主题：.NET 组件和COM组件的交互

一、区别

|  |  |  |
| --- | --- | --- |
|  | COM组件 | .Net组件 |
| 元数据  Metadata | 1.包含在IDL文件中。  Interface Definition Language | 1. 包含在组件中。 2. 可以通过自定义特性扩展。 |
| 释放内存 | 通过AddRef/Release释放 | 通过垃圾回收机制 |
| 接口 | 1. 自定义接口Custom   QueryInterface  AddRef  Release  CustomMethod1  CustomMethod2   1. 调度接口Dispatch   QueryInterface  AddRef  Release  GetTypeInfoCount  GetIDsOfNames  Invoke   1. 双重接口Dual   QueryInterface  AddRef  Release  GetTypeInfoCount  GetIDsOfNames  Invoke  CustomMethod1  CustomMethod2 |  |
| 方法绑定 | 1. 通过IDispatch实现后绑定 | 1.通过反射实现后绑定 |
| 数据类型 | 通过VARIANT数据作为参数 | 通过Object作为参数 |
| 注册 | 注册表HKCR | 共享程序集 |
| 线程 | STA 只允许一个线程访问  MTA 多线程可以同时访问 |  |
| 错误处理 | 通过HRESULT返回值  ISupportErrorInfo | 通过异常 |
| 事件 | IConnectionPointContainer | Event/Delegate |
| 封送 | SAFEARRAY  VARIANT  BSTR  IUnknown\*  IDispatch\* | Array  Object  String  Object  Object |

二、COM对象的创建(ATL)

1. IDL 文件中
2. 声明接口(原数据、接口类型)。
3. 声明COM对象实现的接口及默认的接口。
4. 在COM对象的头文件中
5. 多重继承的接口声明IDispatchImpl<要实现的接口名字,uuid,..>
6. BEGIN\_COM\_MAP中完成IDispatch接口映射
7. 接口函数的声明和实现
8. COM对象事件的声明和实现
9. 声明dispinterface 接口
10. 在COM对象中声明要实现dispinterface接口
11. COM对象实现类中Add Connection Point

三、C#调用COM组件

1.RCW(Runtime callable wrapper) 即生成Interop.接口.dll文件给c#调用

在这个文件里面会完成对IUnknown/IDispatch接口的封装，和引用计数的封装。

2.如果想显示的回收COM组件，则需要调用Marshal.ReleaseComObject(\*);

使用者不需要显式释放COM组件，如果显式调用，则所有的引用都不可以使用。

1. COM对象有STA/MTA

四、.Net COM组件

1.通过COMVisible(true)确定COM的接口和对象

五、.NET调用dll(C++)

可以通过dumpbin /exports <dll路径> | more

导出他所有的导出函数及参数类型