

用 VC 实现控制面板应用程序

赵剑秋^{1,2}, 朱 明¹

(1. 中国科学院长春光机物理所, 中国科学院研究生院, 吉林 长春 130033;

2. 中国人民银行长春中心支行, 吉林 长春 130051)

摘 要: Windows 系统的控制面板是用来对计算机系统进行设置的一个工具集, 是由操作系统提供的。用户可以通过程序来实现定制自己的控制面板。每一个控制面板应用程序实际上是一个扩展名为 cpl 的动态链接库(DLL)组件程序。文中以控制鼠标属性为例, 详细介绍了用 VC++ 实现控制面板应用程序的原理和具体实现方法。

关键词: 控制面板; 组件程序; API; 回调函数

中图分类号: TP311.1

文献标识码: A

文章编号: 1673-629X(2006)06-0110-03

Creating Control Panel Applet with VC++

ZHAO Jian-qiu^{1,2}, ZHU Ming¹

(1. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science, Changchun 130033, China;

2. Changchun Central Branch of the People's Bank of China, Changchun 130051, China)

Abstract: Though Windows provides a number of standard control panel applets, people are allowed to add new ones to suit our requirement. A control panel applet resides in a DLL having an extension cpl. In this article will discuss how to create an applet that will allow you to swap mouse buttons and change typical mouse settings like mouse trails and double-click time.

Key words: control panel; components applet; API; callback function

1 Windows 系统控制面板应用程序的工作原理

Windows 操作系统为人们提供了一些标准的控制面板应用程序集, 通过这些应用程序可以完成诸如改变系统设置、桌面设置、配置网络协议等许多系统管理任务^[1]。控制面板组件是可以扩充的。事实上, 一些软件在安装过程中会自动加入新的控制面板组件。可以通过添加新的控制面板组件程序实现所需的功能^[2]。按照 Windows 用户界面设计原则的规定, 凡是影响到系统的整体行为和界面风格的各项参数都应该通过控制面板来设置, 因此掌握控制面板组件的编程方法是很有必要的。文中将以设置鼠标为例介绍用 VC 实现控制面板程序的方法。利用此方法, 读者可以在自己的应用程序添加自己的控制面板。

每一个控制面板应用程序实际上是一个扩展名为 cpl 的 dll 程序, 且都存放于“Windows\System”目录中, CONTROL.EXE 启动后会依次调入在系统目录下查找到的 CPL 库文件^[3]。每个 CPL 库必须输出一个名为 CPlApplet() 的回调函数供 CONTROL.EXE 调用。CPlApplet() 具有以下原型:

```
typedef LONG (APIENTRY * APPLET_PROC) (HWND  
hWndCpl, UINT msg, LONG lParam1, LONG lParam2)
```

所有发送给控制面板应用程序的消息由 CPlApplet 函数处理^[3]。可以看出, CPlApplet() 与普通窗口处理函数的形式很相似, 事实上, 控制面板正是以发送消息的方式与 CPL 库进行通信。参数 hWndCpl 为控制面板的窗口句柄, msg 为消息标识, lParam1 和 lParam2 为附加的两个参数, 具体的意义视 msg 的值而定。

控制面板用 LoadLibrary() 函数把 CPL 库调入内存以后, 立刻向 CPlApplet() 发送一条 CPL_INIT 消息, 指示 CPL 库做初始化工作。因为这是唯一允许返回失败信息的信息, 所以 CPL 库此时应该分配运行过程中需要的所有内存和资源, 如果因为内存不够或者其它原因不能继续, 就返回零值, 控制面板将不再处理这个 CPL 库, 并自动卸下它。由于 CPL 库可以支持多个组件, 所有 CPL 库初始化完毕后, 控制面板向每个 CPL 库的 CPlApplet() 函数发送一条 CPL_GETCOUNT 消息, 此时 CPL 库返回它所支持的组件数。接下来, 控制面板再针对每一个组件向 CPlApplet() 函数发送多条 CPL_NEWINQUIRE 消息, 目的是取得每个组件对应的图标、名称和提示信息, CPL 库可以在处理这条消息时依次初始化各个组件的对话框。

进行到这一步后, 控制面板显示出所有组件的图标, 并开始接受用户的选择。当用户双击某个组件的图标时, 控制面板就会向该组件所在 CPL 库发送一条 CPL_

收稿日期: 2005-10-09

作者简介: 赵剑秋(1970-), 男, 吉林长春人, 博士研究生, 工程师, 研究方向为图像处理; 朱 明, 研究员, 研究方向为图像处理与自动目标跟踪。

DBLCLK 消息,并指明用户选择的是该 CPL 库中的第几个组件,CPLApplet()函数将会显示相应组件。控制面板在被关闭时会对每个组件发送一条 CPL_STOP 消息,接着对每个 CPL 库发送一条 CPL_EXIT 消息,此时 CPL 库释放在 CPL_INIT 消息中分配的内存和资源。最后控制面板依次卸下各个 CPL 库并退出,这就是控制面板组件的工作原理^[4]。

2 控制面板开发程序开发实例

根据如上所述原理,就可以编写实现控制面板的应用程序。该程序的目的是在控制面板中添加一个能够控制鼠标各种属性的程序组件。利用 VC++ 编写控制面板程序的具体步骤如下:

1)利用 MFC 创建一个名为 SetMouse 的项目文件,应用程序的类型为使用 MFC 的 DLL,MFC 库作为静态库链接,以便于在没有 MFC 库的机器上运行该应用程序^[5]。

2)选择 Project 菜单下的 Settings 选项,在 Link 页中将输出文件名改为 \Setmouse.cpl,在 Debug 页中将“Executable for debug session”改为 windows \ System32 \ Control.exe,以便直接用控制面板运行。

3)用资源编辑器创建一个图标,用于控制面板中显示。

4)创建如图 1 所示的对话框,用 Class Wizard 创建一个 CmyDialog 类,为每个控件创建成员变量,如表 1 所示。

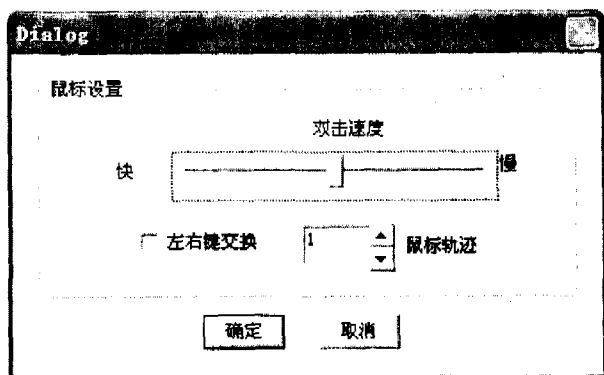


图 1 对话框界面

表 1 控件及其变量

控件	ID	类型	变量
Slider	IDC_SLIDER1	Control	m_sliderctrl
Slider	IDC_SLIDER1	Value(int)	m_sliderval
Spin	IDC_SPIN1	Control	m_spinctrl
Check Box	IDC_SWAP	Value(BOOL)	m_swap

5)为 MyDialog 的成员函数 OnInitDialog()添加如下代码:

```

BOOL mydialog::OnInitDialog()
{
    CDialog::OnInitDialog();
    m_sliderctrl.SetRange(0, 1000);
    m_sliderctrl.SetPos(500);
    m_spinctrl.SetRange(1, 100);

```

```

    m_spinctrl.SetPos(1);
    return TRUE; // return TRUE unless you set the focus to a
    control
    // EXCEPTION: OCX Property Pages should return FALSE
}

```

这段代码分别设置了 Slider 和 Spin 控件的范围和初始值。

6)添加 ONOK()处理函数,其代码如下:

```

void mydialog::OnOK()
{
    CDialog::OnOK();
    ::SystemParametersInfo(SPI_SETMOUSEBUTTONSWAP,
    m_swap, NULL, NULL);
    ::SystemParametersInfo(SPI_SETDOUBLECLICKTIME,
    m_sliderval, NULL, NULL);
    ::SystemParametersInfo(SPI_SETMOUSETRAILS, m_
    spinctrl.GetPos(), NULL, NULL);
}

```

这里,首先通过调用基类的 ONOK()函数来调用 UpdateDate(TRUE),UpdateDate(TRUE)函数将控件的值传递给与之相关的变量。然后,调用 API 函数::SystemParametersInfo(),该函数根据用户输入的数值对系统进行设置。

7)每一个 CPL 文件都需要输出一个回调函数 CPLApplet()作为应用程序的入口点。在 mousecpl.cpp 文件中的 CPLApplet()函数实现如下:

```

#include "stdafx.h"
#include <cpl.h>
#include "mousecpl.h"
#include "resource.h"
#include "mydialog.h"
LONG APIENTRY CPLApplet ( HWND cplhwnd, UINT
msg,
LONG lparam1, LONG lparam2 )
{
    CPLINFO * cplinfo = (CPLINFO*) lparam2;
    mydialog d ( CWnd::FromHandle ( cplhwnd ) );
    switch ( msg )
    {
        case CPL_DBLCLK:
            d.DoModal();
            return 0;
        case CPL_EXIT:
            return 0;
        case CPL_GETCOUNT:
            return 1;
        case CPL_INIT:
            return 1;
        case CPL_NEWINQUIRE:
            return 1;
    }
}

```

```

case CPL_INQUIRE:
    cplinfo -> idIcon = IDI_ICON1;
    cplinfo -> idInfo = IDINFO;
    cplinfo -> IDData = 0;
    cplinfo -> idName = IDNAME;
    return 0;
case CPL_SELECT:
    return 1;
case CPL_STOP:
    return 1;
default:
    break;
}
return 1;
}

```

其中 CPL_DBLCLK 消息通知 CPIApplet() 函数用户双击了对话框的图标。CPL_INQUIRE 消息通知 CPIApplet() 函数提供指定对话框的图标、名称等信息。当包含控制面板应用程序的 DLL 被装载后立即发送 CPL_INIT 消息。这些信息被存储在名为 CPLINFO 的结构当中, IDINFO, IDNAME 的值在字符串标中定义。

CPIApplet() 函数接收控制面板消息请求, 并处理这些请求完成应用程序的初始化、显示和管理对话框、关闭应用程序等工作。

8) 在 mousecpl.h 文件中声明 CPIApplet() 函数:

```

LONG WINAPI CPIApplet ( HWND cplhwnd, UINT
umsg, LONG lparam1, LONG lparam2 );

```

9) 在文件 setmouse.def 中加入 DLL 库名和输出函数名:

```

setmouse.def : Declares the module parameters for the DLL.
LIBRARY "setmouse"
DESCRIPTION 'setmouse Windows Dynamic Link Library'
EXPORTS;
;Explicit exports can go here
CPIApplet

```

编译并执行该工程文件, 将出现如图 2 所示的控制面

板窗口, 里面多了一个图标, 双击该图标就会看到刚才所设计的设置鼠标属性的对话框, 利用该对话框就实现了控制鼠标的功能。



图 2 控制面板窗口

以上程序在 VC++ 6.0, Windows XP 环境下调试通过。

3 结 论

通过上面的例子可以看出, 根据上述原理利用 VC++ 的 API 函数调用, 就可以实现定制自己的控制面板应用程序。

参考文献:

- [1] 康博创作室. Visual C++ 新起点——6.0 实用教程[M]. 北京: 机械工业出版社, 1998.
- [2] 清源计算机工作室. Visual C++ 6.0 开发宝典[M]. 北京: 机械工业出版社, 1999.
- [3] 齐舒创作室. Visual C++ 6.0 用户界面制作技术与应用实例[M]. 北京: 中国水利水电出版社, 1999.
- [4] 乌尼尔, 董海军. Visual C++ 经典例程分析[M]. 北京: 中国电力出版社, 2001.
- [5] 张 力. Visual C++ 高级编程[M]. 北京: 人民邮电出版社, 2002.

(上接第 109 页)

存储必将成为今后网络存储技术的主要发展方向。

总之, 基于 IP 存储协议的好处在于它不关心基础的传输机制是什么而直接提供了一条无需基于光线通道条件下, 实现高速存储网络的方案, 因此为企业提供了高性价比的网络存储解决方案, 总体来说, 随着条件的愈加成熟, 尤其是以太网技术的快速发展, 基于 IP 协议的综合存储将会迎来蓬勃的发展。

参考文献:

- [1] Farley M. Building Storage Networks. USA: Osborne/Mc-

Graw-Hill, 2000.

- [2] NIIT. 存储区域网概念与应用[M]. 北京: 人民邮电出版社, 2002.
- [3] 金 红, 王 煜. 网络存储技术在网络数据备份系统中的应用[J]. 高性能计算技术, 2003(6): 50-53.
- [4] SCSI over TCP/IP[Z]. [s.l.]: Hewlett-Packard Company, 2000. 3-5.
- [5] Clark T. IP SANs: A Guide to iSCSI, iFCP, and FCIP Protocols for Storage Area Network[M]. Boston, MA: Addison-Wesley, 2002. 47-86.