看实例学编程系列丛书

Visual C++ 6.0 数据库开发实例

同志工作室　　编著

人 民 邮 电 出 版 社

需要全本请访问 www.ertongbook.com

内 容 提 要

Visual C++ 6.0 是美国 Microsoft 公司开发的 Microsoft Visual Studio 套件的一部分，是运行于 Windows 平台上的交互式可视化集成开发环境。本书从 Visual C++ 6.0 的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++ 6.0 中开发数据库应用程序的技术，包括 DAO 技术、ODBC 连接、SQL 语言等。

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++ 6.0 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。

看实例学编程系列丛书

Visual C++ 6.0 数据库开发实例

编　　著　　同志工作室

责任编辑　　姚予疆

前　言

Visual C++ 6.0 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 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++ 6.0（为了叙述方便，以下简称为 VC6）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。VC6 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 VC6 开发数据库应用程序，这些示例程序都是作者根据自己使用和开发 VC6 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。

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、无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 语句用某种规则以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，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 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 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 类和 CDaoQueryDef 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 的编程，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。同时，本章还介绍了多媒体程序设计的一般思想，通过本章的学习可以重点地掌握 MMControl 控件的应用，利用它可以设计自己的“媒体播放器”等多媒体应用程序。

本书由尹立民、王兴晶、陈立、孔德祥编写。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。

编著者

编 者 的 话

面向对象技术近年来发展迅速，它被广泛地应用到计算机研究与应用的各个方面，如文件处理、操作系统设计、多媒体技术、网络与数据库开发等。用面向对象技术进行程序设计、开发软件已经成为一种时尚。这种技术从根本上改变了人们以往设计软件的思维方式，从而使程序设计者可以最大限度地摆脱烦琐的数据格式和冗长的研发过程，将精力集中在对要处理的对象的设计和研究上，大大提高了软件开发的效率。为了满足初中级 Windows 程序开发人员、大专院校相关专业师生及业余爱好者学习和应用各种流行程序设计软件的需求，我们同志工作室经过多方调研，在收集了不同层次读者意见的基础上，经过仔细研讨，于 2000 年 5 月份推出了《看实例学编程》系列丛书的前 5 本。

《看实例学编程》系列丛书介绍的软件都是国内外著名软件公司的知名产品，也是国内应用面最广的软件。本套丛书一改以往计算机编程图书枯燥的风格，将软件开发技术融合到程序示例中，采用了由实际到理论、由具体到抽象的逆向写作思路。丛书一经推出，就立即得到了广大读者的好评，同时，也有不少读者建议，能不能以这种方式更深入地介绍软件开发的各专项领域。为了满足广大读者的需求，我们同志工作室的全体成员经过多方讨论，精心策划了下面 10 本专题类图书，收入本套丛书内。它们是《Delphi 5 数据库开发实例》、《Visual Basic 6.0 数据库开发实例》、《Visual C++ 6.0 数据库开发实例》、《C++ Builder 5.0 数据库开发实例》、《Delphi 5 API 函数开发实例》、《Visual Basic 6.0 API 函数开发实例》、《C++ Builder 5.0 API 函数开发实例》、《Delphi 5 多媒体开发实例》、《Visual Basic 6.0 多媒体开发实例》及《C++ Builder 5.0 多媒体开发实例》。

这 10 本书秉承了前 5 本书的特点，但它更侧重于软件开发的具体领域。例如，数据库、多媒体和 API 函数，而不是广泛地学习软件各个方面的知识；不是繁琐冗长的使用手册或枯燥乏味的大本参考书，而是独具实效的实例指南。这 10 本书准确地告诉读者用程序设计软件可以做哪些开发工作以及如何做这些开发工作，内容充实、讲解细致、分析透彻，笔调亲切，绝没有居高临下的架势。而且，我们在编写的过程中尽量省去了枯燥难懂的专业术语，以平和易懂的语言带领大家逐步进入到编程的艺术天堂。这些书以计算机中级程序开发人员为主要的读者对象，为便于读者理解，我们根据自己学习和使用的体会精心挑选了大量的实例，这些实例都是针对程序员在开发过程中最需掌握的技术而特意定制的，能较好地满足读者的需求。

同志工作室

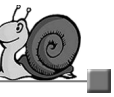
目　　录

第 1 章　基本控件 …… 1

1-1　常用控件 …… 2
静态文本框 …… 2
Picture 控件 …… 3
编辑框 …… 3
按钮/复选框/单选框/组框 …… 10
列表框 …… 16
组合框 …… 17
滚动条 …… 18
进度条 …… 19
滑动控件 …… 20
树控件 …… 25
IP 地址控件 …… 27
其他控件 …… 27
1-2　菜单、工具条和状态条 …… 28
编辑菜单 …… 28
增加菜单单击消息响应函数 …… 31
编辑工具条 …… 32
增加工具条按钮命令消息响应函数 …… 34
菜单、工具条及状态条的 MFC 管理类 …… 34
把菜单、工具条和状态条加入主对话框 …… 39
1-3　公共对话框 …… 41
CFileDialog …… 41
CColorDialog …… 44
CFontDialog …… 52
CPrintDialog …… 59
CFindReplaceDialog …… 60
1-4　非模式对话框 …… 60
1-5　标签化对话框 …… 61
创建属性页 …… 61
创建属性表 …… 62
与属性表交互 …… 63
1-6　小结 …… 64

第 2 章　常用 MFC 类 …… 65

需要全本请访问 www.ertongbook.com



2-1 窗口公共基类 CWnd ····· 66
2-2 功能强大的字符串类 CString ····· 68
CString 的构造函数 ····· 69
CString 的基本操作 ····· 69
2-3 MFC 数组类 ····· 74
数组模板类 ····· 74
非模板数组类 ····· 78
2-4 MFC 链表类 ····· 80
链表模板类 ····· 80
非模板链表类 ····· 85
2-5 CPoint、CRect 和 CSize ····· 85
CPoint ····· 85
CSize ····· 86
CRect ····· 86
2-6 CTime 和 CTimeSpan ····· 89
CTime 类 ····· 89
CTimeSpan 类 ····· 91
2-7 CFile ····· 96
打开文件 ····· 97
文件读写 ····· 98
文件定位 ····· 98
文件相关信息的设置与获取 ····· 99
文件锁定与解锁 ····· 100
文件关闭 ····· 100
2-8 CArchive ····· 104
类的序列化 ····· 104
创建 CArchive 对象 ····· 105
关闭 CArchive ····· 105
CArchive 的读写操作 ····· 105
CArchive 的状态函数 ····· 106
2-9 CStdioFile 类 ····· 106
CStdioFile 的文本存取 ····· 106
用 CStdioFile 类存取文本 ····· 107
2-10 小结 ····· 109

第 3 章 SQL 语言 ····· 111

3-1 启动数据工程项目 ····· 112
3-2 查询语句 ····· 115
Select 语句 ····· 115

Where 子句 …… 119
Group by 子句 …… 125
Order by 子句 …… 126
Having 子句 …… 128
相关的函数 …… 130
3-3 更改记录的查询 …… 136
Select...Into 语句 …… 136
Insert...Into 追加查询语句 …… 140
Update 更新查询语句 …… 144
Delete 语句 …… 146
3-4 表之间的连接 …… 148
Union 运算 …… 148
Left Join 和 Right Join 运算 …… 151
Inner Join 运算 …… 153
3-5 数据定义语言 …… 154
Crtate Table 语句 …… 155
Create Index 语句 …… 155
Alter Table 语句 …… 156
Drop 语句 …… 157
3-6 小结 …… 157

第 4 章 ODBC 程序设计 …… 159

4-1 ODBC 概述 …… 160
4-2 CDatabase 类 …… 161
构建函数 …… 161
属性函数 …… 164
操作函数 …… 166
重载函数 …… 167
CDatabase 类的数据成员 …… 167
4-3 CRecordset 类 …… 167
构建函数 …… 168
属性函数 …… 168
操作函数 …… 169
其他操作函数 …… 170
重载函数 …… 172
数据成员 …… 172
4-4 创建 ODBC 数据库应用程序 …… 173
用 AppWizard 创建应用程序框架 …… 173
为示例程序创建视图 …… 174



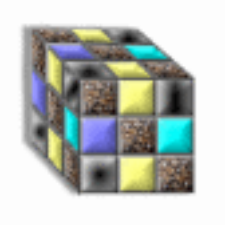
数据库的遍历 176
添加新记录 178
修改记录 179
删除记录 180
记录的筛选和排序 181
4-5 小结 189

第 5 章 DAO 技术 191

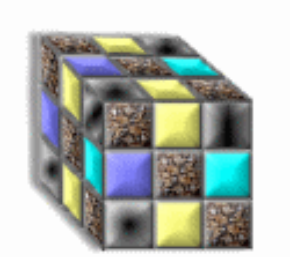
5-1 CDaoWorkspace 类 192
CDaoWorkspace 类概述 192
CDaoWorkspace 类的构造函数 192
工作区属性函数 192
工作区操作函数 194
数据库引擎属性函数 203
CDaoworkSpace 类的数据成员 206
5-2 CDaoDatabase 类 207
CDaoDatabase 类 207
CDaoDatabase 类的构造函数 207
数据库属性函数 207
数据库操作函数 212
CDaoDatabase 类的数据成员 222
5-3 CDaoTableDef 类 223
CDaoTableDef 类概述 223
CDaoTableDef 类的构建函数 223
表定义属性函数 224
表定义操作函数 245
CDaoTableDef 类的数据成员 249
5-4 CDaoQueryDef 类 249
CDaoQueryDef 类 249
CDaoQueryDef 类的构建函数 250
查询定义属性函数 251
查询定义操作函数 255
CDaoQueryDef 类的数据成员 259
CDaoRecordset 类 259
CDaoRecordset 类的构建函数 260
记录集属性函数 261
记录集更新函数 271
记录集定位函数 272
记录集杂项操作函数 276

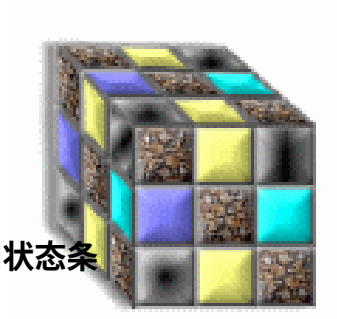
记录集重载函数 …… 283
CDaoRecordset 类的数据成员 …… 284
5-5 小结 …… 285
第 6 章　网络与多媒体 …… 287
6-1 MMControl 控件 …… 288
MMControl 控件属性 …… 289
媒体播放器的制作 …… 290
6-2 MCIWndX 控件 …… 308
MCIWndX 控件 …… 308
动画播放应用程序 …… 309
6-3 MP3 音乐播放器 …… 324
6-4 GIF 动画播放器 …… 332
6-5 网络应用程序设计 …… 343
6-6 小结 …… 355



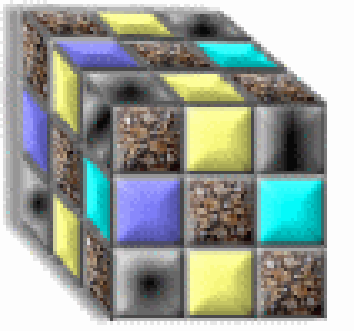


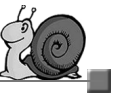
第 1 章
基本控件





- 常用控件
- 菜单、工具条和状态条
- 公共对话框
- 非模式对话框
- 标签化对话框





1-1 常用控件

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包括：

- Caption：静态文本框上所显示的字符串；
- Align text：文本框上字符串的对齐方式，有左对齐、居中对齐和右对齐；
- Center vertically：竖直方向的居中对齐；
- No prefix：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，No prefix 属性使“ & ”字符正常显示；
- No Wrap：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；

需要全本请访问 www.ertongbook.com

- 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，必须设置静态框本身的相应属性，比如调用 SetBitMap 显示位图时要设置 CStatic 的 SS_BITMAP 属性。

一个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。

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。

编辑框

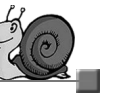
编辑框控件主要用于在程序中输入文本，它的功能由 CEdit 类来实现。

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：

- Multi-Line：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；
- Read-Only：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；



- Auto Vscroll：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；
- Horizontal scroll/Vertical scroll：为编辑框提供水平/竖直滚动条。

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。

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。

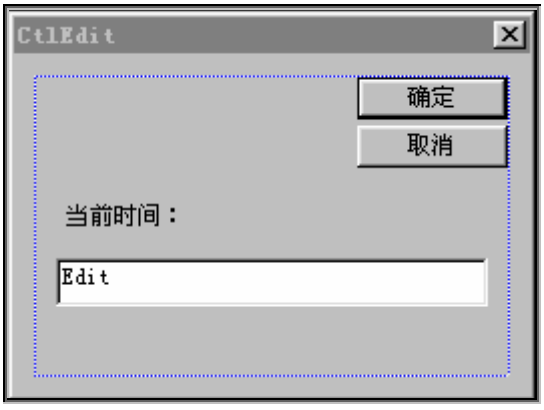
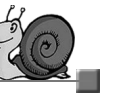


图 1-1　程序界面

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
```

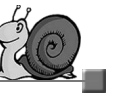
```
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

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)
    //{{AFX_MSG_MAP(CCtlEditDlg)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_WM_TIMER()
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CCtlEditDlg message handlers

BOOL CCtlEditDlg::OnInitDialog()
{
    CDialog::OnInitDialog();
    // Add "About..." menu item to system menu.
    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);
    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
```



```
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }
    // Set the icon for this dialog.  The framework does this automatically
    //  when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE);            // Set big icon
    SetIcon(m_hIcon, FALSE);           // Set small icon
    // TODO: Add extra initialization here
    SetTimer(1000,100,NULL);
    return TRUE;  // return TRUE  unless you set the focus to a control
}

void CCtlEditDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the code below
//  to draw the icon.  For MFC applications using the document/view model,
//  this is automatically done for you by the framework.
void CCtlEditDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting
        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);
```


需要全本请访问 www.ertongbook.com