

利用 MFC 在对话框中使用视图控制

崔华锋 (中国人民银行许昌分行 461000)

焦永梅 张俊锋 (郑州市京广南路 41 号院 450052)

1. 利用 MFC 作为视图控制的优点

对话框应用程序界面清晰直观,程序设计简单快捷,借助按钮、编辑框、滚动杆、组合框等通用控制可以很方便地实现人机交互,特别适合于开发一些中小型应用,因而在各种工具软件及 Windows 系统本身都有很普遍的应用.在 Microsoft Visual C++ 中,基于对话框的应用程序(Dialog-based Application)和单文档应用程序(Single Document Application)及多文档应用程序(Multiple Documents Application)一起被列为三种典型应用程序。更有甚者,在 VB、VFP 和 Delphi 中,几乎所有应用程序的界面都是以窗体(Form)即对话框的形式出现的。

尽管 Windows 系统提供了为数不少的通用控制,但总不能满足用户千差万别的应用要求,笔者在某项应用开发中就遇到了这样的问题,该应用程序主窗口为一个对话框,根据用户选择在对话框中及时显示相应的数据信息,所显示的数据信息包括文字,图表等。要显示这些数据信息,现有的各种通用控制都无能为力(使用静态控制虽然可以显示信息,但无法进行滚动浏览)。笔者曾试图采用动态连接库(DLL)或 ActiveX 控件技术来解决这一问题,虽然可行但都有相当难度,而且会对应用程序的安装和卸载带来麻烦.如果能够把 MFC 为我们提供的视图(如滚动视 CScrollView)作为一个控制(权且称其为视控制)放在对话框中,那么不仅数据显示问题迎刃而解,而且可以很好地利用滚动视图的滚动功能对显示内容进行浏览。

2. 视图控制的实现方法

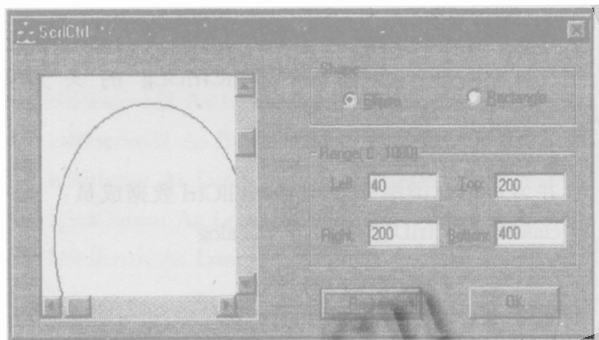
下面通过一个示例程序来具体介绍一下视图控制的实现方法。这个示例程序的执行画面如下图所示。用户通过圆形按钮和编辑框输入要绘制图形的形状和尺寸后,点击 Redraw 按钮即可在左侧视图控制内显示出图形.程序设计的具体步骤如下:

(1)创建一个基于对话框的应用程序,项目名不妨取为 Sctrl;

(2)按上图所示设计对话框,左侧视图控制所在的区域空白;

(3)利用 ClassWizard 为项目添加一个新类(Add Class...),命名为 CScrollCtrl,基类选为 CScrollView;

(4)打开新类 CScrollCtrl 的头文件 ScrollCtrl.h,把其构造函数和析构函数的属性由 protected 改为 public,并添加一些数据成员:



// ScrollCtrl.h : header file

```
class CScrollCtrl : public CScrollView
{
public:
    CScrollCtrl(); //构造函数
    virtual ~CScrollCtrl(); //析构函数
    int shape; //图形形状
    int left, top, right, bottom; //图形尺寸
    ....
};
```

(5)打开新类 CScrollCtrl 的实现文件 ScrollCtrl.Cpp,在其构造函数中对各数据成员赋初值:

// ScrollCtrl.cpp : implementation file

```

CScrollCtrl::CScrollCtrl()
{ shape = IDC-ELLIP; //椭圆形
  left = top = 10; //左上角坐标
  right = bottom = 100; //右下角坐标
}

在其绘制函数 OnDraw() 中添加绘图代码:
void CScrollCtrl::OnDraw(CDC * pDC)
{ //CDocument * pDoc = GetDocument();
  // TODO: add draw code here *
  switch(shape){
    case IDC-ELLIP: //绘制椭圆形
      pDC->Ellipse(left, top, right, bottom);
      break;
    case IDC-RECT: //绘制长方形
      pDC->Rectangle(left, top, right, bottom);
      break;
  }
}

```

(6) 打开对话框类 CScrlCtrlDlg 的头文件 ScrlCtrlDlg.h, 在其前面添上:

```

#include "ScrollCtrl.h"
并为对话框类添加一个 CScrollCtrl 数据成员:
class CScrlCtrlDlg : public CDialog
{ .....
public:
  CScrollCtrl m-sc;
  ...
};

```

(7) 打开对话框类 CScrlCtrlDlg 的实现文件 ScrlCtrlDlg.cpp, 在其初始化函数中加入如下代码以创建视图控制:

```

//ScrlCtrlDlg.cpp: implementation file
#define IDC-SCRLCTRL 101
.....
BOOL CScrlCtrlDlg::OnInitDialog()
{ .....
  // TODO: Add extra init.....
  //视控制尺寸

```

```

RECT rc = {20, 20, 200, 200};
//创建视图控制
m-sc.Create(NULL, NULL,
  WS-VISIBLE|WS-BORDER, rc,
  (CWnd*)this, IDC-SCRLCTRL);
//设置视图控制滚动尺寸
m-sc.SetScrollSizes(MM-TEXT,
  CSize(1000, 1000),
  CSize(100, 100), CSize(10, 10));
return TRUE; // return TRUE .....
}

```

(8) 利用 ClassWizard 为对话框中 Redraw 按钮的 BN_CLICKED 消息加入处理函数:

```

void CScrlCtrlDlg::OnRedraw()
{ // TODO: Add your control .....
  //获取用户设置
  m-sc.shape = GetCheckedRadioButton(
    IDC-ELLIP, IDC-RECT);
  m-sc.left = GetDlgItemInt(
    IDC-LEFT);
  m-sc.top = GetDlgItemInt(IDC-TOP);
  m-sc.right = GetDlgItemInt(
    IDC-RIGHT);
  m-sc.bottom = GetDlgItemInt(
    IDC-BOTTOM);
  //重新绘制视窗口
  m-sc.Invalidate(TRUE);
  m-sc.UpdateWindow();
}

```

(9) 利用 ClassWizard 为对话框 WM-DESTROY 消息加入处理函数:

```

void CScrlCtrlDlg::OnDestroy()
{ CDialog::OnDestroy();
  // TODO: Add your message .....
  m-sc.DestroyWindow(); //撤消视图控制
}

```

(10) 编译、连接并运行该应用程序。

(来稿时间: 1998 年 6 月)