
Subject: Linking to a .net DLL ?

Posted by [nixnixnix](#) on Tue, 26 Jul 2011 20:43:36 GMT

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

Can't find anything on this when I search these fora.

I would like to be able to use some functionality which is proprietary and only available as .net

My colleague can get what I want to compile into a DLL but I am wondering how I then can use that DLL in UPP.

Cheers,

Nick

Subject: Re: Linking to a .net DLL ?

Posted by [dolik.rce](#) on Wed, 27 Jul 2011 05:26:27 GMT

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

I'm not an expert on .net or windows programming, but I have a feeling that what you need is available in Koldos Functions4U package in bazaar.

Best regards,

Honza

Subject: Re: Linking to a .net DLL ?

Posted by [nixnixnix](#) on Thu, 28 Jul 2011 18:09:10 GMT

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

Thanks Dolik and to you too, Koldo.

Exactly what I was looking for

Nick

Subject: Re: Linking to a .net DLL ?

Posted by [nixnixnix](#) on Thu, 28 Jul 2011 21:12:46 GMT

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

Hey Koldo,

Should this work for a .NET compiled DLL? I am trying the following

```

AgiRadar::AgiRadar()
{
// connecting to .net DLL to use the following functions

// Public Shared Function testFunc1() As Integer
//   Return 123
// End Function

// Public Shared Function testFunc2(ByVal x As Double) As Double
//   Return x * x
// End Function

DI turbineSpeedCalc;

String myDllFolder = "D:\\temp\\test\\Debug\\";

////////////////////////////////////

double (*testFunc2)(double);

if (!turbineSpeedCalc.Load(AppendFileName(myDllFolder, "turbineSpeedCalc.dll")))
    throw Exc(Format(t_(""% dll not found"), "turbineSpeedCalc"));

testFunc2 = (double (*)(double))turbineSpeedCalc.GetFunction("testFunc2");

if (!testFunc2)
    throw Exc(Format(t_("Function %s was not found in dll"), "testFunc2"));

double argOut = testFunc2(15.0);

String s = Format("answer is %f",argOut);
PromptOK(s);

////////////////////////////////////

int (*testFunc1)();

if (!turbineSpeedCalc.Load(AppendFileName(myDllFolder, "turbineSpeedCalc.dll")))
    throw Exc(Format(t_(""% dll not found"), "turbineSpeedCalc"));

testFunc1 = (int (*)(()))turbineSpeedCalc.GetFunction("testFunc1");

if (!testFunc1)
    throw Exc(Format(t_("Function %s was not found in dll"), "testFunc1"));

double argOut2 = testFunc1();

```

```
s = Format("answer is %f",argOut2);
PromptOK(s);

}
```

is there a way to list the functions available in a DLL? I figure if I can get these simple functions to work then I should be able to pass structs and arrays to real functions.

It successfully finds and loads the DLL but fails to find either function.

Cheers,

Nick

Subject: Re: Linking to a .net DLL ?
Posted by [Novo](#) on Fri, 29 Jul 2011 03:27:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi nixnixnix,
testFunc1 and testFunc2 most likely have C++ mangling.
I think you can get a list of exported function using either "VC\bin\dumpbin.exe" or dependency walker.

UPP has a nice feature called dli files.
You can find an example here: uppsrc\Oracle\Oci8.dli

I hope this will help.

Subject: Re: Linking to a .net DLL ?
Posted by [koldo](#) on Sun, 31 Jul 2011 21:37:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Nick

In addition, I am not expert in .NET Dll but as DI uses LoadLibrary function, it could only load .NET Dll plain not managed functions outside a class.

Subject: Re: Linking to a .net DLL ?
Posted by [nixnixnix](#) on Sun, 31 Jul 2011 22:51:13 GMT

Thanks.

Dependency walker does not see any functions in the DLL so am thinking there is something that my colleague has neglected to do that is akin to publishing the functions. I have never used DLLs before so am not really sure what am talking about. He is on vacation just now so I will have to ask him when he gets back.

Cheers,

Nick

EDIT: so I just tried linking into the DLL with MS VC++ express and a piece of code from <http://support.microsoft.com/kb/953836>

With some small adjustment I can trigger the function `turbineSpeedCalc::testFunc1`. The main difference appears to be that `Functions4U` allows me to access only global functions within the DLL whereas I appear to need access to a class. If I can make an instance of the class that is within the DLL then I should be able to access its functions.
