**数字图像处理- MFC教程**

**第一课 建立MFC和打开bmp图片**

第一步：建立MFC工程

文件--新建--MFC AppWizardd(exe)工程--单文档—确定。

第二步：设置打开函数

查看--建立类导向(Ctrl+W)--CpictureXSView(类名)--ID\_FILE\_OPEN(IDs列表)--COMMAND(Messages列表)--默认成员函数名为OnFileOpen

--Member Functions(成员函数)中双击该函数进入函数编辑。

建立如下函数：打开图片、保存图片为重点。

|  |  |  |  |
| --- | --- | --- | --- |
| **对象标识符** | **消息** | **函数名** | **所属类** |
| **ID\_FILE\_OPEN** | **COMMAND** | **OnFileOpen** | **CpainterView** |
| **ID\_FILE\_SAVE** | **COMMAND** | **OnFileSave** | **CpainterView** |
| **ID\_FILE\_NEW** | **UPDATE\_COMMAND\_UI** | **OnUpdateFileNew** | **CpainterView** |
| **ID\_FILE\_MRU\_FILE1** | **UPDATE\_COMMAND\_UI** | **OnUpdateFileMruFile1** | **CpainterView** |

注意：在添加Messages列表中：

ON\_COMMAND是点了按钮或菜单项后的响应消息；

ON\_UPDATE\_COMMAND\_UI是用来表示对应的按钮和菜单项的状态的响应消息，比如你打开这个菜单时，处理这个菜单的状态，比如选中、变灰等。

UPDATE\_COMMAND\_UI处理程序的参数是个「指向CCmdUI对象的指针」，而CCmdUI对象就代表着对应的选单命令项，因此你只需呼叫 CCmdUI 所准备的，专门用来处理命令项外观的函式（如Enable或SetCheck）即可。

第三步：View类添加成员函数和成员变量

**//参数定义  
public:  
     CString BmpName;    //保存图像文件文件名  
     CString EntName;     //保存图像文件扩展名  
     CBitmap m\_bitmap;    //创建位图对象  
     void ShowBitmap(CDC\* pDC,CString BmpName);    //用来显示指定位图bmp的函数  
     BOOL ShowJpgGif(CDC\* pDC,CString strPath, int x, int y); //用来显示指定位图jpg、gif的函数**

添加方法：选中ClassView区--选中View类右键--增加成员变量函数。

第四步：为ID\_FILE\_OPEN消息添加打开函数

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*文件打开\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//**

**void CPictureMapView::OnFileOpen()**

**{**

**//四种格式的文件：bmp gif jpg tiff**

**CString filter;**

**filter="所有文件(\*.bmp,\*.jpg,\*.gif,\*tiff)|\*.bmp;\*.jpg;\*.gif;\*.tiff| BMP(\*.bmp)|\*.bmp| JPG(\*.jpg)|\*.jpg| GIF(\*.gif)|\*.gif| TIFF(\*.tiff)|\*.tiff||";**

**CFileDialog dlg(TRUE,NULL,NULL,OFN\_HIDEREADONLY,filter,NULL);**

**//按下确定按钮 dlg.DoModal() 函数显示对话框**

**if( dlg.DoModal() == IDOK )**

**{**

**//获取路径 文件名 扩展文件名**

**BmpName = dlg.GetPathName();**

**//BmpName.Format(\_T("%s",str)); //重点：错误 会出现错误的路径**

**//AfxMessageBox(BmpName); //显示图片的路径：打桩功能**

**EntName = dlg.GetFileExt();**

**EntName.MakeLower(); //此成员函数将此CString对象转换为一个小写字符串**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**/\*\* 使用CString处理字符串可处理Unicode、ANSI \*\*/**

**/\*\* CString中的Format进行字符串转换 \*\*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**Invalidate(); //调用该函数就会调用ONDRAW画图**

**}**

**}**

第五步：新建菜单和最近使用菜单不可用

MFC ClassWizard(ctrl+W)-- CpictureXSView(类名)--ID\_FILE\_NEW(IDs列表)—UPDATE\_COMMAND\_UI(Messages列表)—选择默认成员函数名--Member Functions(成员函数)中双击该函数进入函数编辑。

添加函数：

**//\*\*\*\*\*\*\*\*\*\*\*\* 使新建菜单不可用 \*\*\*\*\*\*\*\*\*//**

**void CPictViewView::OnUpdateFileNew(CCmdUI\* pCmdUI)**

**{**

**// TODO: Add your command update UI handler code here**

**pCmdUI->Enable(FALSE);**

**}**

同理：最近使用菜单

**//\*\*\*\*\*\*\*\*\*\*\*\* 使“最近的文件”不可用 \*\*\*\*\*\*\*\*\*//**

**void CPictViewView::OnUpdateFileMruFile1(CCmdUI\* pCmdUI)**

**{**

**// TODO: Add your command update UI handler code here**

**pCmdUI->Enable(FALSE);**

**}**

第五步：编写显示bmp格式图片函数

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*显示bmp格式图片\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//**

**void CPictureMapView::ShowBitmap(CDC\* pDC,CString BmpName)**

**{**

**//显示图片函数LoadImage**

**HBITMAP m\_hBitmap;**

**m\_hBitmap = (HBITMAP) LoadImage(NULL,BmpName,IMAGE\_BITMAP,0,0,LR\_LOADFROMFILE|LR\_DEFAULTSIZE|LR\_CREATEDIBSECTION);**

**//AfxMessageBox(BmpName);**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**/\*\* 1.要装载OEM图像，则设此参数值为0 \*\*/**

**/\*\* OBM\_ OEM：位图；OIC\_OEM图标；OCR\_OEM：光标 \*\*/**

**/\*\* 2.BmpName要装载图片的文件名 \*\*/**

**/\*\* 3.装载图像类型: IMAGE\_BITMAP-装载位图 \*\*/**

**/\*\* IMAGE\_CURSOR-装载光标 \*\*/**

**/\*\* IMAGE\_ICON-装载图标 \*\*/**

**/\*\* 4.指定图标或光标的宽度和长度 以像素为单位 \*\*/**

**/\*\* 此参数为零并且值LR\_DEFAULTSIZE没有被使用那么 \*\*/**

**/\*\* 函数使用目前的资源宽度 \*\*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**if( m\_bitmap.m\_hObject )**

**{**

**m\_bitmap.Detach(); //m\_bitmap为创建的位图对象**

**}**

**m\_bitmap.Attach(m\_hBitmap);**

**//边界**

**CRect rect;**

**GetClientRect(&rect);**

**//定义并创建一个内存设备环境**

**CDC dcBmp;**

**if( !dcBmp.CreateCompatibleDC(pDC) ) //创建兼容性的DC**

**return;**

**BITMAP m\_bmp; //临时bmp图片变量**

**//MessageBox("图片正在导入!");**

**m\_bitmap.GetBitmap(&m\_bmp); //将图片载入位图中**

**//MFC afxwin1.inl 错误 GetBitmap中**

**CBitmap \*pbmpOld = NULL;**

**dcBmp.SelectObject(&m\_bitmap); //将位图选入临时内存设备环境**

**//图片显示调用函数BitBlt**

**pDC->BitBlt(0,0,m\_bmp.bmWidth,m\_bmp.bmHeight,&dcBmp,0,0,SRCCOPY);**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**/\*\* 参数x、y位图目标矩形左上角x、y的坐标值 居中 \*\*/**

**/\*\* nWidth、nHeigth位图目标矩形的逻辑宽度和高度 \*\*/**

**/\*\* pSrcDC表示源设备CDC指针 \*\*/**

**/\*\* xSrc、ySrc表示位图源矩形的左上角的x、y逻辑坐标值 \*\*/**

**/\*\* dwRop表示显示位图的光栅操作方式 \*\*/**

**/\*\* SRCCOPY用于直接将位图复制到目标环境中 \*\*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**dcBmp.SelectObject(pbmpOld); //恢复临时DC的位图**

**DeleteObject(&m\_bitmap); //删除位图**

**dcBmp.DeleteDC(); //删除后台DC**

**//Invalidate(); //一直调用OnDraw显示图片**

**}**

第六步：实现图片显示

**void CPictureMapView::OnDraw(CDC\* pDC)**

**{**

**CPictureMapDoc\* pDoc = GetDocument();**

**ASSERT\_VALID(pDoc);**

**// TODO: add draw code for native data here**

**/\*在OnDraw函数中实现图片的最后显示功能\*/**

**if(!pDoc)**

**return;**

**if( EntName.Compare(\_T("bmp")) == 0 ) //bmp格式**

**{**

**ShowBitmap(pDC,BmpName);**

**}**

**else if( EntName.Compare(\_T("jpg")) == 0 )**

**{**

**//ShowJpgGif(pDC,BmpName,1,1);**

**}**

**}**