

开发专家

之 **MS.NET**

Visual C++ ..NET

编程指南

飞思科技产品研发中心

编著

致力于解决开发人员在
研发过程中碰到的实际问题

随书光盘
包括了书
中所有范
例的源代
码



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



81

TP312C

F389

开发专家
之MS.NET

Visual C++ .NET

编程指南

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

飞思科技产品研发中心 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING



内 容 简 介

Visual C++.NET 对于为 Windows 和万维网(WWW)创建高性能应用程序来说是一种效率最高的 C++ 工具。本书致力于解决开发人员在研发过程中碰到的实际问题,与同类书籍相比具有突出新特性、突出实用性的特点。本书首先介绍了 Visual C++.NET 的新特性,然后依次介绍 Visual C++.NET 编程的 4 个方面: MFC 编程、ATL 编程、ATL Server 编程和 C++托管扩展编程。在每个部分中都运用具体、实际的例子有侧重地介绍了最常用和最有特点的内容。本书适合 C++/Visual C++/Visual C++.NET 的中高级读者,也可供对 .NET 感兴趣的读者参考。随书光盘包括了书中所有范例的源代码。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Visual C++.NET 编程指南 / 飞思科技产品研发中心编著. —北京: 电子工业出版社, 2003.1
(开发专家之 MS.NET)

ISBN 7-5053-8179-2

I.V... II.飞... III.C 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(2002)第 086172 号

责任编辑: 王树伟 杨 鹂

印刷者: 北京民族印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京海淀区万寿路 173 信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 34 字数: 870.4 千字 附光盘 1 张

版 次: 2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷

印 数: 5 000 册 定价: 49.00 元(含光盘)

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。若书店售缺,请与本社发行部联系调换。电话 68279077

出版说明

“开发专家”是电子工业出版社计算机研发部长期以来精心培育的计算机科学技术类本版品牌。这个品牌是由多个专题系列组成的横向大系列，涵盖了计算机技术的各个方面，特别是一直受到极大关注的程序开发类系列，例如“开发专家之数据库”、“开发专家之网络编程”、“开发专家之 Delphi”、“开发专家之 Sun ONE”及“开发专家之 MS.NET”等。这些专题系列基于各自的角度，从纵向上包含了该专题的所有内容。因此，整个“开发专家”的品牌架构纵横交错，囊括了所有的计算机技术和所有的技术层面，海纳百川而又极具可扩展性。

“开发专家”的作者队伍主要依托于“飞思科技产品研发中心”。“飞思科技产品研发中心”是由专业的策划人员、权威的技术专家和资深的作者队伍共同构成的。在图书的出版上，形成了以研发为基础、以出版为中心、以服务为支持的专业化出版框架和流程。通过深入的市场调查和技术跟踪，在综合了技术需求和读者焦点等因素的基础上，形成各系列丛书的写作重点和大纲，然后聘请业界的最前沿学者进行写作。同时，策划工作全程介入写作进程，严格控制写作质量，用最专业的技术背景、最深刻的理论基础、最具代表性的案例、最能为专业读者接受的形式，为读者提供品质最佳的图书产品。这体现了出版者和著作者的完美结合。

多年来，计算机研发部始终把创造社会效益摆在首位，秉承一切为国内计算机技术专业读者服务的精神，为推动国内 IT 发展、为体现国内技术的原创水平，倾付所有的创意与努力，将出版者的命运与读者的支持紧紧地连在了一起。

在此，我们临出版之残酷竞争而不惧，旌旗猎猎而异军突起，这与广大读者的支持是分不开的。为了使我们的脚步更坚实、使我们的队伍永葆活力和创造力，我们期待着您能为我们的前进贡献出您的意见和建议。同时，我们也在等待着您的加入。

电子工业出版社计算机研发部

前言

关于本丛书

对开发人员而言，.NET 是一个完美的开发平台。它提供了一套公共的运行库，并制定了一套公共语言规范，所有符合该规范的语言都可以无缝使用这套运行库。在.NET 平台下，除了语法上的区别，各种编程语言没有本质的不同。它们共享公共类库，具有类似的编程模型和相差无几的功能。开发人员可以自由选择自己喜爱的语言开发程序。.NET 平台提供大量的服务，包括垃圾自动收集、面向对象的多线程、基于程序集的部署、异常处理、特性编程、远程处理、ASP.NET 网页框架、互操作、安全性等，使开发人员可以快速构架任何应用，从传统的桌面应用到面向 Web 的大型分布式应用，应有尽有。.NET 将彻底改变软件的开发方式、使用方式和发行方式，.NET 将是一场软件革命。

Microsoft .NET 是一个用于构建、运行和体验下一代分布式应用程序的平台。它是跨客户端、跨服务器的开发人员工具，它由以下几部分组成：

- .NET 框架编程模型，它使开发人员能够构建 Web 应用程序、智能客户端应用程序和可扩展标记语言（XML）Web 服务应用程序，它们利用诸如 SOAP、可扩展标记语言（XML）和 HTTP 等标准协议以编程方式通过网络开放其功能。
- 开发人员工具，比如 Visual Studio.NET，该工具为用 .NET 框架进行编程提供了一个迅速集成开发应用程序的环境。
- 一组服务器，包括 Windows 2000、SQL Server 和 BizTalkServer，它可集成、运行、操作和管理 XML Web 服务和应用程序。
- 客户端软件，比如 Windows XP、Windows CE 和 Microsoft Office XP，它有助于开发人员跨多种设备和现有产品，为用户提供深切、引人注目的使用感受。

《开发专家之 MS.NET》系列丛书针对 Microsoft .NET 整个开发体系和特色组织作者进行了编写，它涵盖了 MS.NET 中的方方面面，为那些希望学习 MS.NET 技术的开发人员展示了非常完整的知识体系。在利用 MS.NET 进行项目开发的过程中，本丛书也可以为各种不同需要提供完美的解答。

关于本书

Microsoft Visual C++ 对于为 Windows 和万维网（WWW）创建性能最高的应用程序来说是一种效率最高的 C++ 工具。几乎所有世界级的软件（从占主导地位的 Web 浏览器到任务关键的企业应用程序）都是使用 Microsoft Visual C++ 开发系统生成的。Visual C++ .NET 的出现使 C++ 的生产效率上了一个新台阶，同时保持了它们灵活性、性能和控制。

本书读者对象

本书适合 C++/Visual C++/Visual C++.NET 的中高级读者，也可供对 .NET 感兴趣的读者参考。

本书内容

- Visual C++.NET 的新特性。
- MFC 编程：MFC 中消息映射处理、文档/视图结构、多线程应用开发、DLL 的开发和使用、ODBC 数据库编程及窗体编程。
- ATL 编程：COM 的创建和使用、支持双重接口的 COM 和支持连接点的 COM。
- ATL Server 编程：开发 Web 应用和开发 Web 服务。
- C++托管扩展编程：托管类型、基于委托的事件处理、ADO.NET 数据库编程、基于远程处理框架的分布式应用、Web 服务。

本书特点

跟同类书相比，本书有如下特色：

- 突出新特性

跟以前的版本相比，Visual C++.NET 增加了相当多的新功能。例如 ATL Server 库和 C++托管扩展就是全新的内容。利用新增的功能，开发人员可以开发出强劲的 Web 应用、Web 服务或托管应用。本书用丰富的实例、合理的逻辑、流畅的语言帮助 C++开发人员全面掌握这些新功能。

- 突出实用性

本书致力于解决开发人员在研发过程中碰到的实际问题。Visual C++.NET 中包含了众多的库，对每种库最合适解决的实际问题，本书做了详尽探讨。随书所赠的光盘包含了最近两年来作者对 Visual C++.NET 研讨过程所积累的所有学习案例。除供学习外，读者还可以在开发中直接使用这些代码。

学习建议

学习之前，读者最好安装 Visual Studio.NET 集成开发工具。若没有，至少也要安装 .NET 框架 SDK。微软官方网站提供了免费的 .NET 框架 SDK 下载。

同时对 .NET 编程语言 C#或 Visual Basic 的了解是非常重要的，因为本书对这些编程语言做过很多的介绍。读者可以参考我社出版的同套图书《Visual Basic.NET 编程指南》和《C#编程指南》。

一般而言，按照本书的章节顺序阅读本书对初学者来说是比较合适的。但本书的各章

涉及到独立的专题，因此中、高级读者可以选择从自己感兴趣的章节开始学习。

本书在写作过程中，参考了微软公司的相关资料，特此说明。

本书由飞思科技产品研发中心策划并组织编写，同时参与写作的还有刘晓华、杨艳、李华、谭育红、曾令波、向阳、袁华山、毛云朋、张文举、杨冬、杨华、杨波、覃华伟、占海燕、占兴春、占兴芳、詹兴华、李道海、谭成红、陈亮、陈琳、董耀华、高峰、李飞、雷华平、胡金科、廖先富等。由于编者水平有限，再加上时间仓促，虽然经过再三勘误，但仍难免有纰漏之处，欢迎广大读者予以批评和指正。

我们的联系方式：

电 话：(010) 68134545 68131648

电子邮件：support@fecit.com.cn

飞思在线：<http://www.fecit.com.cn> <http://www.fecit.net>

答疑网址：<http://www.fecit.com.cn/question.htm>

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

飞思科技产品研发中心

目 录

第 1 章 Visual C++.NET 介绍.....	1
1.1 新特性.....	1
1.1.1 特性化编程.....	1
1.1.2 托管代码和面向 .NET 框架	2
1.1.3 库.....	4
1.1.4 新的开发环境.....	4
1.2 创建 Visual C++.NET 应用..	5
1.2.1 创建 Visual C++ 项目	6
1.2.2 用 Visual C++应用 程序向导创建项目	6
1.2.3 往项目中添加项.....	7
1.2.4 编译、生成和分发 Visual C++ 应用	9
1.3 开发实例.....	9
1.3.1 概述.....	10
1.3.2 创建 ATL COM 服务器.....	10
1.3.3 创建简单的 Win32 COM 客户端程序 (ComProdClient.exe) ..	16
1.3.4 向 ATL COM 业务 对象添加.NET 框架支持.....	19
1.3.5 创建简单的 C++托管 扩展客户端程序 (MgdProdClient.exe) ..	22
1.3.6 创建 Visual C# .NET 框架客户端程序	24
1.4 关于 Visual C++.NET 的 常见问题.....	26
1.5 小结.....	28
第 2 章 MFC 编程基础	29
2.1 Windows 应用的基本框架...	29
2.2 MFC 编程领域	33
2.3 MFC 的内容	34

2.4 开发基于 MFC 的应用的 基本方法	36
2.5 应用程序类.....	38
2.5.1 InitInstance 成员 函数.....	39
2.5.2 Run 成员函数	40
2.5.3 ExitInstance 成员 函数.....	41
2.5.4 OnIdle 成员函数.....	41
2.5.5 CWinApp 提供的 特殊服务	42
2.6 窗口对象.....	43
2.7 实例.....	44
2.8 小结.....	46
第 3 章 消息处理.....	47
3.1 基本概念.....	47
3.1.1 消息及其类别	47
3.1.2 消息处理程序 和消息映射	48
3.1.3 用户界面对象 和命令 ID.....	48
3.2 消息处理的基本流程.....	50
3.2.1 非命令消息的处理.....	51
3.2.2 命令消息的处理流程..	52
3.2.3 消息映射表.....	54
3.2.4 消息处理函数.....	56
3.2.5 消息映射范围的 处理程序	58
3.2.6 反射消息的处理函数..	61
3.3 使用自定义消息.....	64
3.4 小结.....	65
第 4 章 开发基于对话框的应用	67
4.1 理解对话框.....	67
4.1.1 创建对话框	67
4.1.2 初始化对话框.....	70
4.1.3 处理 Windows 消息...	70
4.1.4 对话框数据交换.....	72

4.1.5	对话框数据验证	73
4.1.6	关闭对话框	74
4.2	使用有模式对话框	74
4.3	使用无模式对话框	79
4.4	使用属性表	84
4.5	小结	88
第 5 章	使用 CObject 类提供的服务	89
5.1	概述	89
5.2	使用基本 CObject 功能	90
5.2.1	转储	90
5.2.2	断言有效	91
5.3	运行时类信息	93
5.3.1	添加运行时类信息	93
5.3.2	运行时访问对象的类信息	94
5.4	动态创建	95
5.5	序列化	96
5.5.1	创建可序列化的类	96
5.5.2	序列化对象	98
5.6	小结	103
第 6 章	文档/视图结构及其应用	105
6.1	概述	105
6.1.1	文档/视图结构的好处	105
6.1.2	核心类	106
6.1.3	SDI 和 MDI	106
6.2	文档/视图的创建过程	108
6.2.1	创建文档模板	108
6.2.2	文档/视图的创建	109
6.2.3	MFC 对象之间的关系	111
6.3	使用文档	111
6.3.1	从 CDocument 派生文档类	112
6.3.2	用文档数据变量管理数据	112
6.3.3	将数据序列化到文件和从文件序列化数据 ..	113

6.3.4	不使用序列化机制	114
6.3.5	在文档中处理命令	115
6.4	使用视图	115
6.4.1	在视图中绘图	116
6.4.2	通过视图解释用户输入	117
6.4.3	对打印的支持	118
6.4.4	滚动和缩放视图	126
6.5	初始化和清理文档/视图	131
6.5.1	初始化文档和视图	132
6.5.2	清理文档和视图	132
6.6	将多个视图添加到单个文档	133
6.6.1	修改现有应用程序类 ..	133
6.6.2	创建和修改新视图类 ..	134
6.6.3	创建并附加新视图	134
6.6.4	实现切换函数	135
6.6.5	添加视图切换支持	136
6.7	拆分窗口	137
6.7.1	动态拆分	137
6.7.2	静态拆分	139
6.8	小结	144
第 7 章	创建和使用 DLL	145
7.1	理解 DLL	145
7.2	DLL 类型	146
7.3	链接到 DLL	147
7.3.1	隐式链接	147
7.3.2	显式链接	148
7.3.3	定位 DLL	150
7.4	导出和导入	150
7.4.1	使用 .DEF 文件	150
7.4.2	使用 __declspec	152
7.5	创建和使用非 MFC DLL	156
7.6	定义多个应用之间共享的数据	165
7.7	创建和使用静态链接到 MFC 的规则 DLL	166
7.8	创建和使用动态链接到 MFC 的规则 DLL	172

7.9 创建和使用 MFC	
扩展 DLL	175
7.10 小结	182
第 8 章 创建和使用资源	183
8.1 创建和使用纯资源的 DLL ..	183
8.1.1 创建	183
8.1.2 使用	185
8.2 MFC 扩展 DLL 中的资源 ..	190
8.2.1 CdynLinkLibrary	
对象	190
8.2.2 模块定位资源的顺序 ..	191
8.2.3 实例	192
8.3 附属 DLL	193
8.4 小结	197
第 9 章 多线程编程	199
9.1 概述	199
9.2 创建用户界面线程	200
9.3 创建辅助线程	209
9.3.1 启动线程	209
9.3.2 实现控制函数	209
9.3.3 实例	210
9.4 实现线程安全	215
9.4.1 选用合适的同步类	216
9.4.2 实现线程安全的	
基本方法	216
9.4.3 实例	217
9.5 终止线程	224
9.5.1 正常线程终止	224
9.5.2 过早的线程终止	224
9.5.3 检索线程的退出代码 ..	225
9.5.4 实例	225
9.6 小结	232
第 10 章 ODBC 数据库编程	235
10.1 概述	235
10.1.1 ODBC 组件	235
10.1.2 编程步骤	236
10.1.3 在文档/视图结构中	
使用数据库类	238
10.1.4 实例	239
10.2 使用 CDatabase 类	244
10.3 使用 CRecordset	247
10.3.1 浏览记录集	247
10.3.2 参数查询	249
10.3.3 添加、编辑和	
删除记录	252
10.4 小结	257
第 11 章 ATL COM 编程简介	259
11.1 COM 的基本概念	259
11.1.1 接口	260
11.1.2 IUnknown	260
11.1.3 引用计数	260
11.1.4 QueryInterface	261
11.1.5 封送处理	261
11.1.6 聚合	261
11.2 ATL 介绍	262
11.2.1 模板和模板库	262
11.2.2 ATL 的使用范围	262
11.2.3 ATL 还是 MFC	262
11.3 ATL 中的 COM	262
11.3.1 实现 CComObject	
RootEx	263
11.3.2 实现 CComObject、	
CComAggObject 和	
CComPolyObject	263
11.3.3 支持 IDispatch	
和 IErrorInfo	264
11.3.4 支持	
IDispatchImpl	264
11.3.5 更改默认类工厂	
和聚合模型	266
11.3.6 创建聚集对象	266
11.4 使用 COM 特性	267
11.5 用 ATL 创建 COM 实例	273
11.5.1 创建项目	273
11.5.2 添加服务器对象	275
11.5.3 实现服务器	278
11.5.4 添加客户端对象	281
11.5.5 添加客户端接口	282

11.5.6 实现客户端.....	283	15.2.2 向替换方法 传递参数.....	357
11.5.7 使用客户端控件.....	286	15.3 获取用户输入.....	363
11.6 小结.....	288	15.4 输入验证.....	367
第 12 章 支持双重接口.....	289	15.4.1 利用 CValidateObject 进行验证.....	368
12.1 IDispatch 接口的实现.....	289	15.4.2 使用正则表达式.....	371
12.2 调用双接口.....	292	15.5 使用会话状态服务.....	378
12.2.1 通过 IUnknown 接口调用.....	292	15.6 使用缓存.....	392
12.2.2 通过 IDispatch 接口调用.....	294	15.7 使用性能监视服务.....	402
12.3 小结.....	301	15.8 文件上载.....	411
第 13 章 可连接对象.....	303	15.9 创建安全的 Web 应用.....	415
13.1 理解可连接对象.....	303	15.10 创建和使用 ISAPI 过滤器.....	421
13.2 实现连接对象.....	304	15.11 小结.....	427
13.3 基于连接的 COM 事件.....	308	第 16 章 ATL Server 开发 XML Web 服务.....	429
13.4 小结.....	312	16.1 XML Web 服务简介.....	429
第 14 章 ATL 中的窗口编程.....	313	16.2 ATL Server 对 XML Web 服务的支持.....	430
14.1 ATL 中的窗口类.....	313	16.3 Web 服务的使用.....	435
14.2 实现窗口.....	314	16.4 Web 服务中的内存管理.....	438
14.2.1 基于新 Windows 类创建窗口.....	314	16.5 使用 SOAP 标头.....	443
14.2.2 为现有 Windows 类创建超类.....	322	16.6 使用缓存.....	450
14.2.3 为现有窗口 创建子类.....	323	16.7 自定义传输协议.....	457
14.3 实现对话框.....	326	16.8 小结.....	461
14.3.1 使用“ATL 对话框 向导”添加对话框....	326	第 17 章 C++ 托管扩展编程.....	463
14.3.2 手动添加对话框.....	326	17.1 何时使用 C++ 托管扩展..	463
14.3.3 实例：创建基于 ATL 对话框的应用.....	328	17.2 托管 C++ 中的 项目及内容.....	464
14.4 使用 WTL 库进行 ATL 窗口编程.....	336	17.3 托管类型.....	465
14.5 小结.....	347	17.3.1 __gc 数组.....	466
第 15 章 ATL Server 开发 Web 应用.....	349	17.3.2 托管指针类型.....	467
15.1 ATL Server 应用程序结构.....	349	17.3.3 字符串.....	469
15.2 服务器响应文件.....	353	17.3.4 委托.....	469
15.2.1 SRF 基本语法.....	353	17.3.5 MFC 和托管类型.....	470
		17.3.6 托管类型和 main() 函数.....	471

17.4 为现有应用程序添加		17.6 程序集.....	476
C++托管扩展.....	471	17.7 序列化.....	482
17.4.1 修改现有项目设置	471	17.8 基于委托的事件处理.....	485
17.4.2 在现有应用程序中		17.9 互操作.....	491
使用新的		17.10 使用 ADO.NET	
托管扩展功能.....	472	访问数据.....	495
17.5 处理异常.....	473	17.11 创建窗体应用	505
17.5.1 使用托管扩展		17.12 创建多线程应用.....	509
引发异常.....	473	17.13 创建基于远程处理框架的	
17.5.2 使用托管扩展的		分布式应用.....	515
try/catch 块.....	473	17.14 创建 XML Web 服务.....	521
17.5.3 捕捉非托管		17.15 小结.....	528
C++类型.....	474		
17.5.4 托管扩展和 __finally			
关键字.....	475		

第 1 章 Visual C++.NET 介绍

在 Windows 平台下，Visual C++ 一直是最强大和最灵活的开发工具。与 Visual Basic 相比，Visual C++ 对开发人员提出了更高的要求——每一个 Visual C++ 程序员都是在阅读了大量的代码，经历了许多反复后渐渐领会其精妙之处。

Visual C++.NET 带来了震撼人心的感受。与其前期版本相比，它提供了更加完善的类库，并且包含了一种用来开发 .NET 应用的 C++ 托管扩展。另外引入大量的特性语法通过批注的方式大大简化了应用程序的开发。这些特性使开发人员可以在概念的层次而不是编码的层次上进行程序设计。这样，Visual C++ 程序员现在也能享受到 Visual Basic 程序员的乐趣了。

Visual C++.NET 作为最强大的编程工具，已经为未来十年的应用开发做好了准备。有了这么好的编程工具，聪明的开发人员一定会如虎添翼，也能更加淋漓尽致地体会编程的乐趣。

本章主要内容如下：

- Visual C++.NET 的新特性
- 创建 Visual C++.NET 应用的基本方法
- 用 Visual C++.NET 创建应用实例
- 关于 Visual C++.NET 的常见问题及解答

1.1 新 特 性

如果读者是一名 Visual C++ 开发人员，那么最关心的是新版本究竟增添了哪些功能，从而决定是否需把已有的应用升级到或迁移到 Visual C++.NET。对于一名新手而言，可能在学习 Visual C++.NET 前在犹豫是否选用其他语言，例如 C#、Visual Basic 等。

这一节就向新老 Visual C++ 开发人员介绍一下 Visual C++.NET 的若干新特性。相信这些新特性一定会激发大家学习 Visual C++.NET 的愿望。

1.1.1 特性化编程

特性旨在提供一种有效而快捷的方法来简化使用 Visual C++ 的 COM 编程。特性与 C++ 关键字一样，在源文件中使用，并由编译器进行解释。特性可以修改现有代码的行为，甚至可以插入附加框架代码来完成基本任务，例如实现 ActiveX 控件、创建类工厂，或者设置数据库命令的格式。几乎可以将特性应用到任何 C++ 对象（如类、数据成员及成员函数）上，还可以将特性作为独立的语句插入到源代码中。

为了更清楚地看出利用特性化编程是多么方便，请看下面的代码片断：

```
[
    coclass,
    threading("apartment"),
    event_source("com"),
    vi_progid("DispServer.DispServ"),
    progid("DispServer.DispServ.1"),
    version(1.0),
    default("IDispServ"),
    uuid("BD9B1276-EBE9-4E9C-B3F9-07298D8BBF4D"),
    helpstring("DispServ Class")
]
class ATL_NO_VTABLE CDispServ :
    public IDispServ
```

“[”和“]”间的就是若干个特性，这些特性作用在 CDispServ 上，使其变为一个 COM 类。有了特性，IDL 代码也消失了——它们将由编译器根据特性自动产生。

如果不用特性，那么需要先定义一个下面的类：

```
class ATL_NO_VTABLE CDispServ :
    public CComObjectRootEx<CComSingleThreadModel>,
    public CComCoClass<CDispServ, &CLSID_Dispatch>,
    public IDispatchImpl<IDispServ, &IID_IDispServ, &LIBID_DISPSEVERLib, /*wMajor =*/ 1,
/*wMinor =*/ 0>,
    public IConnectionPointContainerImpl<CDispServ>,
    public CProxy_IDispServEvents< CDispServ >
```

然后编写下面一段 IDL 代码：

```
coclass DispServ
{
    [default] interface IDispServ;
    [default, source] dispinterface _IDispServEvents;
};
```

随着学习的深入，读者将进一步熟悉特性在简化 Visual C++.NET 编程方面的好处。而强调用特性简化编程也是本书的一大特色。

1.1.2 托管代码和面向 .NET 框架

.NET 平台是微软未来的计算平台。Visual C++.NET 引入托管扩展支持编写各种托管应用。C++托管扩展简化了 COM 编程，并且使程序能够访问由 .NET 公共语言运行库提供的服务，如垃圾回收和语言互操作性，以及无比丰富的 .NET 基础类库。例如下面的代码利用 .NET 中的 ADO.NET 组件访问数据库：

```
#using <mscorlib.dll>
using namespace System;

#using "System.dll"
```

```
#using "System.Data.dll"

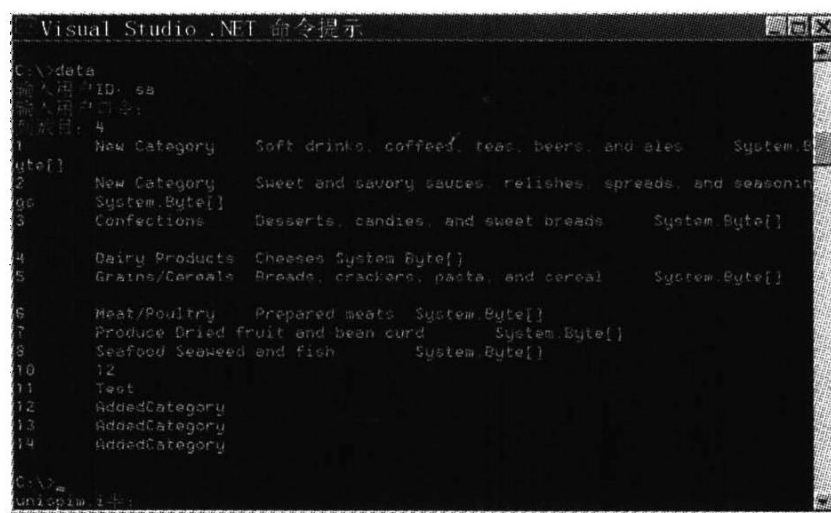
using namespace System::ComponentModel;
using namespace System::Data;
using namespace System::Data::SqlClient;

void main()
{
    // 获取用户名和口令
    Console::Write(S"输入用户 ID: ");
    String *userId = Console::ReadLine();
    Console::Write(S"输入用户口令: ");
    String *password = Console::ReadLine();
    // SQL 查询
    String *query = S"SELECT * FROM Categories";
    // 连接字符串
    //根据自己的 SQL 配置修改下面的连接字符串
    String *connectString = String::Format(S"Data Source=diana\\diana;Database=Northwind;
UID={0};Password={1};", userId, password);
    //生成连接
    SqlConnection* sqlconn = new SqlConnection(connectString);
    //打开连接
    sqlconn->Open();
    SqlCommand *sqlCommand = new SqlCommand(query, sqlconn);
    // 阅读器
    SqlDataReader *dataReader = sqlCommand->ExecuteReader();
    // 列的数目
    int numCols = dataReader->FieldCount;
    // 显示列的数目
    Console::Write(S"列数目: ");
    Console::WriteLine(numCols);
    // 显示返回的记录
    while(dataReader->Read())
    {
        for (int c = 0; c < numCols; c++)
        {
            Console::Write(dataReader->Item[c]->ToString());
            Console::Write(S"\t");
        }
        Console::WriteLine(S"");
    }
}
```

执行下面的命令将源代码编译为可执行文件:

```
cl /clr Data.cpp
```

上面的命令执行后, 将生成 Data.exe, 执行 Data.exe, 输入连接 Northwind 数据库的用户 ID 和口令, 将看到 Categories 表的内容, 如图 1-1 所示。



```
Visual Studio .NET 命令提示
C:\>data
输入用 ID: sa
输入用 ID: sa
创建: 4
1 New Category Soft drinks, coffees, teas, beers, and ales System.Byte[]
2 New Category Sweet and savory sauces, relishes, spreads, and seasonings System.Byte[]
3 Confections Desserts, candies, and sweet breads System.Byte[]
4 Dairy Products Cheeses System.Byte[]
5 Grains/Cereals Breads, crackers, pasta, and cereal System.Byte[]
6 Meat/Poultry Prepared meats System.Byte[]
7 Produce Dried fruit and bean curd System.Byte[]
8 Seafood Seafood and fish System.Byte[]
12
11 Test
12 AddedCategory
13 AddedCategory
14 AddedCategory
C:\>
Unipia 1.1
```

图 1-1 执行结果

大家可能注意到,程序中用`#using`引入程序集(相当于传统的 DLL),用 `using namespace` 引入命名空间,然后就可以自由使用程序集中的类型了。在编译时,要指定 `clr` (公共语言运行时: Common Language Runtime 的首字母缩写) 选项。

在 Visual Studio.NET 集成开发环境下,可以通过设置项目属性指定编译为托管程序集,就会自动指定 `/clr` 编译选项。

托管 C++ 的另一个优势是可以将已有的 C++ 应用平稳过渡到托管平台下。例如,可以用托管 C++ 扩展包装 COM、传统的 DLL 等。在介绍 C++ 托管扩展时,将看到这方面的例子。

1.1.3 库

Visual C++.NET 提供了非常多的类库以支持各种应用的开发。如此繁多的类库一方面使 Visual C++.NET 成为 Windows 平台下最强大的开发语言,但另一方面又增加了 Visual C++.NET 的学习难度。

Visual C++.NET 提供了下面几种类库:活动模板库(ATL)、ATL Server 库、Microsoft 基础类(MFC)、OLE DB 模板库、C 运行时库、共享类、标准 C++ 库、旧 `iostream` 库。对开发人员而言,最有挑战性的是前三个类库。

ATL 主要用来开发轻量级的 COM 组件。ATL Server 类库用来开发 Web 应用,其中还包括前景最看好的 Web 服务。MFC 则是主要用来开发本地的 Windows 窗体应用、DLL 等。当然 MFC 也可以用来开发 COM、Web 应用等,因为 MFC 提供了对 Windows API 的最完整的包装。

在 Visual C++.NET 中,每种应用中都可选用任何一种或多种类库。例如,可以在 MFC 应用中轻松地添加对 ATL 类库的支持。

1.1.4 新的开发环境

Visual Studio.NET 是最新的 Visual C++ 应用集成开发环境。Visual Studio.NET 提供了

开发各种 Visual C++ 应用的模板，如图 1-2 所示。模板已经生成了应用程序的主干框架，这样可以大大提高应用程序的开发效率。

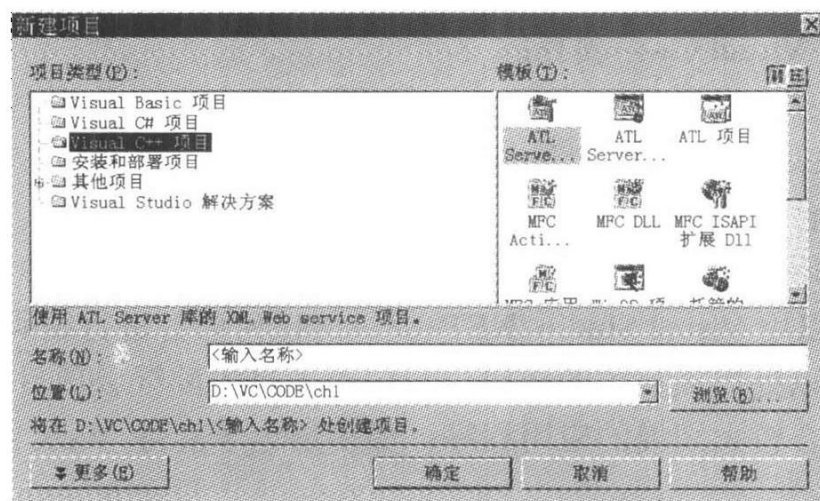


图 1-2 新建 Visual C++ 项目向导

Visual Studio.NET 提供了新的项目属性页指定如何生成和调试 Visual C++ 项目，如图 1-3 所示。利用“属性页”对话框，可以设置支持托管扩展、库目录、程序集目录等。

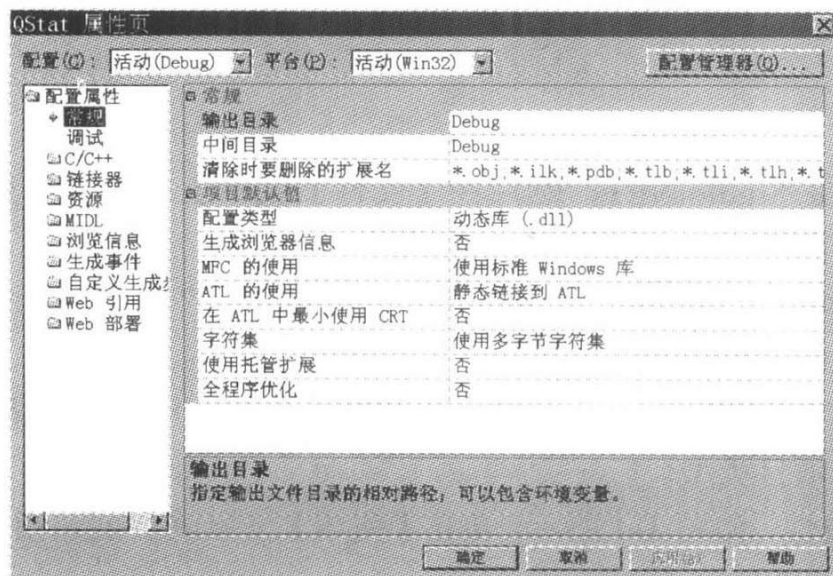


图 1-3 “属性页”对话框

1.2 创建 Visual C++.NET 应用

有两种创建 Visual C++.NET 应用的方式：集成开发环境，命令行编译器 cl。本书主要讲解如何用集成开发环境 Visual Studio.NET 生成 Visual C++.NET 应用。因为 Visual Studio.NET 集成了编辑代码、编译、调试和部署等一系列功能。特别是支持语法的高亮度显示、函数参数自动提示等功能，可以减少输入错误，并免去记忆函数参数的负担。

在 Visual Studio.NET 中，每个 Visual C++.NET 应用就是一个 Visual C++.NET 项目。因此 Visual C++.NET 应用开发过程就是创建 Visual C++.NET 项目，然后向项目中添加功能，最后编译和分发应用。