是比较慢的（这是与首次解释并执行 ASP 网页的情况做比较），但是当 ASP.NET 网页执行过一次后，便会在服务器上保存起来，在第二次执行时，只需花费从 MSIL 编译成机器代码的时间，就可以执行程序了；相比之下，第二次执行 ASP 程序时，服务器还要重新解释，所以与首次执行 ASP 网页的时间相同，如此

便比第二次执行 ASP.NET 网页的时间长。一般网页在制作好后，被浏览执行的次数相当的多，所以虽然首次执行时速度较慢，但在以后多次浏览时节省了更多的时间，这是非常有意义的，请参考图 1-5。

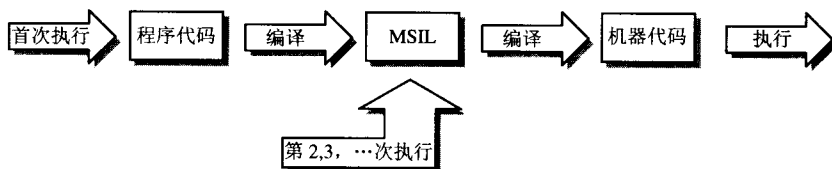


图 1-5

在此补充一下，还记得在图 1-3 中，当您的程序代码经过第一次编译后成为 MSIL 程序代码时（此时称为“Managed Code”），所得到的 MSIL 程序代码与 CPU 所可以执行的机器代码并没有任何关系，也就是不管 CPU 是哪一种类型，所编译出来的 MSIL 程序代码都是一样的。那么，在 MSIL 程序代码中都包含了些什么呢？MSIL 程序代码只包含：

- MSIL 定义的命令集（广泛定义的命令集）
- 对象的加载、保存、初始化以及调用其他方法、组件等命令
- 有关数学运算、逻辑运算、流程控制、内存访问及异常处理等命令

因为上述命令集已经将 ASP.NET 程序的绝大部分都编译好了，所以在执行时，只需转换成 CPU 可以识别的格式便可以执行，因此，利用 JIT 编译器转换 CPU 命令集是非常快速的。

第一次编译后的程序代码为什么称为“Managed Code”呢？这是因为 .NET 是采用编译的方式进行，当利用 JIT 编译成可执行的机器代码后，会将此机器代码保存在内存中以便使用；如果原始程序经修改后又重新编译成为 MSIL 程序代码，就已经纳入管理了，当再次单击或执行此程序时，.NET Framework 就会非常聪明地使用新的 MSIL 程序代码进行编译和执行，所以在此阶段就称为 Managed Code（再次提醒您，在此阶段的程序代码与 CPU 还没有任何关系）。

那什么是 Metadata（元数据）？元数据是存放在内存中或 .NET Framework Portable Executable（PE）文件中的二进制文件（此时已经通过 JIT 编译器生成了 CPU 可以识别的机器代码），JIT 编译器将程序、程序所参考的文件和数据类型编译在一起，也就是把这些元素集中在一起以待执行时使用；因为程序代码在此阶段已起了变化所以称为“Metadata（元数据）”。

运行时因为要管理程序的生命周期（在此生命周期指的是：当原始程序修改、编译成 MSIL 程序代码后，当用户重新执行程序时，将自动移除内存中的旧程序，再将新程序放入执行，而之前旧程序所执行的时间就称为生命周期）、对象的使用与释放等工作，所以在此阶段就称为“Managed Data”。

相信到此阶段您应该了解 CLR 的重要程度吧！先思考一下整个框架，您就会发觉 .NET Framework 的好处确实很多。前面提到的 CLR 可以让您所编写的程序跨平台执行，接下来要说明的是，应用 .NET Framework 的框架，可以将不同程序语言所编写的组件集成到您的

程序当中，也就是跨程序语言。

一般而言，每一个程序设计人员都有自己较为熟悉的程序设计语言，相信您也不例外，如果您所熟悉的程序设计语言符合 MSIL，则您只要将程序编写完毕，就可以编译并执行您的程序。下面从两个角度来说明跨程序语言的做法。

#### 1.1.4 个人独立开发程序

在您独立开发程序时，有时候可能需要引用一些组件来提高开发的效率，当然您所引用的组件必须符合 MSIL 规范，不过您却不必考虑此组件是用何种程序语言开发的。不同的组件有可能是用 VB 编写而成的，也有可能是用 C# 开发完成的，所以，以往使用某种程序语言做开发时，只能引用同一种程序语言所开发的组件，这相当不便。而在 .NET Framework 中，只要程序语言符合 MSIL 规范，跨程序设计语言引用其他程序语言开发的组件是一件非常容易的事。

#### 1.1.5 群体开发程序

与其他伙伴群体一起开发程序时，若全部局限在同一种程序语言进行开发系统，在管理上和技术支持上可以发挥很好的效果，但是在人力资源上会受到许多的限制，因为要找熟悉同一种程序语言的人不是一件很容易的事。有时，明明知道另一种程序语言提供了某种组件，而团队所用的程序语言却必须一行一行地重新编写，这时，您会觉得为什么不用另一种语言开发。应用 .NET Framework 编写时，不必限制每个开发人员使用何种程序语言。例如，有的人可以使用 C#，其他人可以使用 VB 或 ASP.NET，并且开发出来的组件都可以交互使用。这样便加速了程序的开发，也提高了组件的可用性。

谈了好一阵子的 CLR，并说明了跨平台及跨程序语言的好处后，若您仍然觉得犹未尽，建议您到 <http://msdn.microsoft.com/library/default.asp?URL=/library/dotnet/cpguide/cpconthenetframeworkclasslibrary.htm> 参考 .NET Framework 的详细资料，相信您会有更大的收获。

#### 1.1.6 .NET Framework 基本类

在以前开发程序时，有时候必须引用 Win32API，而有时候又必须使用 COM 对象，若是使用 ASP 技术开发要引用 Win32API 或 COM 时，又必须通过 ASP.DLL 来访问这些组件，说起来都非常简单，但是在编写程序时要正确引用这些组件却不是一件容易的事。现在，应用 .NET Framework 的开发方式，这样的困难就不复存在了，请您参考图 1-6。

从图 1-6 中可以看到，应用 .NET Framework 方式开发，都是应用同一种方式引用组件，您不用再为引用 Win32API 还是引用 COM 组件而困扰。现在您只要会使用 .NET Framework 基本类就足够了，为了让您能够快速开发应用程序，.NET Framework 提供了很多组件供您使用，并且这些组件已经包含了数据结构、文件访问、异常处理等诸多方面的功能，因此，您只要学会使用组件，便可以大幅提高开发效率，毕竟“时间就是金钱”。有关如何使用 .NET Framework 组件方面的内容，将在后面的章节中进行介绍。

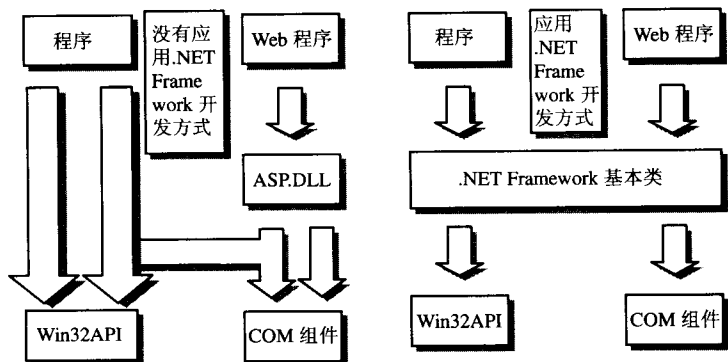


图 1-6

## 1.2 ASP.NET 说明

为什么要使用 ASP.NET 来开发 Web 应用程序？这是 ASP 程序员最常问的问题，先不提应用 .NET Framework 既可以跨平台又可以跨程序语言这个优点，ASP 程序员一般都认为：用 ASP 开发 Web 程序已经非常方便且快速，为什么还要学 ASP.NET 的开发方式？其实，ASP 程序员认为 ASP 已经非常容易使用，这是程序员发挥坚韧不拔的精神，不知不觉地学习、磨练而得来的成果，也就是说，成果是美好的，但是学习过程却是千辛万苦的。然而，学习及编写 ASP.NET 是一个不同的体验，因为微软在开发 ASP.NET 时，就是以方便程序员学习及开发作为首要任务的，这样，程序员即便不太熟悉如何编写 HTML，也能编写出非常完美的程序。以下将说明 ASP.NET 对您来说有着什么样的好处。

### 1.2.1 不再需要编写大量的程序代码

利用 ASP.NET 编写应用程序系统最大的好处在于：不再需要编写大量的程序代码。为什么呢？以往在编写 Web 程序时，程序员除了要编写许多处理程序（含数据库处理部分）外，另外还要将 HTML 的标记语言（Tag Language）包含到程序中，不仅如此，还要费力地将网页上有关输入字段部分，利用 VBScript 或 JavaScript 编写额外的验证程序，这些都完成了才算是编写完毕。而现在，只需要利用 ASP.NET 所提供的组件，加上设置适当的参数便可以完成您要实现的功能，并将前端验证的程序也一并处理完毕，这不仅少写了许多程序，同时也节省了很多时间。不仅如此，以往编写 Web 程序必须将 HTML 的工作一并处理，可是通常程序员对网页编排部分的天分不是很好（笔者就是一个美工白痴），所以写一个 Web 程序，有可能花费 70% 在处理网页的问题，剩下的 30% 才是编写程序，编写完后还要再加上给美工进行修正的工作，一件工作重复处理，浪费了许多宝贵的时间。现在，ASP.NET 让程序员与界面工程师（在此指的是处理用户界面画面的美工人员）可以分开并合作设计，也就是程序员专心在程序设计上，而界面工程师专门处理用户界面，最后再将双方的工作结合起来并使其顺利运作。这样不但可以节省许多程序设计人员的时间（减少



编写处理网页的程序), 并且一件事不必重复处理, 如此亦可以缩短开发周期, 相对地节省了开发成本, 这就是 ASP.NET 所提出的“Code Behind”概念。

以往要在网页间传递信息时, 必须在前一个网页上自行编写一些程序代码以保存状态信息, 以备网页更新时可以应用这些信息而保存原来的网页。例如, 在前一个页面设置好过滤数据库的条件后, 就可能需要记录一下, 以备当下一网页要利用自行编写回上页的链接回来时, 不会因为状态的改变而重新设置过滤的数据。现在方便多了, 因为 ASP.NET 会自动记录状态以便随数据内容的改变而自动改变网页的内容, 而不必重新设置条件。

以往利用 ASP 编写程序时, 您可能已经有意识地尽量使用结构化的方式开发, 尽可能使用过程或函数将程序模块化, 以便将来可以高效地重用这些程序。如果您是这样做的, 说明您的 ASP 功力已经非一般人可比, 不过您回头看一下走过的路, 是不是发觉自己是经过无数个痛苦日子, 披荆斩棘地走过来的? 这条路不好走啊! 若您还没有经历过这样的路, 那就恭喜您了, 因为 ASP.NET 是采用组件方式开发, 所以绝大多数的程序, 都可以使用现有的组件堆叠起来, 并且因为应用组件开发的方式, 当您编写自己的组件后, 以后您可以在其他系统中非常方便地引用这些组件(这就是 ASP.NET 所提出的“User Control”的概念), 也就是提高再利用率的意思, 当您的组件越来越多、越来越完整时, 要组合出一个完整的应用系统就成为一件很简单的事, 不仅快速, 而且不用编写太多的程序, 您所要做的只是设置参数而已。

### 1.2.2 执行方式由解释改为编译

在前一节时, 我们已经说明过 .NET Framework 是利用编译的方式来执行程序的, 而 ASP 是利用解释的方式执行程序, 两者的执行效率问题已经介绍过, 所以在此不再赘述。

### 1.2.3 数据类型的明确化

利用(脚本)类型的程序语言编写程序时, 经常会遇到在程序执行时变量的数据类型发生错误的问题(这是因为这些变量要到执行的时候, 通过用户输入才知道是怎样的数据类型, 就算您事先知道应该是要输入字符串或数字, 但是也是要等到运行时才能判别数据类型)。因为程序是利用解释的方式进行, 不仅解释程序需要花时间判断此变量究竟是何种数据类型, 才能让处理器进行处理, 而且有时候因为数据类型造成处理错误, 调试时不一定能很快找到问题, 于是浪费了许多时间, 尤其在数字部分有时会因为整数或双精度数字而出问题。而 ASP.NET 采用编译的方式执行, 因此, 在执行之前, 必须对变量进行声明, 以便编译程序可以处理。ASP.NET 中变量数据类型的明确化对程序整体执行效率有很大的帮助, 并且在把程序编译成 MSIL 程序代码时, 可以事先将问题找出来, 以避免因在执行时发生问题而发生不可预期的结果。因有时候程序写错了调试了半天, 才发现不是程序逻辑写错, 而是前端的验证程序有问题(分页错误)、或是数据类型转换错误导致写入数据库时发生错误。写了那么多行的程序不要紧, 发生错误一直找不出问题所在才是重挫开发者信心的主要因素。

事实上 ASP.NET 的好处并不只这些, 其他的好处还有: 与 XML 更加紧密的集成; 使