Defining a custom ECOM interface

Overview

This code snippet demonstrates how to define an interface for an ECom component. Clients use the interface to get an instance of the concrete implementation, and plug-ins implement the pure virtual `SayHello(TDes& aString)` method.

Preconditions and important issues

- Due to self-signing, the UID of this interface must be from the unprotected range.
- If the UID3 of the plug-in DLL is changed (and, correspondingly, the name of the RSS file is changed), the ECom framework may not find the plug-in with the new UID. It may be necessary to delete the `ECom.ROM.Z.dat` and `Ecom.idx` files from the development environment folder of the SDK (for example, `C:\Symbian\9.1\S60_3rd_MR\Epoc32\winscw\c\private\10009D8F\`). When the emulator is restarted, the files are recreated.
- The interface header must have a unique instance identifier key (`iDtor_ID_Key`).
- The interface must inform the ECom framework that this specific instance of the interface has been destroyed: `REComSession::DestroyedImplementation(iDtor_ID_Key)`.

MMP file

No MMP file is needed. Only the header and inline source files are defined. ECom components that derive this base class must include these two files.

Header file

```
#include <e32base.h>
#include <ECom.h>
#include "HelloEcomIF.inl" // Proprietary base implementations for ECOM

class CHelloEcomIF : public CBase
    {
public:
    // Instantiates an object of this type
    static CHelloEcomIF* NewL();

    // Destructor
    virtual ~CHelloEcomIF();

    // Requests a list of all available implementations which follow this
    // given interface.
    static void ListAllImplementationsL(RImplInfoPtrArray& aImplInfoArray);

    // Your own pure interface method which the derived class must implement
    virtual void SayHello(TDes& aString) = 0;

protected:
    inline CHelloEcomIF();

protected:
    // Unique instance identifier key
    TUid iDtor_ID_Key;
    };
```

Inline source file

```
inline CHelloEcomIF* CHelloEcomIF::NewL()
    {
    // Finds an ECom with a given implementation UID (KCHelloEcomImplUid)
    // and casts it to CHelloEcomIF*
    return REINTERPRET_CAST(
        CHelloEcomIF*,
        REComSession::CreateImplementationL(
            KCHelloEcomImplUid,
            _FOFF(CHelloEcomIF,iDtor_ID_Key)
        )
    );

    // There are many different solutions to find EComs but this is the
    // simplest. The example uses the default resolver.
    }

inline CHelloEcomIF::~CHelloEcomIF()
    {
    // Destroy any instance variables and inform the framework that
    // this specific instance of the interface has been destroyed.
    REComSession::DestroyedImplementation(iDtor_ID_Key);
```


(C) Copyright Nokia 2013. All rights reserved.

```
    }

inline void CHelloEcomIF::ListAllImplementationsL
(RImplInfoPtrArray& aImplInfoArray)
    {
    REComSession::ListImplementationsL(KCHelloEcomIFUid, aImplInfoArray);
    }
```

ECom plug-in resource file

`KCHelloEcomImplUid` implementation UID for finding the particular ECom component is stated in the ECom resource file (`{{{1}}}`).

```
#include "RegistryInfo.rh"

RESOURCE REGISTRY_INFO theInfo
    {
    // UID for the DLL
    dll_uid = 0xE01F5465;
    // Declare an array of interface information
    interfaces =
        {
        INTERFACE_INFO
            {
            // UID of the interface that is implemented
            interface_uid = 0xE0009DC1;
            implementations =
                {
                // Info for implementation of CHelloEcomIF
                IMPLEMENTATION_INFO
                    {
                    implementation_uid = 0xE0009DC7;
                    version_no = 1;
                    display_name = "ecomexample";
                    opaque_data = "";
                    default_data = "ecomexample";
                    }
                };
            }
        };
    }
```

Postconditions

ECom class header and inline file (or source file) have been defined. The header has at least a constructor, a destructor, and some virtual methods that the derived class must implement.

See also

- File:ECom.zip: Working example code
- Implementing an ECOM interface
- Using an ECOM component
- Nokia Developer Wiki - ECOM
- S60 Platform: ECom Plug-in Architecture
- S60 Platform: ECom Plug-in Examples

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.