

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/msg02378.html>

---

- *From:* "MUS" <[moizshaikh@xxxxxxxxxx](mailto:moizshaikh@xxxxxxxxxx)>
  - *Date:* 25 Oct 2005 01:43:21 -0700
- 

Hello Gaetan!

Well this might not be the perfect Plugin approach (i.e. have a static reference to client assembly) but just an idea, may be you feel worth implementing. Its based on the use of config file and reflection.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<configSections>
<section name="ClientSection"
type="System.Configuration.DictionarySectionHandler" />
</configSections>
<!-- Custom Section -->
<ClientSection>
<add key="BusinessAssembly" value="<your_assembly>" />
<add key="BusinessType" value="<your_type>" />
<add key="BusinessMethod" value="<your_method>" />
</ClientSection>
<!-- / Custom Section -->
</configuration>
```

```
public Object DoAction()
{
try
{
Hashtable configData = (Hashtable)
ConfigurationSettings.GetConfig("ClientSection");
if(configData==null) throw new ArgumentNullException("Unable to read
configuration section");
// Get Assembly
String strAssembly =
Convert.ToString(configData["BusinessAssembly"]);
Assembly assembly = Assembly.Load(strAssembly);
if(assembly==null) throw new Exception("Unable to load assembly:
"+assemblyName);
// Get Type
```

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?

```
String strType = Convert.ToString(configData["BusinessType"]);
Type workingType = assembly.GetType(strType);
if(workingType==null) throw new Exception("Unable to load type:
"+workingType);
// Get Method
String strMethod = Convert.ToString(configData["BusinessMethod"]);
MethodInfo methodInfo = workingType.GetMethod(strMethod);
if(methodInfo==null) throw new Exception(strAssembly+" (type:
"+workingType+" ) doesnot contain "+strMethod);
// Type Instance
Object obj = assembly.CreateInstance(strType, true);
if(obj==null) throw new Exception("Instantiation failed for
"+strType);
// Invoke Method
Object objReturnValue = methodInfo.Invoke(obj, null);
return objReturnValue;
}
catch(Exception ex) { throw ex; }
}
```

Its just a thought, I hope this might be of some help. I cant predict the correctness of the code since i wrote it in "wordpad" [i have a problem with my VS.NET] :-)

Let me know in case of any inconsistancy.

Regards,

Moiz Uddin Shaikh  
Software Engineer  
Kalsoft (Pvt) Ltd

---

• **Follow-Ups:**

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

◇ From: Gaetan

• **References:**

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

◇ From: Gaetan

• Prev by Date: **Re: CreateProcessWithLogonW and ASP.NET**

• Next by Date: **Re: Number to Words..**

• Previous by thread: **Re: 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?**

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

- Index(es):
  - ◆ *Date*
  - ◆ *Thread*