

# COM实用入门教程

## 第五讲

主讲人： 阙海忠

VC知识库网站 ([www.vckbase.com](http://www.vckbase.com)) 拍摄制作

# 这一讲，主要讲解如下要点：

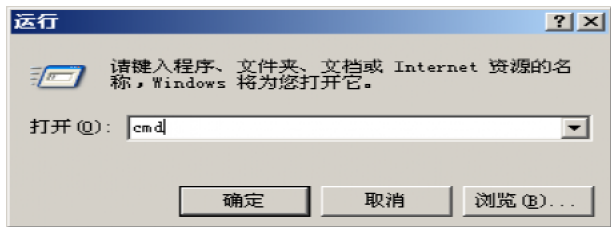
- 一、COM组件的注册；
- 二、智能指针的简介；
- 三、智能指针的使用；
- 四、智能指针与COM接口指针的比较；
- 五、智能指针的注意点。

# COM组件的注册

- 我们目前用ATL制作出来的简单对象是\*.dll的形式，我们在上一讲中能调用这个dll，是因为我们的Visual Studio在编译生成这个dll时，对它进行了注册。

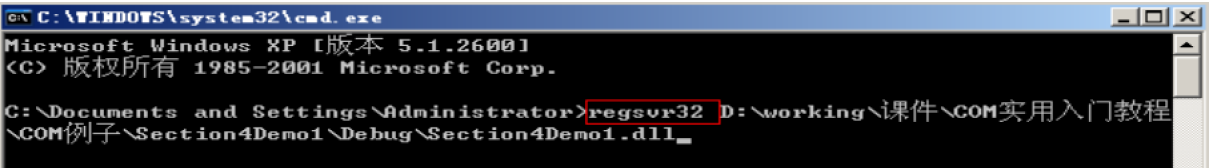
# COM 组件的注册

- 如何对COM组件进行手动注册？  
打开我们的控制台窗口



# COM组件的注册

- 输入如下信息，按Enter键



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>regsvr32 D:\working\课件\COM实用入门教程\COM例子\Section4Demo1\Debug\Section4Demo1.dll_
```

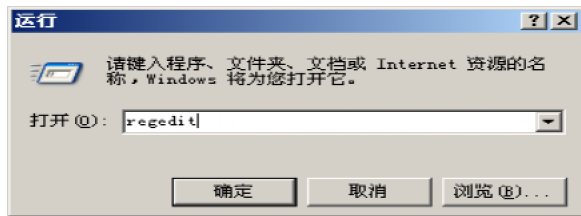
# COM组件的注册

- 提示注册成功！



# COM 组件的注册

- COM 组件被注册后，注册信息在哪里？  
打开注册表编辑器。



# COM组件的注册

打开HKEY\_CLASSES\_ROOT\CLSID\  
搜索我们组件的clsid，比如我这边是  
F65204D1-4E2E-494C-AB52-02D4D1D5438F，





# COM组件的注册



名称	类型	数据
默认	REG_SZ	D:\working\课件\COM实用入门教程\COM例子\Section4Demo1\Debug\Section4Demo1.dll
ThreadingModel	REG_SZ	Apartment

# COM组件的注册



名称	类型	数据
ab (默认)	REG_SZ	Section4Demo1.HelloSimpleObject.1

# COM组件的注册

- 我们从PropID项中，看到了我们COM组件的PropID，PropID是指程序员给某个CLSID指定一个程序员易记的名称。某些计算机语言是通过PropID来标识组件。
- PropID的命名规则：  
<Program>.<Component>.<Version>

# COM组件的注册

注册表的项:

HKEY\_CLASSES\_ROOT\Section4Demo1.HelloSimpleObject

也记录了COM组件的PropID及与之对应的CLSID。

# COM组件的注册

- 对COM组件进行注册的意义是什么？

我们刚才从注册表中看到了，注册信息记录了组件的CLSID，组件的路径，组件的PropID等。

而我们的COM组件是跨应用的，网页，MFC，C#的WinForm，VB需要识别并创建我们的组件。最终需要得到我们组件的路径。而路径是可能改变的，比如用户安装程序时选择不同目录，CLSID跟PropID是不变的，我们采用CLSID或PropID去创建组件，不管COM组件的dll在哪个位置，都跟我们的创建代码无关。

从vista开始，注册需要权限了，我们可以做一个不需要注册的COM。这个内容放到以后高级编程中来讲。

## 这一讲，主要讲解如下要点：

- 一、COM组件的注册；
- 二、智能指针的简介；
- 三、智能指针的使用；
- 四、智能指针与COM接口指针的比较；
- 五、智能指针的注意点。

# 智能指针简介

- C++在调用COM接口指针时是很危险的，因为使用过程需要每一个使用都严格并且正确地调用AddRef()与Release()方法。一旦出现问题，就会造成对象不能被正常释放或者对象被重复删除。
- 所以C++程序员使用COM接口时，得小心翼翼的。有没有办法改变这种现状，让我们的编程过程更简单呢？答案是有的，那就是，智能指针。

# 智能指针简介

- CComPtr是智能指针，是ATL提供的一个模板类，能封装COM接口的AddRef()与Release()方法。
- CComPtr声明出来的变量，是一个对象，这个对象封装了COM接口指针，这个对象的使用访问跟COM接口的使用方法几乎一样。

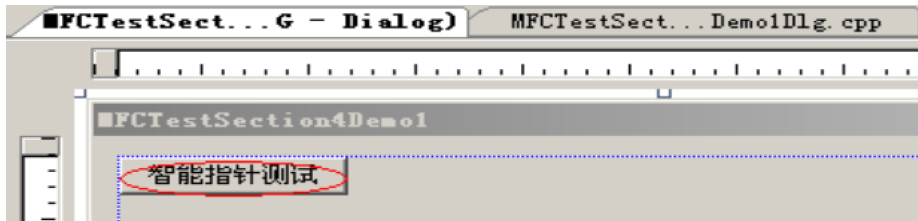


## 这一讲，主要讲解如下要点：

- 一、COM组件的注册；
- 二、智能指针的简介；
- 三、智能指针的使用；
- 四、智能指针与COM接口指针的比较；
- 五、智能指针的注意点。

# 智能指针使用

- 我们在实际的项目中来使用一下智能指针。打开上一讲中的**MFCTestSection4Demo1**项目。
- 在对话框中添加如下按钮，并添加按钮点击事件。



# 智能指针使用

- 我们采用智能指针的方式把之前 `CMFCTestSection4Demo1Dlg::OnInitDialog` 中的简单对象的测试代码的测试流程再写一遍。
- 我们发现智能指针与 **COM** 接口指针的用法很相似，但也有区别的地方。

# 这一讲，主要讲解如下要点：

- 一、COM组件的注册；
- 二、智能指针的简介；
- 三、智能指针的使用；
- 四、智能指针与COM接口指针的比较；
- 五、智能指针的注意点。

## 智能指针与COM接口指针的比较

- `CComPtr<IHelloSimpleObject> spHelloSimpleObject;` 创建了一个智能指针，其它是一个类对象，对象内部有一个 `IHelloSimpleObject*` 的指针变量，并且被初始化为 `NULL`。
- `IHelloSimpleObject *pIHelloSimpleObject,` 一个原始的COM接口指针。

# 智能指针与COM接口指针的比较

二者在CoCreateInstance中的使用方法一模一样，因为对智能指针的&操作，会变转换为对智能指针内部的IHelloSimpleObject \* 变量进行&操作。

## 智能指针与COM接口指针的比较

- 二者对->的操作的用法与意义一样。因为对智能指针的->操作会转换为对  
\_NoAddRefReleaseOnCComPtr<IHelloSimpleObjectSub> \*变量的->操作。  
\_NoAddRefReleaseOnCComPtr<IHelloSimpleObjectSub>是IHelloSimpleObjectSub的子类。

## 智能指针与COM接口指针的比较

- 不过智能指针不能执行`->AddRef()`与`->Release()`。因为`_NoAddRefReleaseOnCComPtr<T>`类把`AddRef()`与`Release()`方法的访问权限设置为`private`。如果智能指针执行`->AddRef()`与`->Release()`，编译时，直接报错。



# 智能指针与COM接口指针的比较

- 智能指针禁用->AddRef()与->Release()的原因是智能指针封装了COM接口指针的AddRef与Release操作，会智能判断何时对象内部会调用COM接口的AddRef，何时调用COM接口的Release。

## 智能指针与COM接口指针的比较

- 之前对COM接口指针的赋值是需要做AddRef操作来添加引用计数，如果采用智能指针，它会智能地执行AddRef我们不必为此担心。
- 之前不再使用COM接口指针时，需要做Release操作来减少引用计数，如果采用智能指针，它会在对象销毁时(析构时)，做COM接口指针的Release操作。

# 智能指针与COM接口指针的比较

智能指针的变量是一个对象。

如果是局部变量，将在这个局部变量生命期结束时，执行智能指针的析构函数；

如果是成员变量，将在成员所在的类对象销毁时，执行智能指针的析构函数；

如果是静态变量，将在程序结束时，执行智能指针的析构函数。

## 这一讲，主要讲解如下要点：

- 一、COM组件的注册；
- 二、智能指针的简介；
- 三、智能指针的使用；
- 四、智能指针与COM接口指针的比较；
- 五、智能指针的注意点。

# 智能指针的注意点

- 如果要释放一个智能指针，直接给它赋NULL值就可以了。这样内部的COM接口指针也会执行Release操作，来减少引用计数。

# 智能指针的注意点

- 当对智能指针取地址时(&运算符操作), 要确保智能指针为NULL, 因为&是要返回内部的COM接口指针的, 如果不为NULL, 则旧的COM接口指针将没有执行Release而直接赋值了一个旧的COM接口指针。
- 不过大家可以放心, 因为如果这时智能指针不为NULL时, 智能指针的代码会通过assert断言报错, 从而提醒程序员进行相应代码的更改。

# 智能指针练习

大家可以尝试采用智能指针的形式，改写 **Section2Demo1** 项目。我们会发现，改写后的新代码对接口的操作方法仍然不变，但不会发现 **AddRef** 与 **Release** 代码。

这个留给大家当练习，不过提醒大家一下，需要包含智能指针的头文件 **atlcomcli.h**。

# 回 顾

- 这一讲，主要讲解如下要点：
  - 一、COM组件的注册；
  - 二、智能指针的简介；
  - 三、智能指针的使用；
  - 四、智能指针与COM接口指针的比较；
  - 五、智能指针的注意点。