

CH 3. COM object (In-process server)



Eun-sung Lee

twoss@mmlab.net

[Multimedia Lab.](#)

Dept. of Electrical and Computer Eng.

University of Seoul

Seoul, Korea

0. Contents

1. Main Objectives
2. Overview
3. How to create DLL
4. Let's make a DLL
5. Reference

1. Main Objectives

- ◆ COM
- ◆ COM Class (In-Process server)



2. Overview

◆ COM Library

- COM Client Server Application
 가 API
- CLSID 가
 , 가
- Co
- Ex) CoInitialize, CoCreateInstance...
- COMPOBJ.DLL

2. Overview

◆ COM Object

1. COM Component

- **Regsvr32** Ex) regsvr32 addback.dll

2. COM Library [CoInitialize]

3. Interface COM Object CLSID

4. COM Object Instance [CoCreateInstance]

5. Interface Pointer

6. method ,

7. COM Library [CoUninitialize]

3. How to Create a DLL

1. **CoCreateInstance**(clsid, iid,...)

2. **CoGetClassObject**()

HKEY_CLASSES_ROOT CLSID clsid
CLSID COM InprocServer32(in-
process)

3. **CoLoadLibrary**()

dll load

4. **DllGetClassObject**()

clsid COM Class Factory

I ClassFactory pointer

I ClassFactory

iid pointer

->CoCreateInstance

I ClassFactory

I ClassFactory::CreateInstance

3. How to Create a DLL

CoCreateInstance(

 CLSID_FilterGraph,/** [In] Class Identifier(CLSID) of the object

 NULL, /** [In] Pointer to controlling IUnknown

 CLSCTX_INPROC,/** [In] Server Context

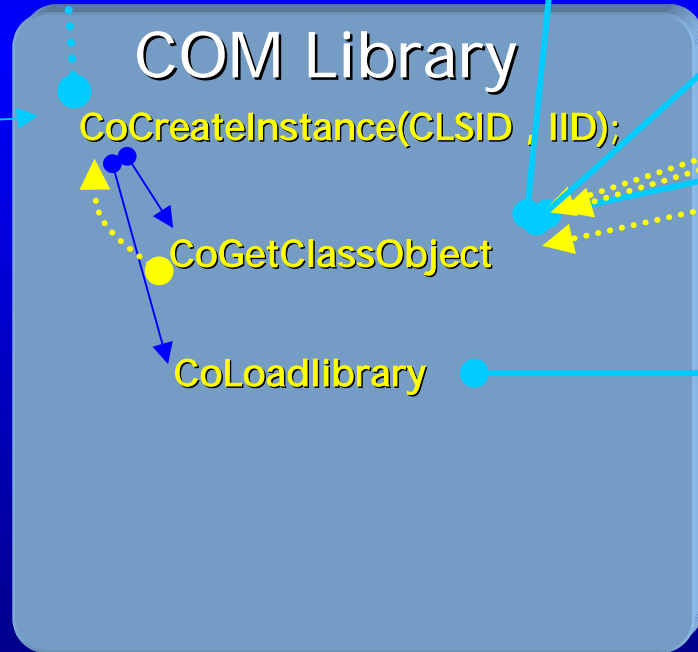
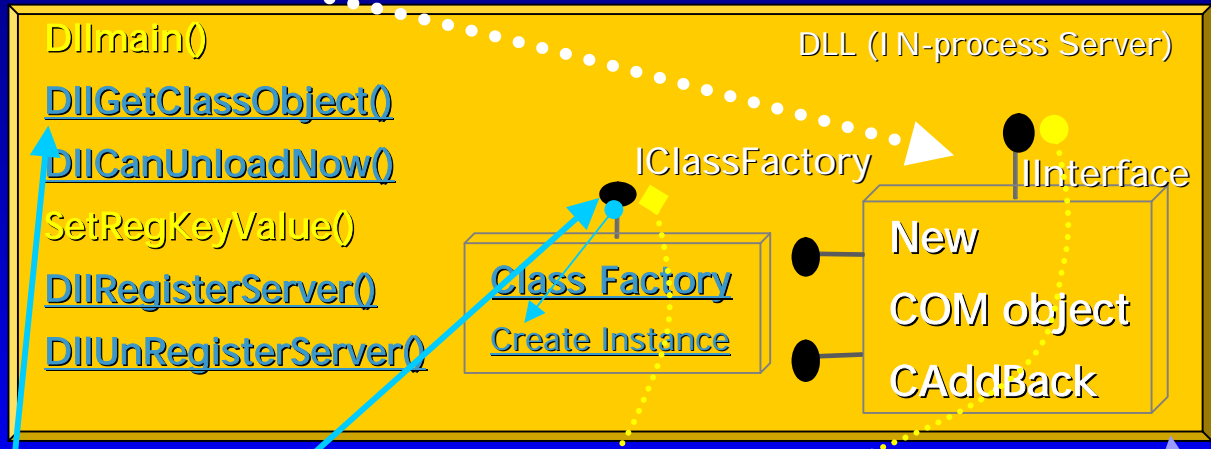
 IID_IGraphBuilder,/** [In] Reference to the identifier of the interface

 (void **)&pGraph);/** [Out] Address of pointer variable that receives the
 interface pointer requested in riid.

Use Case

Client

CoInitialize();
CoCreateInstance(CLSID , IID);
CoUninitialize();



System registry



3. How to Create a DLL - Class Factory

◆ Class Factory

- COM Class Instance
COM Class
- IClassFactory Interface
- IClassFactory Interface COM Interface
IUnknown Interface
- ClassFactory provides two methods :
 - CreateInstance, creates an uninitialized object of a specified CLSID
 - LockServer, locks the object's server in memory, allowing new objects to be created more quickly

3. How to Create a DLL - CreateInstance

- ◆ **I ClassFactory::CreateInstance**

- ◆ Creates an uninitialized object.

```
CLSID  
:  
(CoGetClassObject() CLSID)  
. ■
```

- ◆ **UNKNOWN.H**

```
HRESULT CreateInstance(  
    IUnknown * pUnkOuter, //Pointer to whether object is or isn't part  
    of an aggregate  
    REFIID riid, //Reference to the identifier of the interface  
    void ** ppvObject //Address of output variable that receives  
    the interface pointer requested in riid  
);
```



3. How to Create a DLL - LockServer

◆ IClassFactory :: LockServer()

➤ Client IClassFactory Interface Pointer
Client COM Object Dll
Unload .

```
HRESULT __stdcall CFAddBack::LockServer(BOOL bLock)
{
    if(bLock)
        ++g_cLocks;
    else
        --g_cLocks;

    return S_OK;
}
```



3. How to Create a DLL - DLL Functions

◆ DLL Functions

- **DllMain**: The DLL entry point. The name `DllMain` is a placeholder for the library- defined function name.
- **DllGetClassObject**: Creates a class factory instance.
 -
- **DllCanUnloadNow**: Queries whether the DLL can safely be unloaded.
- **DllRegisterServer**: Creates registry entries for the DLL. Call by `regsvr32.exe`
- **DllUnregisterServer**: Removes registry entries for the DLL. Call by `regsvr32.exe`



3. How to Create a DLL - DllGetClassObject

◆ DllGetClassObject

- COM class instance
COM class Class Factory COM
class Instance
- IClassFactory interface pointer client
- In-process server DllGetClassObject

3. How to Create a DLL

```
STDAPI DllGetClassObject(REFCLSID rclsid,
                        REFIID riid,
                        LPVOID* ppv) {
    HRESULT hr = CLASS_E_CLASSNOTAVAILABLE;
    IUnknown* pUnk = NULL;

    if(rclsid == CLSID_AddBack) {
        hr = E_OUTOFMEMORY;
        pUnk = new CAddBack;
    }

    if(pUnk != NULL) {
        hr = pUnk->QueryInterface(riid, ppv); //I ClassFactory pointer
        if(FAILED(hr))
            delete pUnk;
    }
    return hr;
}
```

3. How to Create a DLL

◆ DllRegisterServer/DllUnregisterServer

- In_process server
REGSVR32.exe
- `DllRegisterServer` export
.
- Ex. >> `regsvr32 <name.dll>`
- `DllUnregisterServer` export
.
- Ex. >> `regsvr32 /u <name.dll>`

3. How to Create a DLL

◆ module-definition (.def) file

- If you are not using the **__declspec(dllexport)** keyword to export the DLL's functions, then the DLL requires a .DEF file

: export all the DLL functions except for the entry-point function. The following is an example *.def file.

Ex>

```
LIBRARY "AddBack.Dll"
```

```
EXPORTS
```

```
    DllCanUnloadNow           @1      PRIVATE
    DllGetClassObject         @2      PRIVATE
    DllRegisterServer         @3      PRIVATE
    DllUnregisterServer       @4      PRIVATE
```

3. How to Create a DLL

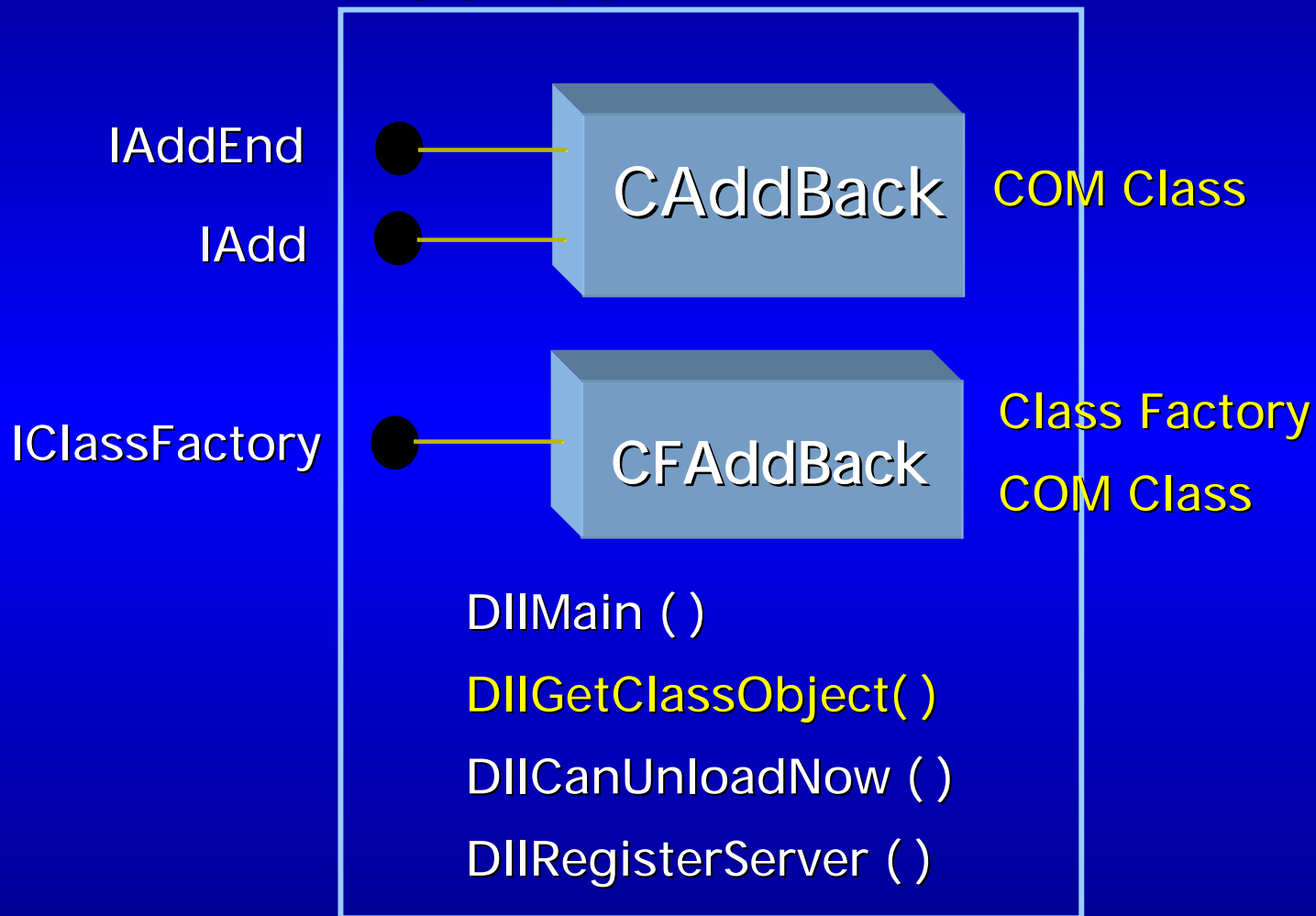
◆ DIICanUnloadNow

- CoFreeUnusedLibraries
 - COM Library API
 - DLL Unload
 -

```
STDAPI DIICanUnloadNow(void)
{
    if(g_cObjects == 0 && g_cLocks == 0)
        return S_OK;
    return S_FALSE;
}
➤ g_cObject
```

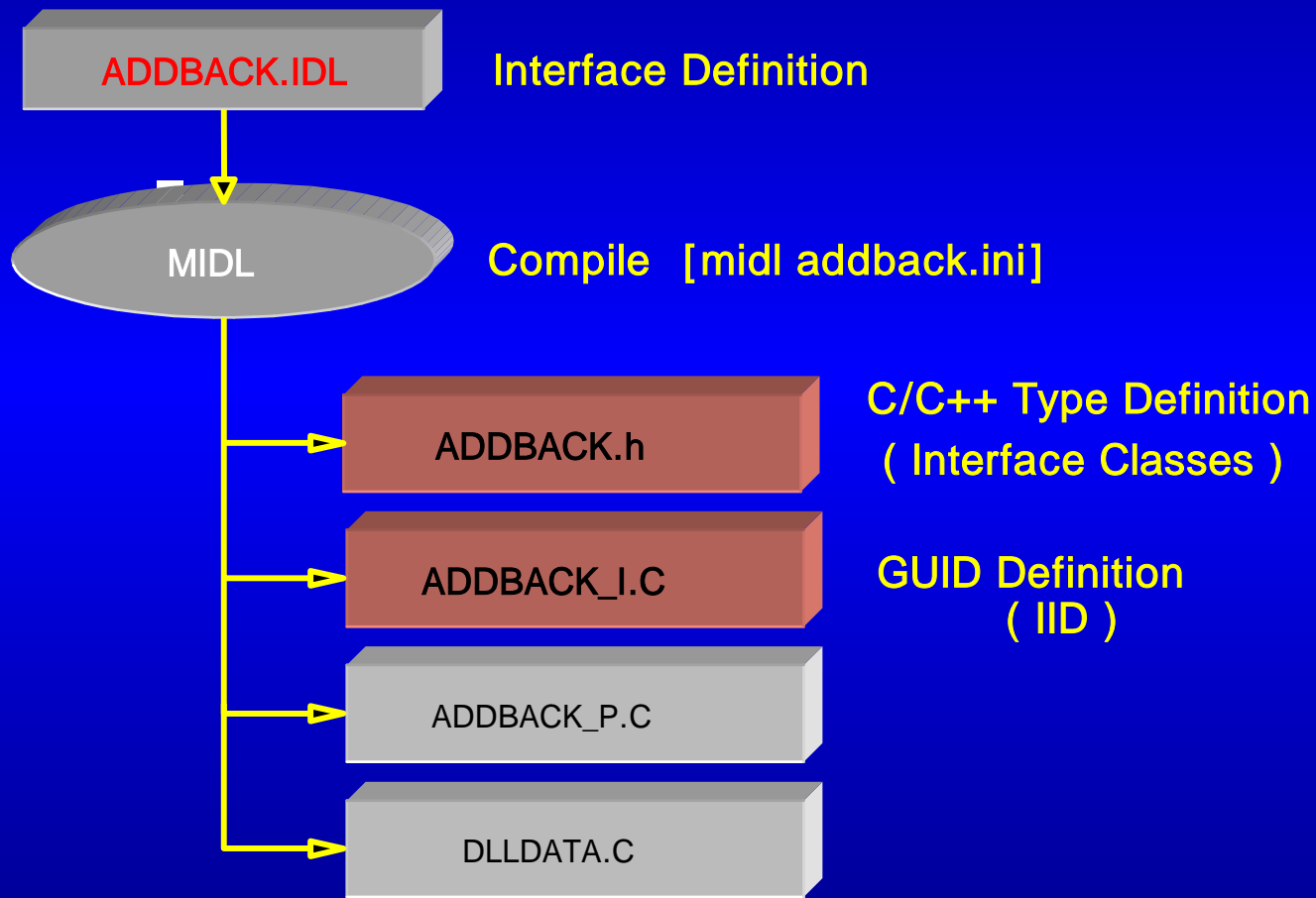
4. Let's make a DLL

AddBack.Dll



4. Let's make a DLL - IDL

1) Definition Interface (I AddEnd, I Add)



4. Let's make a DLL

Ex) interface IAddEnd

```
[
    uuid(C6F96D90-9FCF-11d1-B20A-0060970A3516),
    object
]
interface IAddEnd : IUnknown
{
    import "unknwn.idl";

    HRESULT GetAddEnd([out] short* result);
    HRESULT SetAddEnd([in] short addend);
    HRESULT GetSum([out] short* result);
    HRESULT Clear(void);
};
```

4. Let's make a DLL

◆ In Addback.h

```
#if defined(__cplusplus) && !defined(CINTERFACE)
interface DECLSPEC_UUID("C6F96D90-9FCF-11d1-B20A-
0060970A3516")
IAddEnd : public IUnknown
{
public:
    virtual HRESULT STDMETHODCALLTYPE GetAddEnd(
        /* [out] */ short __RPC_FAR *result) = 0;
    virtual HRESULT STDMETHODCALLTYPE SetAddEnd(
        /* [in] */ short addend) = 0;
    virtual HRESULT STDMETHODCALLTYPE GetSum(
        /* [out] */ short __RPC_FAR *result) = 0;
    virtual HRESULT STDMETHODCALLTYPE Clear( void) = 0
};
```

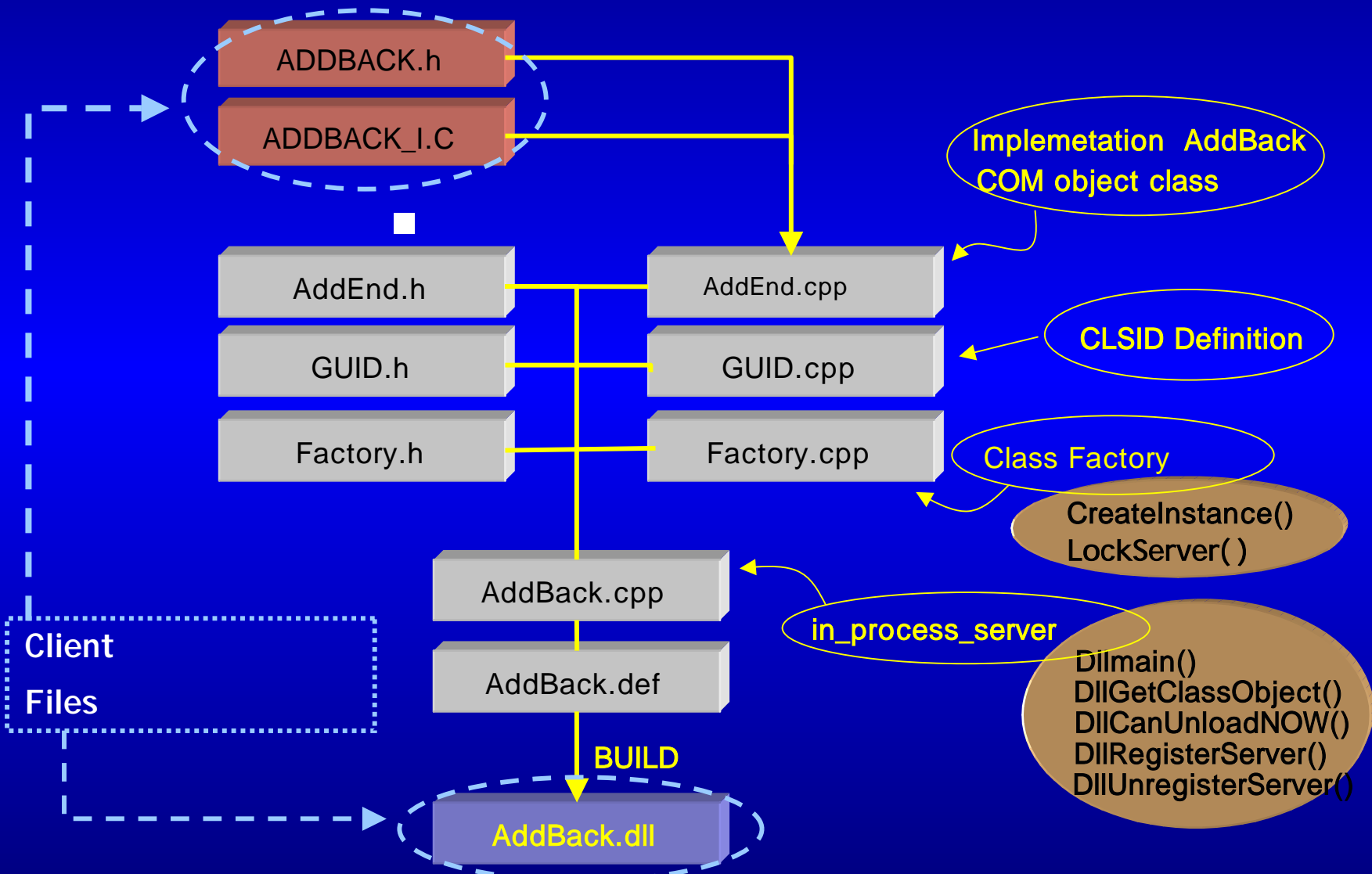
4. Let's make a DLL

◆ In Addback_i.c

```
const IID IID_IAddEnd =  
    {0xC6F96D90,0x9FCF,0x11d1,{0xB2,0x0A,0x  
    00,0x60,0x97,0x0A,0x35,0x16}};
```

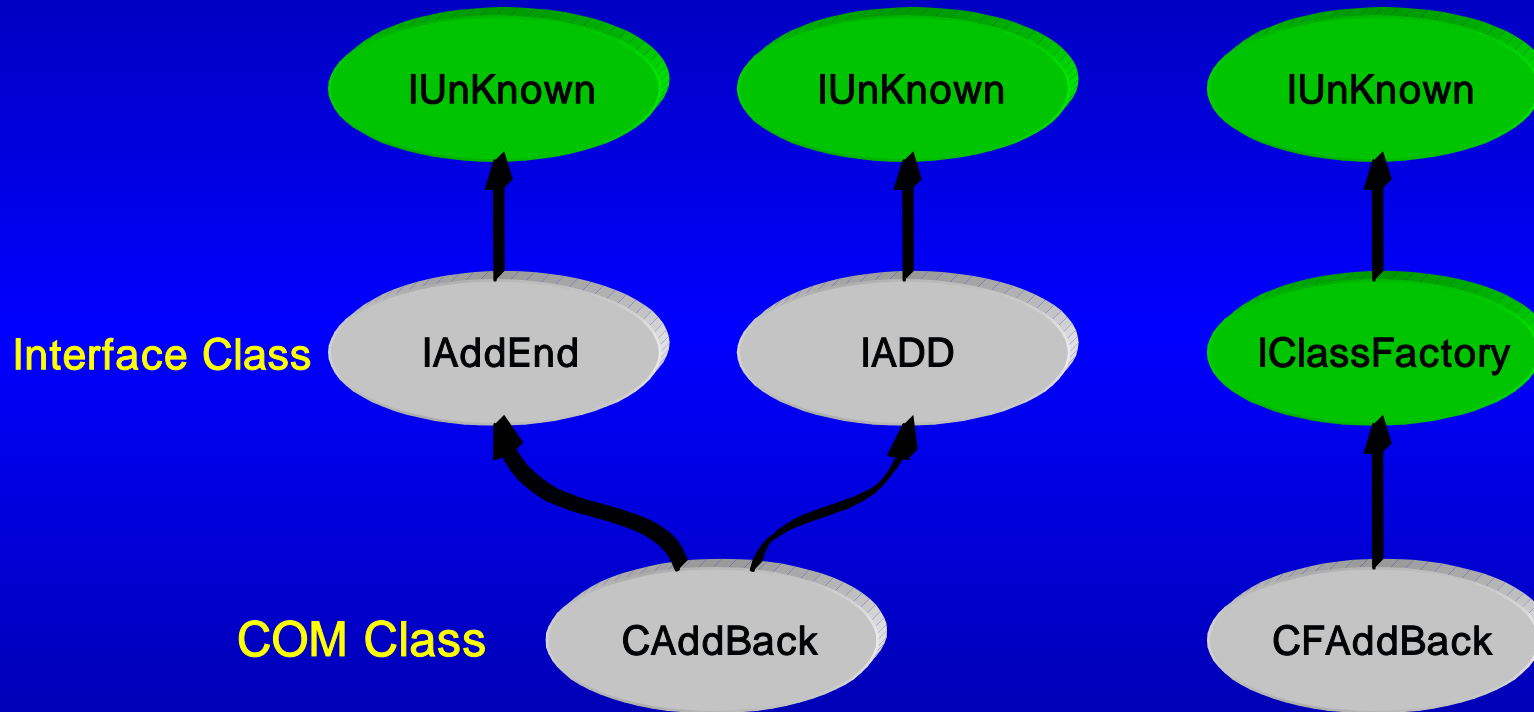
4. Let's make a DLL

2. In_process_server



4. Let's make a DLL

◆ Relationship



5. Reference

- ◆ <http://www.microsoft.com/COM>
- ◆ ATL COM
- ◆ Inside COM
- ◆ Essential COM