## MFC六大关键技术

这一篇主要讲解一下MFC中的6个主要的关键技术，整个Application Framework其实从本质上来说都是建立在这6个关键技术上的。所以能够了解它们的原理对理解MFC整个框架和程序设计有非常大的好处。

1、MFC程序的初始化工作

在MFC中所有的类都来源于一个基类：CObject。MFC程序初始化过程中，其实就是虚函数的调用的过程，分清调用执行的到底是哪一个具体的虚函数，是父类的虚函数，还是基类的虚函数，都是至关重要的。

2、RTTI运行时类型识别

MFC程序运行过程中需要对类的类型进行动态的判断。在实现这一个机制的方法是，在MFC中的每一个类中都有一个CRuntimeClass类（《深入浅出MFC》中是这么命名，具体MFC中具有这个功能的类的真实名称，我还不知道），它用来记录类的基本信息，里面包含了必要的一些数据信息（比如类的名称，父类指针，对象大小，以及维护一个链表的必须指针），最后通过将所有类的CRuntimeClass以串行的方式串联起来，构成一个庞大的“类记录表”。这样在实际的程序运行时，当要判断当前类的信息时，就只要逐个比较类记录表中各个类的信息就可以确定到底是哪个类了。

3、Dynamic Creation（动态创建）

所谓的动态创建，我的理解是：在程序运行时，根据类名称来创建类对象，而这种动态创建的实现还是需要依赖于前面的RTTI中维护的“类记录表”。

4、Persistence机制

这个机制简单的理解，就是一种能够将所有类型的类记录到文件或者从文件读取到类的机制（可能理解的还不够透彻，欢迎指正）。

5、Message Mapping消息映射机制

MFC程序的执行是依赖于消息传递的。所谓的Message Map（消息映射表）其实就是在MFC整个类继承关系图中消息传递攀爬的一个图，这个图中有的消息是从子类攀爬到父类这样向上走的，也有的消息会产生旁流的情况。从形式上来看，是通过类似于RTTI、Dynamic Creation中定义宏的方式，简化消息映射定义的方式的，如：DECLARE\_MESSAGE\_MAP、BEGIN\_MESSAGE\_MAP、ON\_COMMAND、END\_MESSAGE\_MAP。

6、Command Routing命令绕行

MFC 对于消息绕行的规定是： 如果是一般的Windows 消息（WM\_xxx），一定是由衍生类别流向基础类别，没有旁流的可能。

如果是命令消息WM\_COMMAND，就有奇特的路线了：