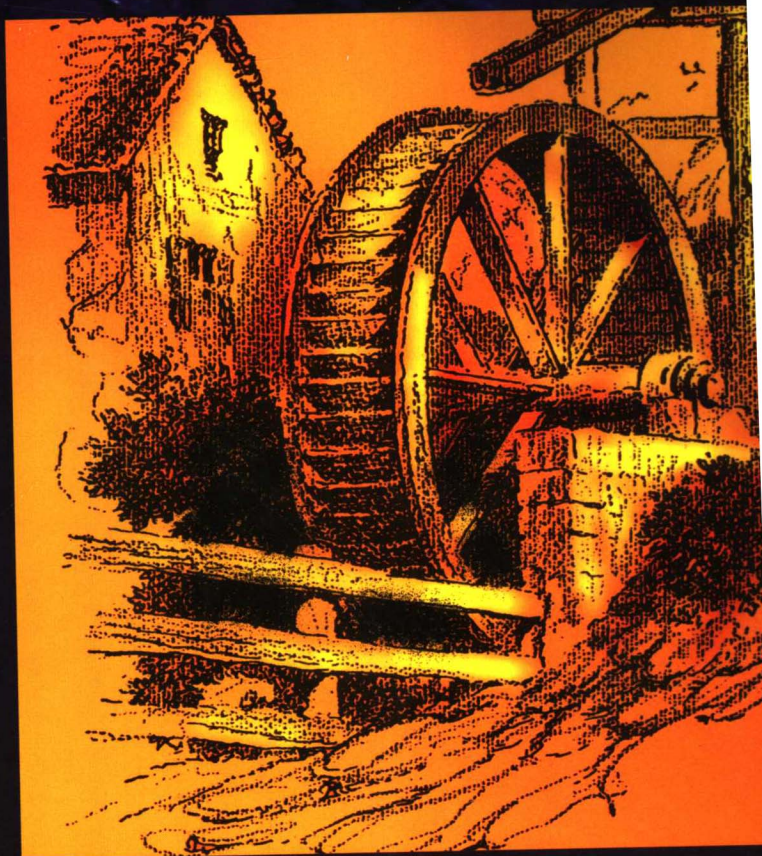
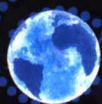




Visual C++ 项目开发实用案例

精选具有实用价值的工具软件开发案例
翔实介绍项目的设计开发过程
深入分析常用的开发技术

刘 瑞
吴跃进 编著
王宗跃



科学出版社
www.sciencep.com

Visual C++项目开发实用案例

刘 瑞 吴跃进 王宗跃 编著

科学出版社

北 京

内 容 简 介

本书由 Visual C++概述和“数据库查询工具”、“数据转换工具”、“图像处理系统”、“系统资源监测器”、“HTTP 日志解析、分析系统”五个案例构成。本书面向有一定 Visual C++编程基础的读者,不仅适合有志于从事软件开发的学生作为课程学习、毕业设计的参考资料,也适合于讲授 Visual C++课程的教师作为案例使用以丰富其讲课内容。对于刚刚开始从事 Visual C++开发的专业技术人员,也可以利用本书提高自己的技术水平。

本书配有光盘,其中提供了所有系统的源程序代码和相关文件,根据书中的讲述,可以建立开发环境,编译运行。同时,本书为了方便更多的初学者使用,详细介绍了 C#核心技术,以便读者可以深入浅出地学习。

本书适合使用 Visual C++进行产品和项目开发的人员,以及高等院校相关专业的师生。

图书在版编目(CIP)数据

Visual C++项目开发实用案例 / 刘瑞等编著. —北京: 科学出版社, 2006

ISBN 7-03-017177-2

I. V … II. 刘… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 040938 号

责任编辑:王淑兰 赵卫江 / 责任校对:耿耘
责任印制:吕春珉 / 封面设计:耕者设计工作室

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

铭浩彩色印装有限公司 印刷

科学出版社发行 各地新华书店经销

*

2006 年 6 月第 一 版 开本: 787×1092 1/16

2006 年 6 月第一次印刷 印张: 19 1/2

印数: 1—4 000 字数: 450 000

定价: 35.00 元 (含光盘)

(如有印装质量问题, 我社负责调换<环伟>)

销售部电话 010-62136131 编辑部电话 010-62130750

前 言

Visual C++是 Microsoft 公司推出的优秀的软件开发工具,即使在软件开发工具百花争艳的今天, Visual C++仍然占有不可动摇的地位。

Visual C++功能强大而全面,能够开发各类应用程序。它向程序员暴露了更多系统内部的实现细节,能够进行更为细致和全面的控制,进行更为底层的开发,其最终生成的代码更小,执行效率更高。对于那些已经掌握 Visual C++的人来说, Visual C++是无所不能的利器;对于那些刚刚开始学习 Visual C++的人来说, Visual C++却意味着困难和复杂,甚至会使人望而却步。

学习软件开发离不开实践,无论你阅读了多少手册和书籍,无论你记住了多少数据结构、算法或者函数的语法,如果不通过一个或者多个案例的设计、开发和调试,你永远都无法学会编程并成为真正的程序员。对于 Visual C++的学习同样如此。

在学习的方法上,不必求全责备,面面俱到,掌握分析、思考和解决问题的方法才是最关键的。当今 IT 技术的发展一日千里,新技术和新方法不断涌现。只有掌握了那些隐藏在语言、工具后面的技术发展的思想和脉搏,掌握软件开发的基本思路和方法,才能从容面对日新月异的变化,跟上时代前进的脚步。

本书就是基于上述思想而编写的。作者试图通过对几个案例的设计、分析和实现,向读者展示 Visual C++开发的思路和方法。在案例的选择上,作者选择的都是一些工具软件,因为这才是最能发挥 Visual C++优点的领域。在案例的规模上,如果规模太小,则复杂度低,锻炼效果有限;如果规模太大,对于初学的读者则难以入门,因此本书选择的案例程序大都在一二千行。

本书由 Visual C++概述和五个案例构成。

在 Visual C++概述中,我们简要地介绍了 Visual C++集成开发环境的构成, Visual C++应用程序开发的基本步骤,讨论了有关 MFC 和事件驱动编程的有关知识。

第一个案例“数据库查询工具”,讲述了一个能够连接多种数据源,并利用 SQL 语句对其进行操作的通用查询工具。介绍了拆分窗口和 TreeView 的开发技术,讨论了如何利用 ADO 和 ADOX 获取表、视图等数据库对象源数据以及操作数据源中数据的方法,展示了 VC 保存和读取文件的技术。

第二个案例“数据转换工具”,讲述了一个可以在多种数据源之间复制数据的数据库工具。介绍了向导程序的编写技巧和获取表结构详细定义的方法,讨论了如何解决不同数据源数据类型之间的转换,以及创建、删除表和复制表中数据的知识。

第三个案例“图像处理系统”,讲述了一个可以对位图等图像进行各种处理操作的图像工具。介绍了位图文件的格式,位图的存取和显示,位图的翻(旋)转、放大、缩小、去除彩色、转为负像等技术,以及如何将 PCX 图像文件转为位图文件,讨论了制作和控制工具条按钮的方法。

第四个案例“系统资源监测器”,讲述了一个可以对系统运行状态进行实时监测和

图形化展示的工具。介绍了 SNMP 协议，以及通过 SNMP 协议对系统中磁盘、CPU、进程、网络等多种资源的状态进行监测的知识，讨论了以多种形象直观的图形化方式，动态、实时地展示资源状态数据的技术，以及利用 Tab Control 控件对不同种类的数据进行组织和分页显示的方法。

第五个案例“HTTP 日志解析、分析系统”，讲述了一个可以对 Web 服务器日志进行解析和统计分析，从而揭示 Web 服务器运行状态和用户使用情况的工具。分析了常见的系统监测方法，介绍了为大多数 Web 服务器所支持的三种日志格式标准，并针对其中 Common Logfile Format 格式的日志文件，给出了数据预处理、日志解析、统计分析乃至保存的技术方法。

本书面向有一定 Visual C++ 编程基础的读者，不仅适合有志于从事软件开发的青年学生作为课程学习、毕业设计的参考资料，也适合于讲授 Visual C++ 课程的教师作为案例使用以丰富其讲课内容。对于刚刚开始从事 Visual C++ 开发的专业技术人员，也可以利用本书提高自己的技术水平。

本书由刘瑞主编，编写工作由刘瑞、吴跃进和王宗跃共同完成。其中，刘瑞编写了第一个和第二个案例；王宗跃编写了第三个案例；吴跃进编写了第四个和第五个案例。限于作者的水平，书中的错误在所难免。对于本书中的不当之处，恳请读者批评指正。

目 录

第 1 章 Visual C++概述	1
1.1 Visual C++简介	2
1.2 Visual C++集成开发环境	2
1.3 Visual C++工程	6
1.4 Microsoft 基础类库 MFC	9
1.5 事件驱动	12
本章小结	13
第 2 章 数据库查询工具	14
2.1 设计目标	15
2.2 系统设计	15
2.3 系统实现过程	16
2.3.1 程序构成	16
2.3.2 创建项目	16
2.3.3 创建拆分窗口框架	16
2.3.4 ADO 和 ADOX 组件	21
2.3.5 修改资源文件	23
2.3.6 CMainFrame 类——创建公共函数	26
2.3.7 CResultView 类——显示查询结果	37
2.3.8 CDBTreeView 类——创建表、视图树列表	38
2.3.9 CSqlView 类——编辑 SQL 语句	61
2.3.10 保存和读取 SQL 脚本文件	63
2.3.11 程序运行效果图	66
2.4 数据库访问技术	66
2.4.1 专用的数据库开发技术	66
2.4.2 ODBC	67
2.4.3 OLE-DB	68
2.4.4 ADO	69
2.4.5 ADOX	71
本章小结	73
第 3 章 数据转换工具	74
3.1 设计目标	75
3.2 系统设计	75
3.3 系统实现过程	76
3.3.1 程序构成	76

3.3.2	创建项目	76
3.3.3	搭建向导程序框架	76
3.3.4	引入 ADO 组件	82
3.3.5	CDTWizard 类——创建和管理属性表	82
3.3.6	CSource 类——获取源数据源和表	86
3.3.7	CTarget 类——将源数据源和表结构映射到目的数据源和表	93
3.3.8	CConfig 类——实现数据转换	133
3.3.9	程序运行效果图	134
	本章小结	135
第 4 章	图像处理系统	136
4.1	设计目标	137
4.2	系统设计	137
4.3	系统实现过程	137
4.3.1	程序构成	137
4.3.2	创建项目	137
4.3.3	修改资源文件	138
4.3.4	定义 PCX 图像文件的格式	141
4.3.5	CDib 类——封装位图操作	142
4.3.6	CImageDoc 类——显示和处理位图	165
4.3.7	CImageView 类——显示位图	170
4.4	程序运行效果图	172
4.5	位图文件格式	174
4.5.1	位图文件格式	174
4.5.2	位图文件结构	174
4.5.3	与位图有关的数据结构	176
4.5.4	常用位图操作算法	177
	本章小结	179
第 5 章	系统资源监测器	180
5.1	需求分析	181
5.2	系统设计	181
5.3	系统实现过程	182
5.3.1	程序的构成	182
5.3.2	系统内函数的功能与相互关系	182
5.3.3	利用 SNMP 命令获取操作系统资源的运行参数	183
5.3.4	制作监测器界面	197
5.3.5	全局变量与新建函数	200
5.3.6	初始化处理	202
5.3.7	实现 Tab Control 的选页操作	215
5.3.8	图像显示部分	222

5.3.9 程序运行效果图.....	233
5.4 简单网络管理协议.....	235
5.4.1 管理站 (MIB) 和管理代理.....	235
5.4.2 SNMP 的管理信息库.....	235
5.4.3 SNMP 的管理信息结构.....	237
5.4.4 SNMP 的协议操作.....	238
本章小结.....	238
第 6 章 HTTP 日志解析、分析系统	239
6.1 设计目标.....	240
6.2 系统设计.....	240
6.2.1 HTTP 日志格式.....	240
6.2.2 统计分析内容.....	243
6.2.3 系统结构.....	245
6.3 系统实现过程.....	246
6.3.1 全局变量与结构体.....	246
6.3.2 日志预处理函数.....	249
6.3.3 日志解析函数.....	279
6.3.4 结果保存函数.....	282
6.3.5 主程序部分.....	287
本章小结.....	302

第1章

Visual C++概述



学习重点

1. 了解 Visual C++集成开发环境的构成；
2. 了解 Visual C++应用程序开发的基本方法；
3. 了解 MFC 类库和事件驱动编程的相关知识。

1.1 Visual C++简介

Visual C++是 Microsoft 公司推出的 C 语言开发工具,是当前在 C++/C 领域内占统治地位的 C++/C 编辑器。Visual C++的前身是 Quick C 和 Microsoft C,但这两种开发工具都基于 DOS 的集成开发环境,直到 Visual C++ 1.0 (Microsoft C/C++ 8.0)才成为真正的 Windows 环境下的集成开发工具。到目前为止,Visual C++已经有了 1.0、1.5、2.0、4.0、5.0、6.0 等多个版本,最新的版本是.NET。

Visual C++具有强大的功能,它支持 Windows 开发所需的各种技术,能够开发各种类型的应用程序。与目前在 Windows 领域流行的其他开发工具,如 Visual Basic 相比,Visual C++功能更强大,它能够向程序员暴露更多系统内部的实现细节,能够进行更为细致的控制,更为底层的开发,其最终生成的代码更小,执行效率更高。当然,与 Visual Basic 等其他开发工具相比,Visual C++的学习难度更大,开发更为复杂,开发时间更长,更容易出现错误,但它却能够完成这些工具难以完成的一些工作。为了扬长避短,Visual C++通常用于诸如操作系统、设备驱动程序、动态链接库以及各类实用工具的开发。

1.2 Visual C++集成开发环境

Visual C++提供了一个图形化的交互式开发环境,称为集成开发环境。在集成开发环境中可以完成创建和管理工程、创建和编辑源程序和资源、编译和调试代码等多项工作。

Visual C++集成开发环境主要由四个部分组成,即菜单栏和工具条、视图、编辑窗口、状态栏,如图 1.1 所示。

1. 菜单栏和工具条

菜单栏提供了 File、Edit、View、Insert、Project、Build、Tools、Windows、Help 等多个菜单,分别提供了创建各种类型的工程、文件,编辑文件和资源,插入类、控件、资源,设置工程的各项配置,编译、创建和执行应用程序,调试程序,访问不同的实用工具,对工具栏、菜单以及集成开发环境进行定制,在不同的窗口之间切换和进行窗口布局,给出相关帮助的功能。

工具条位于菜单栏的下面,由一系列图形按钮构成,它们提供菜单栏最为常用的一些功能。相对于使用菜单的方式,使用工具条按钮不需要多次单击鼠标,在许多菜单项中进行浏览和选择,只需要用鼠标直接单击指定的按钮即可完成相应的功能,因而操作更为简单和方便。工具条中的操作按钮和菜单是相对应的,Visual C++中包含有十几种工具条。默认情况下,屏幕上会显示四个工具条,即 MenuBar、Standard、Build MiniBar 和 WizardBar 工具条。

2. 视图

在工具条下方左侧的窗口包含有三个视图,它们分别是 ClassView (类视图)、ResourceView (资源视图)和 FileView (文件视图)。在 Database Project 中还可以看到第

四种视图，即 **DataView**（数据视图）。可以通过选择视图窗口下方的标签来选择不同的视图，如图 1.2 所示。

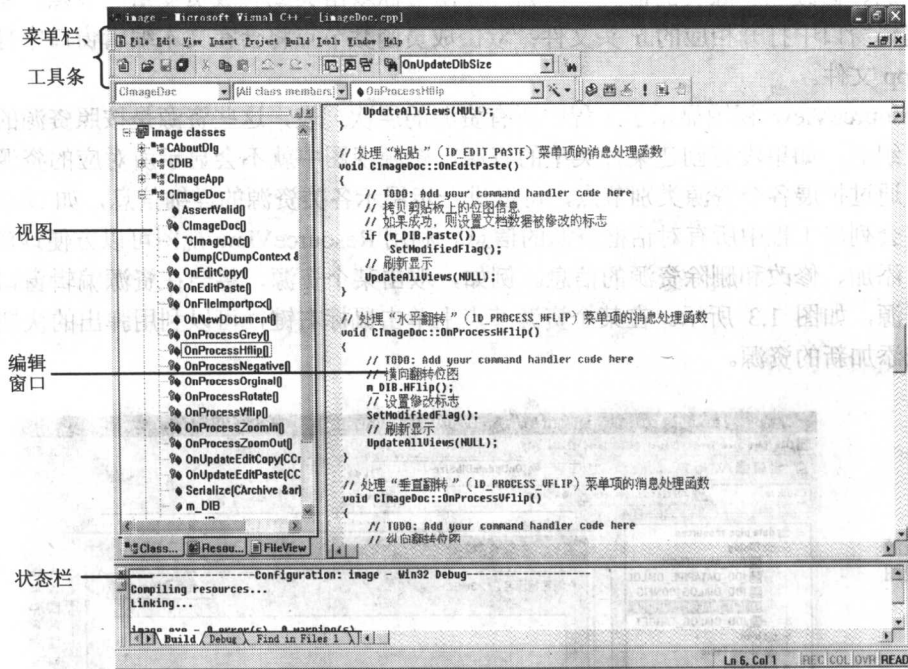


图 1.1 Visual C++集成开发环境

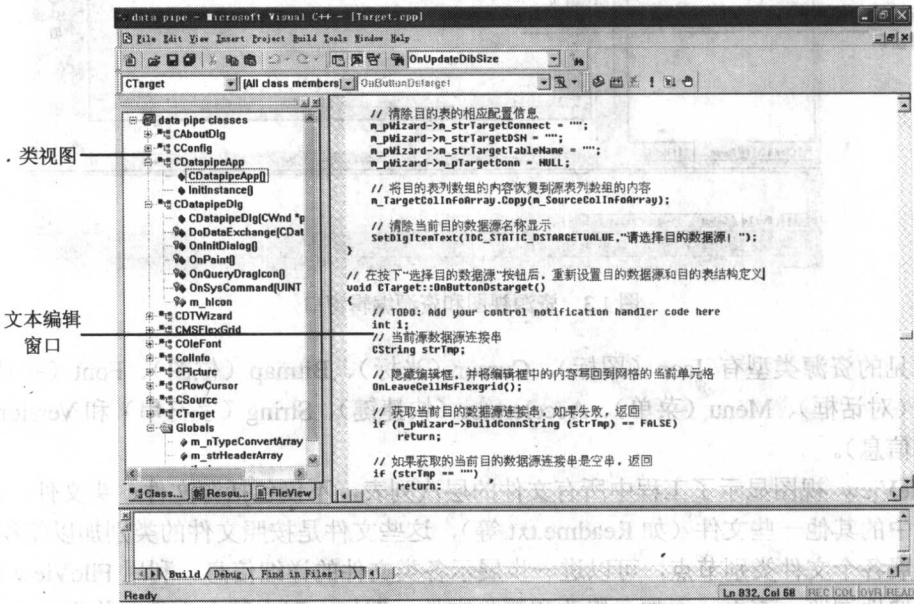


图 1.2 类视图和文本编辑窗口

ClassView 视图显示了工程中所有类的层次列表,通过扩展各个类节点可以进一步显示各个类的成员函数和成员变量的详细信息。利用 ClassView 视图可以方便地浏览、查看、添加、修改和删除类的信息。例如,使用鼠标双击类或者成员变量的节点,将会在文本编辑窗口中打开相应的.h 头文件;双击成员函数节点将会在文本编辑窗口中打开相应的.cpp 文件。

ResourceView 视图显示了工程中所有资源的层次列表,这些资源是按照资源的类别加以组织的,如果没有创建某种类型的资源,资源视图中就不会显示所对应的资源类别节点。通过扩展各个资源类别节点,可以进一步展示各类资源的详细信息,如 Dialog 节点下将会列出工程中所有对话框资源的信息。利用 ResourceView 视图可以方便地浏览、查看、添加、修改和删除资源的信息。例如,双击某个资源,将会在资源编辑窗口中显示该资源,如图 1.3 所示。在某个资源节点上单击鼠标右键,可以利用弹出的快捷方式菜单来添加新的资源。

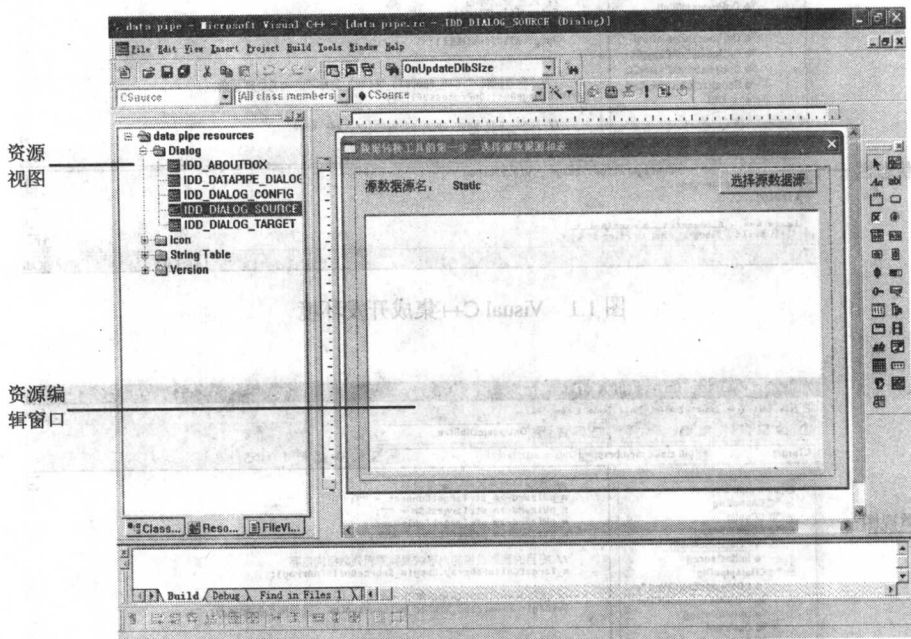


图 1.3 资源视图和资源编辑窗口

常见的资源类型有 Icon (图标)、Cursor (光标)、Bitmap (位图)、Font (字体)、Dialog (对话框)、Menu (菜单)、Accelerator (快捷键)、String (字符串) 和 Versioninfo (版本信息)。

FileView 视图显示了工程中所有文件的层次列表,不仅包括源文件、头文件,也包括工程中的其他一些文件(如 Readme.txt 等),这些文件是按照文件的类别加以组织的。通过扩展各个文件类别节点,可以进一步展示各类文件的详细信息。利用 FileView 视图可以方便地浏览、查看、添加、修改和删除文件。例如,双击某个文件,将会在文本编辑窗口中显示该文件的内容,如图 1.4 所示。在某个文件类别的节点上单击鼠标右键,

可以利用弹出的快捷方式菜单来添加新的文件或者新的文件夹。

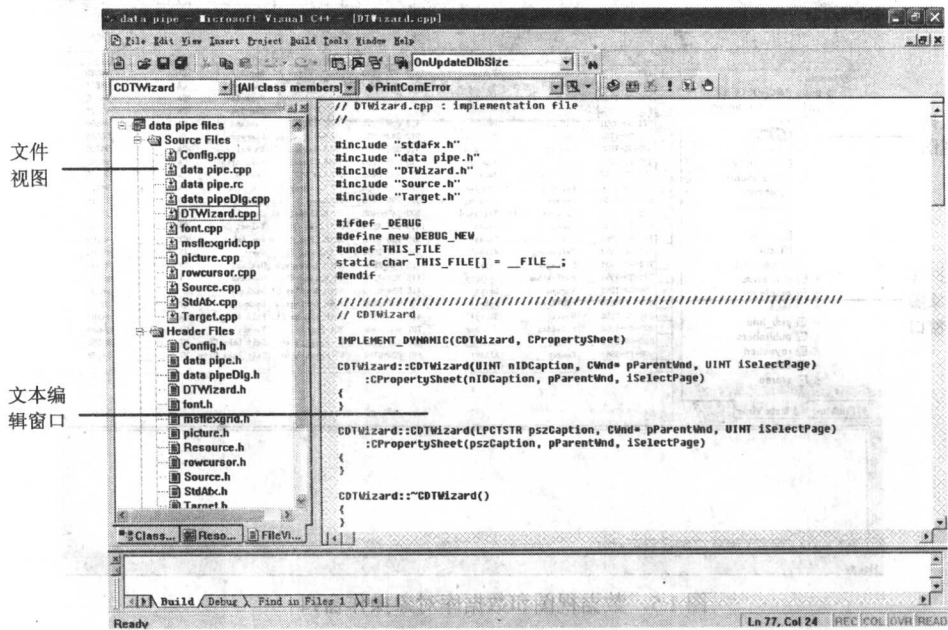


图 1.4 文件视图和文本编辑窗口

在 Database Project 中，DataView 显示了指定的 ODBC 数据源中数据库对象信息的层次列表。这些对象是按照类别加以组织的，包括 Database Diagrams（数据库关系图）、Tables（表）、Views（视图）及 Stored Procedures（存储过程）四个类别。通过扩展各个数据库对象类别节点，可以进一步展示各个数据库对象的详细信息，如图 1.5 所示。利用 DataView 可以方便地浏览、打开、编辑（运行）各个数据库对象及其中的数据，还可以创建、修改和编辑查询脚本。

3. 编辑窗口

编辑窗口位于视图窗口的右侧。Visual C++集成开发环境中主要有两种编辑窗口，即文本编辑窗口和资源编辑窗口。

文本编辑窗口用于编辑、修改源代码或文本文件，它不仅提供了一般编辑器所具有的编辑、复制、粘贴等功能，还专门提供了一些专门用于编程的功能。例如，它将文件中的关键字、注释、代码等不同文字以不同的颜色加以显示，使程序一目了然；它能够自动进行缩进和对齐，从而减少用户在美化格式上所花费的时间；它还可以在用户键入一个函数的名称后，自动显示函数相应的参数及其类型等。

资源编辑窗口用于编辑工程中的各种资源，每种资源都有自己对应的编辑界面。利用资源编辑器，可以利用图形化的界面，以所见即所得的方式轻松地绘制位图和图标，输入字符串资源，创建对话框并插入各种控件，完成调整控件的布局、大小、Tab 顺序及属性等诸多的工作，方便地完成工程中用户界面的设计和维护。

在数据视图中，还有对应的数据库对象操作窗口，可以利用它浏览、编辑表中的数据，或者设计查询脚本。

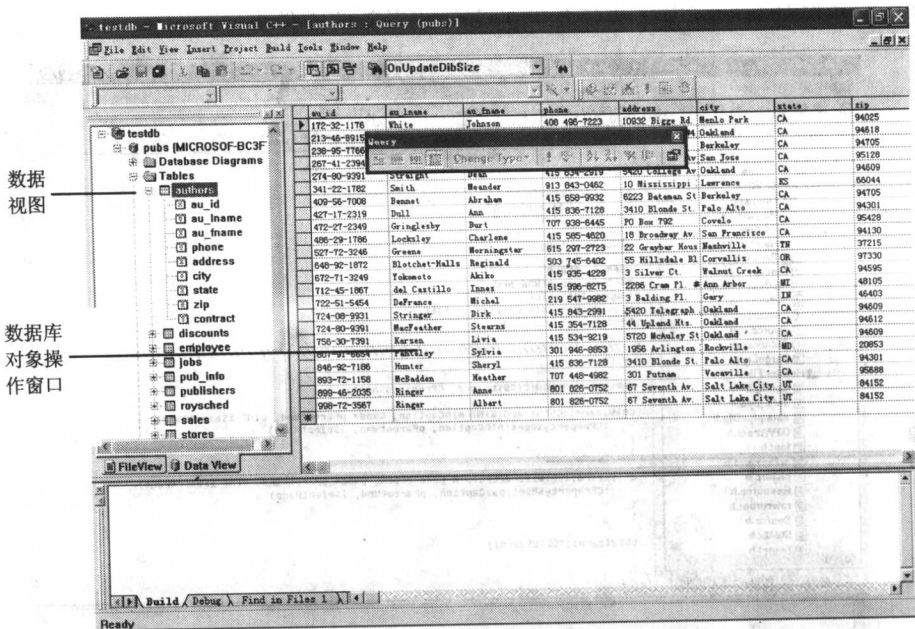


图 1.5 数据视图和数据库对象操作窗口

4. 状态窗口

状态窗口位于视图窗口和编辑窗口下方，主要用于完成以下工作：在编译工程时，显示编译的步骤和结果；显示在文件中查找字符串的结果；在调试工程时，显示调用栈和变量的信息。

1.3 Visual C++工程

Visual C++的编程工作通常包括以下步骤：创建工程，编写代码和资源文件，编译和调试，直到完成所需的应用程序。

创建工程是 Visual C++编程的第一步工作。工程是应用程序的组织单位，利用项目可以对所有与应用程序相关的代码和资源等文件进行管理，并且对与应用程序有关的其他配置信息（如怎样编译应用程序）等进行设置。

在 Visual C++中，可以创建如表 1.1 所列类型的工程。

创建工程的工作相当繁琐，涉及许多文件，还必须保证这些文件之间的一致性。为了简化用户的工作，Visual C++提供了一个名为 AppWizard（应用程序向导）的程序来帮助程序员完成工程的创建工作。

AppWizard 能够根据用户选择的工程类型及其他信息（如是否需要支持数据库，是否需要支持 ActiveX，选择使用对话框、单文档窗口还是多文档窗口等），自动地创建工程，并提供这类应用程序公共的框架代码。这个框架代码自身是一个完整的程序，编译后就可以直接运行，它提供了此类应用程序最基本的功能。现在程序员的工作就是在这

个框架代码中加入和完成解决自己问题的代码，最终实现所需的功能。这种方式既减少了程序员开发重复代码的工作量，又提供了足够的空间供程序员发挥，同时兼顾了效率和灵活性两个方面。

表 1.1 Visual C++提供的工程类型

工程类型	描 述
ATL COM AppWizard	ATL COM 工程。ATL 是 active template library 的简称，它由一套基于模板的 C++类构成，利用这些类可以方便地创建小而高效的 COM (component object model) 对象
Cluster Resource Type Wizard	Cluster Resource Type Wizard 可以创建两种类型的工程：Resource DLL 和 Cluster Administrator extension DLL。可以将它们部署到 Microsoft Cluster Server 上以管理和监测集群上的各种资源（应用程序）
Custom AppWizard	自定义的应用程序向导，只有 Visual C++专业版和企业版才具有此项功能。自定义的应用程序向导与 Visual C++的应用程序向导一样，都可以根据用户的选择，为用户自动生成代码、资源和工程文件。在特定的应用领域，如果需要反复开发特定类型的应用程序，而 Visual C++应用程序又不能提供相应的支持，就可以开发自定义的应用程序向导，为这类应用程序生成公共的基础代码，以提高开发效率
Database Project	数据库工程。利用数据库工程的数据 View，可以连接到指定的数据库，浏览、创建、编辑、管理数据库关系图、表、视图、存储过程、触发器等数据库对象，并操作数据库中的数据
DevStudio Add-in Wizard	插件或者 VBScript，可以编程对 Visual C++集成开发环境中的窗口、文档等对象进行控制，并自动完成一些例行的工作
Extended Stored Proc Wizard	扩展存储过程 (SQL server extended stored procedures, XPROC) 是一种以 Win32 DLL 形式提供的 C 函数，可以在 SQL Server 中使用。扩展存储过程运行在数据库服务器上，可以采用调用存储过程的方式来调用它访问 SQL Server。与使用 SQL 语句的 SQL Server 的存储过程不同，扩展的存储过程使用的是 Win32 API 和 Open Data Services (ODS) 编写的。只有 Visual C++企业版才支持扩展存储过程
ISAPI Extension Wizard	使用 MFC 的 Internet 服务器扩展或过滤器。Internet 服务器扩展或过滤器是一种动态链接库 (DLL)，可以用来增强支持 Internet Server API (ISAPI) 的 Web 服务器的功能。其中，Internet 服务器扩展可以代替 CGI 应用程序；Internet 服务器过滤器可以对发送到 Web 服务器或者从 Web 服务器发出的数据进行筛选
Makefile	Make 文件。一种用于在命令行环境下编译工程的文件
MFC ActiveX Control Wizard	使用 MFC 的 ActiveX 控件程序。ActiveX 控件是一种模块化的程序，可以作为其他应用程序的组成部件，为其他应用程序提供特定的功能
MFC AppWizard(dll)	使用 MFC 的动态链接库。动态链接库是一种二进制文件，动态链接库中的函数可以同时为多个程序所共享和调用
MFC AppWizard(exe)	使用 MFC 的 Windows 可执行程序
Utility Project	Utility Project 是一个可以包含其他文件的容器，可以用它来将多个子工程组合为一个工程，或者导出 MAKEFILE。使用 Utility Project 必须自己创建定制的编译规则和添加相应的文件，Visual C++不会自动完成这些工作
Win32 Application	使用 C 或者 C++编写的、Win32 API 的 Windows 可执行程序
Win32 Console Application	控制台应用程序。控制台应用程序是使用 Console API 编写的，具备 DOS 风格的窗口外观，能够在控制台窗口中提供字符模式支持的应用程序。在控制台应用程序中可以使用标准 C 函数完成输入和输出，如 printf 和 scanf()。使用控制台应用程序可以把某些 DOS 下的代码转移到 Windows 系统之下

工程类型	描 述
Win32 Dynamic-link Library	使用 C 或者 C++编写的、Win32 API 的动态链接库
Win32 Static Library	使用 C 或者 C++编写的、Win32 API 的静态库。静态库是一种函数库文件，当应用程序使用静态库时，在编译时需要将它链接到可执行程序中去。无论是基于 MFC 还是不使用 MFC 的应用程序都可以使用静态库

AppWizard 的第一步的界面如图 1.6 所示。

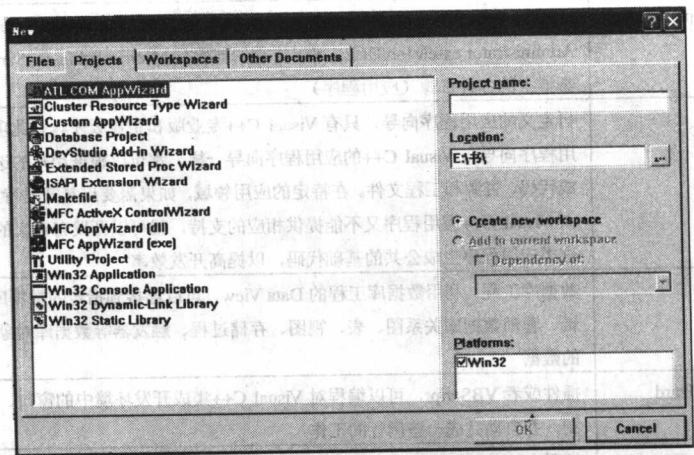


图 1.6 利用 AppWizard 创建工程

经由 AppWizard 生成的 Visual C++ 工程中通常包含如表 1.2 所列的几种类型的文件。

表 1.2 AppWizard 生成的文件的类型

文件类型	描 述
.dsw	Workspace（工作空间）文件。在一个 Workspace 文件中可以包含多个 Project（工程），由 Workspace 文件对它们进行统一的协调和管理
.opt	开发环境所使用的选项文件。用于保存 Workspace 中创建的所有用户选项（如在 Project Setting 对话框中所做的设置或者 Workspace 的布局），在打开 Workspace 时将根据这些选项完成所有的定制工作，使其符合用户的习惯和要求
.dsp	Project（工程）文件。存放的是与特定工程有关的信息，如包含哪些文件。Project 文件是 Workspace 的组成部分之一，每个工程对应于一个.dsp 文件
.clw	ClassWizard 使用的文件。用于保存关于工程中的类的信息
.odl	包含有关控件类型库的 Object Description Language（对象描述语言）源代码。Visual C++使用这个文件来生成类型库，以便使用控件的接口
.h	头文件。主要包括类、常量、函数等的定义
.c 和 .cpp	实现文件。包括类成员函数或者其他函数的代码，其中，stdafx.cpp 和 stdafx.h 文件用于创建预编译头文件 projname.pch
.rc	工程的资源文件。包含大多数 MFC 应用程序通常使用的默认菜单、快捷键、字符串表、About 对话框、图标和工具条等资源
.rc2	对于那些在不同的工程之间共享的资源，可以将它们保存在 RC2 文件中，并使用 include 语句将 RC2 文件包含在工程的 RC 文件中。这样，就不必为每个不同的项目分别创建同样的资源

文件类型	描 述
.def	DLL 工程的模块定义文件。对于控件, 它还提供了有关控件的名称、描述和运行时堆大小的信息
res\profilename.ico	MFC 程序或者控件所使用的图标。当应用程序最小化时, 或者在 About 对话框上会显示这个图标
res\toolbar.bmp	保存应用程序各个工具条按钮所对应图像的位图文件。需要使用 include 语句将它包含在工程的资源文件中
Readme.txt	用于描述 AppWizard 或者 ControlWizard 生成的每一个文件的名称及其用途

1.4 Microsoft 基础类库 MFC

在 AppWizard 中生成了初始的工程和代码之后, 下一步的工作就是在其中加入用户代码以实现自己所选的特殊功能。

要完成这一步的工作, 可以有两个选择, 一个是使用 Win32 API (application programming interface, 应用程序编程接口) 的方式, 另一个是使用 MFC (Microsoft foundation class library, Microsoft 基础类库) 的方式。

Win32 API 的方式就是使用 Microsoft 公司提供的一系列用于 Windows 编程的函数来开发自己的 Windows 应用程序。与开发其他的 C 语言程序类似, 只需要用 include 包含 Win32 API 的头文件, 并在编译时, 链接对应的函数库, 就可以创建自己的 Windows 应用程序了。这些头文件和函数库包含在 SDK (software develop kit, 软件开发工具包) 中, 而 SDK 又包含在 Visual C++ 开发环境中。

Win32 API 是 Windows 的编程基础, 其本身包括了 Windows 的基本概念。由于 Windows 本身结构的复杂性, 许多 API 函数的内在机理也很复杂, API 函数间的关系也很复杂。使用 Win32 API 开发 Windows 应用程序, 会将 Windows 程序运行的所有细节都暴露在程序员的面前。程序员需要了解并掌握应用程序的方方面面, 程序运行的每一个步骤都需要程序员编码才能够实现, 因而开发工作量和难度都较大。

为解决这一问题, Microsoft 公司开发了对 Win32 API 进行封装的 MFC, 为 Windows 程序员提供了一个面向对象的编程接口。MFC 是用于 Windows 编程的一个应用程序框架, 它使用 C++ 编写, 为程序员完成了很多 Windows 编程中的例行性工作, 提供了管理窗口、菜单、对话框, 完成基本的输入、输出, 访问数据库, 通过网络进行通信, 使用 OLE 和 Internet 等诸多工作所需的基本代码。这些代码经过了 Microsoft 公司的精心设计, 凝聚了其多年开发 Windows 程序的经验, 是创建新的 Windows 应用程序的良好而稳定的基础。

MFC 为程序员提供了开发 Windows 应用程序的标准框架和基础代码, 使程序员能够在更高的抽象层次上进行开发, 不必纠缠于非常底层、困难而复杂的技术细节, 也不必重复编写每个 Windows 应用程序都要用到的那些代码, 从而可以将精力集中到解决用户需求所必须实现的业务逻辑上, 只编写那些自己的应用程序所需的专用代码, 从而大大简化了 Windows 编程的工作量, 降低了开发难度, 提高了开发速度和质量。

MFC 在整个 Windows 家族中都是兼容的, 使用 MFC 开发应用程序更容易在不同的