



本书内附光盘

网络精英系列

ASP.NET

行家入门实务

康际科技 王浩然 颜瑞琳 赖光辉 编著

本书特色

- ◆ 详细介绍了什么是ASP.NET、ASP与ASP.NET的差别、ASP.NET的特点，让您充分掌握ASP.NET
- ◆ 以详细的说明，指导您如何构建ASP.NET的运行环境，让您快速上手
- ◆ 通过范例引导您学习ASP.NET的基本语法，让您实现以往在网页中无法实现的功能
- ◆ 通过详细的范例解说，了解ASP.NET HTML控制组件与ASP截然不同的方便之处与强大功能
- ◆ 介绍ADO.NET的语法及使用方法，包括如何产生一个DataSet、DataReader及DataView等

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

ASP.NET

行家入门实务

康际科技

王浩然

颜瑞琳

赖光辉

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: 01-2001-5290 号

版 权 声 明

本书中文繁体字版由台湾金禾资讯股份有限公司出版(2001)。本书中文简体字版经台湾金禾资讯股份有限公司授权由中国铁道出版社出版(2001)。任何单位或个人未经出版者书面允许,不得以任何手段复制或抄袭本书内容。

图书在版编目(CIP)数据

ASP.NET 行家入门实务/王浩然, 颜瑞琳, 赖光辉编著. —北京: 中国铁道出版社, 2002. 1
ISBN 7-113-04503-0

I. A… II. ①王… ②…颜 ③赖… III. 主页制作-程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字(2001)第 096763 号

书 名: ASP.NET 行家入门实务

作 者: 康际科技 王浩然 颜瑞琳 赖光辉

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: 苏 茜

特邀编辑: 王占清

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: 19 字数: 462 千

版 本: 2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-04503-0/TP·663

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

ASP.NET 是 ASP 的最新版本，微软此次更新在功能上有相当大的改进，并不像 ASP 1.0、ASP 2.0、ASP 3.0 一般只做了小幅度的修改。因此，若想要学习网页程序设计，ASP.NET 当然是最佳选择。本书采用大量的范例让您了解 ASP.NET 各项使用方式及如何进行编程，所以当您阅读完一个章节后，即可亲自动手实际操作编写程序，当您阅读到后面的章节时，便能更深刻地领略 ASP.NET 的强大功能，本书编写的方式是由浅入深逐步说明，一定能让您快速上手。

本书所有范例文件均在所附光盘 Sample 文件夹内。

本书由台湾金禾资讯股份有限公司提供版权，经中国铁道出版社计算机图书项目中心审选，张翰文、李自远、马超、杨小平、段小明、廖康良、陈贤淑及孟丽花等同志参与了本书的整稿及编排工作。

2002 年 1 月

目 录

第0章	导读	1
	哪些人应该阅读本书	2
	本书涵盖内容及建议阅读方式	3
	本书的标示方式	4
第 1 章	ASP.NET 介绍及说明	7
	.NET 与 .NET Framework 简介	8
	ASP.NET 说明	13
第 2 章	ASP.NET 使用前准备工作	17
	硬件及软件需求	18
	ASP.NET 开发环境安装	19
第 3 章	ASP.NET 基本语法	27
	ASP.NET 网页可视化组件 (Visual component)	28
	Server Control 组件	35
	Form 的特性与限制	38
	ASP.NET 网页程序流程	38
	<Script>...</Script>——谈 ASP.NET 流程架构	39
	关于 VB.NET	40
	基本程序语句	44
	ASP.NET 事件驱动(Event Drive)	58
	ASP.NET 事件处理程序	73
	结论	82
第 4 章	ASP.NET 对象模型 (Object Model)	83
	Object Model 对象使用语句	84
	Object Model 对象属性/方法	84
第 5 章	HTML 控制组件	95
	HTML Controls	96
	结论	134
第 6 章	Web 控制组件	137
	Web Controls	138
	结论	193

第 7 章	ADO.NET 与数据控制组件	195
	ADO.NET	196
	数据控件	223
	结论	264
第 8 章	数据验证控制组件及其他相关应用	265
	数据验证控制组件	266
	结论	281
附录 A	Windows 2000 Professional 安装说明	283
	硬件需求	284
	开始升级	284
附录 B	IIS 5.0 设置说明	289
	启动 IIS 5.0	290
	Web 站点的设置	292

第0章

导读

ASP.NET 行家入门实务

随着互联网的蓬勃发展，科技的进步也一日千里。或许您会觉得：记得不久前才将 ASP 学习到一个阶段，现在又听到新的名词：“ASP.NET”，是不是又要开始另一段的学习之旅呢？其实不然，如果您已经会使用 ASP 的开发方式，那学习 ASP.NET，并没有您想象地那么困难；若您才准备或刚开始学习 ASP，那么直接跨到 ASP.NET 的领域，这正是时候。为什么这样说呢？刚开始大家并不了解 ASP 能为我们做什么，逐渐有人开始研究 ASP，直到目前的 ASP 3.0，前后历经了四、五年的时间，使得 ASP 成为要在网站上显示动态数据最普遍的开发工具。若您已经学习如何使用 ASP，那非常恭喜您！您已经具备了 ASP.NET 的基本功了，要认识及学习 ASP.NET 对您来说，只要再跨出一小步就可以获得高效率开发动态网站的技巧；若您准备或刚开始学习 ASP，您已经节省了许多摸索时间，因为前人已经将研究改进 ASP 的大树种好了，您只要在树荫下轻松地学习 ASP.NET，即可踏在前人的肩膀上向前迈进。

哪些人应该阅读本书

★ 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，只要您有简单的逻辑概念都可以通过本书学习到 ASP.NET 的基础。

★ 曾有网页编写经验的程序设计人员

曾经使用 IDC、Java、ASP 等动态网页开发工具编写过动态网页程序的人员，本书更是您作为踏入 ASP.NET 的最佳选择。

★ 对编写动态网页非常专业的程序设计人员

若您对编写网页非常专业，曾经有过利用 COM、COM+ 来编写动态网页的经验，本书将更能提供您快速地对 ASP.NET 的认识，并且学习到 ASP.NET 的精神。

★ 网站设计、规划的系统分析师/规划人员

如果您是网站设计、规划的系统分析师或是规划人员，在不了解 ASP.NET 可以提供哪些强而有力的服务时，相信您很难发挥 ASP.NET 的优势，也就无法有效率地开发您所想要的内容。而通过本书您可以了解 ASP.NET 能带给您的所有好处。

★ 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 还有其他您尚未了解的功能，本书详尽地说明可以让您对 ASP.NET 有耳目一新的感觉。

本书涵盖内容及建议阅读方式

为了让您快速领略和学习到 ASP.NET 是如何协助您高效率地建置动态网站的,在本书中分为八个章节,带领您迈向 ASP.NET 的康庄大道。若您已经非常熟悉 ASP 的开发方式,那么建议您可以从第 4 章开始阅读即可顺利进入 ASP.NET 的领域。若您不了解 ASP,那也无妨,依章节顺序阅读便可以学会 ASP.NET。

本书不像一般书籍,通过说明再加上一点范例让您了解什么是 ASP.NET? 本书采取一贯且大量的范例让您了解 ASP.NET 各项的使用方法以及在程序中该如何应用编写,所以当您阅读完一个章节后,便能够马上应用编写程序,这也就是本书编写最主要的目的。所以当您阅读到后面的章节时,便能更深刻地领略 ASP.NET 的强大功能,所以本书编写的方式是由浅入深逐步地说明,能让您快速地上手。

第 1 章

在本章中您可以了解到什么是“.NET”?为什么要使用 ASP.NET? ASP.NET 与 ASP 有什么差异? ASP.NET 的特点是什么?这可以让您对 ASP.NET 有一番了解后,进而让您更加发挥 ASP.NET 的强大功能。

第 2 章

当您已经了解 ASP.NET 后准备要大展身手前,准备一些必要的工具,以便于快速学习及体验 ASP.NET 的威力与快感。本章以详实的说明,教您建置一个学习环境,所以本章是您不可不读的章节,因为“工欲善其事,必先利其器”不是吗?

第 3 章

不论您是否学习过 ASP 语法,本章中将通过范例引导您学习 ASP.NET 的基本语法,当然也提到一点在原来 ASP 及 HTML 中的写法与差异,所以当您阅读完毕后,便可以深刻地了解 ASP.NET 的基本语法,并让您实现以往在网页中无法完成的控制行为。

第 4 章

不论您是否学习过 ASP 的系统对象,在本章您都可以再次或初次了解 ASP.NET 所提供的系统对象,这些对象都比 ASP 所提供的更为丰富及方便,您可以借此来搜集并显示您所需要的信息,让您可以不用编写太多的对象或函数,直接应用系统提供的功能,让程序开发更为便利。

第 5 章

从本章开始进入 ASP.NET 的领域,虽然在第 3、4 章已陆续说明 ASP.NET 基本语法及系统对象,不过正式应用与程序编写都从本章开始,通过范例逐一说明每一种 ASP.NET 的

HTML 控制组件，让您了解 ASP.NET HTML 控制组件与 ASP 截然不同的方便之处与它的强大功能，只要您阅读每一个范例，便可以按部就班地了解每一项的控制，所以从本章开始，请您详细地阅读吧！

第 6 章

本章将说明 ASP.NET 中最方便的 Web 控制组件(Web Controls)，微软已经将 Web Controls 打包的很完整，并以面向对象的概念来设计，且又加入了许多控制的属性、方法及事件，所以 Web Controls 在组件的整体功能变得十分强大。相信经过本章详尽地说明，必能让您从中了解更多的应用方式。

第 7 章

在本章中将介绍 ADO 的升级版本 ADO.NET 及数据控制组件，如 Repeater、DataList 及 DataGrid 等 Web Controls。本章重点在介绍 ADO.NET 的语法及使用方式，包括如何产生一个 DataSet、DataReader 及 DataView 等，而数据控制组件要如何与它结合并以很简单且容易的方法将数据显示在网页上。将使用实际的范例应用，让您与数据库上的应用更加顺利。

第 8 章

在本章中将介绍 ASP.NET 提供的功能强大却简单易用的工具——数据验证组件，及其应用。这对一般在网页上提供在线验证来说是一个强大且方便的工具，当您阅读完本章后，相信您可以快速地开发网页上用户输入验证的方式，而不用使用大量的 VBScript 或 JavaScript 来辛苦地编写检查程序。

附录 A

若您没有安装过 Microsoft Windows 2000 的经验，本章将教您如何简易地安装 Windows 2000，因为本书并不是专述 Windows 2000 的书籍，若有任何其他的设置，例如：如何建置 DHCP、VPN 等问题，请您参阅其他专论 Windows 2000 的书籍。

附录 B

因为在学习 ASP.NET 的过程中，您会想要在浏览器上直接看到成果，但在学习过程中一定会发生问题，但又不知道错误发生在哪里，有时必须借助网站上的设置，将错误信息告诉您，在本章中会教您如何设置 IIS 的参数，好让您可以方便地进行 ASP.NET 的环境设置。

本书的标示方式

为了方便及阅读本书，本书编写标示的意义说明如下：

- 当内容中提到画面上的按钮时，会使用【】括住，例如：【确定】、【取消】。

- 当内容中提到画面上的功能时，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】。
- 当内容中提到一些鼠标单击的过程，会使用下列的表示方式，有些单击时不用鼠标左键点选，但有些功能会需要您使用鼠标左键单击才有作用，这些都是依照 Windows 的惯例，本书不再说明此惯例，除非是使用到鼠标其他的功能，才会特别提出说明。例如：【开始】→【程序】→【管理工具】→【事件查看器】。
这表示请您使用鼠标点到【开始】按钮上，当出现菜单时，移动鼠标指到【程序】，当出现进一步的菜单时，再请使用鼠标指到【管理工具】，出现进一步的菜单后，请单击【事件查看器】功能。
- 当内容中要请您注意的地方，会采用下列方式表示。

在这里会叙述您该注意的内容。



注意

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

在这里会说明一些技巧，请您细细品味一下。



技巧

原书空白页

ASP.NET 介绍及说明

第一章

在介绍“ASP.NET”之前,先让您了解一下什么是“.NET”,因为 ASP.NET 是.NET 架构中的一环,当您有了.NET 全盘的概念后,您就能够了解 ASP.NET 在整个架构中扮演何种的角色,就更能够发挥 ASP.NET 的强大功能。

.NET 与 .NET Framework 的简介

2000 年 6 月,微软在美国召开的 Professional Developers Conference 上揭露了有关微软在 Windows 及因特网方面下一代产品的信息,微软对于未来不论是生活上或是工作上都提供了完整的解决方案,而这个方案就是——“.NET”(您可以浏览微软中国网站内容,<http://www.microsoft.com/china/products/net/default.htm>,可以先从“影片欣赏:迈向.NET 未来新世纪”中领略一下.NET 所涵盖的内容)。

基本上.NET 是架构在网页服务(Web Services)的原则下,微软目前正通过完整.NET 架构平台(包含各项组成部分),以提供网页服务稳固的基础开发架构。而.NET 架构大概可以分为下述几类:

★ .NET Framework (这是开发程序的核心部分)

Framework 提供建置及部署网页服务的环境,这包含了通用语言执行阶段(Common Language Runtime)、多层次的统一对象类链接库(Unified Class Libraries)及 ASP.NET 等部分。

★ .NET Product

所有微软主要产品,从 Windows 到 Office 都被集成进来到.NET 的架构下,产品都通过 XML 来叙述资料,并且利用 SOAP 作为产品间沟通的协议,这也就是.NET 集成的架构。

这样说明好像太笼统了,但如果您只想了解什么是.NET 的话,上述的说明就可以让您概略地知道:将所有产品通过 XML(Extend Markup Language)来描述数据,并通过 SOAP(Simple Object Access Protocol)在产品间交换数据,进而集成产品的新架构,这就是.NET 的基本架构。

到此,若您不想继续深入了解.NET 的架构,可以直接跳到下一节,而可以忽略接下来.NET 架构的说明。不过多花十分钟是非常值得的,因为以下所提到的概念,绝对可以让您对.NET 更靠近一点。

首先,.NET Framework 包含的范围非常广,从系统层到应用程序开发层都包含在其中,.NET Framework 提供许多工具用来集成各层次。其次.NET Framework 整个的架构都是为了让您可以快速地、容易地开发因特网上的应用程序而设计的。所以.NET Framework 提供了许多的组件让您可以有效率地开发应用程序,也就是说,您不用再一行一行地编写程序,只要引用组件或将组件稍加修改就可以符合需要,而创造所需的程序。例如:以往在网页上制作类似 Client/Server 应用程序中的 Grid,必须花费许多时间针对每项的控制做调整或设计,现在您不用那么卖命了,因为.NET Framework 已经提供了这样的组件供您使用,如此便可以快速地做出非常有看头的网页。

请先参考图 1-1,图中是.NET Framework 的简易架构图,在图中就可以发现到.NET

Framework 分成了几个大的层次，每一个层次各司其职分别有不同的作用（当然每一层次也可细分下去）。现在就针对每一个层次的功能进行说明（因为本书的重点在 ASP.NET，若您想要再深入了解 .NET Framework，请参考网址：<http://msdn.microsoft.com/library/default.asp?URL=/library/dotnet/cpguide/cpconthenetframeworkclasslibrary.htm>）。

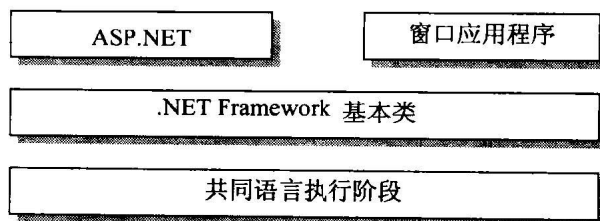
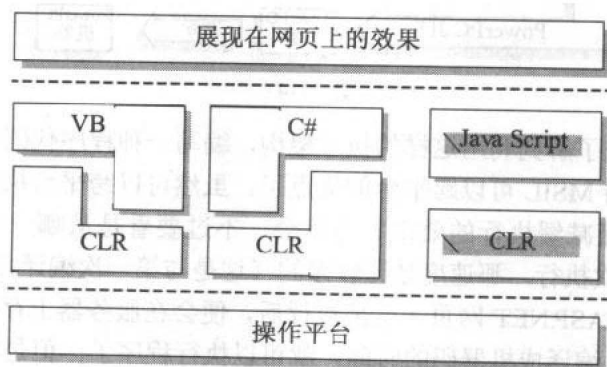


图 1-1

通用语言运行阶段 (Common Language Runtime, CLR)

CLR 是 .NET Framework 的核心部分。一般来说，运行阶段 (Runtime) 提供程序执行的环境，所以 CLR 也提供 .NET 程序编译后 (Compile) 运行的环境。或许您会有疑问，在以前编写 VB 或 C 的程序也是要编译，再加上链接 (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文件一起放在执行程序端才能执行，那 .NET 程序既然有编译，那需不需要链接呢？这个问题非常好（笔者刚开始学习 ASP.NET 时也有这个疑问）！基本上，微软提供 CLR 的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作为开发 .NET 的应用程式，通过 CLR 都可以展现出同样的效果。也就是说，每一种程序语言各有优点，您可以这部分程序用 VB 来编写，另外一部分程序用 C# 来编写，但都可通过 CLR 兼容并蓄地融合在一起，展现出您想要的效果。

图 1-2

也就是说，只要可以编译成 MSIL (Microsoft Intermediate Language) 的程序语言，都可以作为制作 ASP.NET 网页的程序语言；这就告诉您：只要将程序编译后就可以提供执行，而不需要链接成可执行文件（根本也不需要链接成可执行文件）。那什么是 MSIL 呢？请先参考图 1-3 您大概就会了解什么是 MSIL 了。

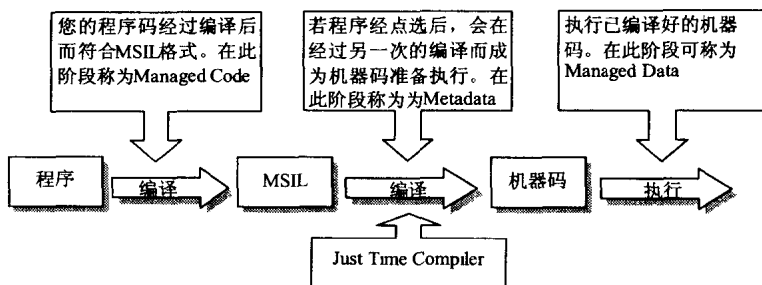


图 1-3

由图 1-3 可以了解到，MSIL 是介于程序代码与机器码之间的语言（所以您并不用编写 MSIL 程序），当编译好 MSIL 的程序代码后，就等待用户点选程序后，便会快速地再次编译后执行。

看到这里您一定会想到：为何不直接将程序编译成机器码，然后再加以执行呢？为什么要如此大费周折地绕了一圈呢（没错！笔者之前也有这样的疑问）？其实，MSIL 程序代码非常接近机器码，从 MSIL 程序代码再编译成机器码加以执行是非常快速地，不同型号的 CPU 就会有不同的 JIT 编译器，如此便可以将同样的 MSIL 程序代码在不同型号的 CPU 机器上执行，请参考图 1-4。

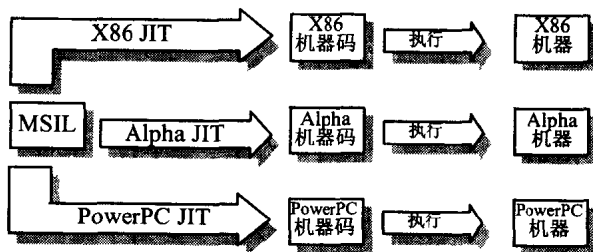


图 1-4

从图 1-4 就可以先了解到利用这样的执行架构，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 可以跨平台的优点了。虽然可以跨平台执行同样的程序固然很好，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看了；若是 ASP.NET 的网页第一次执行，那速度是比较慢的（这是与第一次编译并执行 ASP 网页的情况下做比较），但是当 ASP.NET 网页一旦被执行后，便会在服务器上存储起来，第二次执行时，只需花费从 MSIL 编译成机器码的时间，就可以执行程序了；但是利用第二次执行 ASP 的程序时，服务器又要重新再解译，所以与第一次执行 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多，所以第一次被执行时速度较慢，但在之后节省多次执行的时间，这是非常有效益的（请参考图 1-5）。

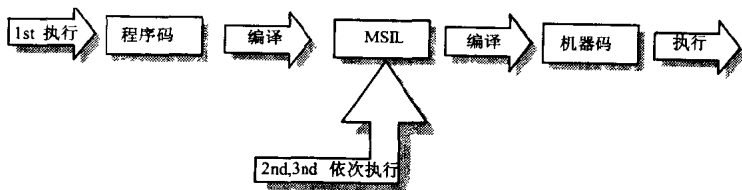


图 1-5

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”，此时所编译的 MSIL 程序代码与 CPU 所可以执行的机器码并没有任何关系，也就是不管 CPU 是哪一种形式，所编译出来的 MSIL 程序代码都一样。那在 MSIL 程序代码中包含了哪些好东西呢？MSIL 程序代码纯粹只包含：

- MSIL 定义的指令集（广泛定义的指令集）
- 对象的加载、存储、起始及调用其他方法、组件等指令
- 有关算术运算、逻辑运算、流程控制、内存访问及异常处理等指令

因为上述指令集已经将 ASP.NET 程序的绝大部分都已经编译好了，所以要执行时，只需转换成 CPU 所认识的格式便可以执行，所以利用 JIT 编译器转换 CPU 指令集是非常快速的。

在此时编译后程序代码为什么称为“Managed Code”？因为 .NET 系采用编译的方式进行，当利用 JIT 编译成可执行的机器码后使用并且将此机器码存在内存中，若源程序修改后再进行编译成为 MSIL 程序代码后，就已经纳入管理了，当使用再次点选或执行此程序时，.NET Framework 就会非常聪明地使用新的 MSIL 程序代码进行编译、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尚未与 CPU 有任何关系）。

那什么是“Metadata”？Metadata 是放在内存中或 .NET Framework Portable Executable (PE) 文件中的二进制文件（此时已经通过 JIT 编译器产生 CPU 已认识的程序代码），JIT 编译器将程序、程序所引用的文件与数据类型编译在一起，在此时起一个融合的作用以待执行时使用；因为程序代码在此阶段已起了变化所以称为“Metadata”（笔者查了字典后，发现 meta-前缀是含有“变化、变换”的意思，正与 Metadata 的含义雷同）。

在执行阶段因为要管理程序中的生命周期（在此生命周期指的是：当源程序修改、编译成 MSIL 程序代码后，当用户重新执行程序时，将自动删除内存中的旧程序，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，以此类推）、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！先思考一下整个架构，您就会发觉 .NET Framework 的好处实在不少。之前提到了 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 的架构，可以集成不同程序语言所编写的组件而成为您所要求的程序，也就是跨程序语言的功能。

一般而言，每一个程序设计人员都有自己较为熟悉使用的程序语言，相信您也不异常，只要您所熟悉的程序语言符合 MSIL，则您只要将程序编写完毕，就可以经过编译后便可以执行您的程序。现在就分别从两种角度来说明跨程序语言的做法。

★ 个人独立开发程序

在您独立开发程序时，有时候会引用一些组件来提升开发的效率，当然您所引用的组件必须要符合 MSIL 规范，可是您却不用在意此组件是用什么程序语言开发的，有可能是用 VB 编写而成、当然也有可能是用 C# 开发完成的，所以在以往使用某种程序语言开发，就只能引用同一种程序语言所开发的组件，这相当麻烦。所以在 .NET Framework 中，只要程序语言符