



配光盘一张

本书特色

以最浅显易懂的方式带您认识新一代的 VB.NET
本书通过大量的实例，从基本的语法到异常
Visual Basic.NET 操作环境的介绍，文件 I/O 的运



普悠玛数位科技 编著

VB.NET 程序开发实务

结构全面剖析 Visual Basic.NET 面向对象的概念以及各种视窗对象的应用。
处理，使你能够轻松学会如何运用新一代的 ADO.NET 来管理数据库，让您充分了解新一代 Visual Basic 的强大功能。
用，让您全方位认识 Visual Basic.NET，一步步迈向 NET 之路。

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



VISUALBASIC.NET

程序开发实务

普悠玛数位科技 编著

中国铁道出版社

2002年·北京

(京)新登字 063 号

北京市版权局著作权合同登记号: 01-2002-1088 号

版 权 声 明

本书繁体字版由第三波资讯股份有限公司出版, 版权归第三波资讯股份有限公司所有。本书简体字中文版由第三波资讯股份有限公司授权中国铁道出版社出版。专有出版权属中国铁道出版社所有, 未经本书原版出版者和本书出版者书面许可, 任何单位和个人不得以任何形式或任何手段复制或传播本书的一部分或全部。

图书在版编目(CIP)数据

Visual Basic.NET 程序开发实务/普悠玛数位科技编著. —北京: 中国铁道出版社出版, 2002. 3

ISBN 7-113-04608-8

I. V… II. 普… III. Basic 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 015817 号

书 名: Visual Basic .NET 程序开发实务
作 者: 普悠玛数位科技
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 郭毅鹏
责任编辑: 苏 茜 徐煜东
封面设计: 孙天昭
印 刷: 北京市兴顺印刷厂
开 本: 787×1092 1/16 印张: 20.75 字数: 499 千
版 本: 2002 年 4 月第 1 版 2002 年 4 月第 1 次印刷
印 数: 1~5000 册
书 号: ISBN 7-113-04608-8/TP·695
定 价: 36.00 元

版权所有 盗版必究

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。

出版说明

想一窥 Visual Basic.NET 世界的读者们是否苦于自己没有 Visual Basic 的底子？现在您也能以最轻松的方式学习了。本书包含许多范例，从基本语法到异常处理，让您能够轻松学会各种对象和语法的使用方法。相信您在看过本书后，一定会对新一代 Visual Basic 的强大功能感到惊讶不已。书中列举的范例文件源代码全收录于随书附赠的光盘中，以便您能更好地学习 Visual Basic.NET，让你一步步迈向.NET 之路。

本书由第三波资讯股份有限公司提供版权，经中国铁道出版计算机项目中心审选。张瀚文、李自运、张森、林霖、李霞、廖康良、孟丽花、陈小娟等同志参与了本书的整稿及排版工作。

2002 年 4 月

目 录

第 1 章 .NET 起源	1
1-1 认识.NET	2
1-1-1 网络应用程序概念	2
1-1-2 .NET 与互联网	2
1-1-3 .NET 的企业蓝图	4
1-2 浅谈.NET Framework.....	5
1-2-1 .NET Framework	5
1-2-2 认识共同语言执行环境	7
1-2-3 什么是 Web Service.....	10
1-2-4 Base Classes 层次概念	11
1-3 .NET 的程序结构.....	11
1-3-1 了解.NET 程序结构	11
1-3-2 认识.NET 开发工具	14
1-3-3 认识可扩展标记语言	16
1-3-4 简单对象访问协议	19
第 2 章 VB.NET 的工作环境.....	21
2-1 安装 VS.NET 平台.....	22
2-1-1 安装环境简介	22
2-1-2 安装步骤	22
2-2 认识 VS.NET 环境.....	29
2-2-1 VS.NET 编辑环境介绍.....	29
2-2-2 VS.NET 各种窗口介绍.....	31
2-2-3 VS.NET 的其他窗口.....	36
2-2-4 VS.NET 的窗口基本操作.....	42
2-3 VB.NET 的项目建立.....	46
2-3-1 建立 VS.NET 项目	46
2-3-2 编译	50
2-3-3 调试	53
第 3 章 VB.NET 的语言结构.....	59
3-1 VB 的语言特性	60
3-2 VB.NET 的变量与基本数据类型	60
3-2-1 认识变量与常量	61
3-2-2 VB.NET 的数据类型.....	61

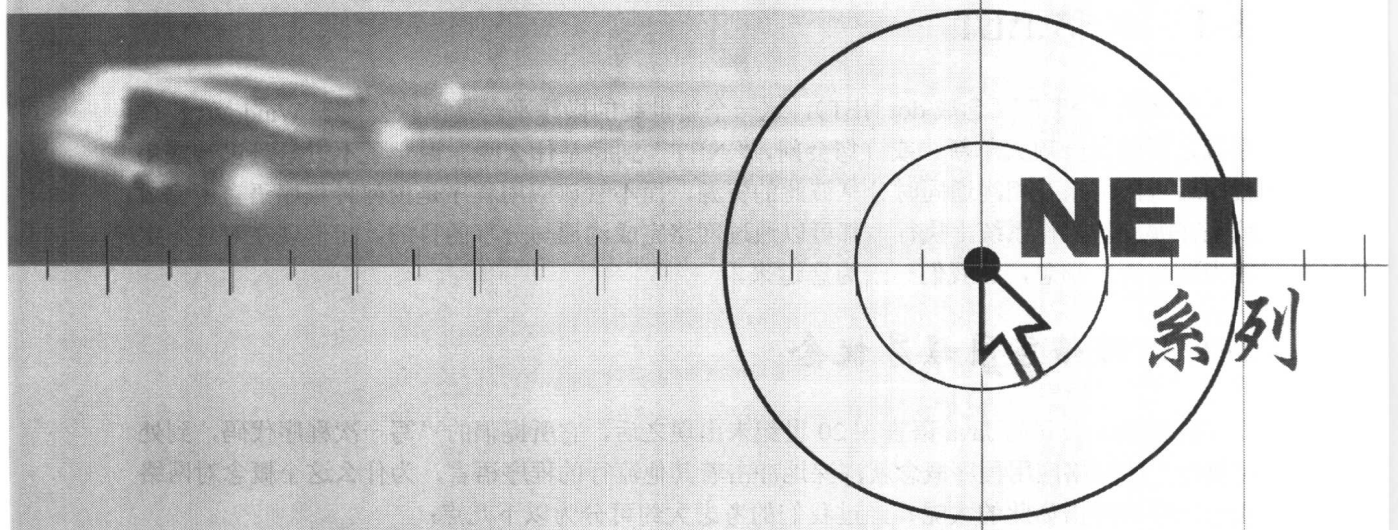
3-3	变量声明法则与转换.....	63
3-3-1	变量声明的方法.....	63
3-3-2	赋予变量的初值.....	63
3-3-3	变量的命名法则.....	64
3-3-4	匈牙利命名法.....	64
3-3-5	变量的作用范围.....	65
3-3-6	变量数据类型的转换.....	66
3-4	运算符.....	68
3-4-1	算术运算符.....	68
3-4-2	赋值运算符.....	68
3-4-3	比较运算符.....	69
3-4-4	连接运算符.....	71
3-4-5	逻辑运算符.....	73
3-4-6	运算符的优先顺序.....	75
3-5	控制结构的运用.....	77
3-5-1	If 判断式.....	77
3-5-2	If...Then...Else 判断式.....	79
3-5-3	嵌套 If...Then...Else 判断式.....	80
3-5-4	Select Case 判断式.....	82
3-5-5	For...Next 循环.....	84
3-5-6	While...End While 循环.....	85
3-5-7	前测型 Do...Loop 循环.....	87
3-5-8	后测型 Do...Loop 循环.....	89
3-5-9	跳离语句 Exit.....	91
3-6	认识函数.....	92
3-6-1	无返回值的 Sub 函数.....	93
3-6-2	有返回值的 Function 函数.....	94
3-6-3	传递参数给函数.....	95
3-7	异常处理.....	99
3-7-1	认识异常处理机制.....	99
3-7-2	异常处理的语法.....	99
3-8	数组.....	102
3-8-1	认识数组.....	102
3-8-2	声明与设置数组值.....	102
3-8-3	改变数组的大小.....	105
3-9	VB.NET 在语法上的改变.....	107
3-9-1	变量声明与设定的改变.....	107
3-9-2	新增的运算式.....	108
3-9-3	数组与字符串的改变.....	109



3-9-4	函数声明及调用的改变	110
3-9-5	关键字的改变	111
第 4 章	.NET 类的应用	113
4-1	自定义类 (Class)	114
4-1-1	认识类与对象	114
4-1-2	认识 Method 与 Fields	114
4-1-3	属性(Property)	119
4-1-4	自定义类	121
4-2	建立类库	124
4-2-1	建立类库	124
4-2-2	在程序中引用类库	128
4-3	事件	131
4-3-1	事件的声明与触发	131
4-3-2	事件处理函数及对象的建立	132
4-4	VB.NET 类应用	134
4-4-1	异常类	135
4-4-2	ArrayList 类	136
4-4-3	时间日期类	139
4-4-4	数学及随机数类	141
第 5 章	剖析 VB.NET 面向对象概念	145
5-1	VB 与面向对象简介	146
5-1-1	面向功能与面向对象	146
5-1-2	类、对象及成员	146
5-1-3	VB.NET 的面向对象功能	147
5-2	继承	149
5-2-1	认识继承结构	149
5-2-2	继承的可视关系	152
5-2-3	其他关键字	156
5-2-4	窗体的继承应用	163
5-3	过载与重载	170
5-3-1	过载	170
5-3-2	重载	172
5-4	构造函数 VS 解析函数	174
5-4-1	构造函数	174
5-4-2	解析函数	177
5-5	多态的认识与应用	181
5-5-1	类与多态	181
5-5-2	接口与多态	181

第6章 各类窗口对象介绍	189
6-1 窗口对象 (Windows Forms) 的基本结构.....	190
6-1-1 与 VB6.0 不同概念的 Windows Forms	190
6-1-2 Windows Forms 的类.....	190
6-2 文本控件.....	194
6-2-1 Label 控件	195
6-2-2 LinkLabel 控件.....	197
6-2-3 TextBox 控件	199
6-2-4 RichTextBox 控件.....	200
6-2-5 ComboBox 控件	200
6-2-6 ListBox 控件.....	202
6-2-7 CheckedListBox 控件	205
6-3 按钮式的控件.....	206
6-3-1 Button 控件	207
6-3-2 CheckBox 控件	207
6-3-3 RadioButton 控件	209
6-4 CommonDialog Class 提供的对话框控件.....	212
6-4-1 OpenFileDialog 类控件	213
6-4-2 FontDialog 控件.....	219
6-4-3 ColorDialog 控件.....	221
6-4-4 PrintDialog 控件	224
6-5 信息对话框.....	226
6-6 图片控件.....	229
6-6-1 PictureBox 控件.....	229
6-6-2 ImageList 控件.....	231
6-7 其他特殊的控件.....	234
6-7-1 Timer 控件	234
6-7-2 日期控件	237
第7章 文件系统操作	241
7-1 数据文件的种类.....	242
7-2 基本的文件存取.....	243
7-2-1 打开文件	243
7-2-2 读取文件内容	244
7-2-3 将数据写入文件	247
7-3 取得文件信息.....	250
7-3-1 使用 File 类取得文件信息	250
7-3-2 使用 FileInfo 类取得文件信息.....	253
7-4 各项文件操作.....	260
7-4-1 复制文件	260

7-4-2	删除文件	261
7-4-3	移动文件	263
7-4-4	文件操作综合应用	264
第 8 章	SDI 与 MDI 程序	273
8-1	使用 Menu 控件	274
8-1-1	MainMenu 控件	274
8-1-2	ContextMenu 控件	276
8-1-3	StatusBar 控件	277
8-1-4	ToolBar 控件	279
8-2	编写 MDI 程序	282
8-2-1	MDI 程序的特征	282
8-2-2	MDI 程序的设置	283
8-2-3	关于图片的 MDI 程序	291
第 9 章	与数据库的交互	295
9-1	使用 ADO.NET 连接数据库	296
9-1-1	数据库连接三步曲	296
9-1-2	使用对象连接数据库	298
9-1-3	使用 Command 对象操作数据库	299
9-1-4	使用 DataAdapter 对象	300
9-1-5	使用 DataSet 对象	302
9-2	使用 Visual Studio 快速连接数据库	308
9-2-1	快速建立数据库连接	309



第 1 章

.NET 起源

1-1 认识.NET

什么是“.NET”（念成 dot NET），这一个新世纪的操作平台是微软公司于 Windows 操作系统之后的另一项大革新。或许您会问，“.NET”到底是什么呢？说穿了不过是可以让应用程序通过网络来互相沟通同时分享彼此的资源，而不管该应用程序是由何种程序语言所完成或是在哪一种操作系统上执行，都可以通过网络完成沟通与分享的目的。相信现在对它一定相当地好奇。别急，让我们一一为您道来。

1-1-1 网络应用程序概念

自从 Sun 公司的 Java 语言在 20 世纪末出现之后，它所提倡的“写一次程序代码，到处可执行”的网络应用程序概念就深深地冲击着其他流行的程序语言，为什么这个概念对网络应用程序的冲击如此的大呢？经过我们的考虑大约可分为以下几点：

◆ 受使用平台的限制太大

目前的网络应用程序通常是为单一的设备或平台所设计的，这样的设计方式所衍生出来的问题就是提高了开发应用软件的成本。

◆ 程序之间的互动不好

现在的网络应用程序，做得最完整的就是人和程序之间的沟通了。不管是目前流行的在线游戏或是提供各项服务的商业网站，对于人与程序之间的沟通发挥无不尽善尽美，但是如果各种网络应用程序，或网站之间可以进行程序或是资源的沟通交流，那不是更完美了吗？可以想象只要在一个网站内提出需求后就自动完成相关的操作吗？这些都是目前应用程序结构所难以完成的。

而 .NET 就是为了让网络应用程序，迈向独立于各种操作平台与实现程序间良好的沟通而产生的。接下来，来看看.NET 与互联网之间的关系。

1-1-2 .NET 与互联网

以微软所提出的结构来看，新一代的网络应用程序都是提供一种网络服务（Web Service），并且都架构在“.NET”这个平台上，基于这个观点，“.NET”将具备以下的功能：

- 进行软件的整合并解决程序间资源共享的问题。
- 提供一个环境让不同语言的程序员都能轻松地编写“.NET”平台的网络应用程序。
- 整合网站之间的信息并根据需求进行整合与应用。

对于第三点，可利用图 1-1 来说明。



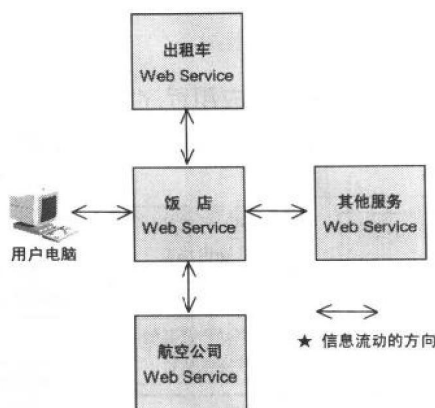


图 1-1

上图中的工作流程是这样的，当进入某饭店网站上提出预订房间的要求之后，网站就会寻问相关的问题，比如乘坐什么交通工具、是否要出租车或是当地的相关景点等信息。而当我们一一地做完选择后，网站便向其他相关的网站提出我们的要求，同时做出一些选择方式供我们做决定，这也就是说，只要打开一个网页就可以完成所有的操作了。

所以可以这样说，“.NET”是新一代综合网络世界的平台。或许会问，“.NET”是利用何种方法来完成这样的强大的功能呢？除了内部结构的革新外，另外最重要的一点是将数据的传输以“XML”的格式来进行封装，图 1-2 就是“.NET”和网络应用程序之间的关系。

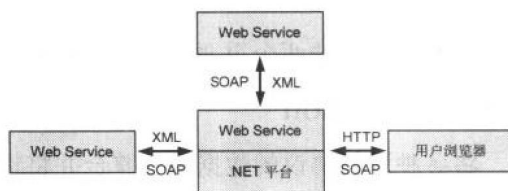


图 1-2

可以看出，在各个 Web Service 之间是以“XML”来封装数据，而 Web Service 间数据传递的通讯协议是以“SOAP”让程序通过网络在不同平台间沟通。图中的 Web Service 是指网络上的各种服务器服务，如 SQL Server 或是 Application Center 等。

**补
给：
站**

- “XML”是一种 World Wide Web Consortium (W3C) 标准下的结构化数据表达语言，提供应用程序可以在不同平台上沟通的数据表达方式。
- “SOAP”全名为“Small Object Access Protocol”，是一种以 XML 为基础的分布式对象通讯协议，提供一个沟通管道给支持“SOAP”的应用程序。
- 使用的通讯协议除 SOAP 和 HTTP 外，还可使用 SMTP (Simple Mail Transfer Protocol) 这个通讯协议。

1-1-3 .NET 的企业蓝图

微软提出“.NET”的目的，就是将一切应用程序都网络化，并且通过网络来进行信息的获取和程序之间的沟通。可以看看“Microsoft.NET”的基本组织图，如图 1-3 所示：

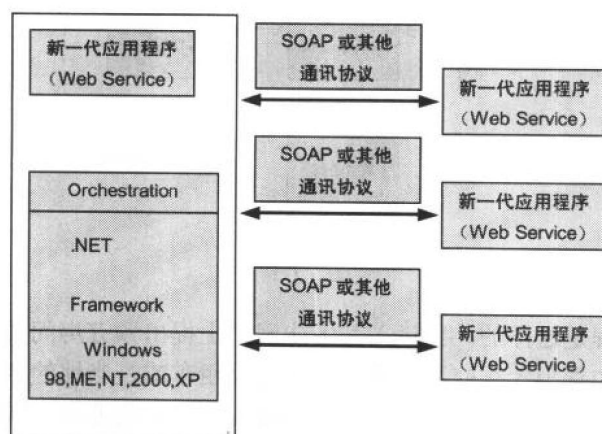


图 1-3

现将上图中的两个重要的部分说明如下：

◆ Orchestration

将各种通讯协议的数据进行整合并将整合过的信息提供给上层的 Web Service。

◆ .NET Framework

提供所需的网络基本功能、编写程序所需的类函数与程序代码的转换。

而由图中可以了解该“.NET”环境可以在不同的平台上执行，比如 Windows 98、Windows ME、Windows 2000 或是即将推出的 Windows XP 的操作系统上。而最底层则为网络数据封装与通讯的各种协议，如 HTTP、SOAP 或是 XML 等。

除了认识“.NET”的内部基本组织，还必须了解构造一个 Web Service 时，可以使用到的部分。依微软公司的蓝图指出，除了本身应用程序的 Web Service 外，还可以使用到下面三种：

◆ “.NET”企业型服务器 (.NET Enterprise Servers)

指一些大型的服务器程序，比如“SQL Server”、“BizTalk Server”、“Application Center”或是“Exchange Server”等。

◆ 已构造的 Service (Building Block Service)

由微软所提供的 Web Service，如日历或是记事本等。

◆ 其他 Web Service

由其他公司所完成的 Web Service，可利用这些 Web Service 的多样性，丰富程序的应用。

补
给
站

- “SQL Server” 提供数据库管理的功能。
- “BizTalk Server” 支持 B2B (business to business) 电子商务结构的服务器。
- “Application Center” 提供构造大型 Web Service 所需的服务应用程序中心。
- “Exchange Server” 是一个文件存储系统, 提供一个数据可以共享的文件服务器。

而对于一个 Web Service 而言, 本身所扮演的角色除了是用户以外, 同时也是一个服务 (Service) 的提供者。以现实的观点来看, 如果能设计出一个功能齐全的 Web Service 让全世界的程序员使用, 那其中的商机真是无限。

本节说明了目前网络应用程序的发展缺憾, 以及介绍微软针对这些缺憾所提出的新一代平台 “.NET” 结构, 也说明了 “.NET” 到底具备了哪些能力来解决网络应用程序发展的瓶颈, 同时对于 “.NET” 的基本结构与和其他网络组件的沟通方式做了一些简单的介绍。当然, 这些都只是入门而已, 下一节, 将针对 “.NET” 的核心结构—— “.NET Framework”, 做一番说明。

1-2 浅谈 .NET Framework

在上一节提到了 “.NET” 可以让应用程序通过网络来互相沟通并同时共享彼此的资源, 而不管该应用程序是由何种程序语言所完成或是在哪一种操作系统上执行。让 “.NET” 可以如此神奇的中心结构就是 “.NET Framework”, 接下来, 来看看 “.NET Framework” 的结构是什么样的。

1-2-1 .NET Framework

“.NET Framework” 提供给程序员一些完整的开发工具, 也提供应用程序一个稳定的执行环境, 而大家所关心的跨平台能力也是通过它来完成的。

另一个大家所关心的话题就是要使用何种语言才能在 “.NET Framework” 上执行呢? 告诉一个好消息, 程序员不再受到语言的限制了。 “.NET” 利用共同语言结构来完成这样的工作, 也就是说, 应用程序将不再是单一语言结构, 可以根据要求调用如 Visual Basic、C、C# 或是 Java 等不同语言所开发完成的函数组件。这也意味着, 软件组件的开发成本将大幅地降低。

接下来, 来看看 “.NET Framework” 内部的结构, 看看到底有哪些组件组成了 “.NET Framework” 如此强大的功能, 如图 1-4 所示:



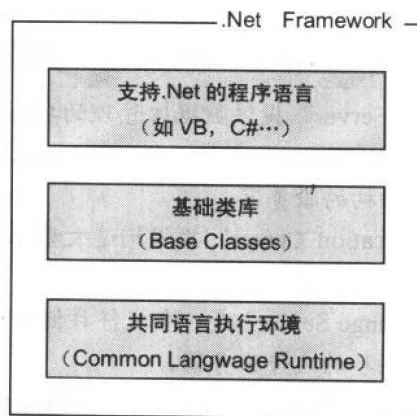


图 1-4

可以将结构图分成三个部分，最上层就是程序语言部分，表示设计程序所使用的语言，而中间的部分是基础类部分，提供一个基本的函数库供不同的程序语言使用，让程序之间的沟通更容易完成，最下层就是共同语言执行环境（Common Language Runtime, CLR），这是整个“.NET Framework”结构最核心的部分。利用这个功能，可以实现跨平台的理想目标，同时也让长久以来困扰程序员的 DLL 陷阱得以解决。

接下来，更进一步地详细讨论“.NET Framework”的各部分结构。

- 在最上层的部分代表完成“.NET”应用程序所使用的程序语言，其中可以使用“Visual Studio.NET”所提供的程序语言（比如说“Visual Basic.NET”或是“C#”）来完成该应用程序。而使用“VS.NET”内的程序语言有什么好处呢？就是两种语言内的类（Class）可以通过本地代码互相继承，这个跨世纪的改变让我们可以针对不同语言的特色来使用该语言的类函数，当 Java、Pascal 或是 COBOL 等其他语言整合到“.NET”环境平台后，语言之间的相互沟通就不再是梦想了。
- 在中间的部分就是“.NET”提供给程序语言的基础类（Base Classes），也就是一般所谓的类函数库（Class Library）。提供这样的类函数库有什么好处呢？每位程序员应该都有这种感觉，就是使用每种程序语言都各有其特性，也有其对应的专门类函数库，如此分明的界限，造成了语言之间的合作非常困难。而在 21 世纪的今天，“.NET”平台让这个梦想实现了，它将一般系统的功能做一个整合，比如说有关网络访问、绘图或是有关窗口操作等分类。所以只要是支持“.NET”平台的程序语言，都可以调用这一个基础类函数库。这代表着当许多人共同开发一个大型应用程序项目时，使用何种语言将不再是争论的焦点了，大家可以选择自己拿手的程序语言进行开发，并将注意力放在应用程序的研发方面。
- 最下面那一层就是整个“.NET Framework”的核心部分，也就是通过这个部分让语言的沟通变成可能。如果是 Java 语言的程序员，或许会发现其实共同语言执行环境（Common Language Runtime）和 Java 本身的虚拟机器（Virtual Machine, VM）的概念很像，也就是在程序的编译过程实现一些技巧来完成这项功能，我们利用下一节来仔细地讨论共同语言执行环境。

1-2-2 认识共同语言执行环境

共同语言执行环境（Common Language Runtime, CLR）是“.NET”平台用来实现程序代码可以跨平台执行的机制。它在进行程序的编译时，将程序代码分成两个部分，也就是中间语言（Intermediate Language, IL）和记载着程序执行所需信息的数据表达部分（Metadata），这种方式和原来的编译方式有何不同呢？如图 1-5 就是一般语言的编译过程：

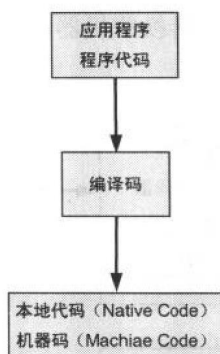


图 1-5

过程中，编译器将程序代码与语言本身提供的函数库结合编译成机器可以执行的本地代码（Native Code）。可以想想，如果此时想拿到别的机器上执行的话，那是不是会因执行的机器或是操作系统的不同而必须重新编译，才能顺利执行程序呢？这也就是为什么先前的应用程序语言不能执行于不同环境的主要原因。接下来，看看如图 1-6 所示的“.NET”平台下的语言编译情形。

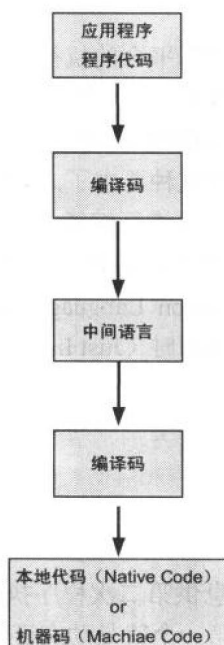


图 1-6

发现了吧！由于“.NET”的编辑器是将程序代码和基础类组合编译成一个中间文件与信息表达文件，而这两个文件的格式都是相同的，所以只要是支持“.NET”平台，都可以执行该应用程序。如此就解决了应用程序执行于不同平台的问题。可以想想看，不管是IBM公司的Mac电脑，或是现在广为流行的个人商务助理（PDA）或是一般的家用电脑，只要这些设备内存在“.NET Framework”，那它所提供的应用程序就可以在该设备上执行，这就成为了只写一次应用程序就可以在各设备执行的最佳平台。

所以如果开发一个“.NET”平台的应用程序，它会被编译两次，第一次是将程序代码编译成中间语言，第二次就是将中间语言转换成机器能够执行的本地代码，如图1-7所示：

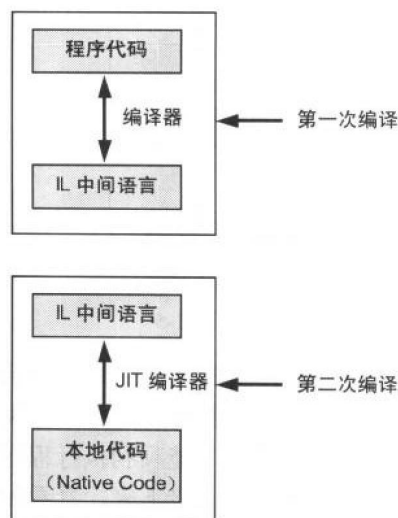


图 1-7

或许会有这样的疑问，如此“.NET”平台的执行效率是不是就比先前的应用程序差？这个想法只对了一半。事实上，当在各种平台第一次执行“.NET”的应用程序时，是需要多一点时间将中间语言编译成各种平台的本地代码（也就是我们常说的机器码“Machine Code”），但是以后在该平台执行程序时就不再有这种要求了。所以现在有两种选择，就是重新编写程序代码或在第一次执行时多花一点时间来进行编译，这两者在当今电脑执行速度已超过“GHz”为单位的硬件性能来看，取舍就相当的清楚了，当然是重新编码较费时间。

另外，从图中可以看出，在“Common Language Runtime”内将中间语言转换成该平台机器能解读的本地代码的最大功臣就是即时（Just In Time, JIT）编译器了，它的主要功能如下所示：

- 检查内存位置的合法性
- 进行类型的检查
- 转换中间语言为本地代码
- 进行运算符的检查
- 记录第一次程序执行时的信息供第二次程序执行时使用

最后还有两个概念要提出的是，第一个就是动态链接函数库陷阱（DLL Hell）的问题，另一种就是语言整合的问题，让我们继续往下进行。