

万水软件项目应用与实例开发丛书

COMPANION
WEB SITE!

Visual C++ .NET

专业项目实例开发

Microsoft Visual C++ .NET Professional Projects

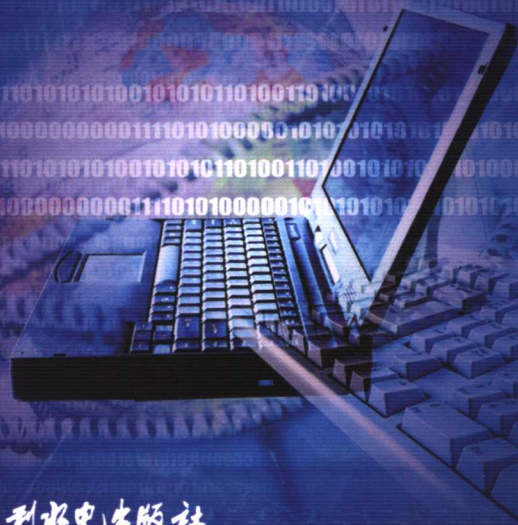
(修订版)

[美] Sai Kishore Sripriya 著

何红波 英宇 等译



中国水利水电出版社
www.waterpub.com.cn



万水软件项目应用与实例开发丛书

Visual C++ .NET 专业项目实例开发 (修订版)

[美] Sai Kishore Sripriya 著

何红波 英 宇 等译

中国水利水电出版社

内 容 提 要

本书凝聚了作者十余年的项目开发与教学经验,是用来打开 Visual C++ .NET 强大功能的金钥匙。本书前几章讲述了 .NET Framework 的基础概念、Visual Studio .NET 编程环境和 Visual C++ .NET 的一些基础知识;后面部分对一些专业项目进行了详细的剖析,这些项目涉及到的内容包括:使用 Visual C++ .NET 进行数据库编程、使用 Managed C++ 创建应用程序、在 Managed C++ 应用程序中实现 ADO.NET、创建托管扩展类库并用于 ASP.NET 应用程序、使用 ATL 创建 COM 组件、创建 ATL Server 应用程序、创建 Web 服务、Managed C++ .NET 程序集和 COM 与 .NET 的互操作性、在 Visual C++ .NET 中使用 XML、Visual C++ .NET 中的联网与远程化等。

本书适合于对 C++ 有一定了解的读者,尤其适合于正在用 Visual C++ .NET 做项目的开发人员。

本书所有实例的源代码可以从中国水利水电出版社网站 <http://www.waterpub.com.cn/softdown/>免费下载。

Authorized translation from English Language Edition published by Premier Press, Inc.
Original copyright © 2002, Premier Press, Microsoft Visual C++ .NET Professional Projects.
Translation by China WaterPower Press, 2002.

北京市版权局著作权合同登记号:图字 01-2002-2334

图书在版编目(CIP)数据

Visual C++ .NET 专业项目实例开发(修订版) / (美)
斯里帕亚(Sripriya, S.K.)著;何红波等译. —2版.
北京:中国水利水电出版社, 2007

(万水软件项目应用与实例开发丛书)

书名原文: Microsoft Visual C++ .NET Professional
Projects

ISBN 978-7-5084-4274-7

I.V... II.①斯...②何... III.C语言—程序设计
IV.TP312

中国版本图书馆CIP数据核字(2006)第144183号

书 名	Visual C++ .NET 专业项目实例开发(修订版)
作 者	[美] Sai Kishore Sripriya 著
译 者	何红波 英 宇 等译
出版 发行	中国水利水电出版社(北京市三里河路6号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266(总机)、68331835(营销中心)、82562819(万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16开本 37印张 913千字
版 次	2007年1月第1版 2007年1月第1次印刷
印 数	0001—4000册
定 价	58.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

修订版序

万水软件项目应用与实例开发丛书是中国水利水电出版社万水公司于 2002 年引进的由全球 IT 培训 15 强之一的 NIIT 资深开发人员编写的一套实用的项目开发类丛书。该套书一经出版就因其创新性的内容安排、实用的案例开发、全面的知识讲解受到了国内广大读者的喜爱和欢迎。

本书是万水软件项目应用与实例开发丛书之一，编写者曾经开发大量的项目，具有丰富的实践经验。该书内容由浅入深，循序渐进。刚开始介绍基本概念，后面部分对一些专业项目进行详细剖析，针对各个项目的开发深入介绍高级知识的应用，使读者能够深入学习并达到实际开发项目的目的。

由于该书第一版出版时间仓促，书中难免存在许多不足，广大读者在该书的使用过程中就该书提出了许多调整建议和改进意见。应广大读者的要求，中国水利水电出版社在 2006 年组织有实力的国内开发人员对该书进行了修改和调整。

修订版对整本书的内容作了调整，从英文版升级为中文版（换图、换文字说明等），更正了第一版中对某些术语的不恰当翻译，规范了代码，在需要的地方加入中文注释，并对第一版中安排不合理的章节进行了调整。

本书既适用于 Visual C++ .NET 初学者，也适用于有一定 Visual C++ .NET 知识但缺乏项目经验的读者。对于前者，可从第一部分开始学习；对于后者，可直接从第二部分开始学习。相信本书中对基础知识深入浅出的阐述以及丰富详实的项目实例会对读者有所帮助。

本书的主要修订工作由周炜、曾凡喜完成，参与程序调试和整理的还有丁昊凯、陈德颢、许靖雯、陈东、童剑、王晓青、黄浩、林晓珊等。如果读者有什么意见和建议可以与我们联系，联系方式为：xinyuanxuan@263.net。

修订者

2006 年 12 月

译者序

Visual Studio .NET 是微软公司最新的应用软件开发平台。与以往版本相比，它发生了很大的变化。对以往的语言做了许多改进，Visual Basic .NET 成为了真正面向对象的语言，Visual C++ .NET 大大增强了对 Web 的集成和对 ATL 的支持，特别是推出了崭新的 C# 语言。开发环境也发生了重大改变，在一个开发环境中集成了多种语言的开发，使得其适应性大大增强。

Visual C++ .NET 是在 Visual C++ 6.0 的基础上开发的，它保留了 Visual C++ 6.0 的原有特点，到目前为止，它是惟一一种能够产生本地代码的 Visual Studio .NET 语言。Visual C++ .NET 还大大增强了 Web 程序开发的能力。相信读者通过本书的学习与实践，一定会深有感触。

本书的前几章讲述了 .NET Framework 的基础概念、Visual Studio .NET 编程环境和 Visual C++ .NET 的一些基础知识。本书的后面部分对一些专业项目进行了详细的剖析。这些设计来自于作者长期的实践与教学，由浅入深，由易到难，涵盖了许多开发课题，既有简单的项目，也有复杂的设计。这些项目涉及到：使用 Visual C++ .NET 进行数据库编程、使用 Managed C++ 创建应用程序、在 Managed C++ 应用程序中实现 ADO.NET、创建托管扩展类库并用于 ASP.NET 应用程序、使用 ATL 创建 COM 组件、创建 ATL Server 应用程序、创建 Web 服务、Managed C++ .NET 程序集和 COM 与 .NET 的互操作性、在 Visual C++ .NET 中使用 XML、Visual C++ .NET 中的联网与远程化等。

本书从实践的角度来介绍 Visual C++ .NET 的应用与开发，适合于已经具备 Visual C++ 早期版本编程知识的读者。如果您是 Visual C++ 的初学者，也可以通过本书中简单项目的学习开始入门并逐渐体会 C++ 的强大功能。如果您是 Visual C++ 的长期实践者，作者捧出自己十多年的编程实践与教学经验与您分享，您一定会获益匪浅。

本书由何红波组织翻译，参与翻译的人有英宇、魏睦棋、梁海华、林棋、侯战友、谢君英、李明、朱云华、徐继伟。由于译者水平有限，加之时间仓促，译误之处在所难免，恳请读者谅解！我们接收读者意见的邮箱为：yingyu@263.net。

译者

2002 年 12 月

前 言

本书目标

本书面向那些已经具备 Visual C++ 早期版本编程知识的读者，从实践的角度来介绍 Visual C++ .NET，特别适合那些能够运用 Visual C++ 任何一种早期版本进行应用程序开发的有经验的开发人员参考。

本书的前面几章概述了 Microsoft 基础类库（MFC）程序设计的主要概念，这些概念保留在 Visual C++ 的新版本中，未做任何改动。这几章运用充足的应用实例基本上重新回顾了 MFC 程序设计的一些概念。

本书的主要部分围绕一些专业性项目进行讲述。这些设计都是建立在实际情况基础上的，由浅入深、由易到难，涵盖了许多具体的课题，指导读者如何将所学的知识付诸实际运用。书中既有简单的项目，如创建一个基于 MFC 的以数据库为中心的应用程序，以及运用 Managed C++ 创建 GDI+ 应用程序，也有复杂的设计，如创建 COM 组件、ATL Server、类库和 Web 服务，以及说明实现与 .NET 组件、XML、网络和远程 COM 互操作性的复杂项目。这些项目通过让读者了解 Visual C++ .NET 在实际情况中的应用来帮助他们达到学习的目的。

除了概述的几章和专业项目外，本书还包含附录，附录中另外提供了一些概念，以便更好地帮助读者了解有关 .NET Framework 和 C#（.NET 语言系列的新成员）的知识。

如何使用本书

本书条理清晰，可以让读者容易地掌握其中的内容。下面是书中使用的一些约定：

分析：本书对所有的代码进行了详尽的分析，对这些代码做什么以及为什么要用到它们逐行作出了解释。

提示：给读者提供一些特殊的建议，或是给出了在使用某种特定产品时与众不同的快捷方法。

注意：给出了附加信息，这些附加信息可能是读者感兴趣的，但不是完成手边项目所必要的知识。

警告：用于警告用户在不正确地执行某项任务时，可能导致灾难性的结果。

新术语定义：所有的新术语都用斜体表示，并在文本中作出定义。

目 录

修订版序
译者序
前言

第一部分 Visual C++ .NET 介绍

第 1 章 .NET Framework 基础 1	3.1.3 执行应用程序..... 33
1.1 .NET Framework..... 1	3.2 定制一个应用程序框架..... 33
1.1.1 公共语言运行环境..... 2	3.2.1 为窗体控件创建变量..... 36
1.1.2 类库..... 4	3.2.2 在控件与变量之间传递数据..... 37
1.1.3 程序集..... 5	3.3 处理按钮事件..... 39
1.1.4 调试器..... 5	3.4 小结..... 40
1.1.5 代码访问安全性..... 6	第 4 章 使用菜单、对话框和窗口控件 41
1.2 Visual Studio .NET 介绍..... 6	4.1 创建菜单..... 41
1.2.1 Visual Studio .NET IDE..... 6	4.1.1 创建菜单资源..... 41
1.2.2 Visual C# .NET..... 7	4.1.2 为显示菜单编写代码..... 41
1.2.3 Visual Basic .NET..... 7	4.1.3 处理菜单..... 44
1.2.4 Visual C++ .NET..... 8	4.2 使用对话框和控件..... 45
1.2.5 Windows 窗体..... 8	4.2.1 创建模式对话框..... 47
1.2.6 Web 表单..... 8	4.2.2 创建无模式对话框..... 54
1.2.7 Web 服务..... 8	4.3 小结..... 57
1.3 Visual C++ .NET 的新特征和增强功能..... 9	第 5 章 文档/视图结构 58
1.4 小结..... 10	5.1 走近文档/视图结构..... 58
第 2 章 Visual C++ .NET 编程基础 11	5.1.1 文档/视图结构的组成..... 58
2.1 MFC 编程概述..... 11	5.1.2 文档/视图类的功能..... 59
2.2 MFC 编程基础..... 11	5.2 创建一个多文档应用程序..... 65
2.2.1 事件处理..... 12	5.2.1 一个多文档应用程序的例子..... 66
2.2.2 设备无关性..... 21	5.2.2 改进应用程序..... 67
2.3 小结..... 27	5.2.3 基于文档/视图结构的应用程序 的控制流程..... 76
第 3 章 创建用户界面 28	5.3 小结..... 77
3.1 MFC 应用程序向导介绍..... 28	第 6 章 线程 78
3.1.1 启动 MFC 应用程序向导..... 28	6.1 线程基础..... 78
3.1.2 创建一个应用程序框架..... 29	

6.1.1 单线程应用程序.....	78	6.3 线程同步.....	86
6.1.2 多线程应用程序.....	78	6.3.1 CSyncObject.....	87
6.2 进一步讨论线程.....	80	6.3.2 CMutex.....	87
6.2.1 用于工作者线程的 AfxBeginThread 函数.....	80	6.3.3 CCriticalSection.....	87
6.2.2 用于用户界面线程的 AfxBeginThread 函数.....	81	6.3.4 CSemaphore.....	87
6.2.3 创建一个单线程应用程序.....	82	6.3.5 CEvent.....	88
6.2.4 创建一个多线程应用程序.....	84	6.3.6 CMultiLock.....	88
6.2.5 终止线程.....	85	6.3.7 CSingleLock.....	89
		6.4 小结.....	89

第二部分 专业项目 1

项目 1 使用 Visual C++ .NET 进行数据库编程...90	8.3.6 用户验收测试.....	104
项目 1 概述.....	8.4 银行业务应用程序.....	104
第 7 章 使用 ODBC 管理数据库概述.....91	8.4.1 创建 Banking 数据库.....	104
7.1 数据库访问机制.....91	8.4.2 配置一个 ODBC 数据源.....	107
7.1.1 ODBC.....91	8.4.3 创建 SaveMyMoney 应用程序.....	108
7.1.2 DAO.....91	8.4.4 编写应用程序代码.....	117
7.1.3 选择一种数据访问技术.....91	8.5 小结.....	156
7.2 MFC ODBC 类.....92	第 9 章 Visual C++ .NET 中的程序调试和 异常处理.....	157
7.2.1 CDatabase 类.....92	9.1 MFC 应用程序的异常处理.....	157
7.2.2 CRecordset 类.....93	9.2 调试 Visual C++ .NET 应用程序.....	159
7.2.3 CRecordView 类.....98	9.2.1 调试基础.....	159
7.3 小结.....98	9.2.2 生成配置.....	161
第 8 章 实现数据访问的技术.....99	9.2.3 任务列表.....	163
8.1 SaveMyMoney 银行案例分析.....99	9.2.4 断点窗口.....	166
8.2 SaveMyMoney 银行解决方案.....99	9.2.5 自动窗口.....	172
8.3 项目生命周期.....99	9.2.6 局部变量窗口.....	172
8.3.1 需求分析.....100	9.2.7 监视窗口.....	173
8.3.2 高层设计.....101	9.2.8 快速查看对话框.....	173
8.3.3 底层设计.....102	9.2.9 其他调试窗口.....	174
8.3.4 构造.....104	9.3 小结.....	175
8.3.5 集成和测试.....104		

第三部分 专业项目 2

项目 2 使用 Managed C++应用程序创建 应用程序.....	176	项目 2 概述.....	176
		第 10 章 托管扩展的介绍.....	177

10.1 .NET 中的 C++ 语言.....	177	11.4 事件和委托.....	196
10.2 使用 C++ 托管扩展编程.....	178	11.4.1 事件.....	196
10.2.1 类.....	179	11.4.2 委托.....	197
10.2.2 接口.....	181	11.4.3 托管代码中的事件处理.....	199
10.2.3 属性.....	181	11.4.4 统一事件模型.....	201
10.2.4 异常.....	181	11.5 小结.....	201
10.2.5 装箱.....	182	第 12 章 创建一个托管扩展应用程序.....	202
10.3 创建一个 GUI 应用程序.....	182	12.1 项目案例分析.....	202
10.4 小结.....	186	12.1.1 需求分析阶段.....	202
第 11 章 GDI+、事件和委托的介绍.....	187	12.1.2 设计阶段.....	203
11.1 GDI+ 介绍.....	187	12.1.3 编码和构造阶段.....	205
11.1.1 GDI+ 提供的功能.....	187	12.1.4 测试阶段.....	206
11.1.2 GDI+ 的新功能.....	188	12.1.5 执行阶段.....	206
11.2 编程模式的变化.....	189	12.2 编写应用程序代码.....	206
11.2.1 在 GDI+ 中重载方法.....	189	12.2.1 创建应用程序.....	206
11.2.2 用于画图 and 填充的不同方法.....	190	12.2.2 创建用户界面窗体.....	206
11.2.3 创建区域.....	190	12.2.3 创建应用程序菜单.....	209
11.3 GDI+ 中重要的类和名字空间.....	190	12.2.4 创建应用程序工具栏和状态栏.....	213
11.3.1 System.Drawing 名字空间.....	190	12.2.5 应用程序的事件处理.....	216
11.3.2 Graphics 类.....	192	12.2.6 委托.....	219
11.3.3 Brush 类.....	194	12.3 Easy Tool 应用程序的完整代码列表.....	223
11.3.4 Pen 类.....	195	12.4 小结.....	268

第四部分 专业项目 3

项目 3 Managed C++ 中的 ADO.NET.....	269	14.1 项目概述.....	284
项目 3 概述.....	269	14.2 数据库规划.....	284
第 13 章 ADO.NET 简介.....	270	14.2.1 创建数据库.....	284
13.1 数据访问技术概述.....	270	14.2.2 创建 Account_Detail 表.....	285
13.2 ADO.NET 概述.....	272	14.2.3 创建 BankLogin 表.....	285
13.2.1 ADO.NET 的功能.....	272	14.3 创建银行业务应用程序.....	285
13.2.2 ADO.NET 的优点.....	273	14.3.1 在托管扩展中创建项目.....	286
13.3 ADO.NET 的体系结构.....	273	14.3.2 创建登录窗体.....	286
13.4 ADO.NET 和 XML.....	283	14.3.3 创建 Account Details 窗体.....	291
13.5 小结.....	283	14.4 小结.....	305
第 14 章 在 Managed C++ 应用程序中使用			
ADO.NET.....	284		

第五部分 专业项目 4

项目 4 创建托管扩展类库并用于 ASP.NET 应用程序.....	306	15.3.4 向表单添加服务器控件	316
项目 4 概述	306	15.3.5 设置 Web 控件的属性	317
第 15 章 ASP.NET 简介	307	15.3.6 处理 Web 控件的事件	318
15.1 ASP.NET 的演变	307	15.4 验证控件	320
15.2 创建 ASP 应用程序的需求	308	15.4.1 使用验证控件	320
15.2.1 ASP.NET 的体系结构	309	15.4.2 使用多个验证控件	325
15.2.2 Web 表单基础	309	15.5 小结	326
15.2.3 Web 表单的组成	310	第 16 章 创建类库并用于 ASP.NET 应用程序	327
15.3 创建 Web 表单	310	16.1 税款计算器应用程序	327
15.3.1 IIS 应用程序根目录	314	16.1.1 项目生命周期	327
15.3.2 ASP.NET Web 表单服务器控件	316	16.1.2 创建 Managed C++ 类库	328
15.3.3 HTML 服务器控件与 Web 服务器控件的比较	316	16.1.3 创建 ASP.NET 应用程序	330
		16.2 小结	334

第六部分 专业项目 5

项目 5 使用 ATL 创建 COM 组件	335	17.4.6 队列组件	356
项目 5 概述	335	17.4.7 COM+ 事件	356
第 17 章 COM 简介	336	17.4.8 自动化事务	357
17.1 COM 的演变	338	17.5 小结	357
17.1.1 什么是 COM	339	第 18 章 基于属性的编程简介	358
17.1.2 COM 要解决的问题	339	18.1 属性和反射的介绍	358
17.2 COM 组件	340	18.1.1 属性类别	358
17.2.1 COM 组件的类型	340	18.1.2 自定义属性	359
17.2.2 COM 库	342	18.1.3 反射	361
17.2.3 接口	342	18.2 小结	363
17.2.4 注册表在 COM 中的作用	351	第 19 章 使用 ATL 实现 COM	364
17.2.5 COM 对象的实例化	352	19.1 Art-Shop——在线美术馆	364
17.3 COM 中的线程	353	19.2 项目生命周期	364
17.4 COM+	355	19.2.1 设计站点	364
17.4.1 内存数据库	355	19.2.2 Checkout 页面	365
17.4.2 负载平衡	355	19.3 创建信用卡验证组件	366
17.4.3 基于角色的安全	356	19.3.1 理解 Luhn 算法	366
17.4.4 新线程模型	356	19.3.2 使用 ATL 项目向导创建基础 COM 组件	366
17.4.5 对象缓冲池	356		

19.3.3 创建 CoClass 类并添加功能.....	368
19.3.4 测试组件.....	375

19.4 小结.....	378
--------------	-----

第七部分 专业项目 6

项目 6 创建 ATL Server 应用程序.....	379
项目 6 概述.....	379
第 20 章 ATL Server 简介.....	380
20.1 什么是 ATL Server.....	381
20.1.1 请求处理程序.....	382
20.1.2 其他功能.....	384
20.2 小结.....	384
第 21 章 创建一个客户簿应用程序.....	385
21.1 案例分析.....	385

21.2 创建基于 HTML 表单的界面.....	386
21.3 创建 ATL Server 应用程序.....	387
21.3.1 理解客户簿项目.....	391
21.3.2 修改 SRF.....	392
21.3.3 表单处理.....	392
21.3.4 添加方法到默认处理程序.....	392
21.4 创建空 XML 文件.....	395
21.5 小结.....	395

第八部分 专业项目 7

项目 7 创建 Web 服务.....	396
项目 7 概述.....	396
第 22 章 Web 服务和相关技术介绍.....	397
22.1 什么是 Web 服务.....	397
22.2 Web 服务结构.....	398
22.3 相关技术.....	398
22.3.1 Web 服务和 XML.....	398
22.3.2 Web 服务和 SOAP.....	406
22.3.3 Web 服务描述语言.....	408
22.3.4 通用说明、发现和集成 (UDDI) ...	408
22.4 .NET 和 Web 服务安全.....	409
22.4.1 在 Internet 上确保 Web 服务的 安全.....	409
22.4.2 在 Intranet 上确保 Web 服务的 安全.....	410
22.5 小结.....	411
第 23 章 使用托管扩展创建 Web 服务.....	412
23.1 Art-Shop 的合作方案.....	412
23.1.1 项目生命周期.....	412
23.1.2 ArtShop 数据库中的 Product 表 ...	413

23.1.3 创建基础 Managed C++ Web 服务.....	413
23.1.4 修改 Web 服务.....	416
23.1.5 用 SQL Server .NET Data Provider 访问数据.....	419
23.1.6 测试 Web 服务.....	422
23.1.7 用 Managed C++ 建立 Web 服务 使用程序.....	422
23.2 小结.....	427
第 24 章 用 ATL Server 创建 Web 服务.....	428
24.1 汇率 Web 服务.....	428
24.2 项目生命周期.....	428
24.2.1 需求分析.....	429
24.2.2 设计.....	429
24.2.3 构造.....	429
24.2.4 测试.....	429
24.3 创建 Web 服务.....	429
24.3.1 创建数据库、表和存储过程.....	429
24.3.2 创建 ATL Server Web 服务.....	430
24.4 测试应用程序.....	440
24.5 小结.....	442

第九部分 专业项目 8

项目 8 Managed C++ .NET 程序集和 COM 与 .NET 的互操作性.....	443	25.6 小结.....	457
项目 8 概述.....	443	第 26 章 实现 COM 和 .NET 的互操作性.....	458
第 25 章 .NET 中的程序集、清单以及与 COM 的互操作性概述.....	444	26.1 Art-Shop——创建一个登录组件.....	458
25.1 程序集.....	444	26.2 项目生命周期.....	458
25.1.1 程序集的类型.....	444	26.3 在 ArtShop 数据库中创建 Customer 表.....	459
25.1.2 全局程序集高速缓存.....	446	26.4 创建登录程序集.....	460
25.1.3 程序集的优点.....	446	26.4.1 用 SQL Server.NET Data Provider 访问数据.....	462
25.1.4 程序集版本控制.....	447	26.4.2 实现方法.....	462
25.1.5 程序集的内容.....	448	26.5 测试组件.....	466
25.1.6 创建程序集.....	448	26.6 小结.....	467
25.1.7 程序集名称.....	449	第 27 章 在 .NET 程序集中使用 COM+ 服务.....	468
25.1.8 公钥和私钥.....	449	27.1 Art-Shop Kiosks——下订单.....	468
25.1.9 程序集位置.....	451	27.2 .NET 和 COM+ 服务.....	469
25.1.10 将专用程序集转换成共享程序集.....	451	27.2.1 使用事务.....	470
25.2 创建 .NET 组件.....	452	27.2.2 创建数据库表.....	471
25.3 COM 互操作性.....	452	27.2.3 创建订单处理程序集.....	472
25.4 COM 可调用的包装器.....	453	27.2.4 用 COM+ 服务注册 ProcessOrder 组件.....	474
25.5 运行环境可调用的包装器.....	454	27.2.5 测试组件.....	475
25.5.1 从 .NET 中调用 COM 对象.....	455	27.3 小结.....	476
25.5.2 从 COM 中调用 .NET 对象.....	456		

第十部分 专业项目 9

项目 9 在 Visual C++ .NET 中使用 XML.....	477	28.3.1 XmlTextReader 类.....	481
项目 9 概述.....	477	28.3.2 XmlValidatingReader 类.....	481
第 28 章 在 Visual C++ .NET 中使用 XML.....	478	28.3.3 XmlReader 类.....	481
28.1 New-Tech 公司——方案.....	478	28.3.4 XmlWriter 类.....	482
28.2 项目生命周期.....	478	28.3.5 XmlDocument 类.....	483
28.2.1 需求分析和设计.....	479	28.3.6 XPathNavigator 类.....	484
28.2.2 应用程序的设计.....	479	28.3.7 XPathDocument 类.....	485
28.2.3 构造.....	480	28.4 创建和测试 Scheduler 应用程序.....	485
28.3 .NET 中的 XML——.NET XML 类.....	480	28.5 小结.....	491

第十一部分 专业项目 10

项目 10 Visual C++ .NET 中的联网与远程化..	492	30.1.1 进程间通信	512
项目 10 概述	492	30.1.2 .NET 远程化框架概况.....	513
第 29 章 实现网络互联	493	30.2 远程对象——服务器端	513
29.1 网络编程介绍	493	30.2.1 主机化远程对象	514
29.1.1 套接字.....	494	30.2.2 配置远程对象	514
29.1.2 IP 地址和端口	494	30.2.3 传递远程对象	515
29.2 使用 Visual C++ .NET 开发一个网络 应用程序	495	30.3 代理对象——客户端	516
29.2.1 System.Net 名字空间	495	30.4 信道服务	517
29.2.2 Socket 类.....	495	30.4.1 信道的工作原理	517
29.2.3 WebRequest 类.....	497	30.4.2 安全通信	518
29.2.4 IPAddress 类	498	30.5 远程化元数据和配置文件	519
29.3 System.Threading 名字空间	499	30.6 对象生存周期.....	519
29.4 案例分析	500	30.7 小结.....	520
29.5 项目生命周期	500	第 31 章 实现远程化.....	521
29.5.1 需求分析.....	500	31.1 NJ Finances——方案	521
29.5.2 设计	500	31.2 .NET 远程化	522
29.5.3 构造	501	31.3 创建远程对象应用程序	522
29.6 小结	511	31.3.1 创建远程服务器应用程序	528
第 30 章 远程化简介	512	31.3.2 创建客户机应用程序	530
30.1 .NET 远程化结构简介	512	31.3.3 测试.....	538
		31.4 小结.....	538

第十二部分 附录

附录 A .NET 的蓝图和目标	539	附录 D 将非托管代码迁移到托管代码	565
附录 B .NET Server 系列	551	附录 E C#基础	571
附录 C 名字空间与属性	561		

第一部分 Visual C++ .NET 介绍

第 1 章 .NET Framework 基础

.NET 现在是 IT 界的时髦词。什么是 .NET，是什么促使了它的出现？与现有的技术相比它有什么不同和新的内容？最重要的是，能为你我以及世界带来什么？这些就是本章要讨论的问题。

毋庸置疑，IT 业始终是不发展的。而一些革命性的变化已经成为这个行业的生存法则。当独立于操作系统的 PC 机进入这个行业时，就像一股清新的空气，接着是网络，想想这些就知道了。最初，我们使用 Intranet（内联网）工作，后来逐渐转用 Internet。迄今为止，Web 无疑是一场最大的革命，因为整个世界都响着 Internet 的旋律。然而每个变化都会伴随着挑战。可以确定地说，Internet 的出现和重大成功导致了计算平台从桌面计算向分布式计算的转移。因为 Web 是异构的，对于程序员来说最大的挑战是创建具有平台无关性和互操作性特征的应用程序。

微软当然也认识到了这种改变并发布了 .Net initiative。 .Net initiative 的目标是使应用程序能通过互联网而相互作用，跨越可能存在的开发平台上的、地理位置上的以及应用程序使用的语言上的不同而形成的障碍。例如，为 Windows 平台开发的应用程序现在可以同一个 Linux 平台下的应用程序交互。类似地，用 Visual C++ 开发的应用程序现在也可以和一个用 Visual Basic 开发的应用程序交互。 .NET 就是为实现上述功能而产生的。

在把焦点转移到 Visual C++ .NET 之前，先来看看 .NET initiative 的一些主要特征： .NET Framework 及其组件，还有 Visual Studio .NET，这是一个支持多种开发语言的开发环境。然后，将介绍 Visual C++ .NET 的新特征和改进之处，包括对 Visual C++ .NET 和 Visual C++ 6.0 的比较，以突出显示这两者之间的差异。

1.1 .NET Framework

.NET initiative 的一个主要目标是提供一个公共的框架，能够帮助用户轻松地开发基于 Web 的应用程序和其他不同于传统应用程序（比如基于 Windows 的应用程序）的服务。为了实现这个目标， .NET Framework 便应运而生了。 .NET Framework 是多种语言的公共框架，例如， Visual Basic .NET、 Visual C++ .NET 以及 Visual C# .NET（C# 读作 C sharp），这些提供了更高层次上的互操作性。除此以外它还提供了一套新的高级功能特征，例如，垃圾收集器（Garbage Collector），可以让用户创建高度安全的应用程序。 .NET Framework 主要通过下面两个组件来提供前述的所有功能特征：

- 公共语言运行环境（Common Language Runtime, CLR）
- 类库（Class Library）

下面将详细介绍这些组件。

1.1.1 公共语言运行环境

和以前的版本不同的是，.NET Framework 为所有 .NET 语言提供一个公共的运行环境，而不是每一种语言都有自己的运行环境。这个公共的运行环境称作 Common Language Runtime (CLR)。除了在运行时管理代码的执行之外，CLR 也为所有适应 CLR 的语言提供一套公共的服务，如异常处理、安全性和调试等。因而，CLR 有助于实现跨语言的互操作性。CLR 管理所有 .NET 语言编译器产生的代码。所以任何在 CLR 下运行的代码被称作托管代码 (managed code)。当编译一个 .NET 应用程序时，相关的编译器将源代码编译成托管代码。编译器用 MSIL (Microsoft Intermediate Language, 中间语言) 表示生成的托管代码。中间语言 MSIL 是与 CPU 无关的。



注意

MSIL 包括用来管理对象的方法（包括方法的加载、初始化和调用）的指令以及用于内存管理和异常处理的指令等。

除了产生中间语言代码外，编译器还产生描述代码的元数据。元数据保存关于类型、成员和引用等的细节信息。CLR 用元数据来控制代码的运行和异常处理，以及提供安全性和跨语言的集成。中间语言代码和元数据都存储在可移植的可执行文件 (Portable Executable, PE) 里。编译之后，在应用程序执行阶段，那些组成托管代码的 MSIL 指令需要先被转换成 CPU 相关的代码。CLR 支持的即时编译器 (JIT Compiler) 把 MSIL 转换成本机代码，即基于特定 CPU 的代码。图 1-1 描述了即时编译器的作用。

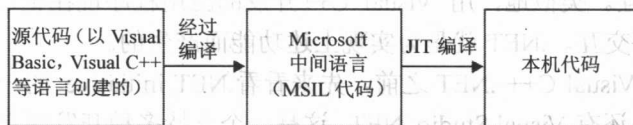


图 1-1 JIT 编译器的作用



注意

MSIL 代码跟 Java 支持的字节代码类似。执行应用程序期间，完全解释字节代码时，由 CLR 提供的 JIT 编译器并不把所有 MSIL 代码编译成本机代码。相反，当某部分代码需要执行时，JIT 编译器就编译这部分特殊的代码。这种实现（指的是随选编译）后面的逻辑就是，在任何时候都不需要执行全部代码。考虑一个具有两个函数的应用程序，其中一个函数用于添加数据，另一个函数用于修改数据。任一时刻，用户只会使用其中的一个函数，因为添加数据和修改数据不能同时发生。在这种情况下，为两个函数都生成本机代码有什么作用呢？这只会导致内存使用的效率低下。因此，JIT 编译器只进行部分编译。一经编译，JIT 编译器就把本机代码存储在内存中以备将来使用。通过这种方式，JIT 编译器保证了应用程序高效地利用系统资源。

如前所述，CLR 最显而易见的优点是支持跨语言的互操作性。简单地说，语言互操作性就是让一种语言写的代码能够访问另一种语言写的代码的功能特性。CLR 中有个支持这个功能特征的组件，称为公共类型系统 (Common Type System, CTS)。

下面简要介绍 CTS 的概况。

1. 公共类型系统 (CTS)

CTS 定义了一套标准的数据类型和一套创建新数据类型的规则。CTS 保证标准数据类型

的实现在所有 CLR 语言中都是相同的, 例如整型。CTS 支持的数据类型大致可以分成两大类:

- 值类型: 这种类型直接存储数据。它下面的子类包括内置的值类型、用户定义的值类型以及枚举类型。
- 引用类型: 与值类型不同, 它存储值的内存地址的引用而不是值本身。它下面的子类有自描述类型 (类和数组)、指针类型以及接口。

虽然在 .NET Framework 里 CTS 使语言的互操作变得容易, 但仅靠它还不足以保证用不同语言开发的应用程序之间的顺利交互 (也就是应用程序互操作性)。为了实现这一点, CLR 提供了另一个叫作公共语言规范 (Common Language Specification, CLS) 的组件。

2. 公共语言规范 (CLS)

应用程序间通过对象来相互作用。为了能顺利地相互作用, 需要保证这些对象只暴露能够被开发语言共同理解的那些功能特征。CLS 实现了这一点。它定义了一套基本语言特征和公共类型系统数据类型, 在不同的 .NET 语言的语法中都必须实现它们。符合这些 CLS 规则的语言被认为是适应 CLS 的。CLS 本身支持几种固有的变量类型。对于固有的那些非 CLS 数据类型 (因此是不适应 CLS 的), CLS 提供了可代替的类型。例如, UInt32 是不适应 CLS 的, 它的替代类型是 Int32。总而言之, 为了实现两个应用程序间的顺利交互, 相关的组件或者对象都应该是适应 CLS 的。

提示

所有的无符号数据类型都是不适应 CLS 的。

下面是一些 CLS 的指导原则:

- 在适应 CLS 的语言里不允许有全局静态变量和方法。
- 无符号数据类型在适应 CLS 的语言里是不允许的。
- 适应 CLS 的语言里使用的名称应该是惟一的。即使使用的语言是区分大小写的, 也必须给不同的变量不同的名称, 并且要注意大小写, 这样其他不区分大小写的语言也能区别这些名称。
- 包含在一个方法的签名中的类型也应该是适应 CLS 的。
- 方法的重载只能基于它们的参数的个数和类型。
- 指针在适应 CLS 的语言里是不允许的。

可以查阅软件文档以了解 CLS 的其他功能特征。

到目前为止, 已经了解了支持语言 and 应用程序互操作性特征的 CLR 组件。下一节将讨论 CLR 的另一个重要特征, 自动内存管理以及支持这个特征的组件, 垃圾收集器 (Garbage Collector, GC)。

3. 垃圾收集器 (GC)

如果用过 C++ 或者 Visual C++, 会很清楚“内存泄漏”。你不会忘记“memory fault core dumped”的消息。在执行使用了指针的应用程序时一定收到过这个信息。动态内存分配提高了应用程序使用内存的效率, 但也带来了严重的缺点, 即内存泄漏。

为了解决内存泄漏问题, .NET Framework 的 CLR 提供了垃圾收集器。它能够进行自动内存管理, 这样就从根本上解决了内存泄漏问题。然而, 这个特征只能用于托管代码。在解释垃

圾收集器的任务之前, 需要理解对于一个托管应用程序, 内存是怎样分配的。当运行应用程序时, 就给它分配一个地址空间。这个地址空间叫做托管堆 (managed heap)。当在应用程序中建立新的对象时, 运行时环境在这个地址空间里给对象分配内存空间。当这个地址空间被填满后就不能再建立对象, 除非创建新的空间。这时垃圾收集器就发挥作用了。

垃圾收集器扫描不再使用的对象的内存, 删除该对象并释放其所占的内存区。为了识别那些不再使用的对象, 垃圾收集器跟踪每个应用程序维护的根 (root)。每个根要么指向地址空间中的一个对象, 要么设置成 null。JIT 编译器维护着一个应用程序的活动根列表。垃圾收集器利用这个列表跟踪这些对象。那些不能通过根获得的对象被标识为“不再使用的”并被从内存中删除。

注意

只有当地址空间满时垃圾收集器才开始起作用。一旦清理完毕, 垃圾收集器就处于闲置状态了。

CLR、CLR 特征和它的组成组件是 .NET Framework 最重要的组成部分之一。 .NET Framework 另一个组成部分是类库, 接下来就将介绍类库。

1.1.2 类库

类库 (Class Library) 是 .NET Framework 的一部分。使用 .NET Framework 提供的这些面向对象的类, 易于创建基于 Windows 和基于 Web 的应用程序。这个类库称为基础类库。除了基础类, 类库也包括一些接口和值类型, 这些值类型提供多种对系统功能的访问。类库的显著特征是同一个库能供不同的语言进行程序开发, 例如 Visual C++ 和 Visual Basic, 这在以前是不可能的。但现在 .NET Framework 类库实现了这种可能, 它提供了一套能跨语言使用的公共函数。

.NET Framework 的类库层次化地组织在名字空间中。一个名字空间包含一些具有某种共同特性的类。根据应用程序所需要的类, 在程序中引入相应的名字空间。使用名字空间的一个优点在于能使用名称相同但属于不同名字空间的类。

.NET Framework 用一个点把名字空间的名称和名字空间中类的名称分隔开, 从而清楚地描述了层次结构。比如 System.Console, Console 是一个类 (类型), 而 System 是一个名字空间。在 .NET Framework 类库中, System 名字空间在层次结构的最上面, 包含类 (例如 Console 类和 Array 类)、接口和结构体。表 1-1 描述 .NET Framework 提供的一些名字空间, 表 1-2 描述 System 名字空间中的一些重要的类。

表 1-1 类库中的名字空间

名字空间	用途
Microsoft.Csharp	用 Visual C#.NET 编译并生成代码
Microsoft.Jscript	用 JScript 编译并生成代码
Microsoft.VisualBasic	用 Visual Basic .NET 编译并生成代码
Microsoft.Win32	操纵系统注册表, 管理操作系统事件
System	管理异常和定义数据类型、事件、事件处理程序、接口和属性信息