MDI多视图的编程

一、原理

为一个多文档应用程序加入多个视图窗口，这些视图都是属于同一个文档的。基本思想是添加多个文档类模板，添加命令菜单，为其添加响应函数，以显示相应的视图窗口。

二、实现

1、为APP类添加相应的模板类指针变量

2、在APP的InitInstance()函数中实例化文档类模板指针

3、添加菜单命令，并为其添加响应函数。

4、默认启动程序后会自动创建新文档，需屏蔽之，再改写OnFieNew函数。

三、实例

1、为APP类添加两个模板类指针变量

public:

CMultiDocTemplate \*m\_pTemplateDocB;

CMultiDocTemplate \*m\_pTemplateDocA;

2、在InitInstance()函数中，改写文档类模板指针实例化代码，如下：

m\_pTemplateDocA = new CMultiDocTemplate(IDR\_MDIMultiViewDemTYPE,

RUNTIME\_CLASS(CMDIMultiViewDemoDoc),

RUNTIME\_CLASS(CChildFrame),

RUNTIME\_CLASS(CViewA));

if (!m\_pTemplateDocA)

return FALSE;

AddDocTemplate(m\_pTemplateDocA);

m\_pTemplateDocB = new CMultiDocTemplate(IDR\_MDIMultiViewDemTYPE,

RUNTIME\_CLASS(CMDIMultiViewDemoDoc),

RUNTIME\_CLASS(CChildFrame),

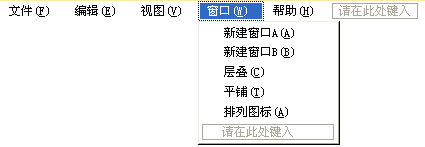
RUNTIME\_CLASS(CViewB));

if (!m\_pTemplateDocB)

return FALSE;

AddDocTemplate(m\_pTemplateDocB);

3、在IDR\_XXXTYPE菜单的窗口菜单下，添加菜单项：



添加函数代码：

void CMainFrame::OnWindowNewa()

{

CMDIChildWnd\* pActiveChild = MDIGetActive();

CDocument\* pDocument;

if (pActiveChild == NULL ||

(pDocument = pActiveChild->GetActiveDocument()) == NULL)

{

TRACE(traceAppMsg, 0, "Warning: No active document for WindowNew command.\n");

AfxMessageBox(AFX\_IDP\_COMMAND\_FAILURE);

return; // command failed

}

CDocTemplate\* pTemplate =((CMDIMultiViewDemoApp\*)AfxGetApp())->m\_pTemplateDocA;

ASSERT\_VALID(pTemplate);

CFrameWnd\* pFrame = pTemplate->CreateNewFrame(pDocument, pActiveChild);

if (pFrame == NULL)

{

TRACE(traceAppMsg, 0, "Warning: failed to create new frame.\n");

return; // command failed

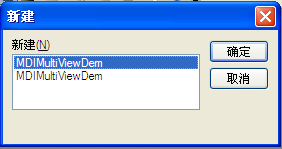
}

pTemplate->InitialUpdateFrame(pFrame, pDocument);

}

第二个菜单项的响应函数类似，只是把m\_pTemplateDocA改成了m\_pTemplateDocB，这段代码是从MFC的OnWindowNew中拷贝过来，然后改写的。

4、程序默认启动后会启动如下对话框，让用户选择文档类模板，然后创建一个新的文档：



由对话框可见，对话框是中列表元素是以程序名来标识的，而两个文档类模板都属于同一个程序，所以并不友好，为避免发生这种情况，需要进行如下处理：

4.1 设置cmdInfo变量：

cmdInfo.m\_nShellCommand=CCommandLineInfo::FileNothing;

进行如上设置后，启动程序后不再自动新建文档。

4.2 为CMainFrame和CChildFrame类添加创建新文档的OnFileNew函数：

void CMainFrame::OnFileNew()

{

CDocTemplate\* pTemplate = (CDocTemplate\*)((CMDIMultiViewDemoApp\*) AfxGetApp())->m\_pTemplateDocB;

ASSERT(pTemplate != NULL);

ASSERT\_KINDOF(CDocTemplate, pTemplate);

pTemplate->OpenDocumentFile(NULL);

}

void CChildFrame::OnFileNew()

{

CDocTemplate\* pTemplate = (CDocTemplate\*)((CMDIMultiViewDemoApp\*) AfxGetApp())->m\_pTemplateDocB;

ASSERT(pTemplate != NULL);

ASSERT\_KINDOF(CDocTemplate, pTemplate);

pTemplate->OpenDocumentFile(NULL);

}

这里的代码是仿照MFC默认的OnFileNew函数写的，这里为直接将m\_pTemplateDocB设置为默认的文档类模板，避免了上述的选择对话框，当然也可以自己创建一个对话框来让用户选择以哪个文档类为模板创建。

注意：文档类模板指针在ExInstance函数中不要主动删除，应为在默认的ExitInstance函数内部还会检查模板指针列表，调用AssertValid函数，如果将文档类指针删除，则列表中指针无效，于是报错。