**深入挖掘IDR\_MAINFRAME**-----王 王 著

http://www.360doc.com/content/10/0729/09/1066008\_42190420.shtml  
相信每个用过MFC的文档类的读者，特别是使用Application Wizard创建的单文档模板，都能在资源中的String Table（字符串资源）中找到IDR\_MAINFRAME这个ID，那么这是个什么呢，究竟有什么神奇的功效？待我们一步一步揭开它神秘的面纱。

首先，我们可以发现在字符串都是用很多“\n”隔开的，这些字符都用来表示什么呢？看来它的意义不简单啊，好的，我们使出第一招---MSDN  
  
所谓期望越大，失望也就越大，在MSDN中竟然没有它的身影，不得不让人感到失落，那么它到底是何方神圣，连“生死薄”都没有，看来来历不小。  
看来我们不得不使出第二招---在代码中搜索，即使MSDN没有，但是在代码中总会调用它的，如果没调用它，我想也没有什么特殊的意义了。  
  
  功夫不负有心人啊，很快就找到它了，这段代码存在于CWinApp::InitInstance（）中，于是我们不禁又要问CSingleDocTemplate又是什么类呢，我们MSDN它。  
  我们可以得到以下信息：  
  
  翻译一下，就是定义一个可以生效一个单文档接口的文档模板。它的基类就是CDocTemplate。  
我们再看它的构造函数：  
  
  也就是说，在这个类里传进去四个参数，根据信息我们可以知道它们分别是字符串资源ID，文档类指针，窗口框架类指针，视图类指针。  
我们重点关系第一个参数，先查一下nIDResource是用来干啥的，我们继续看。

翻译一下，指向一个与文档类型一起用的资源ID，它可能包括菜单，图标，加速键列表，字符串资源。这个字符串资源由多达用‘\n’隔离七个字符子链组成，这个字符描绘文档类型，更多信息请看CDocTemplate::GetDocString()。  
哦，我们知道了一个大概了，这七个字符串子链都用来表示文档类型，那么又有问题了，它们都分别表示什么文档类型呢？我们把问题放在这里。（对于后面的三个参数，就是把它们连接起来，由于与本主题关系不大，所以就不深究了。）  
现在，我们可以关心一下它的父类，特别是前面提到的那个成员函数GetDocString()，关于它的介绍如下：  
  
 翻译一下，获得一个被文档类型关联的字符串。  
  我们重点关心index,这是一个枚举量，总共有七个成员，有眉目了，这个七个成员想必就与前面提到的问题有关了，我们看描述。  
  
大概说明一下：  
CDocTemplate::windowTitle  
  主窗口标题栏上的字符串，（仅在SDI程序出现，MDI程序将以IDR\_MAINFRAME字符串为默认值。）  
CDocTemplate::docName  
  缺省文档的名称。缺省是无标题。  
CDocTemplate::fileNewName  
  文档类型的名称。如果应用程序支持多种类型的文档，此字符串将显示在"File/New"对话框中。如果没有指定，就不能够  
CDocTemplate::filterName  
  文档类型的描述和一个适用于此类型的通配符过滤器。这个字符串将出现在“File/Open”对话框中的文件类型列表框中。要和CDocTemplate::filterExt一起使用。  
CDocTemplate::filterExt  
  文档的扩展名。如果没有指定，就不能够在“File/Open”对话框中处理这种文档。要和CDocTemplate::filterName一起使用。  
CDocTemplate::regFileTypeId  
  如果你以::RegisterShellFileTypes向系统的注册表注册文件类型，此值会出现在HKEY\_CLASSES\_ROOT之下成为其子项，并仅供Windows内部使用。如果没有指定，这种文件类型就无法注册。  
CDocTemplate::regFileTypeName  
  这也是存储在注册表中的文件类型名称。它会显示于程序中用以访问注册表的对话框内。  
  终于挖掘到它的意义了，是不是很有成就感呢，其实相对于结果来说，更重要的是我们得到这个结果的过程。