

管理信息系统开发案例系列丛书

Visual C++ .NET

管理信息系统开发案例

武新华 安向东 编著
杜虹 魏志勤

典型的开发案例

使你学习 Visual C++ .NET 能尽快上手
更是你毕业设计的第一手参考资料

西安电子科技大学出版社

<http://www.xduph.com>



管理信息系统开发案例系列丛书

Visual C++.NET 管理信息系统 开 发 案 例

武新华 安向东 编著
杜 虹 魏志勤

西安电子科技大学出版社

2005

内 容 简 介

本书融知识性、实用性于一体,通过详细介绍5个利用 Visual C++.NET 语言进行管理信息系统开发的典型实例,全面讲解了使用 Visual C++.NET 语言进行程序设计的方法和步骤。书中开篇就以实例构成,将 Visual C++.NET 数据库及系统的开发渗入到实例设计中进行讲解。实例的讲解主要分为需求分析、功能设计、数据库结构设计、模块设计、系统开发过程、相关开发技术及最终运行测试等几个方面,并在讲解中提供编译好的安装程序与源程序供读者参考。

本书实例丰富,内容详实,涵盖了 Visual C++.NET 语言编程的难点和热点。本书适于作为大中专院校计算机与相关专业的 Visual C++.NET 语言的教学辅导或参考书,也可以作为 Visual C++.NET 语言培训班的教材和从事管理信息系统设计人员的参考资料,更是广大毕业生进行毕业设计的第一手参考资料。

图书在版编目(CIP)数据

Visual C++.NET 管理信息系统开发案例 / 武新华等编著.

—西安:西安电子科技大学出版社,2005.3

(管理信息系统开发案例系列丛书)

ISBN 7-5606-1487-6

I. V… II. 武… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 001453 号

策 划 李惠萍

责任编辑 李惠萍

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2005年3月第1版 2005年3月第1次印刷

开 本 787毫米×1092毫米 1/16 印张 19.75

字 数 467千字

印 数 1~6 000册

定 价 26.00元

ISBN 7-5606-1487-6/TP·0790

XDUP 1758001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

本书是一本关于如何使用 Visual C++.NET 编程语言进行数据库管理信息系统应用实例开发的指导教程。书中通过典型实例全面介绍了 Visual C++.NET 数据库管理信息系统的基础知识。其主要内容包括：Visual C++.NET 数据库编程基础，表和数据库的操作，数据库管理程序的设计，数据库的查询，报表和标签的设计，面向对象程序设计，控件、表单、菜单设计，以及类的设计方法。全书以“应用案例→分析如何解决问题→处理实际问题”为主线，重点在于培养数据库管理信息系统开发者的编程能力和解决实际问题的分析、设计能力。

为照顾初学者，作者特意对书中内容进行了精心编排，尽力做到以图例讲解代替大段枯燥的代码和文字说教，使得各个层面的读者，甚至是那些从未接触过管理信息系统开发和没有太多 Visual C++.NET 数据库编程语言基础的读者也能够在阅读完本书后轻松入门。

为进一步方便广大读者的阅读，本书完全采用图例步骤式的讲解方法，使得图文能够紧密结合，尽可能地减少了长篇累牍的文字说明，理论讲解深入浅出，同时强调应用技能的快速掌握。阅读本书您一定能感受到本书不仅简单易读，而且具有很强的实践性。显然，本书无疑是提高广大管理信息系统开发者知识水平与设计、开发技巧的一本不可多得的工具书。

我们编写本书的初衷不是单纯地教会大家应该如何进行某一个管理信息系统的开发，而是要大家尽可能多地掌握数据库管理信息系统的开发精髓，了解别人的编程思路，通过分析这些不同管理信息系统开发的程序原理，从中充分体会到 Visual C++.NET 的编程思想，掌握其中的设计理念和实现技巧，并且做到举一反三，最终达到能够根据自己的实际情况和设计需求，开发出适合具体需求的管理信息系统。

我们相信，有这样一本书置于您的案头，那许许多多在过去看来很难实现的事情，现在您会突然发现原来它们是那么的触手可及。

本书不仅可以作为大中专院校相关课程的教学辅导书，也可作为 Visual C++.NET 爱好者及培训班的参考书，更是相关专业毕业生进行毕业设计或从事管理信息系统开发设计人员的第一手参考资料。

本书由几位经验丰富的高校教师编写，同时也得到了众多网友的支持，在此一并表示衷心的感谢。本书的编写情况是：武新华负责第 1、3、5 章，魏志勤负责第 2 章，杜虹负责第 4 章，李建伟负责第 6 章，安向东负责第 7 章，最后由武新华、安向东统审全稿。

我们虽满腔热情，并对全书内容反复锤炼，几经改写，但限于自己的水平，书中的疏漏之处难免，欢迎广大读者批评指正。

武新华

2005 年元月

目 录

第 1 章 Visual C++.NET 系统开发基础	1
1.1 Visual C++.NET 的新特性	2
1.1.1 什么是 C++托管扩展	2
1.1.2 Visual C++与 Visual C #的区别	3
1.1.3 为何选用 Visual C++.NET 开发语言	4
1.2 Visual C++.NET 的集成开发环境	5
1.2.1 Visual C++.NET 的启动界面	6
1.2.2 Visual C++.NET 的解决方案管理器	7
1.2.3 工具箱与浮动条	8
1.2.4 Visual C++.NET 的代码编辑窗口	9
1.2.5 资源视图和类视图	13
1.2.6 Visual C++.NET 的属性窗口	14
1.2.7 服务器资源管理器和注释窗口栏	16
1.3 Visual C++.NET 程序的调试与错误处理	17
1.3.1 创建一个简单的调试项目	18
1.3.2 更正编译过程中产生的错误	20
1.3.3 为正在调试的程序加入断点	21
1.3.4 步进执行代码的两种方法	22
1.3.5 监视与更改变量的值	23
1.4 Visual C++.NET 程序的编译与运行	24
1.4.1 生成项目的可执行文件	25
1.4.2 运行已经编辑好的程序	25
1.4.3 使用控制台应用程序编写普通 C++程序	26
第 2 章 房产楼盘信息查询系统	27
2.1 系统开发的思路分析	27
2.2 创建相应的数据库结构	28
2.3 程序框架和用户界面	29
2.3.1 创建 MFC 应用程序	29
2.3.2 用 AfxOleInit()来初始化 COM 库	32
2.3.3 创建 Connection 对象并连接数据库	33
2.3.4 执行 SQL 命令并取得结果记录集	36
2.3.5 记录集的遍历与更新	41

2.3.6 保存与载入数据的设计实现	46
2.3.7 实现响应用户操作	47
2.3.8 释放 ADO 对象	50
2.4 ADO 技术的高级应用	50
2.4.1 通过 ADO 进行数据源的连接	50
2.4.2 出错情况下错误信息的获取	54
2.4.3 使用 UDL 文件创建 ADO 连接	54
2.5 相关开发技术	56
2.5.1 Visual C++.NET 数据库开发技术的比较	56
2.5.2 Visual C++.NET 中的 ADO 技术	59
2.5.3 在 Visual C++.NET 中实现对 ADO 的操作	62
2.6 运行最终的设计结果	67
第 3 章 通用仓库管理信息系统	69
3.1 系统开发的思路分析	69
3.2 创建系统所需的数据库	70
3.3 系统应用程序的创建	76
3.4 设计系统中的各应用模块	77
3.4.1 登录窗口的设计和程序初始化	78
3.4.2 创建系统的主对话框	79
3.4.3 操作日志模块的设计	80
3.4.4 设备代码管理窗口的设计	82
3.4.5 创建库存信息管理窗口	89
3.4.6 设备入库信息管理窗口的设计	91
3.4.7 创建本系统的设备入库信息窗口	96
3.4.8 设备采购报表管理的设计	99
3.4.9 设计系统的库存报警管理窗口	101
3.4.10 设计系统的帮助模块	104
3.5 相关开发技术	105
3.5.1 用 Visual C++ 进行 COM 编程的技术基础	105
3.5.2 Visual C++ 中的 COM 包装	109
3.5.3 在 Visual C++ 程序中接收 COM 事件	112
3.6 工程实例的最终运行测试	115
第 4 章 远程数据库信息查询系统	117
4.1 系统开发的思路分析	117
4.2 程序框架和用户界面设计分析	118
4.2.1 创建程序的框架	118
4.2.2 实现窗口的拆分	121

4.2.3	添加相应的框架资源	123
4.2.4	DAO 框架的创建	125
4.2.5	实现数据库在 CDaoTreeView 中的结构显示	127
4.2.6	在 CDaoListView 中实现数据显示	136
4.2.7	使用 CSocket 添加网络传输功能	143
4.2.8	使用 Socket 传输结果集	148
4.3	通过自定义消息控制远程数据库	153
4.4	相关开发技术	156
4.4.1	DAO 用法简介	156
4.4.2	利用 MFC 进行 DAO 数据库编程	157
4.4.3	WinSocket 网络编程简介	160
4.4.4	利用 MFC 建立网络通信的过程	166
4.5	工程实例的最终运行测试	167
第 5 章	公司考勤管理信息系统	169
5.1	系统开发设计的思路分析	169
5.2	创建系统所需的数据库	170
5.3	系统应用程序框架的生成	172
5.4	创建系统中相应的各功能模块	174
5.4.1	生成系统的登录认证对话框	174
5.4.2	创建系统的主对话框	176
5.4.3	上下班时间设置对话框的设计实现	179
5.4.4	创建系统的考勤修改对话框	181
5.4.5	生成系统的修改出勤记录属性页	183
5.4.6	生成系统的加班记录属性页	189
5.4.7	生成系统的请假记录和出差记录属性页	193
5.4.8	生成系统的考勤统计对话框	196
5.5	相关开发技术	205
5.5.1	ODBC 的部件组成	206
5.5.2	在系统中配置 ODBC 数据源	207
5.5.3	MFC 对 ODBC 的支持	209
5.5.4	ODBC 的数据库类	210
5.5.5	MFC 的 ODBC 数据库编程	215
5.6	工程实例的最终运行测试	218
第 6 章	出租车信息定位管理系统	220
6.1	系统开发的思路分析	221
6.2	程序框架和用户界面设计分析	222
6.2.1	加入串口通信控件	222

6.2.2 串口操作类的创建	223
6.2.3 解码数据包的创建	227
6.2.4 串口控件的收发设计	232
6.3 如何进行串口调试	242
6.4 相关开发技术	245
6.4.1 串口通信基础知识	245
6.4.2 Visual C++对多线程的支持	247
6.4.3 实现串口通信的两种方法	250
第 7 章 网络信息过滤管理系统	258
7.1 系统开发的思路分析	258
7.2 创建系统案例的程序框架	259
7.3 编制一个标准的浏览器	261
7.3.1 创建本浏览器的工具栏	262
7.3.2 用下拉列表框创建地址栏	268
7.3.3 创建状态栏和动画控件	270
7.3.4 调用活动栏 (CReBar) 控件	271
7.3.5 创建下拉菜单	273
7.3.6 为系统添加 Favorite 菜单	274
7.4 使用 CHtmlView 实现浏览功能	280
7.5 定制网络信息浏览过滤的设计实现	285
7.5.1 过滤 IE 的弹出式广告窗口	285
7.5.2 记录用户浏览的网站地址	286
7.5.3 阻止用户浏览不良网站	287
7.5.4 利用 CHtmlView 制作精美用户界面	288
7.6 相关开发技术	292
7.6.1 WinInet 技术简介	292
7.6.2 编写 HTML 源码浏览对话框	293
7.6.3 WinInet 相关类的介绍	301
7.6.4 使用 WinInet 类的几点学习体会	304
7.7 工程实例的最终运行测试	306



第 1 章

Visual C++.NET 系统开发基础



本章主要介绍下列内容：

- ◆ Visual C++.NET 的新特性
- ◆ Visual C++.NET 的集成开发环境
- ◆ Visual C++.NET 程序的调试与错误处理
- ◆ Visual C++.NET 程序的编译与运行

Visual C++作为微软为 Windows 系统量身定做的超级开发语言，无疑是 C++各种版本的语言之中最优秀的。在强大的 Windows 系列操作系统的支持下，Visual C++已经成为用户最多、功能最强的程序开发语言之一。

尽管学习、掌握甚至精通 Visual C++是每个 Windows 程序员的目标，但要想学好 Visual C++的编程却并不简单。笔者就曾经阅读过许多讲授 Visual C++的工具书，却总是感到不得要领，许多看似明白的函数和控件到了真正使用时却不知道从何下手。更有甚者，在读者进行了大量的阅读和实验之后，却发现自己仅仅还是停留在编写“Welcome”或“Hello World”的水平。

笔者认为，学习 Visual C++就是要力求能窥见 Windows 操作系统的内部结构；能够欣赏精制的 Windows 消息循环机制；能灵活运用 Windows 提供的各种各样的强大工具。因为只有达到这种水平，才能够有信心拍拍自己的胸脯说：“我学会了 Windows 编程。”

其实，要想学好编程最好的办法就是分析典型的源代码，并且对其进行改进，直至最终写出自己的程序，这同时也是很多编程高手的经验。然而当真正拿到源程序时，却发现漫天盖地都是代码，初学者根本就不知从何下手；尤其对于 Visual C++的 MFC 程序，几乎 60%的代码都是机器自动生成的，哪些该看、哪些不用看都是初学者头疼的问题。

大家可能注意到了，微软已经进入了 .NET 时代，MFC 时代大约就快过去了(Visual Studio.NET 2003 是 Microsoft 的第二代开发工具，用于构建和部署 Microsoft .NET 的软件，在这个最新平台上，包括了 Visual C++、Visual Basic 和 Visual C#等开发工具)，但这同时也是一个莫大的机遇。首先，MFC 的框架还可以在 Windows 环境下继续使用，而且还推出了新的 MFC7.0 版本；另一方面，.NET 框架组件又提供了功能更为强大的编程组件，Visual C++程序员可以不用学习太多东西就能够使用这些优秀的组件，这就使得用 Visual C++开发程序变得更加方便、快捷。

综上所述，Visual C++.NET 无论从速度还是从安全方面考虑都具有非常广阔的市场，它不仅是 Windows 桌面开发与 .NET 框架网络开发的最佳利器，而且“进”可以用托管扩展语言开发 .NET 框架程序，“退”可以用原有的 Visual C++语言与 MFC 联手开发桌面程序。



1.1 Visual C++.NET 的新特性

Visual C++.NET 包括许多新的特性,如 Web 应用程序、托管 C++扩展、ATL、DCOM、MFC 等方面的增强以及 ATL Server、Windows .NET Framework 的支持等。

尽管如此,Visual C++在 Windows 桌面应用程序上的编程方法仍可分为两类:

(1) 非 Window 编程。如控制台程序(一种与早期 DOS 相兼容的程序,用来学习基本 C++的编程思想)。

(2) Window 编程。这一类编程方法又可分为两种。一种是直接调用 Windows 提供的 Win32 API(应用程序接口)函数开发 Windows 应用程序;另一种是使用 Visual C++所提供的 MFC(Microsoft Foundation Class, 微软基础类)库中的“类”及其功能函数。

由于 MFC 对 Win32 API 编程方式的极好封装,使得用户不需要理解 Windows 内部的编程机制就可以通过“应用程序向导”创建相应的应用程序框架,这些框架能满足绝大多数开发者的需求,并且使得用户不必添加太多的代码就能够实现最常用的程序功能。因此,MFC 是 Visual C++中影响最深的一种编程方式,并且这个编程方式将会一直存在下去。

微软 .NET 的本质是全面支持各种面向网络体系结构的应用,其中最主要的是 Web 服务、Web 应用和多层分布式计算等。Visual C++.NET 是 Visual C++ 6.0 的后续版本,也称为 Visual C++7.0。笔者在这里提醒读者千万不要被其华丽的名称所迷惑,也不要怀疑新版本对以往 Visual C++程序的兼容性。因为 Visual C++.NET 仍然是在 Windows 系统下的应用软件开发工具,只不过其名称更加商品化了。与以往版本最大的不同就是 Visual C++.NET 有一个重要的功能:全面支持 .NET 平台,尤其是网络应用系统的高效率开发。

在新一版的 Visual C++中,除了对原有的界面和工具做了较大的调整之外,新增的托管扩展代码语言无疑是 Visual C++.NET 的最大特点。

1.1.1 什么是 C++托管扩展

C++托管扩展是一套帮助 Visual C++开发者编写 .NET 框架程序的 C++语言扩展。托管扩展允许用户在同一个程序里同时使用托管代码和非托管代码。用托管代码编写的程序可以兼有非托管代码和托管代码的优点,用托管扩展可以很方便地将现有的组件封装为 .NET 框架组件,从而在将其整合到 .NET 平台时保护已有的投资。

托管扩展是下面所述情况下的最好选择。

1. 快速将现有的非托管 C++程序移植到 .NET 框架下

如果开发者有一个现成的 C++程序,托管扩展提供了平滑地将其移植到 .NET 框架下的手段,因为它可以在同一个程序里甚至同一个文件里混合使用非托管代码和托管代码。可以一个一个地把组件移植到 .NET 框架下,或者可以继续使用非托管 C++编写组件,利用 C++语言的强大和高度的适应性,然后用托管扩展写出简洁、高效的代码并将它封装起来,使 .NET 框架组件能调用它。

2. 从支持 .NET 框架的语言访问 C++组件

托管扩展支持从任何一种支持 .NET 框架的语言访问 C++类,这就提供了这样一种可能



性：用户可以通过托管扩展编写简单的封装，使自己的 C++ 类和方法表现得像托管类。而封装是完全的托管类，可以被任何一种支持 .NET 的语言调用。封装类作为映射层工作在托管类和非托管 C++ 类之间，它只是简单地直接将对方法的调用传递给了非托管类。正如调用非托管类一样，托管扩展支持调用任何非托管的动态连接库或静态库。

3. 从非托管代码中访问 .NET 框架的类

通过托管扩展，用户可以从 C++ 代码中直接创建和调用一个 .NET 框架的类。在写 C++ 代码时可以像对待托管 C++ 类一样对待 .NET 框架组件。

4. 在同一个可执行文件中的托管代码和非托管代码的转换

在托管代码和非托管代码之间，Visual C++ 编译器能自动对数据、指针、异常和控制流进行转换，这就使得托管代码可以无缝地与非托管 C++ 代码一起进行工作。

1.1.2 Visual C++ 与 Visual C #的区别

微软为了推行 .NET 战略，特别为 .NET 平台设计了一种语言——C#，并称 C# 是由 C 和 C++ 派生而来的一种“简单、流行、面向对象、类型安全”的程序设计语言，其综合了 Visual Basic 的高效率和 C++ 的强大功能。然而更多的人感觉 C# 更类似于 Java。事实上，C# 融合了大量的 Java 思想，C# 是 .NET 的关键性语言，它是整个 .NET 平台的基础。与 C# 相比，.NET 所支持的其他语言显然是配角身份，包括 Visual C++.NET 在内。

但是微软并没有打算放弃 Visual C++.NET，相反微软对 Visual C++.NET 有着另一番独特的打算。Visual C++.NET 的定位与 C# 不完全重合，Visual C++.NET 的应用范围仍强于 C#，这一点无论对于微软公司还是软件业的应用现状都非常重要。

Visual C++.NET 的重要使命是在尽可能支持 .NET 计划的同时，还要继续强化自身功能，保证传统 Win32 下应用系统开发的绝对权威。这也就让人理解为什么 Visual C++.NET 被微软设计成惟一的一个能够产生优化 MSIL (Microsoft Intermediate Language, 微软中间语言) 指令的语言，也是惟一的一个可以在同一个映像中包含受控和不受控代码的语言。

Visual C++.NET 的现在和未来都将在微软的战略中扮演最重要的角色，Visual C++.NET 才是 .NET 的系统语言。为了适应这些新的要求，Visual C++.NET 较 Visual C++ 6.0 有着许多重要的改变与增强。

1. 集成开发环境

集成开发环境 (IDE) 整合了基于服务端开发模式和大大改进的 C++ 编译器。同时，MFC 和 ATL 也都做了重要的改进，许多公用的类被二者共享。Visual C++.NET 需要一个新的 MFC 支持库 MFC70.DLL，这个支持库不再与 MFC42.DLL 兼容，但是源程序依然可以让 Visual C++.NET 和 Visual C++ 6.0 兼容。在 Visual C++.NET 中，消息映射机制变得更加安全，但也要对某些代码稍作修改。

在 Visual Studio.NET 一整套开发平台中，Visual C++.NET 是惟一需要有额外支持库的语言，即需要 MFC 支持库 MFC70.DLL，这样 Visual C++.NET 就既可以开发出面向 .NET 的应用系统，也可以开发出传统的基于 Win32 和 MFC 的应用系统。

2. 添加了新类

在 Visual C++.NET 中，系统的头文件 (.h) 都根据最新的 SDK 平台做了相应的修改，增



加了支持 Windows 2000 和 Windows XP 中的相关特性。MFC 和 ATL 中在许多领域也增加了一些实用的类,例如,在安全方面(增加了 CDac1、CSac1、CSid、CAccessToken、CToken Groups、CTokenPrivileges、CSecurityDesc、CSecurityAttributes 等多个类)、Web 服务方面、数据库访问方面等,尤其是通过采用新的 OLE DB 类和属性使访问数据库的速度与性能获得大幅度提高。

ATL7.0 中增加了 5 种类模板:CAtlArray、CAtlList、CAtlMap、CRBMap、CRBMultiMap。

为了避免像千年虫那样的错误发生,Visual C++.NET 对时间和日期采用了 64 位记法,在类 CTime 和 CTimeSpan 中也增加了支持 64 位记法的方法,例如 CTimeSpan::GetDays64 和 CTime::GetTime64。

另外,Visual C++.NET 也支持线程池(Thread Pools)机制,通过线程池来提高应用服务器的响应能力。为此,Visual C++.NET 在 MFC7.0 中提供了 CThreadPool 类。

3. GDI+及相关 API

GDI+及其相关 API 提供了大量新的图形技术支持函数,不过这套 API 函数仅支持 .NET 平台下的应用。为了使用 GDI+,需引用头文件 gdiplus.h 和链接库 gdiplus.lib。GDI+与传统的 GDI 编程模型不大相同,不再用选择刷子和笔到某个设备环境 DC(Device Context),而是可以将特定的笔和刷子传递给任何一个画图命令,而且笔和刷子也不再像以前那样总是绑定在一起,现在可以通过不同的函数分开使用它们。

值得关注的是在 GDI+中,Visual C++.NET 提供了一组支持 DHTML 编辑的类以及 DHTML 和对话框体类(CDHtmlDialog、CmultiPageDHtmlDialog),这些丰富的控件和函数使得高效开发 Web 应用成为可能。

4. Web 服务及应用

在 Web 服务、Web 应用和多层应用方面,Visual C++.NET 也在 MFC7.0 中提供了一系列的类来建立 Web 服务,高效率生成 HTML 代码,支持轻量级的 HTTP 客户端,支持 SMTP 和 MIME 编码的消息,支持发送、接收电子邮件等。

微软的 .NET 平台主要是针对网络,.NET 的精髓就在于通过基于 XML 的 SOAP 远程过程调用并通过 UDDI/WDSI 自动发现网络服务器,来实现整个分布式网络应用系统。而 Visual C++.NET 在这一点上与其他 .NET 开发工具相比毫不逊色,相反可以利用的手段更多。

因此,虽然 Visual C++.NET 在微软的 .NET 计划中并不是被大力宣传的部分,但它的地位仍然不能被撼动。微软更希望向一般开发者和最终用户兜售它的 .NET 与 C#,微软希望大多数视窗应用软件的开发转向 CLR 和 .NET 平台,而 C#似乎成为这个 .NET 平台中无所不能的面向对象的网络语言,但事实上 Visual C++.NET 仍然是 Windows 系统中最锐利的武器,利用 Visual C++.NET 或许还能够透过已经被微软高度操作封闭的 CLR 窥见 Windows 的一些本来面目,而惟有这样,才能更有利于深入了解及把握操作系统的现在与未来。

1.1.3 为何选用 Visual C++.NET 开发语言

首先需要明确的一点是:.NET 是为 Client/Server 和基于 Web 的应用程序设计的。如今,应用程序越来越趋向于 Client/Server 架构,今后还会有更多的应用程序看起来更像是浏览器,.NET 架构正是顺应了这样的潮流。



需要明确的一点是：.NET 是为 Client/Server 和基于 Web 的应用程序设计的，它并不适用于桌面的应用程序，特别是当大家都很关心保护自己知识产权的时候更显出其局限性，因为在桌面上使用 .NET 不能可靠地保护用户的代码。如果开发者是一个 Visual Basic 或 C# 的开发人员，则不可能在 .NET 中写出受到完全保护的代码。

因为这个限制，如果想保护自己的桌面程序代码，就需要用 Visual C++ 写代码。在 .NET 框架下，保护知识产权的惟一方法就是把自己的非托管 C++ 代码封装成控件或 COM，然后在托管代码中调用，但是这并不适用于 Visual Basic 或 C# 程序员。

综上所述，Visual C++.NET 无论从速度还是安全方面考虑，都是具有广阔市场的。而且以目前的情况，从桌面应用到完全的 Internet 应用，完全实现 .NET 的 Client/Server 架构，完全达到微软提出的“软件服务”模式还差得很远，无论从技术上还是市场上都有很多的障碍，这种变化可能要经过好几年的时间。所以桌面应用程序的开发还是有非常广阔的前景的，而 Visual C++.NET 无疑是桌面开发与 .NET 框架网络开发的最佳利器。

1.2 Visual C++.NET 的集成开发环境

为了配合微软的 Microsoft .NET 架构以及其在框架上的革命性软件开发计划，Visual C++.NET 对开发界面做出了很大的调整，各个开发工具的组织形式和样子与原来都有很大不同。当然，还诞生了一些新的功能强大的开发工具。

下面就来全面介绍一下 Visual C++.NET 的全新开发界面。

如图 1-1 所示，Visual C++.NET 的开发界面大体由以下几部分组成：代码编辑区、服务器资源管理器、工具箱、类视图、属性窗口、解决方案资源管理器和动态帮助区等。

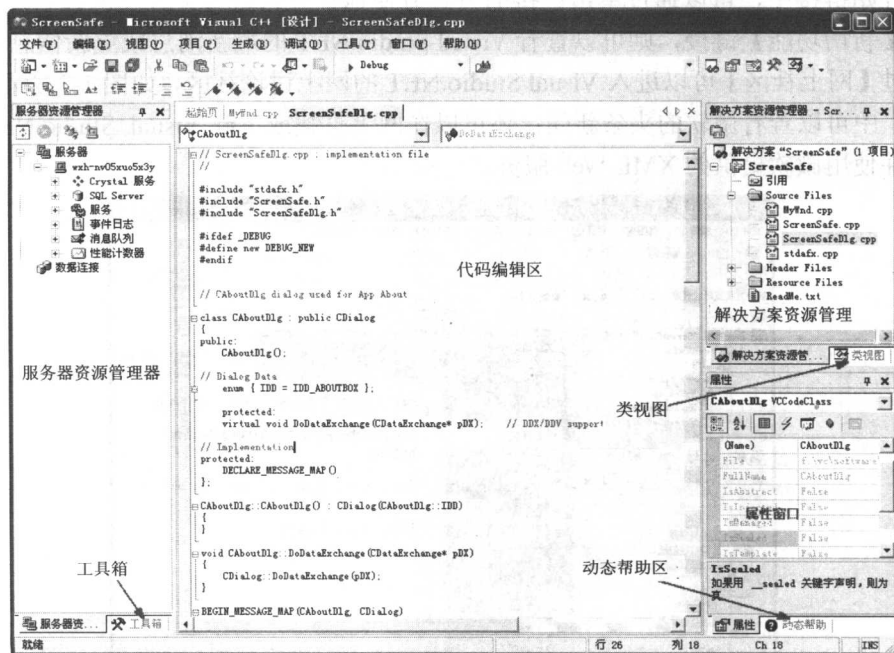


图 1-1 Visual C++.NET 的开发界面



正是由于这些工具的协同工作,才使得用户可以更清楚地查看源代码,更方便地创建程序框架,更全面地掌握程序资源,更容易地插入程序组件和更快速地得到在线帮助。总之,利用 Visual C++.NET 的开发界面将使用户从浩如烟海的代码中解放出来,从而更容易地开发出功能强大的程序。

1.2.1 Visual C++.NET 的启动界面

每次启动 Visual C++.NET 时,就会进入 Visual C++.NET 的启动界面,如图 1-2 所示。这个启动画面实际上就是启动时默认进入的主页,在 Visual C++.NET 中正是通过这种很流行的网页浏览形式组织信息的。

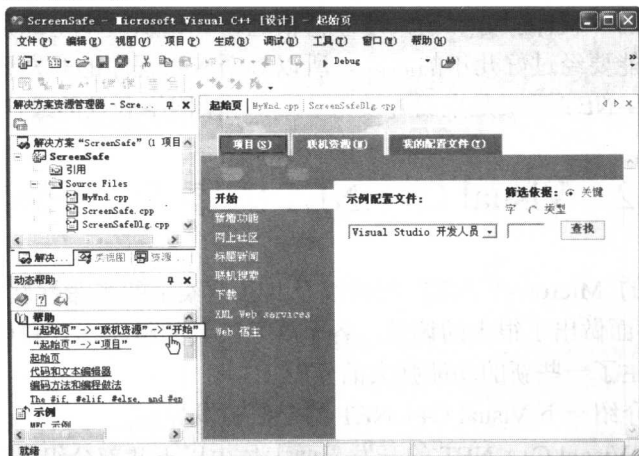


图 1-2 Visual C++.NET 的启动界面

在开始链接中,可以通过单击链接的方式方便地打开最近曾经开发的项目。如果点击左边的【新增功能】链接,则可以查看 Visual Studio.NET 的最新消息或更新信息。

通过【网上社区】可以进入 Visual Studio.NET 的网上讨论社区,如图 1-3 所示。在【标题新闻】中可以查看微软的头条新闻;还可以在网查找或下载 Visual Studio.NET 的相关资源,并使用微软提供的 XML Web 服务。

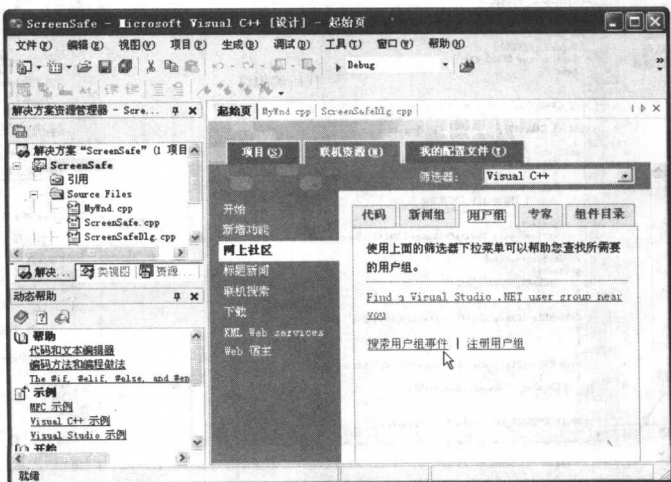


图 1-3 进入 Visual Studio.NET 的网上讨论组



最后,还可以在【我的配置文件】中设置在 Visual C++.NET 中使用何种风格来布局各种开发工具和快捷键。例如可以在这里设置沿用 Visual C++ 6.0 时的窗口布局和快捷键,这样可以使 Visual C++ 的老用户们不致于较大改动自己的编程习惯。如图 1-4 所示。

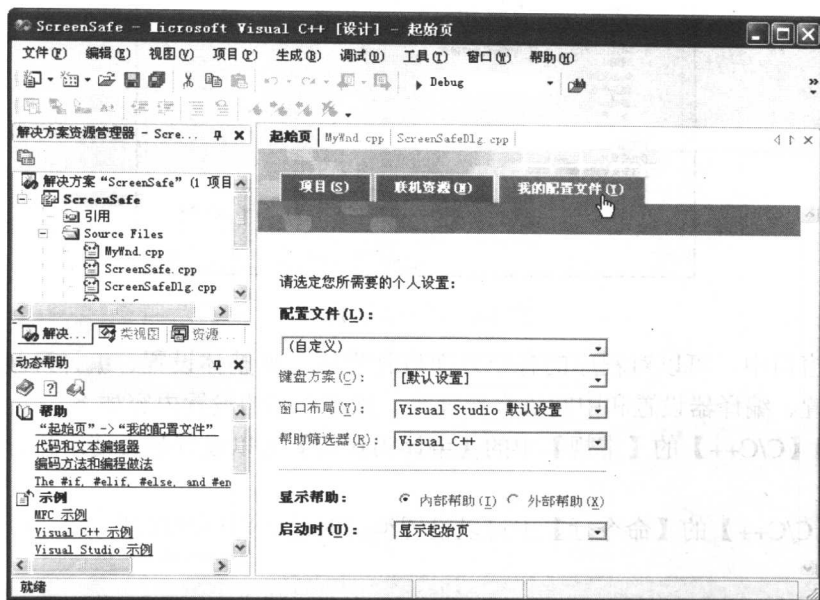


图 1-4 设置 Visual C++.NET 的界面

同时,在这里还可以设置是使用内部浏览器查看帮助还是使用外部浏览器,还有 Visual C++.NET 的起始界面是显示启动网页还是打开前一次的项目等。

1.2.2 Visual C++.NET 的解决方案管理器

Visual C++.NET 的解决方案管理器(Solution Explorer)如图 1-5 所示,它包含用于创建应用程序的全部项目,这些项目可以是 Visual C++.NET、C#、Visual Basic .NET 等不同语言开发的项目(一个解决方案可以由几个组件项目和引用它们的用户界面组成)。

在解决方案管理器中,可以从树状视图中清晰地查看程序的实现文件、头文件和资源文件,而更重要的是在解决方案管理器中可以对各个项目进行整体设置。

设置的方法是在项目图标上点击右键,选择【属性】或直接点击左上方的工具图标就可以调出程序设置窗口,如图 1-6 所示。



图 1-5 解决方案管理器

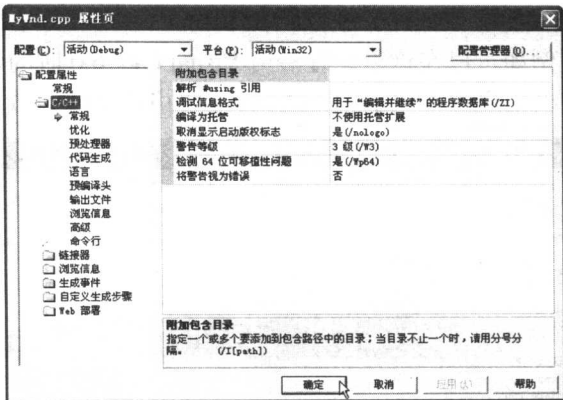


图 1-6 项目设置窗口

在这个窗口中，可以对程序的各个方面进行设置，如整体设置、编译环境设置、C++ 语言属性设置、编译器设置和中间语言设置等。比较常用的设置内容如下：

- 利用【C/C++】的【常规】中的【编译为托管】可以设置是否使用托管扩展来编译程序。
- 在【C/C++】的【命令行】中可以设置编译时的特殊编译选项，也可以设置要链接的库文件。
- 利用【链接器】的【常规】中的【附加库目录】可以设置附加链接库文件的目录，如图 1-7 所示。

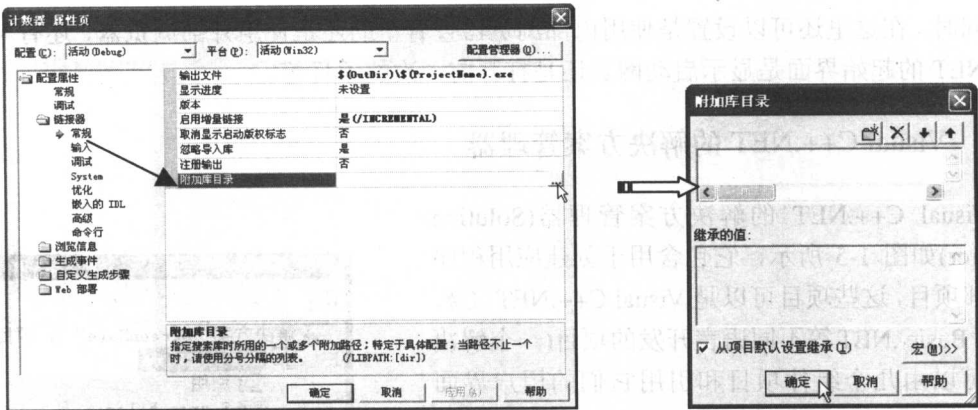


图 1-7 设置库文件目录

同时，在解决方案管理器的右键菜单中，还有很多实用的命令，如在项目中加入新的文件，编译单个实现文件，更改工程名等。

1.2.3 工具箱与浮动条

通过使用 Visual C++.NET 工具箱和浮动条可以大大提高开发人员的工作效率，尤其在设计对话框时，使用工具箱中的可视化控件无疑使开发变得十分方便。

在 Visual C++.NET 中，工具箱往往是浮动的，即只有当鼠标停留在浮动条的“工具箱”上，工具箱才会弹出来，而一旦完成操作，工具箱又会自动缩进去。这对于屏幕资源日益

