**Identifier '????????' is not CLS-compliant**

## description

is there a solution for the problem described here: <https://confuser.codeplex.com/workitem/8079> ?  
I try out the Confuser.exe the first time for my simple Winform-project, but I got a error at runtime:  
Message: Identifier '????????' is not CLS-compliant.  
Parameter name: ident  
(and a little more...)  
  
I exame my code and found exactly the 'faulty' line: The code is to deserialize a any object from a xml-file.

public static T ReadFile<T>(string FileName) where T : class

{

T classObj = null;

// for debugging only

System.Windows.Forms.MessageBox.Show(FileName); <<<<<< here: Filename is displayed all right and can be accessed in the next line

if (System.IO.File.Exists(FileName))

{

using (FileStream fs = new FileStream(FileName, System.IO.FileMode.Open))

{

XmlSerializer xser = new XmlSerializer(typeof(T)); <<<<<< here it happends -> exception !!!

classObj = (T)xser.Deserialize(fs);

}

}

else

{

throw new Exception("File '" + FileName + "' could not be found!");

}

return classObj;

}

I also tried out to serialize an Object to a file. The file is created, but not could not be filled as an error accures when it comes to create a instance of the XmlSerializer it:

public static bool WriteFile<T>(T classObj, string FileName) where T : class

{

bool writeSuccess = false;

using (FileStream fs = new FileStream(FileName, FileMode.Create))

{

System.IO.TextWriter tw = new System.IO.StreamWriter(fs);

XmlSerializer xser = new XmlSerializer(typeof(T)); <<<<<<<<<<< this line cause the exception

xser.Serialize(tw, classObj);

tw.Flush();

fs.Flush();

writeSuccess = true;

}

return writeSuccess;

}

I also tried to use  
[assembly: System.CLSCompliantAttribute(true)] and [CLSCompliantAttribute(true)]  
to attribut the class (from which the object should be serialized), but I got no success to avoid the problem in the obuscated Assembly. Of course, I've allready set the [Serializable()] Attribute to that class too.  
  
[yck1509](http://www.codeplex.com/site/users/view/yck1509) wrote Jul 15, 2013 at 3:44 PM

Most probably the serialized class have their members renamed to unreadable characters and XmlSerializer doesn't support it. You can exclude the serializable class from renaming and it should work.

[IndustryMaxi](http://www.codeplex.com/site/users/view/IndustryMaxi) wrote Jul 17, 2013 at 4:03 PM

Thank you for that tip! You are right. Excluding the serializable class and as well the serializing methods from being confused (option "none" under the advanced tab) cause a different behavior. Now the Programm throws "the type initializer for ' module ' threw an exception" ...   
  
I hope to find the right solution with a small test project soon..   
here the solution, how it works in my case. All settings are done in the 'Advanced Settings'-Tab:

* for the assembly whcih contains the XmlSerializer.Deserialize(fs) / XmlSerializer.Serialize(..):  
  check all but exclude 'Method Proxy Confusion'  
  don't forget to press "Apply to Members"
* for the assembly which contains the serializing class,   
  check all but exclude 'Name Confusion' and 'Reduce Metadata Confusion'  
  don't forget to press "Apply to Members"

When I did the tests some settings lead to run apparently, but the serialized data are set all to default values (e.g. int --> 0, strings -> "", etc.). So perform tests of you application after dot-confusion!