

# C++Builder中如何动态设置系统的显示模式

彭耶萍

(吉首大学 信息管理与工程学院,湖南 张家界 427000)

**【摘要】**本文介绍C++Builder中如何通过系统API函数ChangeDisplaySettings()设置指定显示设备的显示模式的过程。

**【关键词】**C++Builder;显示模式;API

**【中图分类号】**TP311.56 **【文献标识码】**A **【文章编号】**1673-1891(2010)01-0063-03

## 引言

在C++Builder的编程过程中如果需要动态调节系统的显示模式,可以通过系统API函数ChangeDisplaySettings()设置指定显示设备的显示模式。本文主要介绍了API函数的参数设定并给出一个具体示例。

## 1 函数ChangeDisplaySettings

函数ChangeDisplaySettings()的声明如下:

```
LONG ChangeDisplaySettings(  
LPDEVMODE lpDevMode, // 显示模式  
DWORD dwFlags //显示模式选项  
);
```

## 2 参数的设定

参数LPDEVMODE lpDevMode包含将要设置的现实模式信息的DEVMODE结构的指针,可以设置为下列成员:dmBitsPerPel、dmPelsWidth、dmPelsHeight、dmDisplayFlags、dmDisplayFrequency、dmPosition。参数DWORD dwFlags指定DEVMODE结构中的哪些成员将用于改变显示模式的设置。可以取下列值之一:

0:动态改变当前屏幕的显示模式。

CDS\_FULLSCREEN:临时设置。

CDS\_GLOBAL:设置结果对所有用户有效。

CDS\_NORESET:显示模式设置将被保存在注册表中,但不影响当前的显示模式,该标志必须与CDS\_UPDATEREGISTRY标志一起使用。

CDS\_RESET:表示显示模式设置将被改变,即使所请求的显示模式和当前的设置相同。

CDS\_SET\_PRIMARY:指定显示设备将成为主显示设备。

CDS\_TEST:测试所请示的显示模式是否可以被设置。

CDS\_UPDATEREGISTRY:动态改变当前屏幕的显示模式,并把该显示模式保存到注册表的配置信息中。

返回的值可能包括:

DISP\_CHANGE\_SUCCESSFUL:设置成功。

DISP\_CHANGE\_BADDUALVIEW:设置成功,Windows XP专用。

DISP\_CHANGE\_BADFLAGS:参数flags设置错误。

DISP\_CHANGE\_BADMODE:不支持的显示模式。

DISP\_CHANGE\_BADPARAM:参数设置有误。

DISP\_CHANGE\_FAILED:设置失败。

DISP\_CHANGE\_NOTUPDATED:无法更新注册表,仅适用于Windows NT/2000/XP。

DISP\_CHANGE\_RESTART:必须重新启动计算机才能使设置生效。

## 3 程序示例

本文程序使用系统函数ChangeDisplaySettings()设置指定显示设备的显示模式的部分代码如下:

收稿日期:2010-03-05

作者简介:彭耶萍(1981-),女,湖南龙山人,讲师,研究生,研究方向为数据挖掘。

1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

```

Void_fastcall TForm1::BitBtn1Click(TObject *Sender)
{
    If(MessageBox(Handle,“是否将当前显示模式设置为:1024*768*32?”,
“信息提示“,MB_YESNO)= =INDO)
return;
DEVMODE *lpDevMode;
lpDevMode=new DEVMODE;
int i=0,nindex;
BOOL Result=EnumDisplaySettings(NULL,i,lpDevMode);
While(Result)
{
    if (lpDevMode->dmPelsWidth= =1024&& lpDevMode->dmPelsHeight= =768&& lpDevMode->
dmBitsPerpel= =32)
    {
        lpDevMode->dmFields=DM_BITSPERPEL|DM_PELSWIDTH|DM_PELSHEIGHT;
        LONG lResult=changeDisplaySettings(lpDevMode,0);
        If(lResult= =DISP_CHANGE_SUCCESSFUL)
        MessageBox(Handle,“设置显示模式成功“,“信息提示“,MB_OK);
        else if(lResult= =DISP_CHANGE_RESTART)
        MessageBox(Handle,“设置显示模式成功“,“信息提示“,MB_OK);
        else
        MessageBox(Handle,“设置显示模式失败“,“信息提示“,MB_OK);
    }
    lpDevMode= =new DEVMODE;
    Result=EnumDisplaySettings(NULL,++i,lpDevMode);
}
delete lpDevMode;
}

Void __fastcall TForm1:BitBtn2Click(TObject *Sender)
{
    if(MessageBox(Handle,“是否将当前显示模式设置为:800*600*32?”,
“信息提示“,MB_YESNO)= =IDNO)
return;
DEVMODE *lpDevMode;
lpDevMode=new DEVMODE;
int i=0,nIndex;
BOOL Result=EnumDisplaySettings(NULL,i,lpDevMode);
while(Result)
{
    if(lpDevMode->dmPelsWidth= =800&& lpDevMode->dmPelsHeight= =600&&
lpDevMode->dmBitsPerPel= =32)
    {
        lpDevMode->dmFields=DM_BITSPERPEL|DM_PELSWIDTH|DM_PELSHEIGHT;
        LONG lResult=ChangeDisplaySettings(lpDevMode,0);
        if(lResult= =DISP_CHANGE_SUCCESSFUL)

```

```
MessageBox(Handle, “设置显示模式成功”, “信息提示”, MB_OK);  
else if(IResult==DISP_CHANGE_RESTART)  
MessageBox(Handle, “设置显示模式成功”, “信息提示”, MB_OK);  
else  
MessageBox(Handle, “设置显示模式失败”, “信息提示”, MB_OK);  
}  
lpDevMode=new DEVMODE;  
Result=EnumDisplaySettings(NULL, ++I, lpDevMode);  
}  
lpDevMode=new DEVMODE;  
Result=EnumDisplaySettings(NULL, ++i, lpDevMode);  
}  
delete lpDevMode;  
}
```

#### 注释及参考文献:

- [1]康向东. C++Builder 新起点[M]. 北京: 机械工业出版社, 2000.
- [2]陈周造, 陈灿煌. 精通 C++Builder5 程序设计高级教程[M]. 北京: 中国青年出版社, 2001.
- [3]洪国胜, 张建原, 洪月里. C++Builder 程序设计轻松上手[M]. 北京: 清华大学出版社, 2001.

## How to Dynamically Set the System's Display Mode in C++ Builder

PENG Ye-ping

(*Information Management and Engineering College, Jishou University, Zhangjiajie, Hunan 427000*)

**Abstract:** This article describes how to set the display mode of the specified display device through the system API functions `Change Display Settinge()` in C++Builder.

**Key words:** C++Builder; Display mode; API