
Subject: Re: DLLs

Posted by [zaurus](#) on Sat, 18 Oct 2008 15:28:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

I had a similar problem some time ago, when I had to integrate a third party .NET DLL. After many days and nights googleing I found that there is a way by accessing the DLL via DCOM.

In your .NET DLL you need to expose an Interface.

using System.Runtime.InteropServices;

```
namespace TestClass
{
    [InterfaceType(ComInterfaceType.InterfaceIsDual)]
    public interface ITestInterface
    {
        bool Function1OfDLL(string sParameter);
        string Function2OfDLL(string sAnotherParameter);
    }
    public class TestClass : ITestInterface
    {
        public TestClass()
        {
            //Init code;
        }

        bool Function1OfDLL(string sParameter);
        {
            //Do something here
        }
        string Function2OfDLL(string sAnotherParameter);
        {
            //Do something here
        }
    }
}
```

Then you need to register your DLL with "RegAsm.exe TestDLL.dll". The RegAsm tool is part of the .NET-Framework.

In VC++ 6.0 it goes like this to access the DLL. Something similar should be possible in U++.

```
#import "TestDLL.tlb"
using namespace TestClass;
HRESULT hr = CoInitialize(NULL);
ITestInterface* m_DLL;
m_DLL = NULL;
CLSID clsid;
```

```
REFIID refiid = __uuidof(ITestInterface);  
hr = CLSIDFromProgID(OLESTR("TestDLL.TestClass"), &clsid);  
hr = CoCreateInstance(clsid, NULL, CLSCTX_ALL, refiid, (LPVOID*) &m_DLL);
```

Access functions of DLL.

```
m_DLL->Funtion1OfDLL();
```

To close connection with DLL do this.

```
m_DLL->Release();  
CoUninitialize();
```

I hope this helps.

Good luck.

Zaurus
