

Re: How can I detect the presense of an optional assembly and call a method within it?

Re: How can I detect the presense of an optional assembly and call a method within it?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005-10/msg02193.html>

- *From:* "Peter Rilling" <peter@xxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sun, 23 Oct 2005 22:03:29 -0700
-

You need more than just the assembly name, you also need the class name that will be invoked against.

Basically you will need to use Reflection. The idea being that reflection allows you to dynamically create instances at runtime and invoke method. Take a look at reflection and see if this helps.

"Gaetan" <someone@xxxxxxxxxxxxxxxx> wrote in message news:8ugol1pf2fqu9gl55udr43oijp3bqg2afj@xxxxxxxxxxxx
>I would like to extend the capabilities of my application by calling a user
>method
> residing in a client provided assembly without having to recompile my
> application.
>
> Things would work like this:
>
> 1- Read a configuration file where the client provides the assembly name
> (DLL) and the
> method name.
>
> 2- Detect the existence of the assembly file.
>
> 3- Invoke the client method.
>
> I'm not sure how to handle item #3. I cannot add a reference to the client
> DLL in my
> application since I do not know in advance what the assembly name will be
> nor the method
> name. Can someone provide me a clue on how I should handle it?
>
> Thanks.

Re: How can I detect the presense of an optional assembly and call a method within it?

- **References:**

- ◆ **[How can I detect the presense of an optional assembly and call a method within it?](#)**

- ◆ *From:* Gaetan

- Prev by Date: **[Re: Timer Execution](#)**

- Next by Date: **[Re: Base64 encoding/decoding](#)**

- Previous by thread: **[How can I detect the presense of an optional assembly and call a method within it?](#)**

- Next by thread: **[Re: How can I detect the presense of an optional assembly and call a method within it?](#)**

- Index(es):

- ◆ **[Date](#)**

- ◆ **[Thread](#)**