

编程宝典
2002
8

北京希望电子出版社 总策划
[美] Keith Franklin 编著
王海峰、李位星、李新鹏 等译

Microsoft VB.NET 开发人员指南

理解为什么 VISUAL BASIC 是为 .NET 框架的首选语言
探索 VISUAL BASIC.NET 语言的新特性
深入了解 .NET 框架如何工作
探索 .NET 框架的能力

SAMS

中国科学院出版集团
北京希望电子出版社

736

“十五”国家重点电子出版物规划项目·计算机知识普及和软件开发系列

TP393.4
F-83C

编程宝典 2002 8

北京希望电子出版社
[美] Keith Franklin
王海峰、李位星、李新鹏

总策划
编著
等译

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载，
也可到视听部复制

Microsoft VB.NET 开发人员指南

理解为什么 VISUAL BASIC 是为.NET 框架的首选语言
探索 VISUAL BASIC.NET 语言的新特性
深入了解.NET 框架如何工作
探索.NET 框架的能力

SAMS



中国科学出版集团

北京希望电子出版社

内 容 简 介

本书是一本帮助 Visual Basic 程序员迅速跟上 Microsoft .NET 开发平台/框架所带来变化的优秀指导书。本书共分 14 章, 涵盖的内容包括: .NET 框架、Visual Studio.NET、面向对象的编程思想、方法属性和事件、对象的创建和销毁、创建 VB.NET 工程、数据和对象类型、控制逻辑流、错误处理、ADO.NET、.NET 应用程序开发设计结构和实现、互操作能力、从 VB 6 移植以及 .NET 的发展趋势。

本书从多个角度把 Visual Basic 和 .NET 框架下的 Visual Basic.NET 做了深入比较, 同时给出资深程序员多年积累的大量经典示例和源代码。本书条理清晰, 层次安排合理, 使读者对 .NET 框架有清楚的认识和了解。

本书适合于有 Visual Basic 编程经验的开发人员, 同时对于有 Java、C++ 和 Delphi 经验的开发人员或 .NET 框架有兴趣的其他程序员来说也有很大的参考价值。

版 权 声 明

本书英文版名为“VB.NET for Developers”, 由 Sams publishing 出版, 版权归 Sams publishing 所有。本书中文版由 Sams publishing 授权出版。未经出版者书面许可, 本书任何部分不得以任何形式或任何手段复制或传播。

盘 书 系 列 名 : “十五”国家重点电子出版物规划项目 计算机基础知识普及和软件开发系列
编 程 宝 典 2002 (8)

盘 书 名 : Microsoft VB.NET 开发人员指南

总 策 划 : 北京希望电子出版社

文 本 著 作 者 : [美] Keith Franklin 编著 王海峰、李位星、李新鹏 等 译

C D 制 作 者 : 希望多媒体开发中心

C D 测 试 者 : 希望多媒体测试部

责 任 编 辑 : 大成

出版、发行者 : 北京希望电子出版社

地 址 : 北京中关村大街 26 号, 100080

网 址 : www.bhp.com.cn

E-mail: lwm@hope.com.cn

电话: 010-62562329, 62541992, 62637101, 62637102, 62633308, 62633309

(图书发行) 010-62613322-215 (门市) 010-62547735 (编辑部)

经 销 : 各地新华书店、软件连锁店

排 版 : 希望图书输出中心 董淑红

CD 生 产 者 : 北京中新联光盘有限责任公司

文 本 印 刷 者 : 北京广益印刷有限公司

开 本 / 规 格 : 787 毫米×1092 毫米 16 开本 16.875 印张 287.5 千字

版 次 / 印 次 : 2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

印 数 : 0001-5000 册

本 版 号 : ISBN 7-900088-17-2

定 价 : 33.00 元 (本版 CD)

说明: 凡我社产品如有缺页, 可执相关凭证与本社调换。

编辑的话

一本优秀的计算机图书前面总是存在这样的无休无止的简介、前言、序、序、序……。这常常让我感到迷惑，甚至怀疑这样做有没有必要：把自己的想法强加给别人是否正确？但是当编辑完这本书以后，我完全明白撰写那些我曾经认为无聊东西的人的想法：当你成功塑造了一件满意的作品时，你根本没有办法抗拒将自己对这件杰作的想法与大家分享的冲动！

曾经有位微软 ATC 培训中心的讲师到出版社讲课，我提了一个问题：.NET 框架及其相关概念的含义和联系。可能我的天赋有限，我没能完全理解讲师的解答。但是从编辑这本书的过程中，我每前进一页，对.NET 框架的理解就更深一层。作者的思路简直清楚得玲珑剔透，他对读者需求的把握之准确令人惊讶。

感谢微软创造了.NET 这样匪夷所思的东西，感谢 Keith Franklin 在第一时间为我们提供了第一手权威资料，同时也庆幸我们找到王海峰等这样的有大量开发经验的朋友来将这份资料精心塑造成中文版……

带着这本书给我的东西，我可以轻易的和一些优秀的程序员交流关于.NET 框架的一些问题，以至于他们中的一些把我当成高手，直到现在还有程序员和我探讨一些技术问题，当然，大多数情况下交流无法进行下去。

作为一个编辑，最大的乐趣就是拿到这样一本有价值的书，冲上一杯咖啡，然后享受他们……

大成

前言

自从 1991 年引入 Visual Basic 1.0 以来, Windows 应用程序开发一直被基于 RAD 的开发模式支配。但随着我们进入基于 Internet 应用程序开发的成熟时代, Windows 和 Internet 发展对更有效的程序开发模型的需要就很明显了。

为了利用 Internet, Microsoft 改变了 ActiveX 的含义和技术结构, 使得这项技术嵌入浏览器时更轻便。Microsoft 为客户端开发生产出了 Visual Basic 的轻型版本——VBScript。对 Visual Basic 也做了些改进, 使它成为首要的组件开发工具。

现在为第二次浪潮的准备工作已在进行之中, 其中包括 Java 和动态服务器页面(Active Server Page)开发的更新工具。同时, Microsoft 开始为 Windows 和 Internet 开发设计一种全新的程序开发模型。其努力的结果是 .NET 框架。 .NET 框架在 2000 年 7 月的 Microsoft 专业开发者会议(Microsoft Professional Developers Conference)上引出的。

.NET 框架是建立用于 Internet 时代应用程序的综合性平台, 它仍然允许开发人员建立使用 Windows 平台的完全富于新特色的应用程序。 .NET 框架包含一个内容极其广泛的类框架, 为程序开发人员做的大部分一般的任务提供服务。此外, .NET 框架为安全性、跨平台通信、资源管理和扩展性提供了完整的服务。

选择 .NET 作为将来 Windows 发展的平台, 同时选择 Microsoft 作为建立 Internet 应用程序的一套办法, Visual Basic .NET 从一个新的起点被重新编写。 Visual Basic 开发者会对此既兴奋又不安。正如 Yoda 在星球大战 4: 帝国反击战(Star Wars Episode V: The Empire Strikes Back)中说的: “你肯定对所了解的不了解。”

在很多情况下不是这样。 .NET 框架在几乎所有情形下会提供逻辑更明确的方法来完成 Visual Basic 早期版本中需要自己编写代码的任务。本书的目标是帮助你提速到 Visual Basic .NET, 同时指出很多在把 Visual Basic 搬进 .NET 框架时所做的改动。

谁应该阅读这本书?

本书主要是帮助现在的 Visual Basic 开发者迅速跟上 .NET 框架带来的变化。此外, 有 Java、C++ 和 Delphi 经验的开发人员或对 Visual Basic 和 .NET 框架有兴趣的其他应用程序开发工具有经验的人来说也会发现这本书很有意义。

虽然这本书致力于 Visual Basic .NET, 它主要的目标是介绍 Visual Basic 和 .NET 框架如何协调工作。它还介绍 .NET 框架带来的开发的新变化, 包括 ASP.NET、Windows Form 和 Web Service。

在开始之前需要什么？

在开始用 Visual Basic.NET 开发应用程序之前，需要从 Microsoft 处订购 CD。

我的建议是在 Windows 2000 或 Windows XP 机器上运行该产品。

现在有很多公司已经选用它来调度程序。Bill Gates 在亚特兰大 2001 年 6 月的 TechEd 上说这是 he 看到过的最稳固的使用品。

本书内容快速浏览

第 1 章 .NET 框架

第 2 章 Visual Studio.NET

第 3 章 面向对象的编程思想

第 4 章 方法、属性和事件

第 5 章 对象的创建和销毁

第 6 章 创建 VB.NET 工程

第 7 章 数据和对象类型

第 8 章 控制逻辑流

第 9 章 错误处理

第 10 章 ADO.NET

第 11 章 .NET 应用程序开发设计、结构和实现

第 12 章 互操作能力

第 13 章 从 VB 6 移植

第 14 章 .NET 今后的发展

附录 A WINDOWS FORM 应用程序

附录 B ASP.NET WEB SERVICE 与 WEB FORM 应用程序

关于作者

Keith Franklin 是 IL Naperville 的 Empowered Software (ESS) 的主席和资深设计师。他是 Visual Studio (以前是 Visual Basic Programmers Journal)、.NET 杂志的定期撰稿人。他还是 VSLive 和 BVITS 的发言人。Keith 是 Chicago.NET Users Group 的理事和 www.dot-netrocks.com 网站的特色设计师之一, 该网站是他和 Rocky Lhotka 创建的。作为 ESS 的设计师, Keith 磋商、教授并为建立高度可伸缩性的 Internet、电子商务和分布式客户 / 服务器应用程序的顾客设计解决方案。自从 Beta 1 发布以来, Keith 和 EES 一直使用 .NET 框架、ASP.NET 和 VB.NET 为客户开发应用程序。你可以通过 EES (www.dot-netrocks.com) 或借助发 E-mail 到 ka_franklin@empowered.com 与 Keith 联系。

Rebecca M.Riordan 有几乎 20 年的软件设计经验, 且赢得了作为咨询家、系统分析师和数据库工作支持系统的设计人员的国际声誉。她的经历包括: 在美国、澳大利亚和欧洲领导工程开发; 管理澳大利亚的一个最大的计算机零售商的专业服务部门以及 Microsoft 的资深技术工程师。目前她居住在新墨西哥城, 是 Designing Relational Database System、SQL Server 2000 Programming Step by Step 以及即将出版的 ADO.NET Step by Step 的作者。

谨把此书献给

我把这本书献给我的妻子和孩子们。在过去的六个月中, 我把时间都灌注在写稿上 (尤其是有时信息似乎藏在表面现象之下), 我几乎把他们逼疯 (尤其是 Tami)。Tami, 你使得一切都很值得, 我全心地爱你, 你是我的惟一。你创建了一个温暖幸福的家庭, 对我来说它比任何东西都珍贵; 没有你, 什么都会失去意义。

我有两个可爱的小儿子, 而且就要有第三个孩子了。无论什么时候我听到他们的笑声, Scottie 和 Jeffie 使得太阳好像更加明亮。Scottie, 当我看到你跑过你的第一任棒球教练时 (完全不管他有气无力的叫你停下来), 我禁不住大笑和流泪: 我笑是因为你是那么的可爱; 我流泪是因为你长得是那么快。Jeffie, 你是 Daddy 的小滑稽演员。虽然你看上去更像你的妈妈, 但你的智慧和会挖苦人肯定像我。当我看你给我你那胡闹的傻笑时, 我好像也年轻了。你这个小家伙是 Daddy 的心肝。

我还想把这本书献给我的父亲、母亲、兄弟和妹妹。尤其是我的父亲: Dad, 你从未意识到你的孩子们直接从你那儿得到了多少智慧和同情心。无论我们会怎么跟你争论芝加哥的球队, 我们都爱你, 并尊重你和你的智慧。但最重要的是: 因为你, 我们都长成富于同情心和善良的人。我爱你 Dad, 你的孙子们崇拜你。

致谢

写一本关于某种新产品的书, 是一项很难完成的工作。这需要很多细心的人来完成, 我想感谢他们中的一些。首先, 我要感谢 Neil Rowe, 我的收集编辑, 是他把所有的一切组

织起来并使得更进一步。我还要感谢 Rebecca Riordan，我的开发编辑，他帮助组织书的结构，使信息更清晰。

此外，我需要感谢 Microsoft 创建了这样一个匪夷所思的产品。我特别需要感谢的是 Visual Studio.NET 和 .NET 框架开发组。这种开发平台或开发工具远远超出曾经开发出来的任何产品。特别的感谢要给作者组，是他们花费了如此长的时间和精力与我分享信息。

最后，我需要感谢我那些聪慧过人的伙伴和有强大工作能力的 Empowered Software Solution 开发人员：这仅仅是证明我们建立一个致力于开发人员的面向极度高质量的咨询公司的目标是正确的初步证据。

告诉我们你的看法

作为本书的读者，你是我们最重要的评论家。我们十分重视你的见解，并想知道我们哪里做得好；哪里应该改善；你希望我们出版哪方面的书籍以及其他任何你想送给我们的至理名言。

作为 Sams 公司的联合发行人，我欢迎您的评论。你可以发传真、E-mail 或直接写信给我，让我知道你喜欢和不喜欢本书的地方——以及我们应该在哪方面做工作，使得书更健全。

请注意我不能就与本书主题有关的技术问题帮助你。同时，因为收到的邮件很多，我也许不能回复每封信。

在写信时，请一定要包括本书的题目和作者以及你的姓名、电话或传真号。我会仔细阅读你的评论，并把它和本书的作者和编辑们分享。

传真： 317-581-4770
Email: feedback@sampublishing.com
邮箱: Jeff Koch.Associate Publisher
Sams
201 West 103rd Street
Indianapolis, IN 46290 USA

译者序

自从 Visual Basic 1.0 发行以来，到今天已经有 10 多年的历史了，现在普遍使用的是 VB 6 版本。但随着网络应用的突飞猛进，Visual Basic 在很多方面显得过于单薄，比如安全性方面、跨平台操作、与数据库通信、伸缩性以及使用 Web 服务等方面，已不能适应技术发展的新的需要。Microsoft 正是看到了这一点，努力推出适应 Internet 发展需要和利用 Web 服务的开发平台。这就是 .NET Framework。这种平台与 Visual Basic 语言的结合就是 Visual Basic.NET。

本书从多方面把 VB 6 和 Visual Basic.NET 做了比较，集中阐述了 .NET 在安全性、跨平台通信、资源管理和扩展性提供的完整服务。并对把用 Visual Basic 6 编写的代码移植到 Visual Basic.NET 的可行性和移植方法做了介绍。书中语言逻辑性强，思路清晰，内容前后连贯，并且始终与 Visual Basic 的早期版本作比较，使读者（尤其是对 VB 有经验的开发者）很容易理解 .NET 框架的新特性。同时，本书作者使用了先介绍再应用的思路，使读者对新技术有了理性认识后，然后从感性上加以强化，大大提高了读者的吸收水平。书中有很多应用 .NET 框架的示例，有助于读者加深理解。

虽然这本书致力于 Visual Basic.NET，它主要的目标是介绍 Visual Basic 和 .NET 框架如何协调工作。但它还介绍 .NET 框架给开发带来的新变化，包括 ASP.NET、Windows Form 和 Web Service。

我们相信，本书不仅对有 Visual Basic 工作经验的开发者有帮助，对于那些有 Java、C++ 和 Delphi 开发经验的人们，也会发现这本书对他们很有参考价值。

书中很多示例代码可以从书中提供的网站下载。值得一提的是在本书的附录部分有一个有趣的 .NET 框架应用实例，读者可以从中看到 VB.NET 的轻便性和使用 Web 服务的简单性。

全书由王海峰、李位星、李新鹏、邵月慧、郑洁、莫伟锋、鹿卫国、赵将、孙同江、于法国、苏琳、李瑞、王海卫、初元芳、于大勇、代桂平、李峰、高岩、韦利国、安柏杉、臧力龙等进行翻译、录排、校对等工作。本书最后由王海峰统稿。由于时间仓促，且译者的水平有限，在翻译过程中难免会出现一些错误，请读者批评指正。

译者

目 录

第 1 章 .NET 框架	1
1.1 构件 (Assemblies)	2
1.1.1 私有构件	3
1.1.2 公共构件	3
1.1.3 元数据	4
1.2 通用运行时语言 (The Common Language Runtime)	4
1.2.1 及时编译 (Just-In-Time Compilation, JIT)	5
1.2.2 版本管理 (Versioning)	6
1.3 命名空间 (Namespaces)	7
1.4 .NET 框架类库	9
1.5 部署 (Deployment)	10
1.6 小结	11
第 2 章 Visual Studio .NET	12
2.1 预览 Visual Studio .NET	12
2.2 Visual Studio .NET IDE	18
2.3 Visual Studio .NET 编译器	20
2.3.1 可折叠的代码块 (Collapsible Code Blocks)	20
2.3.2 自动套用表单和完成	22
2.3.3 代码和注释中嵌入的 URL	22
2.4 其他的编译器	22
2.5 VB 和 .NET 框架	22
2.5.1 VB.NET 的介绍	23
2.5.2 VB.NET 的设计目标	23
2.5.3 访问能力	24
2.5.4 OOP 支持	25
2.5.5 现代语言构造	26
2.5.6 安全与安全性	26
2.5.7 多种语言综合	26

2.5.8 COM 和 Windows API 互操作能力	26
2.6 小结	27
第 3 章 面向对象的编程思想	28
3.1 Visual Basic .NET 抽象	28
3.2 VB.NET 封装	30
3.3 VB.NET 多态性	31
3.4 VB.NET 继承	34
3.4.1 实现继承 (Implementation Inheritance)	34
3.4.2 包容器 (Containment)	37
3.5 基于接口 (Interface-Based) 的编程	37
3.5.1 VB.NET 接口的变化	38
3.6 小结	41
第 4 章 方法、属性和事件	42
4.1 方法	42
4.1.1 参数的改变	42
4.1.2 程序调用	42
4.1.3 返回值	43
4.1.4 静态程序	44
4.1.5 公共方法	44
4.2 函数重载	45
4.3 属性	45
4.3.1 共享属性 (Shared Property)	47
4.4 事件	50
4.4.1 代理 (Delegate)	50
4.4.2 事件句柄	51
4.4.3 创建事件	52
4.4.4 激活事件	53
4.4.5 使用 EventArgs	53
4.5 特性	54

4.6 小结	56
第 5 章 对象的创建和销毁	57
5.1 构造函数 (Constructor)	57
5.2 析构函数 (Destructor)	58
5.3 可访问性限定符 (Accessibility Modifier)	60
5.3.1 类的限定符	60
5.3.2 方法和属性的限定符	61
5.4. 小结	61
第 6 章 创建 VB.NET 工程	62
6.1 不支持的工程类型	62
6.2 类库	62
6.3 Web Form 应用程序	67
6.3.1 创建一个 Web Form 应用程序	68
6.3.2 Web Form 控件	74
6.4 Web 服务工程	80
6.5 Windows 应用程序	84
6.6 Windows 控件 (Windows 控件)	90
6.7 Windows Service (Windows 服务)	95
6.8 Console Application (控制台 应用程序)	97
6.9 小结	98
第 7 章 数据和对象类型	99
7.1 .NET 框架中的类型	99
7.1.1 数值型	99
7.1.2 引用型	101
7.2 字符串处理	102
7.2.1 System. String 类	103
7.2.2 System. Text. StringBuilder 类	105
7.2.3 Boxing 和 Unboxing (封装和 解封装)	106
7.3 VB.NET 中的固有数据类型	107
7.4 Class (类)	109
7.5 Structure (结构)	111
7.6 小结	113

第 8 章 控制逻辑流	114
8.1 Short-Circuited (短路) 表达式	114
8.2 选择	115
8.2.1 IF 语句	115
8.2.2 Select Case	117
8.3 Iteration (循环)	118
8.3.1 For Next	119
8.3.2 For Each	120
8.3.3 Do Loop	124
8.3.4 While Loop	126
8.4 删除的语句	126
8.5 小结	126
第 9 章 错误处理	127
9.1 System. Exception 类	128
9.2 Try...Catch...Finally 语句	128
9.2.1 Try 子句	129
9.2.2 Exit Try 语句	129
9.2.3 Catch 子句	129
9.2.4 Finally 子句	129
9.2.5 Throw 语句	130
9.3 实现结构化的异常句柄	130
9.3.1 引发异常	131
9.3.2 确保清理代码的执行	132
9.3.3 创建定制的异常类型	132
9.4 全局异常句柄	134
9.5 有效使用结构化的异常句柄	135
9.5.1 任意创建定制的异常类型	135
9.5.2 把所有东西封装进异常句柄	136
9.5.3 用异常句柄代替条件检验	138
9.5.4 把所有异常回抛给调用者	139
9.6 使用异常处理的 Windows Form 应用 程序示例	139
9.7 小结	142
第 10 章 ADO.NET	143
10.1 从 DAO 到 ADO.NET	143

10.1.1 远程数据对象.....	144	11.3.2 把 Interop 限制到最小.....	174
10.1.2 OleDb 和 Active Data Objects (动态数据对象)	145	11.3.3 使用数据类型	174
10.1.3 ADO.NET	146	11.3.4 不要过度使用 Fianlize 方法.....	175
10.2 ADO.NET 对象模型.....	147	11.3.5 避免使用 On Error	175
10.2.1 OleDbConnection.....	147	11.3.6 使用.NET 框架类库.....	175
10.2.2 OleDbCommand.....	148	11.4 结构方面的考虑	175
10.2.3 OleDbDataReader	148	11.4.1 Windows DNA、DNA 2000 和.NET.....	175
10.2.4 OleDbDataAdapter	149	11.4.2 .NET 框架的分层结构.....	176
10.2.5 OleDbException.....	149	11.4.3 .NET 企业服务器 (Enterprise Server)	178
10.2.6 批更新 (Batch Updating)	150	11.5 XML 整合	180
10.2.7 DataSet (数据集)	150	11.5.1 SOAP	180
10.2.8 DataTable (数据表)	150	11.5.2 Web 服务.....	181
10.2.9 DataRelation (数据关系)	150	11.5.3 创建 Web 服务.....	182
10.3 Visual Studio.NET 中的 ADO.NET 支持	151	第 12 章 互操作能力.....	184
10.3.1 Strongly Typed DataSets (强 类型的 DataSet)	160	12.1 调用 Windows API	184
10.4 小结.....	164	12.1.1 Declare 语句.....	185
第 11 章 .NET 应用程序开发设计、 结构和实现.....	165	12.1.2 控制调度 (Controlling Marshalling)	187
11.1 安全性 (Security)	165	12.1.3 Declare 格式化结构.....	189
11.1.1 代码标识	166	12.2 在 COM 应用程序中使用.NET 框架.....	190
11.1.2 代码访问安全.....	166	12.2.1 创建健壮的名字	190
11.1.3 说明安全机制.....	167	12.2.2 安装全局构件缓存 (Global Assembly Cache)	190
11.1.4 强制安全机制.....	167	12.2.3 注册	191
11.1.5 基于角色的安全机制.....	167	12.2.4 创建类型库	191
11.1.6 远程安全机制.....	168	12.2.5 检查	191
11.1.7 加密	168	12.2.6 从.NET 框架应用程序中使用 COM 组件	193
11.2 垃圾收集与资源管理	168	12.3 COM+服务.....	196
11.2.1 基于 COM 的资源管理.....	169	12.3.1 COM+提供的服务	197
11.2.2 .NET 框架资源管理.....	172	12.3.2 编写与 COM+服务互操作的 .NET 框架组件.....	198
11.2.3 封装的 CleanUp	173		
11.3 性能的考虑	174		
11.3.1 扩展性与性能.....	174		

12.4 小结	208
第 13 章 从 VB 6 移植	210
13.1 VB.NET 从 VB 6 中删除了什么	210
13.1.1 工程类型	210
13.2 语言结构	211
13.2.1 非文档特性	213
13.3 语言的变化	213
13.4 从 VB 6 移植到 VB.NET	214
13.4.1 移植过程	214
13.4.2 移植的考虑	218
13.5 创建新的 VB 6 应用程序	218
13.5.1 使用标准文档	218
13.5.2 培训	218
13.5.3 进行显式声明	219
13.5.4 计划改变	219
13.5.5 使用 XML	219
13.6 小结	220
第 14 章 .NET 今后的发展	221
14.1 Cross Platform.NET	221

14.2 附加的基本 WebService	221
14.2.1 Hailstorm 综述	221
14.3 .NET 和.NET 企业服务器间的 更好集成	223
14.4 小结	223
附录 A Windows Form 应用程序	224
A.1 体育运动统计数据入口	224
A.1.1 应用程序的数据源	227
A.1.2 加载树	227
A.1.3 菜单	231
A.1.4 创建新记录	232
A.1.5 Visual Inheritance	232
A.2 小结	235
附录 B ASP.NET Web Service 与 Web Form 应用程序	236
B.1 运动统计 Web 服务	237
B.2 Super Stats Live 入口	243
B.3 Bob 的梦幻棒球	246
B.4 小结	249

第 1 章 .NET 框架

在二十世纪九十年代，由于计算硬件价格的急剧下跌以及 Internet 的膨胀，使得整个产业的发展模式迅速变化。作为对这些变化的反应，Microsoft 为 Windows 平台改变了开发模式。这一思路最终导致了 .NET 框架（.NET Framework）的发展。

Microsoft 最初是在 1997 年引进对象运行时环境的，当时引进的是 Microsoft 事务服务器（Microsoft Transaction Server, MTS）。这之后不久，它们又开始开发 Windows NT 5，最终是 Windows 2000；下一版本的 COM，称之为 COM+。

COM+起初包含两个组件：COM+ Service（COM+服务），它允许借助管理工具提供服务，而不是通过直接把代码写入 API 来提供服务；另一组件是 COM+ Runtime（COM+运行时），它把所有 COM 的繁琐代码从开发者转移到操作系统中。

同时，Microsoft 还在积极地开发一些技术，比如：Java 和 XML。Microsoft 意识到他们不是惟一有好想法的开发者，所以 Microsoft 注意其他开发者的工作，选择好的想法并扩展到自己的产品中。

所有的一切最终形成了我们所知的下一代 Web 服务（Next Generation Web Services, NGWS），它最终就是 .NET 框架。

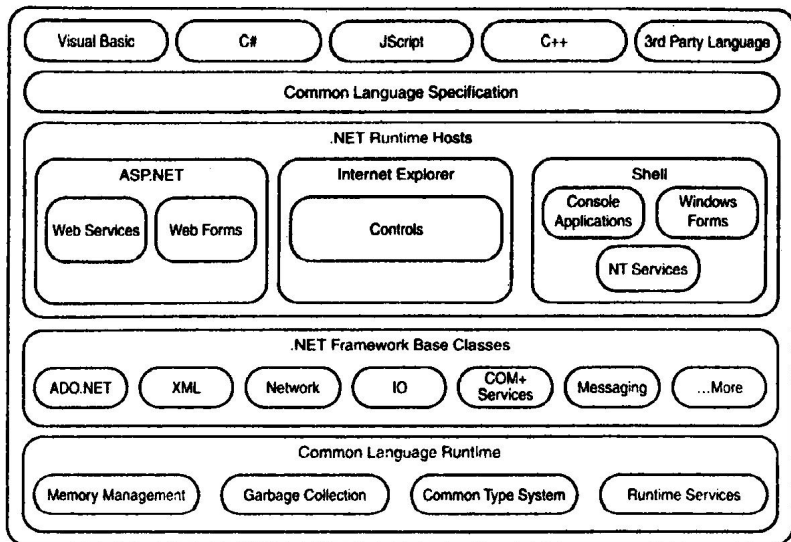


图 1-1 .NET 框架的组件

1.1 构件 (Assemblies)

在 Windows 环境中, 配置应用程序时经常出现问题, 其中最常见的问题是 DLL Hell (DLL 地狱)。

实际上, DLL Hell 有许多版本。第一个并且是问题最严重的版本出现在应用程序覆盖已经存在的旧版本的 DLL 的时候, 会使其他应用程序中断。DLL Hell 的另一个版本出现在新版本的 COM DLL 中断旧版本的应用程序时, 组件的实际运行与以前的版本不同。

首先, 非 COM DLL 可以从系统上的任何地方加载的事实, 迫切需要非 COM 问题。如果一台给定的机器上有多种版本的 DLL, 一个应用程序可能从某个位置加载, 而另一个可能从不同的位置加载。然而, 因为 Windows 总是在加载另一个之前尽量使用已加载的 DLL, 因而应用程序可能以不可预测的顺序中断, 中断的顺序依赖于它们被加载的顺序。

在 COM 中, Microsoft 用两种方法攻克 DLL Hell 问题。其一, 所有的 COM 组件需要注册。这样它们只从一个位置加载。其二, 当 COM 组件增加新功能时, 它们既要实现新接口, 又要支持旧接口。但仍然出现问题, 通常是因为组件的较新版本锁定了旧版本客户依赖的行为。

Windows 2000 中, Microsoft 引进了并行部署的思想来固定 DLL Hell 问题。Windows 2000 保护系统 DLL, 不好的安装程序不能替换旧版本的系统 DLL。 .NET 框架把并行部署和版本管理提升到新的水平。

在 .NET 框架中, 管理代码以构件的形式进行部署。你可以把构件看成是 VB 中的 DLL 或 EXE, 但构件远远大于简单的 DLL 和 EXE, 它们基本上只是二进制文件。

与传统的 DLL 和 EXE 不同, 构件可以分解为多个 DLL。这可用来分离程序中经常使用和很少使用的部分。这个技术对于从网络或 Internet 传来的应用程序很有好处, 因为程序中很少使用的部分可以不下载, 等到用的时候再下载。

在 COM 的世界里, 可以通过类型库访问组件, 类型库描述对象和接口。类型库由一种叫接口定义语言 (Interface Definition Language, IDL) 的语言编写而成, IDL 很像 C/C++。COM 的问题之一是类型库和它所描述的 DLL 和 EXE 相互之间分离。

相反, 构件是自我描述的。构件中的信息告诉组件和接口定义语言 (IDL) 构件能做什么, 以及在什么环境下能被允许作某些事情。

构件中描述一个组件的信息如下:

- 版本信息
- 代码访问安全性信息
- 易于跨语言完整性信息

在 .NET 框架中构件有两种类型：公共构件和私有构件。

1.1.1 私有构件

缺省情况下构件是私有的。 .NET 框架安装后，私有构件只能由这样的应用程序使用，即应用程序的目录或子目录中有该构件的拷贝。这与 COM 明显不同。在 COM 中，组件要在系统范围内注册。私有构件能够减轻 DLL Hell，因为两个应用程序可以使用同一组件的不同版本，而不用担心在使用者机器上安装的是何种版本的组件。

1.1.2 公共构件

一台机器上的所有应用程序都能访问公共构件。当应用程序要加载一个组件时，CLR 先试图找到私有组件，只有在找不到的情况下才寻找公共构件。

公共构件必须安装到全局构件缓存（Global Assembly Cache, GAC）。图 1-2 显示了用 .NET 框架安装的壳扩展，并显示了安装到机器 GAC 中的内容。其他适合 .NET 框架的应用允许你把构件安装到 GAC 中，正如 Microsoft 安装器。

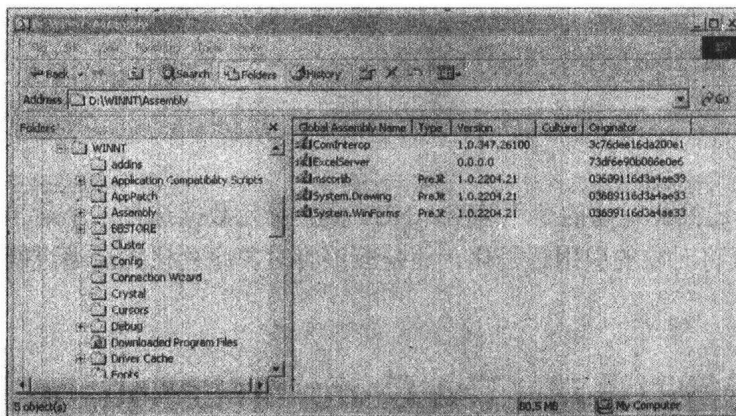


图 1-2 全局构件缓存

多数情况下你会用到私有构件，但是第三方的组件也可能被安装到 GAC 中。GAC 的另外的一个功能是能够持有同一个构件的多种版本，这就是为什么 DLL Hell 问题没有重新出现。

1.1.3 元数据

与早期的开发平台不同，.NET 框架的体系结构设计者们认为所有的代码应该能够自我描述，应该能够为其他开发工具提供信息，从而使开发工具能够决定是否以及怎样与代码相互作用。在.NET 框架中，存储在构件中的元数据（Metadata）提供这一信息。

元数据在如何使得设计时间（Design-time）和运行时环境与构件一起工作中起关键作用。Visual Studio.NET 和开发者的代码通过一个反射（Reflection）过程查看构件的元数据。在.NET 框架中有一套类来简化反射。

元数据在使.NET 框架的安全系统控制在给定环境中如何使用何种类型中，也是关键因素。我们将在 11 章“.NET 应用程序开发设计，结构和执行”详细地讨论这个问题。

构件元数据包含在构件中称为“货单”（Manifest）的区段中。“货单”中包含以下关于该构件的信息：

- 标识
- 文件列表
- 引用构件以及版本信息
- 输出类型
- 字段、方法、属性和所有类型事件的声明
- 输出资源
- 许可请求

最后，元数据还简化对象的远程策略。.NET 框架用元数据选择一个适当的途径来远程化对象：应该仅仅将对象的引用远程化，还是应该将对象序列化然后通过网络，再在另一端将其反序列化。

1.2 通用运行时语言（The Common Language Runtime）

在.NET 框架的背后是一个叫管理环境（Managed Environment）的思想。管理环境作为运行时的管理者，控制着运行什么应用程序代码以及允许做什么。在管理环境中，开发者从编写所有确保完善，灵活和确保应用程序安全的繁琐工作中解放出来。

在.NET 框架中，这个运行时就是通用运行时语言（Common Language Runtime, CLR）。CLR 为.NET 框架提供了全部的运行时服务。虽然运行时对于 Visual Basic 开发者来说没有任何新鲜的东西，但 CLR 对.NET 框架来说完全是新事物。

与 Visual Basic 的运行时不同，CLR 负责所有的管理代码与操作系统的相互工作。在