

Introduction of COM

CH 1. Why Com?

Hyun-Soo, Kim

jfet@mmlab.net

Multimedia Lab.

Dept. of Electrical and Computer Eng.

University of Seoul

Seoul, Korea

Document History

Originator : bluejfet [2001.12]

Revised by twoss[2002.12]

0. Contents

1. Main Objectives of the presentation.
2. Overview
3. What is COM ?
4. What is Interface ?
5. IUnknown Interface
6. GUID
7. HRESULT
8. Reference

COM

- ◆ (Component-based Distributed Computing)
COM/DCOM(Component Object Model/ Distributed Component Object Model) CORBA(Common Object Request Broker Architecture)
- ◆ Microsoft가 COM/DCOM
Windows
Unix , IBM
OMG(Object Management Group) CORBA
UNIX
sun Microsystems Java Beans
Enterprise Java Beans가 가

COM

- ◆ Microsoft
COM .
Microsoft

- ◆ COM Microsoft

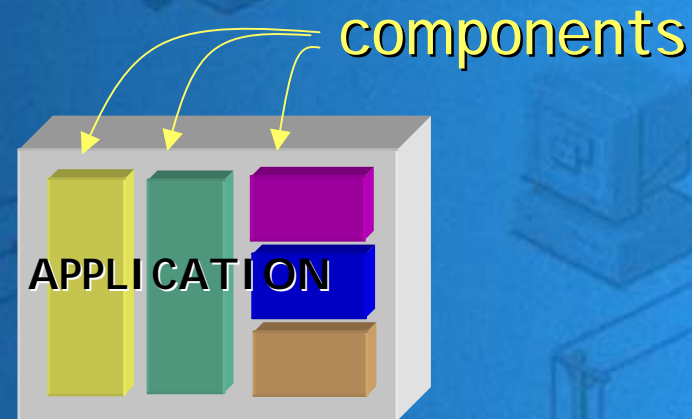
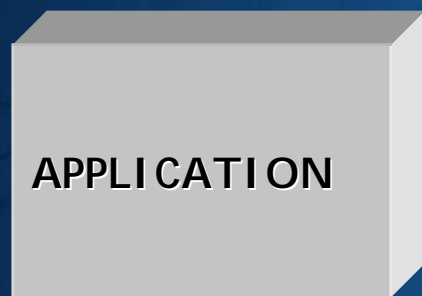
1. Main Objectives

- ❑ Introduction COM
- ❑ Interface
- ❑ IUnknown Interface
- ❑ GUID ■ HRESULT

2. Overview

◆ Components

- component Application Program component



- Component가 application



가

.

2. Overview



1. (Object Oriented Design)

2. OOD

1. Binary object

2.

3.

component(ex.dll)가
component Client Program
compile

3.

COM

2. Overview

- ◆ Component Software
 - COM/DCOM(Distributed Component Object Model)
 - CORBA(Common Object Request Broker Architecture)

3. What is COM?

- ◆ COM (Component Object Model)
 - ~defines a standard for component interoperability, is not dependent on any particular programming language, is available on multiple platforms, and is extensible.
 - COM
가 .

3. What is COM?

1. Defines a binary standard for component interoperability
2. Is programming language-independent
3. Is provided on multiple platforms
(Microsoft® Windows®, Microsoft Windows NT™, Apple® Macintosh®, UNIX®)
4. Is extensible
5. Location transparency
6. Version compatibility

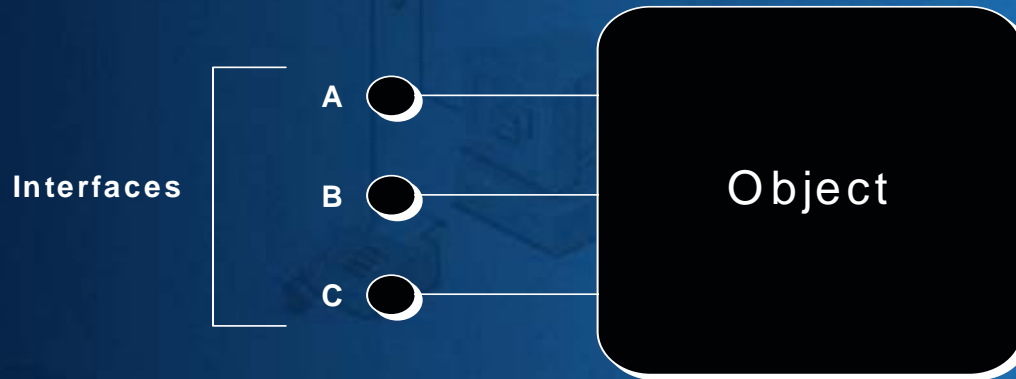
4. What is Interface?

- ◆ In COM, applications interact with each other and with the system through collections of functions called *interfaces*.
- ◆ ~is a strongly-typed group of semantically-related functions, also called "interface member functions."
- ◆ Interface names begin with "I" by convention
- ◆ function . 가 pure virtual abstract class virtual function table(VTBL)

4. What is Interface?

- ◆ (Interface) COM 가
- ◆ Client interface ■ COM object com object가

4. What is Interface?



A typical picture of a component object that supports three interfaces A, B, and C.



4. What is Interface?

◆ Implementing a COM Interface

◆ Ex)

```
Class I X {  
    public:  
        virtual void Fx1() =0;  
        virtual void Fx2() =0;  
};  
Class I Y {  
    public:  
        virtual void Fy1() =0;  
        virtual void Fy2() =0;  
};
```

4. What is Interface?

```
◆ #include <objbase.h>
interface IX {
    virtual void __stdcall Fx1() = 0;
    virtual void __stdcall Fx2() = 0;
    virtual void __stdcall Fx3() = 0;
    virtual void __stdcall Fx4() = 0;
};

interface IY {
    virtual void __stdcall Fy1() = 0;
    virtual void __stdcall Fy2() = 0;
};
```

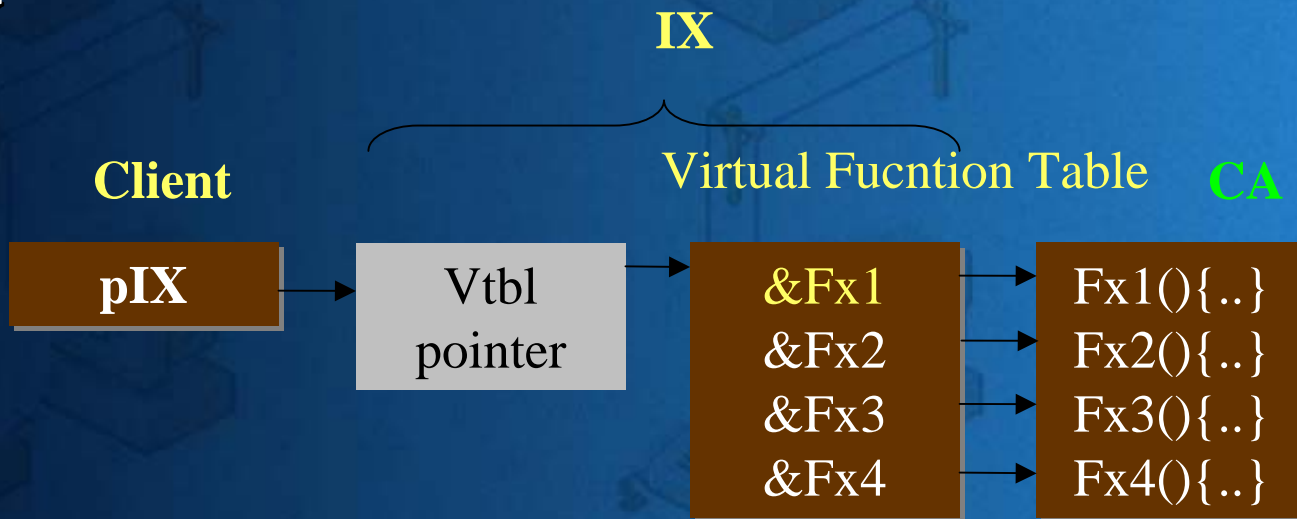
4. What is Interface?

◆ Implementing a COM CLASS

```
class CA : public IX,  
          public IY  
{  
public:  
    /** Implement interface IX.  
    virtual void STDMETHODCALLTYPE Fx1(void) {cout  
    <<"CA::Fx1"<<endl;}  
    virtual void STDMETHODCALLTYPE Fx2(void) {cout  
    <<"CA::Fx2"<<endl;}  
    ..  
    /** Implement interface IY.  
    virtual void STDMETHODCALLTYPE Fy1(void) {cout  
    <<"CA::Fy1"<<endl;}  
    virtual void STDMETHODCALLTYPE Fy2(void) {cout  
    <<"CA::Fy2"<<endl;}  
};
```

4. What is Interface?

```
int main() {  
    CA* pA = new CA;  
    /*Get an IX pointer  
    IX* pIX = pA;  
    ..  
}
```



5. IUnknown Interface

◆ IUnknown Interface 가

- COM Object가
Pointer

Client

→ QueryInterface()

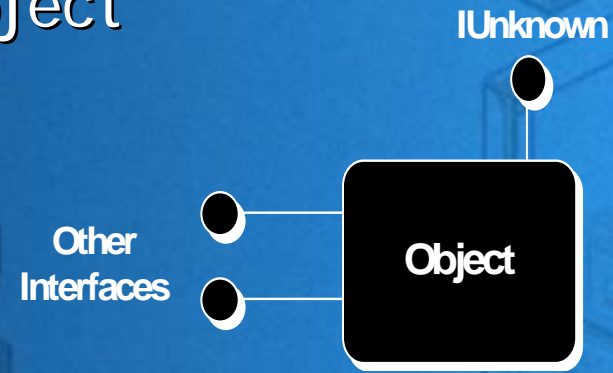
- COM Object Client가
, COM Object
. COM Object

→ AddRef()

→ Release()

◆ COM Interface IUnknown Interface

Interface



5. IUnknown Interface

- ◆ Defined in "UNKNOWN.H"

```
interface IUnknown {  
    virtual HRESULT __stdcall  
    QueryInterface(const IID& iid, void**  
    ppv)=0;  
    virtual ULONG __stdcall AddRef() = 0;  
    virtual ULONG __stdcall Release() = 0;  
};
```

5. IUnknown Interface

- ◆ IDL(Interface Definition Language)
 -
- ◆ MIDL(Microsoft Interface Definition Language)
 - Microsoft IDL Compiler
 - Ex. >> midl addback.idl
 - 4 File
 - ~.h : C/C++
 - ~_i.c : GUID
 - ~_p.c , DIIData.c :

6. GUID (Globally Unique Identifier)

- ◆ GUIDs identify objects such as interfaces, manager entry-point vectors (EPVs), and class objects.
- ◆
- ◆ A GUID is a 128-bit value consisting of one group of 8 hexadecimal digits, followed by three groups of 4 hexadecimal digits each, followed by one group of 12 hexadecimal digits
- ◆

```
typedef struct _GUID {  
    DWORD Data1;  
    WORD Data2;  
    WORD Data3;  
    BYTE Data4[8];  
} GUID;
```

ex) {5B5ACFA6-9A05-4084-AF35-549AF8E20927}

6. GUID (Globally Unique Identifier)

- ◆ IID (Interface Identifier)

-
- typedef GUID IID;

- ◆ CLSID (class Identifier)

- COM Object
- typedef GUID CLSID;

- ◆ GUID

- ◆ Tools

- UUI DGEN.exe (dos)
- GUI DGEN.exe (window)

ID

ID

6. GUID (Globally Unique Identifier)

- ◆ `DEFINE_GUID(IID_IFoo, 0x5b5acfa6, 0x9a05, 0x4084, 0xaf, 0x35, 0x54, 0x9a, 0xf8, 0xe2, 0x9, 0x27);`

=

Extern "C" const GUID IID_IFoo = {0x5b5acfa6, 0x9a05, 0x4084, {0xaf, 0x35, 0x54, 0x9a, 0xf8, 0xe2, 0x9, 0x27}}

➤ INITGUID.H

- ◆ GUID :
operator==, IsEqualGUID(),
IsEqualIID(), IsEqualCLSID()

- ◆ GUID 가

7.HRESULT

- ◆ The facility field indicates the system service responsible for the error
- ◆ Values of HRESULTs
 - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/com/error_899v.asp
- ◆ `HRESULT hr = pl ->QueryInterface(xx , xx);`
 - `if(hr == S_OK)` → (X)
 - `if(SUCCEEDED(hr))` → (O)
- ◆ `#define SUCCESS(hr) (long(hr) >=0)`
- ◆ `#define FAILED(hr) (long(hr) <0)`

8. Reference

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