

软件学院丛书

Visual C++ .NET 程序设计与应用

张 炜 等编著

展示 .NET 开发环境

介绍 C++ 基本知识

C++ 在各方面的应用

丰富的编程实例

Visual C++ .NET
程序设计与应用

电子工业出版社



电子工业出版社
Publishing House of Electronics Industry
www.phei.com.cn

软件学院丛书

Visual C++ .NET 程序设计与应用

张 炜 等编著

電子工業出版社

Publishing House of Electronics Industry

北京 · Beijing

内 容 简 介

本书详细介绍了微软新一代程序开发工具.NET的使用方法和特性,并结合C++的最新版本,通过具体的编程,探讨了C++在各个领域的应用。

本书在该细微处讲的非常详细,对.NET框架的整体开发环境、C++的基本语句、以及该语言在各方面的应用,都进行了深入细致的讲解,而且实例也很结合实际,有利于读者自学和提高。

本书除了介绍.NET开发环境外,还介绍了C++基本概念、类、对象、应用程序设计、菜单设计、控件设计、对话框设计、多线程程序设计、数据库编程、ActiveX编程和网络应用程序设计等方面的知识。

本书适合初学者和大专院校学生学习使用,也可作为刚使用.NET环境编程的人员的参考书。

本书版权归电子工业出版社所有,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,翻版必究。

图书在版编目(CIP)数据

Visual C++ .NET 程序设计与应用 / 张 炜等编著. — 北京: 电子工业出版社, 2002.9

(软件学院丛书)

ISBN 7-5053-7941-0

I. V... II. 张... III. C 语言 - 程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 062434 号

责任编辑: 郝志恒

印 刷: 北京市增富印刷有限责任公司

出版发行: 电子工业出版社 www.phei.com.cn

北京市海淀区万寿路 173 信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787 × 1092 1/16 印张: 29 字数: 742 千字

版 次: 2002 年 9 月第 1 版 2002 年 9 月第 1 次印刷

定 价: 39.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,请向购买书店调换。若书店售缺,请与本社发行部联系调换。联系电话: 88211980 68279077

出版说明

为什么要编写这套丛书

随着 Internet 的高速发展, 计算机走进了千家万户, 从而实现了安全和快速的电子商务平台, 同时也为软件开发提出了更高的要求。

微软公司为了适应新形势的变化, 在原来 Visual 可视化编程软件的基础上, 依据新的 Web 标准, 推出了具有全新设计理念的 .NET 系列软件。 .NET 的主要目标是在 Internet 站点之间通过分布式的协作, 为用户提供更完美的服务。为此, 微软公司开发了 .NET 架构, 它是未来软件行业发展的方向, 相信不久的将来一定会成为软件领域的新亮点。

根据这种新的发展趋势, 我们组织国内长期从事编程的软件工程师, 编写了这套深入介绍 .NET 编程的丛书。

丛书的读者对象

本套丛书定名为“软件学院”, 也意喻着本书的读者对象是在校学习的学生和有一定基础的软件编程人员。

丛书的作者和编委

本套丛书的作者有大学教授, 有研究所高级工程师, 也有公司软件设计师, 他们都有使用 Visual 可视化开发环境开发应用系统的丰富经验, 对软件编程有很深的理解。作者在详细介绍新软件特性的基础上, 考虑到 Visual 可视化编程环境的继承性差异, 将 .NET 技术的编程思想融入到自己的实际编程工作中。本书通过实例将作者的设计思想和程序实现过程原原本本地展现给读者。

本套丛书的编委会成员为: 袁建洲、林丽闽、李兰友、万振凯、梁普选、梁心东、唐树才、张玮、翟战强、刘树声。

丛书的特点

本套丛书与实际应用结合紧密, 通过大量实例提高读者掌握编程技术的能力和技巧, 使读者在编程实践中掌握程序语言的内涵。

对于程序语言的各种功能及应用, 本套丛书都通过实例来进行介绍, 更加方便了读者自学。为了照顾初学者, 我们还对如何进入 .NET 编程环境等基础知识进行了简单介绍, 使初学者不至于落入无从下手的境地。

编 者

前 言

.NET是微软最新推出的开发平台,它所支持的Visual C++ .NET作为最流行的编程语言,较原来的 Visual C++ 6.0 版本而言,在很多方面都做了改进,相信读者使用过一段.NET 开发平台后,一定会深有感触。

在编写本书时,作者充分考虑到了初学者学习程序语言的特点,在对基本知识的讲解上,力求做到深入细致,并结合大量示例,用操作步骤的形式,一步步向读者揭开 Visual C++ .NET 的神秘面纱。

本书的特点就是该细微的地方讲的很细,尤其是在讲解编程实例时更是如此。作者把自己的编程经验和对.NET 技术的了解,通过示例呈现给读者,培养读者的实际编程能力,并让读者在编程中,掌握.NET 的技术。

本书作者张炜、唐树才、袁建洲,非常熟悉在.NET框架基础上构架企业级应用程序的方法,对可视化的软件架构设计、数据库应用设计方法与实现、数据库建模方法、高级 Web 服务应用开发设计、开发下一代安全 Web 应用方面很有研究和经验。他们所开发的项目曾获得过很多奖项。

由于本书编写比较仓促,难免有一些问题,欢迎广大读者批评指正,我们将不胜感激。

本书作者除张炜、唐树才、袁建洲外,还有翟战强、薛刚逊、杨航、魏勇、邸颜强、陈大勇、李强、江光德、谢应邦、刘花果、梁辉、周锋、刘和、佟美莹,另外,田东坡、李黎、张明珠、郭景新、唐海涛、王长宁、陈惠明、陈惠芬、李东升、卢云强、董波、腾为肇也对本书的出版提供了很多帮助。

编 者

2002.7

目 录

第 1 章 Visual C++ .NET 集成开发环境	1
1.1 Visual C++ .NET 概述	1
1.1.1 .NET 开发框架	1
1.1.2 Visual C++ .NET 新特点	5
1.1.3 解决方案和项目	6
1.2 Visual Studio .NET 的启动和退出	6
1.3 菜单栏	14
1.3.1 “文件” 菜单	14
1.3.2 “编辑” 菜单	15
1.3.3 “视图” 菜单	16
1.3.4 “项目” 菜单	18
1.3.5 “生成” 菜单	20
1.3.6 “调试” 菜单	21
1.3.7 “工具” 菜单	22
1.3.8 “窗口” 菜单	25
1.3.9 “帮助” 菜单	26
1.4 工具栏	27
1.5 窗口	30
1.5.1 “解决方案资源管理器” 窗口	30
1.5.2 “类视图” 窗口	31
1.5.3 “资源视图” 窗口	32
1.5.4 “属性” 窗口	33
1.5.5 代码和文本编辑器窗口	33
1.5.6 服务器资源管理器	35
1.6 小结	35
习题（一）	36
第 2 章 C++ 语言概述	37
2.1 C++ 语言元素	37
2.1.1 标识符	37
2.1.2 关键字	37
2.1.3 基本数据类型	38
2.1.4 变量和常量	39
2.1.5 分隔符	39

2.1.6 运算符和表达式	39
2.1.7 数组	41
2.1.8 指针	42
2.1.9 字符串	43
2.1.10 自定义数据类型	43
2.1.11 数据类型转换	44
2.2 基本语句	45
2.2.1 空语句	46
2.2.2 块语句	46
2.2.3 选择语句	46
2.2.4 循环语句	50
2.2.5 break 和 continue 转移语句	51
2.2.6 预处理指令	51
2.2.7 注释语句	53
2.3 函数	53
2.3.1 C++ 标准库	54
2.3.2 自定义函数	54
2.3.3 函数重载	56
2.3.4 函数调用	57
2.4 类和对象	58
2.4.1 面向对象的编程	58
2.4.2 类	62
2.4.3 对象	64
2.4.4 基类和派生类	66
2.4.5 虚函数和抽象类	68
2.4.6 运算符重载	68
2.5 小结	70
习题 (二)	71
第3章 Windows 应用程序快速设计	73
3.1 MFC 基础知识	73
3.1.1 MFC 概述	73
3.1.2 MFC 类的层次结构	74
3.1.3 CObject 类	76
3.1.4 CWinApp 类	79
3.1.5 CWnd 类	80
3.1.6 CFrameWnd 类	80
3.1.7 CView 类	81
3.1.8 CDocument 类	81
3.2 Windows 的消息响应机制	81
3.3 快速设计 Windows 应用程序	83

3.3.1 创建 MFC 应用程序框架结构	83
3.3.2 编辑源文件	90
3.3.3 运行应用程序	104
3.4 小结	107
习题 (三)	107
第 4 章 菜单、工具栏和状态栏	110
4.1 菜单	110
4.1.1 创建菜单资源	110
4.1.2 操作菜单资源	113
4.1.3 动态修改菜单	114
4.1.4 菜单消息响应	116
4.1.5 菜单的 UI 机制	118
4.1.6 使用菜单示例	119
4.2 工具栏	128
4.2.1 工具栏资源	129
4.2.2 工具栏类	130
4.2.3 消息传递机制	132
4.2.4 创建工具栏	132
4.3 状态栏	134
4.3.1 CStatusBar 类	134
4.3.2 创建和处理状态栏	135
4.3.3 状态栏提示信息	137
4.4 小结	158
习题 (四)	159
第 5 章 控件	160
5.1 常用控件概述	160
5.1.1 控件种类	160
5.1.2 控件的工作机制	162
5.2 常用控件使用	165
5.2.1 编辑框控件	165
5.2.2 按钮	166
5.2.3 列表控件	168
5.2.4 组合框	170
5.2.5 滚动条	171
5.2.6 微调按钮	171
5.2.7 滑块控件	172
5.2.8 进度条	173

5.2.9 动画控件	173
5.2.10 树形控件	174
5.3 创建使用控件的应用程序	175
5.4 小结	188
习题 (五)	188
第 6 章 对话框	190
6.1 对话框	190
6.2 通用对话框类	191
6.2.1 CFontDialog 类	191
6.2.2 CFileDialog 类	193
6.2.3 CPrintDialog 类	194
6.2.4 CColorDialog 类	195
6.2.5 CFindReplaceDialog 类	196
6.3 创建对话框	198
6.3.1 创建对话框模板	198
6.3.2 用 MFC 类向导生成对话框类	199
6.3.3 创建并显示对话框	200
6.3.4 初始化对话框	201
6.3.5 处理消息	201
6.3.6 对话数据交换和验证	202
6.3.7 访问对话框控件	204
6.3.8 关闭对话框	205
6.3.9 管理对话框的 MFC 函数	206
6.4 创建对话框应用	207
6.5 小结	216
习题 (六)	216
第 7 章 文档 / 视图结构	218
7.1 文档 / 视图结构	218
7.1.1 文档和视图	218
7.1.2 文档类	219
7.1.3 视图类	220
7.1.4 框架窗口类	220
7.1.5 文档模板	221
7.1.6 文档 / 视图结构的相互关系	222
7.2 多文档和多视图	223
7.2.1 多文档	223
7.2.2 多视图	224
7.3 创建一个单文档应用程序	225

7.4 小结	239
习题（七）	239
第8章 OpenGL 编程	241
8.1 OpenGL 基本概念	241
8.2 OpenGL 程序基础	244
8.2.1 OpenGL 工作原理	244
8.2.2 OpenGL 数据类型	250
8.2.3 OpenGL 函数	250
8.3 编写 OpenGL 应用程序	256
8.4 小结	279
习题（八）	279
第9章 多线程应用程序的设计	281
9.1 概述	281
9.1.1 进程和线程	281
9.1.2 线程的分类	283
9.2 创建和使用线程	283
9.2.1 启动线程	283
9.2.2 初始化线程	286
9.2.3 悬挂和恢复线程	286
9.2.4 使线程休眠	286
9.2.5 终止线程	287
9.2.6 启动工作者线程的实例	288
9.3 设置线程优先级	298
9.4 实现线程同步	300
9.4.1 信号灯类 CSemaphore	301
9.4.2 事件同步类 CEvent	301
9.4.3 临界区类 CCriticalSection	302
9.4.4 互斥信号灯类 CMutex	302
9.5 创建多线程应用	304
9.6 小结	313
习题（九）	313
第10章 动态链接库	314
10.1 DLL 概述	314
10.2 使用 DLL	316
10.2.1 DLL 入口函数	316
10.2.2 DLL 的导入和导出函数	316
10.2.3 链接 DLL	317

10.3 编写 DLL	319
10.4 小结	335
习题 (十)	335
第 11 章 ActiveX 编程	337
11.1 OLE 技术概述	337
11.1.1 组件对象模型	338
11.1.2 结构化存储	339
11.1.3 Monikers	340
11.1.4 统一数据传输	340
11.1.5 OLE 文档	341
11.1.6 OLE 自动化	341
11.2 ActiveX 技术概述	342
11.2.1 ActiveX 文档	343
11.2.2 ActiveX 容器	343
11.2.3 ActiveX 控件	346
11.2.4 创建 ActiveX 控件	353
11.2.5 插入 ActiveX 控件	357
11.3 ActiveX 控件应用实例	358
11.4 小结	361
习题 (十一)	361
第 12 章 数据库编程	363
12.1 关系数据库基本知识	363
12.1.1 基本概念	363
12.1.2 SQL 语言	365
12.2 ODBC/MFC 编程	367
12.2.1 ODBC 体系结构	367
12.2.2 CDatabase 对象	369
12.2.3 CRecordSet 对象	371
12.3 DAO/MFC	375
12.3.1 CDaoWorkspace 类	376
12.3.2 CdaoDatabase 对象	377
12.3.3 CDaoRecordset 对象	378
12.3.4 CDaoTableDef 对象	379
12.3.5 CDaoQueryDef 对象	380
12.4 使用 OLE DB 编程	380
12.5 使用 ADO 编程	381
12.6 数据库编程示例	382
12.6.1 创建数据源	382

12.6.2 创建 ODBC 应用程序	384
12.6.3 编辑 ODBC 应用程序	386
12.6.4 增加表中记录的功能	389
12.6.5 删除表中记录的功能	394
12.6.6 查找记录的功能	395
12.7 小结	406
习题 (十二)	406
第 13 章 网络通信应用程序的设计	408
13.1 网络通信基础知识	408
13.2 Windows Sockets 编程	409
13.2.1 套接字的定义	410
13.2.2 端口和地址	411
13.2.3 阻塞	413
13.2.4 CAsyncSocket 类	413
13.2.5 CSocket 类	417
13.3 WinInet 编程	420
13.3.1 MFC WinInet 类	420
13.3.2 创建 MFC WinInet 客户程序	421
13.4 创建网络应用程序	428
13.5 小结	444
习题 (十三)	444
附录 Visual C++ .NET 的安装	446
习题答案	451

第 1 章

Visual C++ .NET 集成开发环境

本章包括:

- ◆ Visual C++ .NET 概述
- ◆ Visual Studio .NET 的启动和退出
- ◆ 菜单栏
- ◆ 工具栏
- ◆ 窗口

Visual C++ 是一个基于 Windows 平台的 C++ 集成开发环境, 该开发环境由许多部件组成, 这些部件协同工作可大大简化应用系统的开发过程, 它继承并兼容标准 C 语言、标准 C++ 语言等语言标准。由于它是微软公司推出的开发工具, 因此它能够较好地与 Windows 平台 (包括 Windows 95、Windows 98、Windows NT、Windows 2000、Windows CE 等操作系统) 接合, 从而可深入地访问操作系统的内部和底层, 实现高级程序设计要求, 提高应用程序的运行效率和质量。

如果读者用过别的一些软件开发工具, 例如 Borland C++、Delphi、Visual Cafe 等, 那么对于这种 IDE 集成开发环境 (Integrated Development Environment) 就不会陌生。同这些集成开发环境相比, Visual C++ 有许多自身的特点。

我们知道, 微软公司的软件开发环境以其见长的人机界面的友好性, 吸引了大量的不同层次的软件设计人员使用其程序开发工具, 而 Visual C++ 正是这种开发环境。它操作简单, 而且界面结构和风格明晰, 功能设计也很符合程序员的开发习惯, 同时它的在线帮助内容丰富, 使用灵活。因此, 掌握这种开发工具的操作使用方法和界面特点, 有利于进行应用系统的开发。本章将介绍 Visual C++ .NET 的集成开发环境和使用方法。

1.1 Visual C++ .NET 概述

Visual C++ .NET 是微软公司于 2002 年推出的应用和解决方案的开发环境软件, 是 Visual Studio .NET 系列的一个重要成员, 它是 Visual C++ 6.0 的更高级版本。

1.1.1 .NET 开发框架

Visual Studio .NET 将在网络应用解决方案方面具有更为强大而有力的功能支持, 主要用

于开发企业规模的 Web 应用程序,也可快捷而简便地开发桌面应用程序。

.NET 开发平台由公共语言运行时、基类库、数据库访问技术 ADO.NET 和 XML、网络应用开发技术 ASP.NET 和 Windows 编程技术 Windows 窗体、公共语言规范及编程语言等部分组成,如图 1-1 所示。

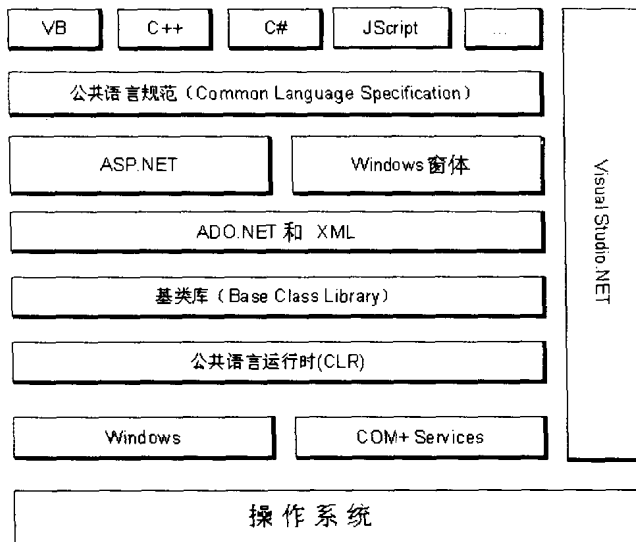


图 1-1 Microsoft .NET 开发平台

.NET 开发平台包括 .NET 编程语言和 .NET 框架 (Framework) 等部分。.NET 开发平台可以缩短应用开发时间,简化发布和管理,提高运行效率。其主要优点为与平台无关、多语言开发、简化开发过程、简单的部署、无缝的集成、以 XML/SOAP 为核心、提高应用的可靠性和性能更加优化等特点。

1. 公共语言运行时

.NET 开发平台给编程开发人员提供一种全新的开发框架,而公共语言运行时 (Common Language Runtime, CLR) 是此框架的基础。公共语言运行时是一个软件引擎,用来加载应用程序,确认它们没有错误运行,进行相应的安全许可验证,执行应用程序,然后在完成后将它们清除。基于公共语言运行时开发的代码称为可管理代码,可管理代码在执行代码和运行时 (Runtime) 引擎之间定义了一个接口,如创建对象、方法调用等任务都可由运行时引擎来完成,它同时向执行代码提供一些附加的服务功能。

在 .NET 开发平台中,编程开发人员可以使用 VC、VB、C# 等编程语言编写应用程序,编译器产生的可管理代码并不是 x86 机器码或其他机器码,而是 Microsoft 中间语言 (Microsoft Intermediate Language, MSIL) 代码,MSIL 指令不能在 CPU 上直接执行,因此,运行时引擎必须首先编译可管理的 MSIL 指令,使其成为本地 CPU 指令。CLR 提供不平行的跨语言集成,包括跨语言继承代码的能力。所有使用 CLR 的语言共享一个通用类型系统,它能让使用多种语言开发应用程序变得更简单。

CLR 加载并运行应用程序的步骤如下:

1. 类加载器 (Class Loader) 将应用程序的汇编加载到内存中。汇编包括 Microsoft 中间语言代码、描述应用程序的汇编中软件组件的元数据以及其他应用程序所需的组件。类加载器使用应用程序汇编的元数据, 试图加载任何应用程序所需组件的支持汇编, 如支持桌面应用程序的图形用户接口控制的汇编。类加载器使用版本策略 (Versioning Policy) (由编程人员或系统管理员指定) 来确定加载什么版本的汇编, 这就消除了组件版本问题。
2. 一旦加载应用程序和支持的汇编后, 验证器 (Verifier) 就检查其内容以确保它是类型安全的 (type-safe), 并确定应用程序的安全许可。
3. 本机编译器将 MSIL 代码转换为可控制的本机代码, 这是处理器相关代码, 它知道如何与 CLR 提供的服务 (如碎片整理) 或 CLR 安全系统等进行交互。

2. 基类库

基类库向程序员提供用来编写在 CLR 控制下运行的代码的软件组件, 它们按照单一有序的分级组织提供了一个庞大的功能集, 包括从文件系统到对 XML 功能的网络访问的每一样功能。

在 .NET 技术前, 编程人员使用类库 (Class Library) 来提高应用程序的开发效率, 包括使用 COM 组件、API 和编程语言本身的类库, 这些类库在调用上不完全相同, 有各自不同的约定, 从而给应用程序稳定性带来了潜在的威胁。为了解决这一问题, .NET 编程技术提出了基类库 (Base Class Library) 的概念, 它包括从输入输出到数据访问等各方面, 提供了一个统一的面向对象的、层次化的、可扩展的编程接口。

由于基类库比较庞大, 因此为了容易查找和使用这些基类库, .NET 编程技术提出了命名空间 (Namespace) 的概念。命名空间是一个层次结构, 分门别类地放置不同的类, 使用一种点符号将它们分隔, 在不同的命名空间中可以使用相同名称的类。常用的基类库如表 1-1 所示。

表 1-1 .NET 框架常用基类库

分类	命名空间	功能
常用任务	System.Collections	集合对象, 包括队列、数组、哈希表、链表等。
	System.IO	简单数据流访问与管理, 包括文件 I/O、内存 I/O 等。
	System.Text	字符编码、转换和字符串处理。
	System.Text.RegularExpressions	全面支持正则表达式。
	System.Threading	多线程支持, 包括锁定和同步。
客户端 GUI	System.Drawing	丰富的 2-D 功能和 GDI+ 支持。
	System.Windows.Forms	Windows 传统应用程序的丰富界面特性支持。
Web 服务	System.Web	支持 Web 服务器和 Client 管理、通信与设计。提供 ASP.NET 的核心支持, 包括 Web Forms。
	System.Web.Services	基于 SOAP 的 Web Service 的客户与服务器端支持。
.NET 框架安全	System.Security	访问 .NET Framework 安全系统的基本机制。
	System.Security.Cryptography	编码及解码服务, 包括数据的编码、解码、随机数生成、消息认证、数字化签名的支持。
数据	System.Data	访问、管理数据和数据源。

(续表)

分类	命名空间	功能
框架服务	System.Xml	处理 XML 支持。
	System.Xml.Serialization	对象到 XML 的双向映射。
	System.Diagnostics	跟踪调试代码支持。
	System.DirectoryServices	访问活动目录。
	System.Management	服务与应用程序管理工具。
	System.Messaging	微软消息队列 (MSMQ) 的访问与管理, 消息的接收与发送。
	System.ServiceProcess	安装、执行基于 Windows 的服务程序, 不能访问特定服务, 诸如 Active Directory、Web Service。
	System.Timers	定时器、其他更复杂的应用程序时间调度。

在使用时, 编程开发人员在应用中添加所需基类库的引用, 然后就可以使用这个类库中的所有方法、属性等等。跟传统的 Windows 编程相比, 使用和扩展基类库非常容易, 从而可以高效、快速地构建基于下一代互联网的网络应用。

3. ADO.NET 和 XML

绝大多数应用程序都需要访问数据, 从简单的文本文件到大型关系型数据库。在 .NET 技术框架中访问数据库的技术是 ADO.NET。

ADO.NET 提供一组用来连接到数据库、运行命令和返回记录集类库。在 .NET 框架中, ADO.NET 增加了对 XML 语言的支持。ADO.NET 还引入了数据集 (DataSet) 的概念, 可以驻留在内存的数据缓冲区中, 提供对数据的关系型视图。这样, 无论是关系型数据库, 还是 XML 文档, 都可以使用统一的编程模型来创建和使用它, 从而提高了程序的交互性和可扩展性, 尤其适合于分布式的应用场合。ADO.NET 还引入了一些新的对象, 如 DataReader 可以用来高效率地读取数据, 产生一个只读的记录集等等。

ADO.NET 通过一系列新的对象和编程模型, 并与 XML 紧密结合, 使得在 Microsoft .NET 中的数据操作十分方便和高效。

4. ASP.NET 和 Windows 窗体

ASP.NET 是 Microsoft .NET 中的网络编程结构, 它可以方便和高效地建造、运行和发布网络应用。在 ASP.NET 中可以支持多种语言, 支持脚本语言和通用语言运行时支持的所有语言, 并将代码和内容分开, 从而提高开发效率。在 ASP.NET 中通过引入服务器端控件, 可以快速开发网络应用, 提高构建网络应用的效率, 同时服务器端控件是可扩展的, 编程开发人员可以根据需要构造服务器端控件。

ASP.NET 应用不再是解释脚本, 而是编译运行, 再加上灵活的缓冲技术, 从根本上提高了性能。由于 ASP.NET 的应用框架基于通用语言运行时, 发布一个网络应用, 仅仅是一个拷贝文件的过程, 即使是组件的发布也是如此, 更新和删除网络应用, 可以直接替换/删除文件。编程开发人员可以将应用的配置信息存放到 XML 格式的文件中, 管理人员和编程开发人员对应用管理和开发可分开进行, 它提供了更多样的认证和安全管理方式, 并在可靠性等多

方面都有很大提高。

传统的基于 Windows 窗体 (Windows Forms) 仍然是 .NET 战略中不可或缺的一部分。在 .NET 中开发传统的基于 Windows 的应用程序时,除了可以利用现有的技术如 ActiveX 控件以及 Windows 接口外,还可使用 ADO.NET 和 Web Service 等。Windows 窗体为 .NET 框架提供了美观的客户端图形用户界面。

Visual Studio .NET 包括 C#、Visual C++、Visual Basic 等编程语言开发工具,通过公共语言规范 (Common Language Specification) 实现代码复用。Visual Studio .NET 提供了一套简化联合设计开发解决方案的技术支持,同时配以强有力的 MSDN 这一重要的软件开发所需的电子参考手册,将使得企业应用级的程序员有如鱼得水的感觉。同时也使得初级程序员能够较快地掌握解决开发方案的思路,并尽快地上手和应用编程。

1.1.2 Visual C++ .NET 新特点

Visual C++ .NET 同 Visual C++ 6.0 相比,具有许多新的特点,主要有:

1. 以 .NET 框架为目标

Visual C++ .NET 着重提供一种有力而简单的途径来设计一个以 .NET 框架公用语言运行时 (CLR) 为目标的应用程序。当使用 Visual C++ .NET 开发代码时,可以管理代码来编译应用程序或组件,生成一种 CLR 支持的代码。

2. 采用自然代码编程

除了支持新型的管理代码模式,Visual C++ .NET 还显著增强了采用自然 C++ 代码进行程序开发的能力。包括改良的程序库、更快更小的执行体、对于 Windows 和 Web 开发的支持。

3. ATL 服务器

ATL 服务器是一组类和集成开发环境 (IDE) 的特性,可以用来帮助用户建立服务器应用,特别是基于 Web 的 ISAPI 应用。在自然 C++ 代码中,如同其他 ATL 类一样,ATL 服务器类被设计用来为应用程序中应用最为普遍的组成部分提供高性能代码。

ATL 服务器 ISAPI 类在 ISAPI 的 EXTENSION_CONTROL_BLOCK 顶端提供一个 C++ 薄层,作为分隔静态和动态内容的模型。它使用服务器响应文件 (SRF)、一些 cookies 和上载文件的高度抽象以及请求和响应对象。

ATL 服务器提供一些服务器类,包括线程池、缓冲类、性能监视助手、加密服务、SMTP 邮件客户功能等类,用来结合配置或独立配置。

4. 新型的集成调试器

Visual Studio .NET 为所有的 Visual Studio 语言提供一个单一集成调试器。一个新的统一的界面整合了旧版 Visual C++ 和 Visual Basic 调试器,其特别为:可使用 C#, C++ 或 Visual Basic 方式跨越多种语言,并由自然 C++ 代码 (或脚本) 到管理代码进行单步调试;绑定并调试已经运行的进程;调试多线程和多程序。

5. 编译器转换开关

Visual C++ .NET 编译器包括一些新的简单的开关,使得用户调试并解决问题的速度快于以往,另外使得代码更加紧密和轻巧。例如:能够通过使用 /clr 实现 C++ 的管理扩展;使用