

Assembly References – User Compiled

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2006-05/msg00801.html>

- *From:* "Quin" <QuinP@xxxxxxxxxxxxxx>
 - *Date:* Sat, 27 May 2006 14:47:56 -0700
-

Greetings,

I am compiling a class with the CSharpCodeProvider, then compiling a class which uses it, having troubles with the Method.Invoke.

CLASS1:

```
using System;
namespace MySpace {
public class MyClass {
public long MyProperty {
get { return 999; } set {}
}
}
}
```

After compile, reflection shows:

```
Module[0] MyClass.dll
Type[0] MySpace.MyClass
Method[0] Int64 get_MyProperty()
Method[1] Void set_MyProperty(Int64
Method[2] System.Type GetType()
Method[3] System.ToString()
Method[4] Boolean Equals(System.Object)
Method[5] Int32 GetHashCode()
```

CLASS2:

```
using System;
using MySpace;
namespace MySpace {
public class ProcessingClass {
public static void Process_MyClass(MyClass mc) {
Console.WriteLine("Value: " + mc.MyProperty.ToString());
}
}
}
```

After compile, reflection shows:

```
Module[0] ProcessingClass.dll
Type[0] MySpace.ProcessingClass
```

Assembly References – User Compiled

```
Method[0] Void Process_MyClass(MyClass)
Method[1] System.Type GetType()
Method[2] System.ToString()
Method[3] Boolean Equals(System.Object)
Method[4] Int32 GetHashCode()
```

then invoking:

```
Type[] types;
```

```
//Get MyClass Instance
```

```
types = class1Assemble.GetExportedTypes();
object myclass = Activator.CreateInstance(types[0]);
```

```
//Invoke Process_MyClass Method on passing myclass
```

```
types = class2Assembly.GetExportedTypes();
MethodInfo mi = types[0].GetMethod("Process_MyClass");
mi.Invoke(null, new object[] { myclass });
```

This operation fails with the following message:

```
"Object of type 'MySpace.MyClass' cannot be converted to type
'MySpace.MyClass'."
```

Rather Odd. I can change the MyClass definition to MyClass(long val),

change the Process_MyClass to;

```
Console.WriteLine("Value: " + val.ToString());
```

and the following works fine;

```
mi.Invoke(null, new object[] { 999 });
```

```
--
```

.