开发技术整合应用丛书

Visual C++ .NET 网络数据库整合应用

精锐创作组　　编著

人民邮电出版社

需要全本请访问 www.ertongbook.com

图书在版编目（CIP）数据

Visual C++.NET 网络数据库整合应用 / 精锐创作组编著. －北京：人民邮电出版社，2001.10
（开发技术整合应用丛书）
ISBN 7-115-09713-5

Ⅰ.V…Ⅱ.精…Ⅲ.C 语言—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2001）第 069087 号

内 容 提 要

本书从 Visual C++ .NET 的网络数据库编程基础开始，以示例的形式由浅入深地介绍了在 Visual C++ .NET 中开发网络数据库应用程序的技术，包括 DAO 技术、ODBC 连接、查询语言、Internet 编程等。

本书通俗易懂，讲解细致，并对大量示例进行了透彻的分析，适合中级程序开发人员学习使用。对于从事 Visual C++ .NET 网络数据库开发与应用的广大科研人员和高校相关专业的师生而言，本书也不失为一本有价值的自学读物或教学参考书。

开发技术整合应用丛书

Visual C++.NET 网络数据库整合应用

◆ 编　　著　精锐创作组
责任编辑　张瑞喜
执行编辑　牛　磊

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ pptph.com.cn
网址　http://www.pptph.com.cn
读者热线：010-67129212　010-67129211（传真）
北京汉魂图文设计有限公司制作
北京　　印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：25
字数：607 千字　　2001 年 10 月第 1 版
印数：1 –　000 册　　2001 年 10 月北京第 1 次印刷

ISBN 7-115-09713-5/TP・2511

定价：35.00 元

本书如有印装质量问题，请与本社联系 电话：（010）67129223

前　言

Visual C++ .NET 是运行于 Windows 平台上的交互式的可视化集成开发环境，它是美国 Microsoft 公司开发的 Microsoft Visual Studio .net 套件的一部分。像其他的可视化集成开发环境（如 Visual Basic、Delphi、C++ Builder）一样，Visual C++ .NET（为了叙述方便，以下简称为 VC，或 VC.NET）集程序的代码编辑、编译、连接和调试等功能于一体，给编程人员提供了一个完整方便的开发界面和许多有效的辅助开发工具。VC 的应用程序向导可以为很大一部分类型的程序提供框架代码，用户不用书写程序代码，只需按几个按钮就可以生成一些完整的可以运行的程序。

本书通过丰富的示例程序向读者介绍如何使用 VC 开发网络数据库应用程序，这些示例程序都是作者根据自己使用和开发 VC 程序时的体会精心编写的，是针对程序员在开发过程中最迫切需要、使用频率最高的内容特意定制的，可以说比较贴切地符合了初级和中级程序员的需求。另外，本书中所有示例程序的代码都经过了严格的调试和测试，读者只要按照书中的步骤往下做，最终一定能够圆满地完成程序。

本书主要包括以下内容。

第 1 章是本书的基础部分，介绍了有关对话框编程的话题，包括常用的对话框控件、菜单、工具栏和状态条、公共对话框、非模式对话框和标签化对话框等。

第 2 章首先分类列举了 MFC 中的常用类，然后对 CWnd、MFC 数组链表类、CString、CTime、CTimeSpan、CRect、CSize、CPoint、CFile、CArchive 和 CStdioFile 等重要的成员函数以示例的形式进行了讲解。

第 3 章全面地介绍了 SQL 语言。SQL 是一种数据库查询和编程语言，同时也是对数据库中的数据进行组织、管理和检索的工具。用户想检索数据库的数据时，就可以用 SQL 语言编写出提取信息的 SQL 语句。SQL 语句用某种规则获取数目有限的特定的记录。

尽管查询是 SQL 的核心，但是 SQL 不仅仅是一个查询工具，它还可以控制数据库管理系统提供给用户的所用功能，包括定义数据存放的结构、数据更新、数据的检索、实现数据共享和规定数据的完整性等。SQL 已经成为关系型数据库的标准语言，它广泛应用于客户机/服务器的结构中。

第 4 章介绍了 ODBC 技术。Microsoft 公司于 1991 年 1 月发布了 ODBC，并迅速得到了广大用户的认可，各主要数据库厂商也纷纷推出支持 ODBC 的驱动程序。事实证明 ODBC 作为 Microsoft 公司的重大成果，已经成功地为 Windows 操作系统建立了一个开放的接口。

第 5 章通过较长的篇幅介绍了 CDaoWorkspace 类、CDaoDatabase 类、CDaoTableDef 类、CDaoQueryDef 类和 CDaoRecordset 类，在讲解的过程中穿插了很多示例，通过这些示例，读者可以更加深入地理解 DAO 技术在 VC 中的应用。

Windows 提供了许多 API 函数来支持 Internet 编程，但是这需要较多的网络知识，因此，

我们在第 6 章中着重介绍了使用 MFC 中的 WinInet 类进行编程的方法，并举例说明了如何建立自己的网络应用程序。

参加本书编写的人员有尹立民、王兴晶、韩雪松、杨瑞多、诸葛伟、赵建明、张松、梁研、刘晓亮、高博、胡彦军、周述康、晏奇、王雷、黄山、戴祝运、陈圆、袁骏、冯海涛、屈岩松、涂彬、张牧、郭琳、段明媛、朱嘉音等。由于编写时间紧，作者水平有限，书中难免存在一些不足之处，恳请读者批评指正。

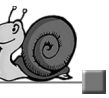
编者

需要全本请访问 www.ertongbook.com

目　　录

第 1 章　控件技术 1
1.1　**常用控件** 2
1.1.1　**静态文本框** 2
1.1.2　Picture **控件** 3
1.1.3　**编辑框** 3
1.1.4　**示例程序** 4
1.1.5　**按钮/复选框/单选框/组框** 10
1.1.6　**示例程序** 11
1.1.7　**列表框** 15
1.1.8　**组合框** 16
1.1.9　**滚动条** 18
1.1.10　**进度条** 18
1.1.11　**滑动控件** 19
1.1.12　**示例程序** 20
1.1.13　**树控件** 24
1.1.14　IP **地址控件** 26
1.1.15　**其他控件** 26
1.2　**菜单、工具栏和状态条** 27
1.2.1　**编辑菜单** 27
1.2.2　**编辑工具栏** 30
1.2.3　**菜单、工具栏及状态条的** MFC **管理类** 33
1.2.4　**把菜单、工具栏和状态条加入主对话框** 37
1.3　**公共对话框** 39
1.3.1　CFileDialog 39
1.3.2　CColorDialog 42
1.3.3　**示例程序** 44
1.3.4　CFontDialog 50
1.3.5　**示例程序** 53
1.3.6　CPrintDialog 57
1.3.7　CFindReplaceDialog 57
1.4　**非模式对话框** 58
1.5　**小结** 59

第 2 章　典型 MFC 编程技术 61
2.1　**窗口公共基类** CWnd 62
2.2　**功能强大的字符串类** CString 64

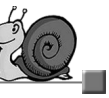


2.2.1 CString 的构造函数 65
2.2.2 CString 的基本操作 65
2.3 MFC 数组类 69
2.3.1 数组模板类 70
2.3.2 非模板数组类 74
2.4 MFC 链表类 75
2.4.1 链表模板类 75
2.4.2 非模板链表类 80
2.5 CPoint、CRect 和 CSize 80
2.5.1 CPoint 81
2.5.2 CSize 81
2.5.3 CRect 82
2.6 CTime 和 CTimeSpan 84
2.6.1 CTime 类 84
2.6.2 CTimeSpan 类 86
2.7 CFile 91
2.7.1 打开文件 91
2.7.2 文件读写 93
2.7.3 文件定位 93
2.7.4 文件相关信息的设置与获取 94
2.7.5 文件锁定与解锁 95
2.7.6 文件关闭 95
2.8 CArchive 98
2.8.1 类的序列化 99
2.8.2 创建 CArchive 对象 99
2.8.3 关闭 CArchive 100
2.8.4 CArchive 的读写操作 100
2.8.5 CArchive 的状态函数 100
2.9 CStdioFile 类 101
2.9.1 CStdioFile 的文本存取 101
2.9.2 用 CStdioFile 类存取文本 102
2.10 小结 103

第 3 章 数据查询 105
3.1 准备工作 106
3.2 查询语句 108
3.2.1 Select 语句 108

3.2.2 简单 Select 语句示例 109
3.2.3 Where 子句 111
3.2.4 Where 子句的应用 112
3.2.5 Group by 子句 116
3.2.4 Order by 子句 117
3.2.5 Having 子句 119
3.2.6 相关的函数 121
3.3 更改记录 127
3.3.1 Select...Into 语句 127
3.3.2 Insert...Into 追加查询语句 128
3.3.3 Update 更新查询语句 132
3.3.4 Delete 语句 133
3.4 表之间的连接 135
3.4.1 Union 运算 135
3.4.2 Left Join 和 Right Join 运算 138
3.4.3 Inner Join 运算 139
3.5 数据定义 140
3.5.1 Crtate Table 语句 141
3.5.2 Create Index 语句 141
3.5.3 Alter Table 语句 142
3.5.4 Drop 语句 143
3.6 小结 143

第 4 章　ODBC 技术在 VC 中的应用 145
4.1 ODBC 概述 146
4.2 CDatabase 类 147
4.2.1 构建函数 147
4.2.2 属性函数 150
4.2.3 操作函数 152
4.2.4 重载函数 152
4.2.5 数据成员 153
4.3 CRecordset 类 153
4.3.1 构建函数 153
4.3.2 属性函数 154
4.3.3 操作函数 155
4.3.4 其他操作函数 156
4.3.5 重载函数 157

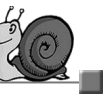


4.3.6 数据成员 158
4.4 创建 ODBC 数据库应用程序 159
4.4.1 创建应用程序框架 159
4.4.2 创建视图 159
4.4.3 数据库的遍历 161
4.4.4 添加新记录 163
4.4.5 修改记录 164
4.4.6 删除记录 165
4.4.7 记录的筛选和排序 166
4.5 小结 175

第 5 章 DAO 技术在 VC 中的应用 177
5.1 CDaoWorkspace 类 178
5.1.1 CDaoWorkspace 类概述 178
5.1.2 CDaoWorkspace 类的构造函数 178
5.1.3 工作区属性函数 178
5.1.4 工作区操作函数 180
5.1.5 数据库引擎属性函数 188
5.1.6 CDaoworkSpace 类的数据成员 192
5.2 CDaoDatabase 类 192
5.2.1 CDaoDatabase 类 192
5.2.2 CDaoDatabase 类的构造函数 193
5.2.3 数据库属性函数 193
5.2.4 数据库操作函数 197
5.2.5 CDaoDatabase 类的数据成员 207
5.3 CDaoTableDef 类 207
5.3.1 CDaoTableDef 类概述 207
5.3.2 CDaoTableDef 类的构建函数 208
5.3.3 表定义属性函数 209
5.3.4 表定义操作函数 228
5.3.5 CDaoTableDef 类的数据成员 233
5.4 CDaoQueryDef 类 233
5.4.1 CDaoQueryDef 类 233
5.4.2 CDaoQueryDef 类的构建函数 233
5.4.3 查询定义属性函数 234
5.4.4 查询定义操作函数 238
5.4.5 CDaoQueryDef 类的数据成员 242

5.5 CDaoRecordset 类 242
5.5.1 CDaoRecordset 类的构建函数 243
5.5.2 记录集属性函数 244
5.5.3 记录集更新函数 252
5.5.4 记录集定位函数 254
5.5.5 记录集杂项操作函数 257
5.5.6 记录集重载函数 264
5.5.7 CDaoRecordset 类的数据成员 264
5.6 小结 266

第 6 章　用 VC 开发网络应用程序 267
6.1 Windows Sockets 简介 268
6.1.1 什么是 Windows Sockets 规范 268
6.1.2 Microsoft Windows 和针对 Windows 的扩展 268
6.1.3 Windows Sockets 协议栈安装检查 269
6.1.4 套接口 270
6.1.5 阻塞 273
6.2 Windows Socket 1.1 函数参考 274
6.2.1 库函数 274
6.2.2 数据函数 298
6.2.3 Windows 扩展函数 303
6.3 Windows Socket 2 的扩展特性 309
6.3.1 同时使用多个传输协议 309
6.3.2 与 Windows Socket 1.1 应用程序的向后兼容性 310
6.3.3 在 Windows Sockets 中注册传输协议 311
6.3.4 协议无关的名字解析 312
6.3.5 重叠 I/O 和事件对象 312
6.3.6 使用事件对象异步通知 315
6.3.7 服务的质量（QOS） 316
6.3.8 套接口组 317
6.3.9 共享套接口 317
6.3.10 连接建立和拆除的高级函数 318
6.3.11 扩展的字节顺序转换例程 318
6.3.12 分散/聚集方式 I/O 318
6.3.13 协议无关的多点通信 319
6.3.14 新增套接口选项一览 319
6.3.15 新增套接口 ioctl 操作代码 319

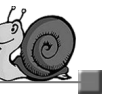


6.3.16 新增函数 320
6.4 Internet 编程 342
6.4.1 WinInet (HTTP, FTP, Gopher)类编程简介 342
6.4.2 创建一个互联网客户应用程序 343
6.4.3 HTTP 应用 344
6.4.4 FTP 应用 355
6.4.5 Gopher 应用 364
6.5 小结 387

需要全本请访问 www.ertongbook.com



第 1 章
控件技术



1.1 常用控件

1.1.1 静态文本框

静态文本框是一种静态控件，可以显示字符串、矩形（Rectangle）、图标（Icon）、光标（Cursor）、位图（Bitmap）以及元文件（Metafile）等，常被用来作为其他控件的标签，也可以作为容器容纳其他控件，还可以分割控件。静态文本框使用户无须编写代码就可以方便地在静态控件上显示字符串。

静态文本框的属性

静态文本框的重要属性如下：

- Caption：静态文本框上所显示的字符串。
- AlignText：文本框上字符串的对齐方式，有左对齐、居中对齐和右对齐。
- CenterVertically：竖直方向的居中对齐。
- NoPrefix：在通常情况下，文本框上的“ & ”字符并不显示出来，而是表示紧跟在它后面的字母下面加下划线，NoPrefix 属性使“ & ”字符正常显示。
- NoWrap：显示的文本到达控件边界时是否自动换行。
- Sunken：文本框凹陷显示。
- Border：给文本框加一个黑框作为边界。

CStatic 类的成员函数

静态控件（其中包括静态文本框）由 MFC 类 CStatic 来管理，它是窗口类 CWnd 的子类。CStatic 的主要成员函数如下：

- SetBitMap：设置要在静态控件上显示的位图，参数是位图句柄，返回以前与控件相联系的位图句柄。
- GetBitMap：得到通过函数 SetBitMap 设置的与控件相联系的位图句柄。
- SetIcon：设置要在静态控件上显示的图标，参数是图标句柄，返回以前与控件相联系的图标句柄。
- GetIcon：得到通过函数 SetIcon 设置的与控件相联系的图标句柄。
- SetCursor：设置要在静态控件上显示的光标，参数是光标句柄，返回以前与控件相联系的光标句柄。
- GetCursor：得到通过函数 SetCursor 设置的与控件相联系的光标句柄。
- SetEnhMetaFile：设置要在静态控件上显示的元文件，参数是元文件句柄，返回以

前与控件相联系的元文件句柄。

- GetEnhMetaFile：得到通过函数 SetEnhMetaFile 设置的与控件相联系的元文件句柄。

使用上述函数时要注意，要想让静态框显示某种类型的对象，必须设置静态框的相应属性，比如调用 SetBitMap 显示位图时要设置 CStatic 的 SS_BITMAP 属性。

静态文本框通常不接受鼠标和键盘的输入，但是如果在属性对话框中设置了其 Notify 属性，当鼠标单击时，它会给父窗口一个 BN_CLICKED 事件通知。

1.1.2 Picture 控件

Picture 控件也是一种静态控件，由 CStatic 类管理。与静态文本框相比较，此控件提供了一种方便地显示图像的办法：只要在其属性对话框上选择要显示的图像类型（位图、光标、图标）和要显示的图像，就可以直接显示出来，而不用写任何代码；

Picture 控件的重要属性包括：Center image、Real size image、Type、Image 和 Color 等。

1.1.3 编辑框

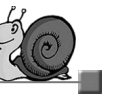
编辑框控件主要用于在程序中输入文本，它的功能由 CEdit 类来实现。

编辑框的属性

编辑框可以是单行的(Single-Line)或多行的(Multi-Line)，多行编辑框允许用户在控件中输入多行文本。通常情况下，在一个编辑框中按【Enter】键相当于单击对话框的默认按钮。如果要让用户在多行编辑框中通过按【Enter】键增加一行，那么应该设置其 AutoHScroll 或者 Want return 属性。

编辑框的主要属性如下：

- MultiLine：设置为多行编辑框。
- AutoHScroll：如果设置此属性，当用户输入的文本到达控件右边界时，控件自动水平卷动，允许用户输入更多的内容。如果没有设置此属性，单行编辑框将不允许再输入多余的字符。
- Number：只允许输入数字。
- Password：输入的任何字符都显示成字符“*”。
- Uppercase/Lowercase：所有输入的字符都以大写/小写字母显示。
- ReadOnly：设置编辑框属性为只读，即用户无法在编辑框输入字符和修改编辑框内容。
- AutoVscroll：当用户在编辑框内按【Enter】键或当输入的文本到达控件边界时自动到下一行。
- HorizontalScroll/VerticalScroll：为编辑框提供水平/垂直滚动条。



CEdit 类的成员函数

CEdit 类（编辑框属于 CEdit 类）的常用成员函数如下：

- GetLineCount：得到多行编辑框的行数。
- GetSel：得到编辑框中所选中字符串的起始位置和结束位置。
- SetSel：在编辑框中根据起始位置和终止位置选中一个字符串。
- SetLimitText/GetLimitText：设置/得到编辑框所能容纳的字符数。
- Undo/Clear/Copy/Cut/Paste：剪贴板操作。
- SetReadOnly：根据参数设置控件的只读属性。

编辑框的常用事件

编辑框的常用事件如下：

- EN_UPDATE：控件即将显示改变之后的文本内容时触发。
- EN_CHANGE：控件文本内容改变时触发。
- EN_SETFOCUS：控件即将获得输入焦点时触发。
- EN_KILLFOCUS：控件即将失去输入焦点时触发。
- EN_MAXTEXT：控件已经不能再接收输入，新的输入被删除时触发。

1.1.4 示例程序

下面是一个利用编辑框显示当前日期和时间的示例程序。

程序界面

示例程序的主界面是对话框，上面有一个静态文本框、两个按钮和一个编辑框，界面如图 1-1 所示。

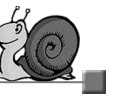


图 1-1 程序界面

添加代码

程序响应代码如下所示：

```
// CtlEditDlg.cpp : implementation file
//
#include "stdafx.h"
#include "CtlEdit.h"
#include "CtlEditDlg.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
/////////////////////////////////////////////////////////////////////////////
// CAboutDlg dialog used for App About
class CAboutDlg : public CDialog
{
public:
 CAboutDlg();
// Dialog Data
 //{{AFX_DATA(CAboutDlg)
 enum { IDD = IDD_ABOUTBOX };
 //}}AFX_DATA
 // ClassWizard generated virtual function overrides
 //{{AFX_VIRTUAL(CAboutDlg)
 protected:
 virtual void DoDataExchange(CDataExchange* pDX);       // DDX/DDV support
 //}}AFX_VIRTUAL
// Implementation
protected:
 //{{AFX_MSG(CAboutDlg)
 //}}AFX_MSG
 DECLARE_MESSAGE_MAP()
};
CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
```



```
 //{{AFX_DATA_INIT(CAboutDlg)
 //}}AFX_DATA_INIT
}
void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
 CDialog::DoDataExchange(pDX);
 //{{AFX_DATA_MAP(CAboutDlg)
 //}}AFX_DATA_MAP
}
BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
 //{{AFX_MSG_MAP(CAboutDlg)
        // No message handlers
 //}}AFX_MSG_MAP
END_MESSAGE_MAP()
/////////////////////////////////////////////////////////////////////////////
// CCtlEditDlg dialog

CCtlEditDlg::CCtlEditDlg(CWnd* pParent /*=NULL*/)
 : CDialog(CCtlEditDlg::IDD, pParent)
{
 //{{AFX_DATA_INIT(CCtlEditDlg)
        // NOTE: the ClassWizard will add member initialization here
 //}}AFX_DATA_INIT
 // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
 m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CCtlEditDlg::DoDataExchange(CDataExchange* pDX)
{
 CDialog::DoDataExchange(pDX);
 //{{AFX_DATA_MAP(CCtlEditDlg)
 DDX_Control(pDX, IDC_CtlEdit, m_ViewTime);
 //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CCtlEditDlg, CDialog)
```


需要全本请访问 www.ertongbook.com