C++ Builder 与 Windows API 经典范例

郑　明　郑世伟　编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

WinAPI 函数实例在 Windows 操作平台上具有跨程序语言的共性，也是除 C++ Builder 之外程序员必须参考的程序。本书内含 132 个 WinAPI 函数及 206 个完整、可执行的实例。在实例中，针对其中使用最频繁的 WinAPI 函数提出可行的实例，同时提供窗口程序员简易快捷的参考资料，让读者翻阅时可即时提醒思考方向，减少回忆函数用法的时间。

实例中还有利用多个不同函数协调完成一项功能的范例，可引导程序员使用函数用法。另外，还包含具有完整注释的鼠标和键盘的 Hook 连接文件与可执行文件程序代码。适合在 Windows 操作系统中，编写窗口程序而需使用 WinAPI 函数的程序员和其他相关技术人员使用或参考。

书　　名：C++ Builder 与 Windows API 经典范例
作　　者：郑　明　郑世伟　编著
责任编辑：马　丽
出 版 者：清华大学出版社(北京清华大学学研大厦，邮编 100084)
http://www.tup.tsinghua.edu.cn
印 刷 者：北京牛山世兴印刷厂
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16　**印张**：31.25　**字数**：750 千字
版　　次：2002 年 8 月第 1 版　2002 年 8 月第 1 次印刷
书　　号：ISBN 7-900643-29-X
印　　数：0001～5000
定　　价：49.00 元(含 1 张光盘)

目　录

第1章 文件函数

1.1 AreFileApisANSI 函数, SetFileApisToANSI 函数, SetFileApisToOEM 函数

以下 3 个函数，性质相似且相互依存出现，本节以相同范例演示。

- AreFileApisANSI
  函数语法格式：BOOL AreFileApisANSI (VOID)
  功能简易说明：判断 Win32 函数是否为 OEM/ANSI 字符集
- SetFileApisToANSI
  函数语法格式：VOID SetFileApisToANSI (VOID)
  功能简易说明：设置文件 API 为 ANSI 字符集
- SetFileApisToOEM
  函数语法格式：VOID SetFileApisToOEM (VOID)
  功能简易说明：设置文件 API 为 OEM 字符集

1. 范例目的

单击 Button1 按钮，判断现在程序是否为 ANSI 字符集或 OEM 字符集，并可转换 OEM 字符集或 ANSI 字符集。本范例参见附书光盘“配盘范例\Ch1_File Functions\ch01-01-1”文件夹。

2. 程序代码

Unit1.h

```
#01 //---------------------------------------------------------------------
#02 #ifndef Unit1H
#03 #define Unit1H
#04 //---------------------------------------------------------------------
#05 #include <Classes.hpp>
#06 #include <Controls.hpp>
#07 #include <StdCtrls.hpp>
#08 #include <Forms.hpp>
#09 #include <ComCtrls.hpp>
#10 #include <ToolWin.hpp>
#11 //---------------------------------------------------------------------
#12 class TForm1 : public TForm
#13 {
#14 __published:    // IDE-managed Components
#15     TToolBar *ToolBar1;
#16     TButton *Button1;
#17     TMemo *Memo1;
#18     TStaticText *StaticText1;
#19     TButton *Button2;
```

```
#20     TButton *Button3;p
#21     void __fastcall Button1Click(TObject *Sender);
#22     void __fastcall Button2Click(TObject *Sender);
#23     void __fastcall Button3Click(TObject *Sender);
#24 private:    // User declarations
#25 public:     // User declarations
#26     __fastcall TForm1(TComponent* Owner);
#27 };
#28 //---------------------------------------------------------------------
#29 extern PACKAGE TForm1 *Form1;
#30 //---------------------------------------------------------------------
#31 #endif
```

Unit1.cpp

```
#01 //---------------------------------------------------------------------
#02 #include <vcl.h>
#03 #pragma hdrstop
#04
#05 #include "Unit1.h"
#06 //---------------------------------------------------------------------
#07 #pragma package(smart_init)
#08 #pragma resource "*.dfm"
#09 TForm1 *Form1;
#10 //---------------------------------------------------------------------
#11 __fastcall TForm1::TForm1(TComponent* Owner)
#12     : TForm(Owner)
#13 {
#14 //默认按钮状态
#15 Button1->Enabled = true;
#16 Button2->Enabled = false;
#17 Button3->Enabled = false;
#18 }
#19 //---------------------------------------------------------------------
#20 void __fastcall TForm1::Button1Click(TObject *Sender)
#21 {
#22 //删除 Memo1 中所有文字
#23 Memo1->Text = "";
#24 //判断现在程序是否为 ANSI 字符集或 OEM 字符集
#25 //这项判断功能对 8 位输入输出设备极为有用
#26 if(AreFileApisANSI())
#27    {
#28    Memo1->Lines->Add("现在程序是 ANSI 字符集！");
#29    Memo1->Lines->Add("为 WIN32 函数使用。");
#30    Memo1->Lines->Add("若单击 Button2 按钮，则转成 OEM 字符集，");
#31    Memo1->Lines->Add("可以直接由 8 位输入输出设备使用。");
#32    Button2->Enabled = true;
#33    Button3->Enabled = false;
#34    }
#35 else
#36    {
#37    Memo1->Lines->Add("程序是 OEM 字符集！");
#38    Memo1->Lines->Add("可以直接由 8 位输入输出设备使用。");
#39    Memo1->Lines->Add("若单击 Button3 按钮，则转成 ANSI 字符集，");
#40    Memo1->Lines->Add("为 WIN32 函数使用。");
#41    Button2->Enabled = false;
```

```
#42    Button3->Enabled = true;
#43    }
#44 Memo1->Lines->Add("以上都会影响现有函数返回的结果。");
#45 Button1->Enabled = false;
#46 }
#47 //---------------------------------------------------------------------
#48 void __fastcall TForm1::Button2Click(TObject *Sender)
#49 {
#50 //由 ANSI 变成 OEM
#51 SetFileApisToOEM();
#52 Memo1->Text = "由 ANSI 字符集转成 OEM 字符集";
#53 //重设按钮状态
#54 Button1->Enabled = true;
#55 Button2->Enabled = false;
#56 Button3->Enabled = false;
#57 }
#58 //---------------------------------------------------------------------
#59 void __fastcall TForm1::Button3Click(TObject *Sender)
#60 {
#61 //由 OEM 变成 ANSI
#62 SetFileApisToANSI();
#63 Memo1->Text = "由 OEM 字符集转成 ANSI 字符集";
#64 //重设按钮状态
#65 Button1->Enabled = true;
#66 Button2->Enabled = false;
#67 Button3->Enabled = false;
#68 }
#69 //---------------------------------------------------------------------
```

3. 运行结果

程序执行前画面如图 1.1 所示，程序执行后画面如图 1.2 所示。

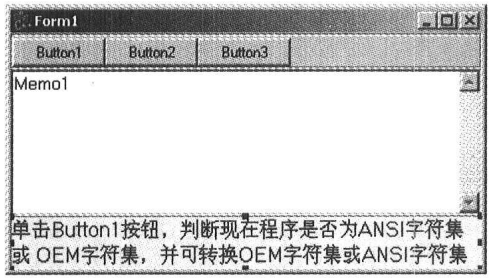

图 1.1

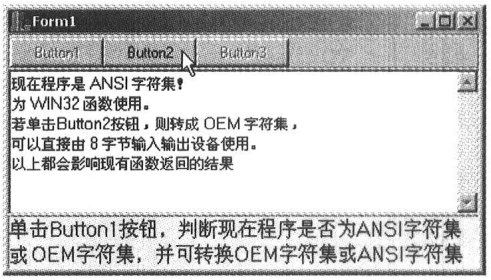

图 1.2

1.2　CopyFile 函数

函数语法格式：BOOL CopyFile(LPCTSTR lpExistingFileName, LPCTSTR lpNewFileName, BOOL bFailIfExists)

功能简易说明：复制指定文件

1. 范例目的

单击按钮，将上方选取的文件复制到下方目录中。本范例参见附书光盘“配盘范例\Ch1_File Functions\ch01-02-1”文件夹。

2. 程序代码

Unit1.h

```
#01 //---------------------------------------------------------------------------
#02 #ifndef Unit1H
#03 #define Unit1H
#04 //---------------------------------------------------------------------------
#05 #include <Classes.hpp>
#06 #include <Controls.hpp>
#07 #include <StdCtrls.hpp>
#08 #include <Forms.hpp>
#09 #include <ComCtrls.hpp>
#10 #include <FileCtrl.hpp>
#11 //---------------------------------------------------------------------------
#12 class TForm1 : public TForm
#13 {
#14 __published:   // IDE-managed Components
#15     TButton *Button1;
#16     TStaticText *StaticText1;
#17     TDirectoryListBox *DirectoryListBox1;
#18     TDriveComboBox *DriveComboBox1;
#19     TFileListBox *FileListBox1;
#20     TEdit *FileEdit;
#21     TFilterComboBox *FilterComboBox1;
#22     TDirectoryListBox *DirectoryListBox2;
#23     TFileListBox *FileListBox2;
#24     TEdit *Edit1;
#25     TFilterComboBox *FilterComboBox2;
#26     TDriveComboBox *DriveComboBox2;
#27     void __fastcall Button1Click(TObject *Sender);
#28
#29 private:   // User declarations
#30 public:        // User declarations
#31     __fastcall TForm1(TComponent* Owner);
#32 };
#33 //---------------------------------------------------------------------------
#34 extern PACKAGE TForm1 *Form1;
#35 //---------------------------------------------------------------------------
#36 #endif
```

Unit1.cpp

```
#01 //---------------------------------------------------------------------------
#02 #include <vcl.h>
#03 #pragma hdrstop
#04
#05 #include "Unit1.h"
#06 //---------------------------------------------------------------------------
#07 #pragma package(smart_init)
#08 #pragma resource "*.dfm"
#09 TForm1 *Form1;
```

```
#10  //---------------------------------------------------------------------
#11  __fastcall TForm1::TForm1(TComponent* Owner)
#12      : TForm(Owner)
#13  {
#14  }
#15  //---------------------------------------------------------------------
#16
#17  void __fastcall TForm1::Button1Click(TObject *Sender)
#18  {
#19  //汇集来源路径
#20  AnsiString target = DirectoryListBox2->Directory+"\\"+FileEdit->Text;
#21  //将指定文件复制到目的目录下
#22  if(::CopyFile(FileListBox1->FileName.c_str(),//目的路径
#23              target.c_str(),// 源路径文件
#24              true) == 0)
#25          {
#26          ShowMessage(" 指定目录已有相同文件，\n 所以无法复制文件！ ");
#27          }
#28  //将目的区域更新
#29  FileListBox2->Update();
#30  }
#31  //---------------------------------------------------------------------
```

3. 运行结果

程序执行前画面如图 1.3 所示，程序执行后画面如图 1.4 所示。

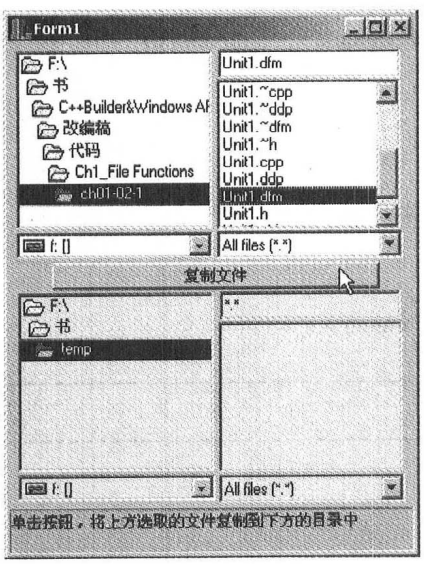

图 1.3

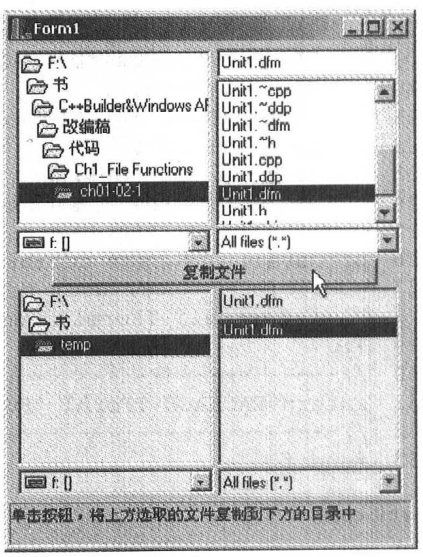

图 1.4

1.3　CreateDirectory 函数，CreateDirectoryEx 函数

函数语法格式：BOOL CreateDirectory(LPCTSTR lpPathName,LPSECURITY_ATTRIBU TES lpSecurityAttributes)

功能简易说明：建立一个新目录

1. 范例目的

单击按钮，在指定目录下增加一个新目录。本范例参见附书光盘“配盘范例\Ch1_File Functions\ch01-03-1”文件夹。

2. 程序代码

Unit1.h

```
#01 //---------------------------------------------------------------------------
#02 #ifndef Unit1H
#03 #define Unit1H
#04 //---------------------------------------------------------------------------
#05 #include <Classes.hpp>
#06 #include <Controls.hpp>
#07 #include <StdCtrls.hpp>
#08 #include <Forms.hpp>
#09 #include <FileCtrl.hpp>
#10 #include <Grids.hpp>
#11 #include <Outline.hpp>
#12 #include <ExtCtrls.hpp>
#13 //---------------------------------------------------------------------------
#14 class TForm1 : public TForm
#15 {
#16 __published:   // IDE-managed Components
#17     TStaticText *StaticText1;
#18     TDirectoryListBox *DirectoryListBox1;
#19     TDriveComboBox *DriveComboBox1;
#20     TBevel *Bevel1;
#21     TStaticText *StaticText2;
#22     TEdit *Edit1;
#23     TButton *Button1;
#24     void __fastcall Button1Click(TObject *Sender);
#25
#26 private:   // User declarations
#27 public:        // User declarations
#28     __fastcall TForm1(TComponent* Owner);
#29 };
#30 //---------------------------------------------------------------------------
#31 extern PACKAGE TForm1 *Form1;
#32 //---------------------------------------------------------------------------
#33 #endif
```

Unit1.cpp

```
#01 //---------------------------------------------------------------------------
#02 #include <vcl.h>
#03 #pragma hdrstop
#04
#05 #include "Unit1.h"
#06 //---------------------------------------------------------------------------
#07 #pragma package(smart_init)
#08 #pragma resource "*.dfm"
#09 TForm1 *Form1;
#10 //---------------------------------------------------------------------------
#11 __fastcall TForm1::TForm1(TComponent* Owner)
#12     : TForm(Owner)
#13 {
```

```
#14  }
#15  //---------------------------------------------------------------------
#16  void __fastcall TForm1::Button1Click(TObject *Sender)
#17  {
#18  //若已经输入新路径名称
#19  if(Edit1->Text !="")
#20     {
#21     AnsiString dir = DirectoryListBox1->Directory;
#22     //分辨是否在根目录下
#23     if(dir.Length()>4)
#24        dir = dir + "\\" + Edit1->Text;
#25     else
#26        dir = dir + Edit1->Text;
#27     //在程序中显示新目录的完整路径
#28     StaticText1->Caption = "新建目录路径: " + dir;
#29     //若新建目录失败
#30     if(!::CreateDirectory(dir.c_str(), NULL))
#31        ShowMessage("新建目录失败! ");
#32     //若新建目录成功，则更新显示
#33     DirectoryListBox1->Update();
#34     }
#35  else
#36     ShowMessage("空白无效! 请输入有效目录名称。");
#37  }
#38  //---------------------------------------------------------------------
```

3. 运行结果

程序执行前画面如图 1.5 所示，程序执行后画面如图 1.6 所示。

图 1.5

图 1.6

1.4　CreateFile 函数

函数语法格式：HANDLE CreateFile(LPCTSTR lpFileName,DWORD dwDesiredAccess, DWORD dwShareMode,LPSECURITY_ATTRIBUTES lpSecurityAttributes, DWORD dwCreationDistribution, DWORD dwFlagsAndAttributes, HANDLE hTemplateFile)

功能简易说明：打开、建立一个新文件或新对象

1.4.1　范例一

1. 范例目的

单击按钮，在指定目录下增加一个新文本文件(aaa01.txt)。本范例参见附书光盘“配盘范例\Ch1_File Functions\ch01-04-1”文件夹。

2. 程序代码

Unit1.h

```
#01 //---------------------------------------------------------------------------
#02 #ifndef Unit1H
#03 #define Unit1H
#04 //---------------------------------------------------------------------------
#05 #include <Classes.hpp>
#06 #include <Controls.hpp>
#07 #include <StdCtrls.hpp>
#08 #include <Forms.hpp>
#09 //---------------------------------------------------------------------------
#10 class TForm1 : public TForm
#11 {
#12 __published:   // IDE-managed Components
#13     TStaticText *StaticText1;
#14     TButton *Button1;
#15     void __fastcall Button1Click(TObject *Sender);
#16 private:   // User declarations
#17 public:        // User declarations
#18     __fastcall TForm1(TComponent* Owner);
#19 };
#20 //---------------------------------------------------------------------------
#21 extern PACKAGE TForm1 *Form1;
#22 //---------------------------------------------------------------------------
#23 #endif
```

Unit1.cpp

```
#01 //---------------------------------------------------------------------------
#02 #include <vcl.h>
#03 #pragma hdrstop
#04
#05 #include "Unit1.h"
#06 //---------------------------------------------------------------------------
#07 #pragma package(smart_init)
#08 #pragma resource "*.dfm"
#09 TForm1 *Form1;
#10 //---------------------------------------------------------------------------
#11 __fastcall TForm1::TForm1(TComponent* Owner)
#12     : TForm(Owner)
#13 {
#14 }
#15 //---------------------------------------------------------------------------
#16
#17 void __fastcall TForm1::Button1Click(TObject *Sender)
#18 {
#19
#20 AnsiString FileName="C:\\aaa01.txt";
#21 char TheText[MAX_PATH]="这是 CreateFile 函数测试文字内容。";
#22 HANDLE TheFile;
#23 DWORD BytesWritten;
#24
#25 TheFile=CreateFile(FileName.c_str(),
#26                     GENERIC_WRITE,FILE_SHARE_READ,
#27                     NULL,
#28                     CREATE_ALWAYS,
```

```
#29                     FILE_ATTRIBUTE_NORMAL,
#30                     NULL);
#31
#32  if(TheFile==INVALID_HANDLE_VALUE)
#33     {
#34     ShowMessage("无法建立新文件！");
#35     return;
#36     }
#37
#38  if(!WriteFile(TheFile,TheText,sizeof(TheText),&BytesWritten,NULL) ||
#39        sizeof(TheText)!=BytesWritten)
#40     {
#41     ShowMessage("文件无法写入新数据！");
#42     }
#43
#44  // 关闭一个打开的对象 handle
#45  ::CloseHandle(TheFile);
#46  }
#47  //---------------------------------------------------------------------------
#48
```

3. 运行结果

程序执行前画面如图 1.7 所示，程序执行后画面如图 1.8 和图 1.9 所示。

图 1.7

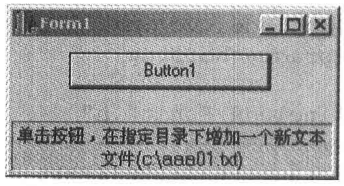

图 1.8

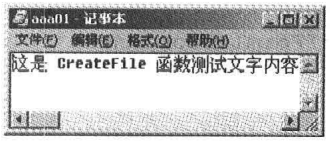

图 1.9

1.4.2 范例二

1. 范例目的

单击按钮，取得所有 COM 端口的设备状态。本范例参见附书光盘“配盘范例\Ch1_File Functions\ch01-04-2”文件夹。

2. 程序代码

Unit2.h

```
#01  //---------------------------------------------------------------------
#02  #ifndef Unit2H
#03  #define Unit2H
#04  //---------------------------------------------------------------------
#05  #include <Classes.hpp>
#06  #include <Controls.hpp>
```

```
#07 #include <StdCtrls.hpp>
#08 #include <Forms.hpp>
#09 #include <ComCtrls.hpp>
#10 #include <ToolWin.hpp>
#11 //---------------------------------------------------------------------------
#12 class TForm1 : public TForm
#13 {
#14 __published:   // IDE-managed Components
#15     TToolBar *ToolBar1;
#16     TButton *Button1;
#17     TStaticText *StaticText1;
#18     TListBox *ListBox1;
#19     void __fastcall Button1Click(TObject *Sender);
#20 private:   // User declarations
#21 public:        // User declarations
#22     __fastcall TForm1(TComponent* Owner);
#23 };
#24 //---------------------------------------------------------------------------
#25 extern PACKAGE TForm1 *Form1;
#26 //---------------------------------------------------------------------------
#27 #endif
```

Unit2.cpp

```
#01 //---------------------------------------------------------------------------
#02 #include <vcl.h>
#03 #pragma hdrstop
#04
#05 #include "Unit2.h"
#06 //---------------------------------------------------------------------------
#07 #pragma package(smart_init)
#08 #pragma resource "*.dfm"
#09 TForm1 *Form1;
#10 //---------------------------------------------------------------------------
#11 __fastcall TForm1::TForm1(TComponent* Owner)
#12     : TForm(Owner)
#13 {
#14 }
#15 //---------------------------------------------------------------------------
#16
#17 void __fastcall TForm1::Button1Click(TObject *Sender)
#18 {
#19 //删除所有项目
#20 ListBox1->Clear();
#21 //默认检测 COM 端口的返回值
#22 HANDLE h = INVALID_HANDLE_VALUE;
#23 //逐一进行 COM 端口检测
#24 for(int i=1;i<=32;i++)
#25    {
#26    AnsiString comname = "\\\\.\\COM" + String(i);
#27    //打开 COM端口，并返回检测值
#28    h = ::CreateFile( comname.c_str(),
#29                      GENERIC_READ | GENERIC_WRITE,
#30                      0,
#31                      0,
#32                      OPEN_EXISTING,
#33                      FILE_ATTRIBUTE_NORMAL,
#34                      0);
#35    //若无法打开指定 COM 端口
```

```
#36     if (h == INVALID_HANDLE_VALUE)
#37        {
#38        //取得错误信息
#39        DWORD err = ::GetLastError();
#40        //若 err 为 2 ，即此 COM 端口尚未安装
#41        if (err!=2)
#42           {
#43           ListBox1->Items->Add("COM" + AnsiString(i) +
#44                                " 尚未使用，错误信息为 (" +
#45                                AnsiString((int)err) + ")");
#46           }
#47        }
#48     else
#49        {
#50        ListBox1->Items->Add("COM" + AnsiString(i) + " 正在使用中！");
#51        }
#52     //释放 handle
#53     ::CloseHandle(h);
#54     }
#55
#56  }
#57  //---------------------------------------------------------------------------
```

3. 运行结果

程序执行前画面如图 1.10 所示，程序执行后画面如图 1.11 所示。

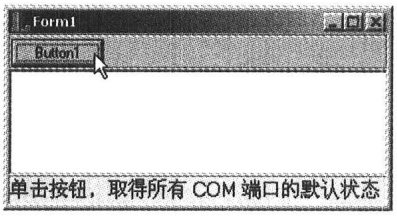

图 1.10

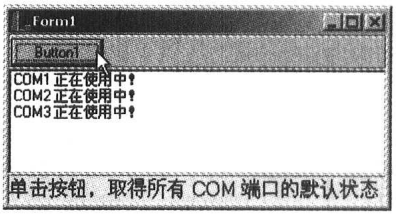

图 1.11

1.4.3 范例三

1. 范例目的

单击按钮，由上次打开“设置”文件时间重新设置系统时间。本范例参见附书光盘“配盘范例\Ch1_File Functions\ch01-04-3”文件夹。

2. 程序代码

Unit3.h

```
#01  //---------------------------------------------------------------------
#02  #ifndef Unit3H
#03  #define Unit3H
#04  //---------------------------------------------------------------------
#05  #include <Classes.hpp>
#06  #include <Controls.hpp>
#07  #include <StdCtrls.hpp>
#08  #include <Forms.hpp>
#09  #include <ComCtrls.hpp>
#10  #include <ToolWin.hpp>
```