

GOTOP

# Visual Studio .NET

## 高手攻略

许进标 编著  
袁鹏飞 改编



内附光盘

人民邮电出版社  
POSTS & TELECOMMUNICATIONS PRESS

# Visual Studio .NET 高手攻略

许进标 编著

袁鹏飞 改编

人民邮电出版社

## 图书在版编目 (CIP) 数据

Visual Studio.NET 高手攻略 / 许进标编著; 袁鹏飞改编. —北京: 人民邮电出版社, 2003.3  
ISBN 7-115-11172-3

I. V... II. ①许...②袁... III. 计算机网络—程序设计 IV. TP393

中国版本图书馆 CIP 数据核字 (2003) 第 004359 号

## 版 权 声 明

本书为台湾蕃峰资讯股份有限公司独家授权的中文简化字版本。本书专有出版权属人民邮电出版社所有。在没有得到本书原版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形式(包括资料和出版物)进行传播。

本书原版版权属蕃峰资讯股份有限公司。

版权所有, 侵权必究。

### Visual Studio .NET 高手攻略

- 
- ◆ 编 著 许进标  
改 编 袁鹏飞  
责任编辑 陈 昇
  
  - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
读者热线 010-67132705  
北京汉魂图文设计有限公司制作  
北京密云春雷印刷厂印刷  
新华书店总店北京发行所经销
  
  - ◆ 开本: 787×1092 1/16  
印张: 26  
字数: 630 千字 2003 年 3 月第 1 版  
印数: 1-4 000 册 2003 年 3 月北京第 1 次印刷

著作权合同登记 图字: 01-2002-5938 号

ISBN 7-115-11172-3/TP·3385

定价: 46.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

# 内 容 提 要

本书由最简单的面向对象程序设计概念说起，让您可以轻松地使用面向对象程序设计及 Visual Basic .NET 来编写程序，并写出一个好程序。全书共 14 章，前 3 章是工作环境介绍及程序语言基础；接下来的 5 章，以简单的范例讨论面向对象程序设计的各种概念；最后的 6 章则是实际操作部分，包括如何使用 ADO.NET 及 ASP.NET 等。

本书内容实用，适于初中级程序开发人员阅读。通过阅读本书，读者可以快速、全面地掌握 Visual Basic .NET 程序设计技术。

# 序

Visual Studio .NET 是 Microsoft 最新一代的应用程序开发工具。它可以让您在同一个环境之下，开发各种不同的应用程序。Visual Studio .NET 支持多种不同的程序语言，其中最简单易学的就是 Visual Basic .NET。本书详细介绍如何在 Visual Studio .NET 的环境下，使用 Visual Basic .NET 来开发应用程序。

新一代的 Visual Basic .NET 已完全支持面向对象程序设计。它是一套功能强大的程序语言，提供了非常多的模板与范例，适用于各行各业。不过，如果您不懂得面向对象程序设计，只是随便找个模板套用，知其然而不知其所以然，那么您就无法充分发挥 Visual Basic .NET 的功能。

本书由最简单的面向对象程序设计概念说起，让您可以轻松地使用面向对象程序方法及 Visual Basic .NET 来撰写程序，并写出一个好程序。建议有心想学好 Visual Basic .NET 的人，应该花一点时间研读本书的第 4 至第 8 章，一定可以大大提高您编写程序的功力。

本书的编排，前 3 章是工作环境介绍及程序语言基础。接下来的 5 章，以简单的范例讨论面向对象程序设计的各种概念，帮您建立 Visual Basic .NET 的扎实基础，使用户不会一开始就迷失在一大堆操作步骤中，非常适合教学之用。最后的 6 章则是实际操作部分，包括如何使用 ADO .NET 及 ASP .NET 等。由于 Visual Basic .NET 所提供的数据太多，不能完全纳入本书。这 6 章的主题是笔者认为比较有代表性的，也可用来印证前几章的理论部分。至于其他各种实现，只要您有了前 8 章的基础，再配合 Visual Studio .NET 所提供的文件说明，就很容易了解其操作步骤以及为何要如此操作。

在本书编撰期间，感谢蓉峰资讯的蔡彤孟经理的鼓励与关怀，江佳慧小姐以及出版部同仁的支持。另外要感谢爱妻丽嫔的全力配合，才能使本书如期完工。本书仓促付梓，若还有疏漏之处，请读者见谅，并请不吝指教！

许进标

2002 年 8 月

# 光盘使用说明

本书所附赠的光盘内容主要包括本书的所有范例，读者可将其直接复制到硬盘后使用。若您的硬盘空间不足，也可直接从光盘片中打开。不过要注意，从光盘中读取的文件为只读文件，必须更改其属性之后才可修改。更改属性的方法如下：

1. 在**资源管理器**中选取要更改的文件，然后按右键，点选 **属性** 命令。
2. 将**属性**中的只读复选框取消。

使用范例时，请注意下列各项：

● 有些范例文件必须连接到数据库（Microsoft SQL Server）才能正常作用，请参考书上的相关章节，依据您的网络实际情况，重新建立数据源，连接到数据库中。如果您无法连接到 Server，可能无法将数据更新，甚至看不到正确的结果。

● 本书中的有些范例中会用到存在硬盘中的图形文件，作者已尽量将范例中用到的图形文件复制到每一章的范例文件夹中，读者可以按自己的安装位置重新指定图形文件的路径，才能让范例程序顺利执行。

● 本书中的范例必需在计算机中安装有 Visual Studio .NET 才能执行。

● 本书中每一章的范例文件，若有多个窗体或模块，必需自己依不同的章节重新设置启动对象，才能看到正确的结果。

# 目 录

<b>第1章 认识.NET 框架</b> .....	1
1.1 .NET 框架简介.....	1
1.1.1 .NET 框架的主要组件.....	2
1.1.2 通用语言运行时.....	3
1.1.3 类库.....	5
1.1.4 .NET 框架的相关文档.....	6
1.2 系统需求.....	9
1.2.1 .NET 框架的系统需求.....	9
1.2.2 Visual Studio.NET 的版本.....	9
1.2.3 Visual Studio.NET 的系统需求.....	10
<b>第2章 Visual Studio.NET 的开发环境</b> .....	13
2.1 建立第一个项目.....	13
2.1.1 配置起始页.....	13
2.1.2 快速建立第一个应用程序.....	15
2.2 Visual Studio.NET 窗口介绍.....	19
2.2.1 动态帮助窗口.....	19
2.2.2 解决方案与项目的关系.....	20
2.2.3 使用程序代码窗口.....	22
2.2.4 任务列表.....	28
2.2.5 属性窗口.....	30
2.2.6 添加引用.....	30
2.2.7 输出窗口.....	31
2.2.8 类视图.....	32
2.3 项目的属性设置.....	33
2.3.1 项目的一般属性配置.....	34
2.3.2 配置启动项目.....	35
2.3.3 其他属性配置.....	36
2.4 Visual Studio.NET 的文件架构.....	37
2.4.1 SLN 文件.....	38
2.4.2 VBPROJ 文件.....	38
2.4.3 suo 及 vbproj.user 文件.....	42
2.5 调试与配置设定.....	43

2.5.1	配置管理器	44
2.5.2	配置的属性值设置	46
<b>第3章</b>	<b>Visual Basic.NET 程序语言重点摘要</b>	<b>48</b>
3.1	在模块中输入程序	48
3.1.1	新建控制台应用程序	48
3.1.2	重新命名模块名称	50
3.1.3	使用变量	52
3.1.4	Option 属性设置	53
3.2	数据类型	54
3.2.1	Visual Basic.NET 的数据类型及声明	54
3.2.2	数据类型转换	56
3.2.3	字符串与字符串函数	60
3.2.4	数据格式化	63
3.3	数学运算及数学函数	66
3.3.1	数学运算	66
3.3.2	数学函数	67
3.4	循环及条件语句	68
3.4.1	循环结构	68
3.4.2	条件语句	71
3.4.3	Exit 语句	72
3.5	数组的使用	74
3.6	程序与函数	75
3.6.1	Function 程序	75
3.6.2	Sub 程序	77
3.6.3	使用数组自变量	78
<b>第4章</b>	<b>面向对象程序设计</b>	<b>80</b>
4.1	面向对象程序设计	80
4.1.1	面向对象程序简介	80
4.1.2	封装、继承及多态性	81
4.1.3	将程序对象化	83
4.2	如何建立对象	85
4.2.1	使用 New 方法实例化	85
4.2.2	参数型构造函数	87
4.3	命名空间	92
4.3.1	导入命名空间	92
4.3.2	使用.NET 框架的命名空间	94
4.3.3	System.Collections 命名空间	97
4.4	使用对象变量	103



---

4.4.1	对象变量的声明与使用	103
4.4.2	以 ByVal 和 ByRef 传递对象变量	106
<b>第 5 章</b>	<b>自定义类</b>	<b>108</b>
5.1	建立类及其成员	108
5.1.1	建立类	108
5.1.2	存取范围设置	111
5.1.3	添加重载的类成员	111
5.1.4	添加其他构造函数	114
5.1.5	设置默认属性	115
5.1.6	变量范围的设置	116
5.1.7	共享变量	118
5.2	对象的生存期	121
5.3	数值类型和引用类型	123
5.3.1	枚举	124
5.3.2	结构	125
5.4	调试	128
<b>第 6 章</b>	<b>继承与接口</b>	<b>131</b>
6.1	建立和使用继承	131
6.1.1	建立继承	132
6.1.2	重写属性及方法	136
6.1.3	在继承架构中使用多态性	139
6.1.4	隐藏	141
6.1.5	抽象基类	143
6.2	Object 类	147
6.2.1	常用的 Object 类成员	148
6.2.2	继承的危机: DLL Hell	153
6.3	接口的基本认识	162
6.4	常用接口	166
6.4.1	使用 ICloneable 接口	166
6.4.2	使用 IDisposable 接口	168
6.5	常用集合接口	169
6.5.1	IEnumerable 及 IEnumerator 接口	169
6.5.2	ICollection 接口	172
6.5.3	IList 接口	172
6.5.4	IDictionary 接口	173
6.5.5	IComparable 接口	175
6.5.6	IComparer 接口	177

<b>第7章 事件与代理</b> .....	180
7.1 简单的事件处理方式.....	180
7.2 建立自定义事件处理程序.....	183
7.2.1 自定义简单事件处理程序.....	183
7.2.2 自定义事件类.....	187
7.2.3 动态处理事件.....	189
7.3 代理.....	191
7.3.1 建立代理.....	191
7.3.2 多点传送的代理事件.....	193
7.3.3 代理与排序.....	195
<b>第8章 结构化异常处理</b> .....	199
8.1 结构化异常处理.....	200
8.1.1 结构化异常处理概念.....	200
8.1.2 结构化异常处理范例.....	202
8.2 使用 Exception 类成员.....	205
8.2.1 Exception 类成员.....	206
8.2.2 使用 Throw 关键字.....	207
8.2.3 系统或应用程序的异常情况.....	208
8.3 处理异常情况的正确观念.....	209
<b>第9章 使用控件建立 Windows 应用程序</b> .....	211
9.1 常用控件介绍.....	212
9.1.1 TextBox 控件.....	213
9.1.2 使用 ButtonBase 控件.....	218
9.1.3 CheckedListBox 控件的使用.....	222
9.1.4 ListBox 及 ComboBox 控件.....	226
9.1.5 定位顺序的设置与修改.....	227
9.2 高级控件应用.....	229
9.2.1 使用 TrackBar 控件.....	229
9.2.2 使用 MonthCalendar 控件.....	231
9.2.3 使用 Spin 控件.....	234
9.2.4 使用 Panel 控件.....	235
9.2.5 使用 ToolTip 控件.....	236
9.2.6 使用 ErrorProvider 控件.....	237
9.2.7 Anchor 与 Dock 属性设置.....	239
9.3 使用自定义菜单及状态栏.....	241
9.3.1 建立一般菜单.....	241
9.3.2 建立 PopUp 菜单.....	244
9.3.3 将菜单项目与命令结合.....	245

9.3.4 建立及使用状态栏	246
9.4 自定义对话框	249
9.4.1 继承窗体	249
9.4.2 以窗体作为对话框	251
9.4.3 取得对话框的结果	253
9.5 自定义控件	257
9.5.1 新增控件	257
9.5.2 新增属性	261
<b>第 10 章 输入与输出</b>	<b>264</b>
10.1 文件与目录管理	264
10.1.1 使用 DirectoryInfo 类	265
10.1.2 使用 Directory 类	269
10.1.3 使用 FileInfo 类	271
10.1.4 使用 File 类	273
10.2 使用 Stream 读取和写入文件	275
10.2.1 使用 FileStream 写入及读取数据	276
10.2.2 使用 MemoryStream	279
10.2.3 使用 StreamWriter 及 StreamReader	279
10.3 串行化	282
10.3.1 将对象串行化	282
10.3.2 还原串行化	287
<b>第 11 章 Visual Basic.NET 的绘图功能</b>	<b>289</b>
11.1 如何在窗体上绘图	289
11.1.1 重写 OnPaint 方法	290
11.1.2 建立窗体的 Paint 事件	291
11.1.3 使用 Graphics 类	294
11.2 常见的绘图问题	298
11.2.1 坐标的平移与旋转	299
11.2.2 选择颜色	302
11.2.3 字体问题	305
11.2.4 设置不同的画笔	312
11.2.5 使用不同的画刷	318
11.3 点击测试	324
11.3.1 PictureBox 控件的点击测试	324
11.3.2 矩形范围及不规则形状的点击测试	327
<b>第 12 章 使用 ADO.NET 存取数据库</b>	<b>332</b>
12.1 存取 SQL Server 数据库	332

---

12.2 数据适配器简介.....	342
12.2.1 ADO.NET 基本概念.....	342
12.2.2 数据适配器简介.....	344
12.3 使用数据集.....	345
12.3.1 使用可视化工具建立关联.....	346
12.3.2 直接使用程序建立关系.....	351
12.4 数据库存取高级范例.....	356
12.4.1 参数查询范例.....	356
12.4.2 使用多个数据适配器.....	362
<b>第 13 章 建立 ASP.NET Web 应用程序.....</b>	<b>372</b>
13.1 使用 Web 窗体存取 Access 数据库.....	372
13.2 使用 SqlConnection 查询及更新数据.....	385
<b>第 14 章 建立安装程序.....</b>	<b>396</b>
14.1 建立 Web 应用程序的安装程序.....	396
14.2 建立 Windows 应用程序的安装程序.....	400

# 第 1 章 认识.NET 框架

Visual Studio.NET 是微软开发的最新一代的应用程序开发工具，其中包括 Visual Basic.NET, Visual C# (读做 C-Sharp) 以及 Visual C++.NET 3 种程序语言。您可以用其中的任何一种程序语言来开发您的应用软件。Visual Studio.NET 的基础是建立在 .NET 框架上。本章将简单地说明 .NET 框架、Visual Studio.NET 及 Visual Basic.NET 之间的关系。

## 1.1 .NET 框架简介

多年以来，应用软件的开发工具日新月异，不断地创新、改良，但仍有许多不如人意的地方。美国一位知名的作家 (Andrew Troelsen) 曾经比较几种常用的开发工具，并列其所面临的问题。他认为 Windows 应用软件的发展，可分成几个阶段，分别说明如下：

- **使用 C 语言及 Windows API:** C 语言是一种结构化的程序语言，相当简洁。不过，它必须手动处理内存的管理工作，而且它也没有具备面向对象编程 (Object-Oriented Programming, 简称 OOP) 的特性。所以，虽然 C 语言搭配 Win32 的 API，可以开发出适当的应用软件；但随着程序的加大，其 Bug 也愈多。
- **使用 C++ 程序语言及 MFC:** C++ 程序语言是由 C 语言改良而成的，具有 OOP 的特性 (有关 OOP 的进一步说明，请参考第 4 至第 8 章)。不过，由于它是源于 C 语言，所以仍保有 C 语言在内存管理上的缺点。使用 C++ 程序语言，配合 Microsoft Foundation Classes (简称 MFC) 所提供的许多 C++ 类，程序开发者也可以开发出其所需要的软件。
- **使用 Visual Basic 6.0:** Visual Basic 是一种比较容易学习的程序语言，而且可以很轻易地整合 Windows 组件，开发出适当的软件。不过，即使到了 Visual Basic 6.0，它仍然不是一种面向对象的程序语言，这成了它的主要缺点。本书将介绍的 Visual Basic.NET 则没有这方面的缺点。
- **使用 Java 语言:** Java 本身是一种很好的程序语言及开发平台。但 Java 有一种排他性，它希望只要使用 Java 就可以开发出各式各样的软件。但实际上有些程序只是使用 Java 还无法达到最佳效果。Java 在不同程序语言的整合上并不成功，所以即使您知道有些程序是现成可用的，您也无法将其纳入 Java 程序中。
- **使用 COM 来开发软件:** COM 是 Component Object Model 的缩写。COM 可以说是一种架构，它定义出一些规则，只要依照 COM 规则所建立的类 (Class)，都可重复使用。COM 的特点是可以跨程序语言使用：使用 C++ 所开发出来的类可以供 Visual Basic 使用，也可以给其他程序使用。不过，COM 的重用性是有限制的，它并不能用继承方式来使用对象。使用 COM 来开发软件，在软件的开发及发布上也不太顺利。
- **使用 Windows DNA 开发软件:** Windows DNA (Distributed iNternet Architecture) 是 Microsoft 为了适应 Internet 时代所推出来的解决方案，它包括了很多技术及程序语言 (例

如 ASP、HTML、XML、JavaScript、VBScript、COM(+) 及数据存取技术 ADO 等等)，可以用来开发 Web 应用程序。Windows DNA 的缺点是不同的程序之间无法沟通，每一种程序语言都有它自己的类型 (Type) 系统。同样是 int，在 JavaScript 及在 C 语言中却有不同的定义。

综合以上说明，一个程序开发者所希望拥有的开发平台，应该要有下列特点：

- 要具有面向对象编程的特性，可以将对象重复使用。
- 要能够让不同的程序语言可以自由沟通，让开发出来的对象可以轻易地与其他人分享。
- 除了可以开发一般软件之外，还要能开发最新的 Web 应用软件或 Web 服务。

Microsoft 的 .NET 框架就是针对以上目标所开发出来的开发平台，它不但可以满足以上 3 点需求，还解决了面向对象程序中常见的“DLL Hell”问题。接下来，本章将参考 Microsoft 的文件，简单地介绍 .NET 框架的基本概念。

### 1.1.1 .NET 框架的主要组件

参考图 1-1，当您在安装 Visual Studio.NET 时，其中有一个选项为  .NET 框架 SDK。只要您选择完全安装，即可取得整套的 .NET 框架开发平台参考文件。

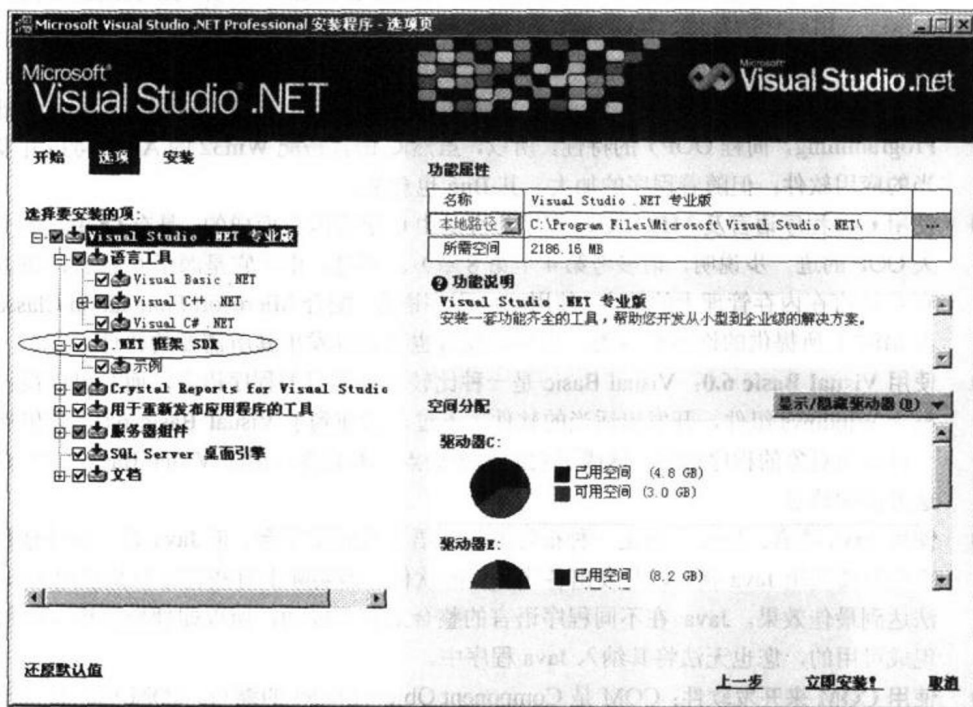


图 1-1 安装 Visual Studio.NET 及 .NET 框架 SDK



说明

Visual Studio.NET 有数种不同版本，本文所用的是中文专业版。选择完全安装约需要 2266MB 的硬盘空间。

.NET 框架包括通用语言运行时 (Common Language Runtime, 简称 CLR) 和类库 (Class Library) 两个主要组件, 以及一个通用语言规范 (Common Language Specification, 简称 CLS), 如图 1-2 所示。

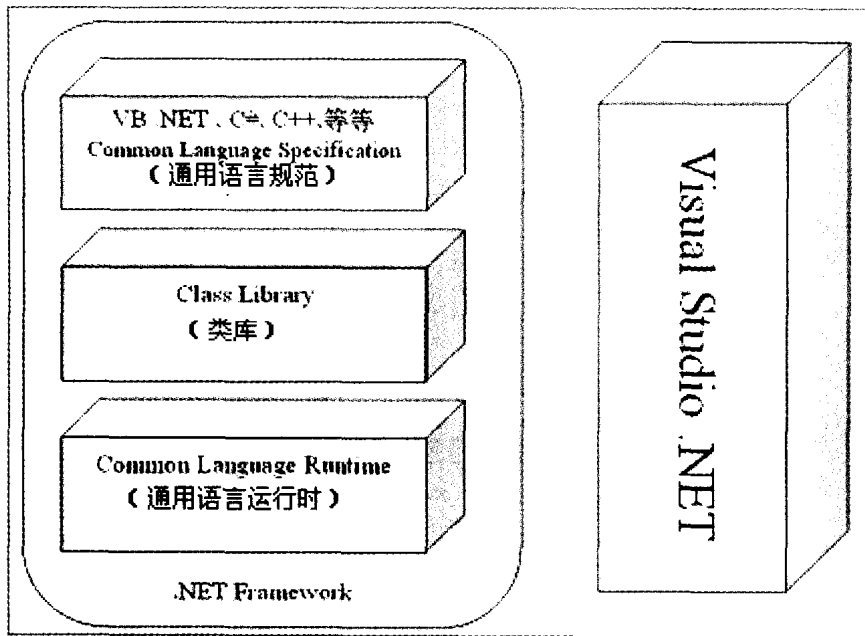


图 1-2 .NET 框架与 Visual Studio.NET

.NET 框架是一种跨程序语言的运算平台, 只要满足 CLS 规范的程序语言, 都可在 .NET 框架上执行。根据统计, Visual Studio.NET 已支持超过 25 种程序语言。



说明

在 Visual Studio.NET 中提供 Visual Basic.NET、Visual C# 及 Visual C++ 等程序语言。本书的范例及说明大部分都是使用最简单易学的 Visual Basic.NET。

### 1.1.2 通用语言运行时

Common Language Runtime (CLR) 是 .NET 框架中最重要的一个组件。虽然以前版本的 Visual Basic 及 C 语言也有各自的 Runtime Library (例如 MSVBVMXX.DLL 及 MSVCRTXX.DLL), 不过其功能比不上 .NET 框架的 CLR, 不同程序语言之间也无法互相沟通。

CLR 负责管理内存、编译、执行程序代码、程序代码安全验证和其他的系统服务。参考图 1-3, 这是一个典型的 .NET 应用程序执行流程。在本节中我们将按照这个流程来说明 CLR 的功能。

当程序开发者使用任何一种与 .NET 框架兼容的程序语言 (例如 Visual Basic.NET) 完成原始程序代码之后, 要先用各自的程序语言编译器 (Compiler) 检查语法是否正确, 然后编译成 .NET 框架的 DLL 文件或 EXE 文件。请注意, 编译出来的 DLL 或 EXE 并不像以前 Visual Basic 的 P-Code, 也不是本地 (Native) 机械语言, 而是一种新的格式, 称为 Microsoft Intermediate Language (简称

MSIL 或 IL)。不论您是用 Visual Basic.NET 或是 Visual C#来编译，其结果都是 IL 格式。

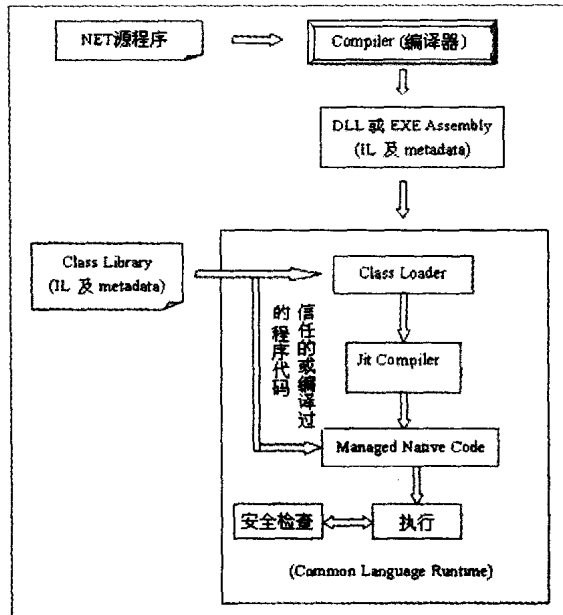


图 1-3 CLR 执行流程

MSIL 是一种包括可用加载、存储、初始化和调用对象方法的指令集，它同时可以自动管理内存的存取，也可进行错误的例外处理。MSIL 指令集与 CPU 的类型或性能无关，所以在程序代码可以执行之前，它必须再编译一次，转换成 CPU 特定的程序代码（此时才算是本地机器码）。

参考图 1-3，原始程序代码编译之后，除了产生 MSIL 之外，还会产生一些元数据（Metadata）。MSIL 与 Metadata 结合便成为编译后的组件（Assembly），通常是以.DLL 或.EXE 为扩展名。元数据与组件在.NET 框架上扮演着相当重要的角色，稍后将有更详细的说明。

在图 1-3 中，将 MSIL 转为可以执行机器码的是一个称为 Just-in-Time（简称 JIT）的编译器。不同的 CPU 有不同的 JIT 编译器。参考图 1-3，有人认为原始程序代码经过第一次编译之后还要经过 JIT 的编译才能执行，怀疑这种两阶段的编译可能会影响执行速度。但是，实际上当您的程序代码第一次调用对象的方法时，它才要通过 JIT 编译；当您第二次调用对象的同一个方法时，它可以直接使用编译过的机器码来执行，所以影响相当有限。

如果您的程序中引用到其他类库或组件，则在 JIT 编译前会先通过一个类加载器将相关的类加载之后，再进行编译。最后，当您的程序要执行前，它还会进行安全检查，确认程序代码中所调用的对象类型是否正确，以及是否有足够的权限来执行此程序。

### 什么是 Metadata（元数据）



元数据是一种二进制信息，记录了模块或组件中所参考或定义的类型及成员。在执行程序代码时，CLR 会将元数据加载到内存，并依元数据的描述来处理类、成员及继承等等信息。元数据存储下列信息：

- 组件说明：包括组件的版本、输出的类型、是否参考其他组件以及组件的安全性使用权限。



- 类型说明：包括成员（方法、属性、事件等等）、名称、基类（Base Class）及实际接口等。
- 属性：修改类型和成员的额外描述项目。

由于元数据所存储的信息非常多，包括与其他模块或组件互动所需的一切数据，所以在 CLR 中甚至不需要登录操作系统。由于元数据可以充分地描述应用程序，故在执行时有较高的可靠性。

另外，元数据也让 CLR 可以执行各种不同的程序语言。Visual Basic.NET 的程序开发者可以继承 Visual C# 所建立的对象；同一个项目中，也可以使用不同的程序语言来建立各自的程序。借助元数据的帮忙，这些程序语言可以互相沟通，彼此共享。

### 什么是组件（Assembly）

在图 1-3 中，经过初步编译的 DLL 或 EXE 文件，在 .NET 应用程序中称为 **组件**。组件是组成 .NET 应用程序部署（Deploy）、版本控制、重复使用、启动范围设置和安全权限的基本单位。组件中所包含的元数据也称为**组件信息清单**（Assembly Manifest），其中包含关于组件内容、版本控制和相关性的信息。由于组件中含有版本信息，所以可以解决过去因 DLL 文件版本不同所产生的 DLL 冲突（有人称之为 DLL HELL）。

另外，由于组件中的信息充足，故不必登录操作系统即可正常运作。这样可以使您的应用程序更可靠，也更容易部署。通常您只要将您完成的应用程序复制到客户的计算机，就可以安装 .NET 框架的应用程序了。



说明

由于 .NET 框架应用程序在执行之前要经过 Common Language Runtime 的再处理，所以客户端的计算机必须要安装 .NET 框架才能执行您的 .NET 应用程序。

## 1.1.3 类库

.NET 框架的两大组件，除了 Common Language Runtime 之外，另一个就是类库（Class Library）。类库是以支持 Web 标准与应用为首要设计目标，它的特点是可以统一应用程序开发程序，使用简单，且具有高度的扩充性。

类库的程序代码是以层次式的**名字空间**（Namespaces）与**类**（Class）组织而成，它不仅是面向对象的，而且也是面向组件的（Component Oriented）。类库中的每一个类都是对象，也是组件。您可以利用其属性、方法及事件等调用、使用选定的类。

类库的涵盖范围很广，参考图 1-4，不管您是要开发 Windows 应用程序、ASP.NET Web 应用程序、Web 服务或 Windows 服务，都有相对类库及模板可以使用。

您不需要去了解到底类库中总共有多少个链接库，因为 Microsoft Visual Studio.NET 已经将这些类库整理分类，做成很多模板。您只要选择适当的模板，就可以使用相关的类库。图 1-4 右上方的模板清单列出了 Visual Studio.NET 专业版所提供的范本，不同版别可能会有不同的模板清单。