

InvalidCastException and Remoting version compatibility

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.remoting/2007-01/msg00038.html>

- *From:* schaf@xxxxxxxx
 - *Date:* 16 Jan 2007 09:41:29 -0800
-

Hi NG !

I use your help very urgent!

2 years ago a remote application was created.

The DataService (DataService.dll) contains a IRemoteFactory, which allows the client to get access to the corresponding interface (here the implementation):

```
public sealed class VRemoteFactory : MarshalByRefObject,
IRemoteFactory {
private DataObject m_DataObj = null;

public IDataObject iDataObject {
get {
if (null == m_DataObj) {
m_DataObj = new DataObject();
}
return m_DataObj;
}
}
```

The class DataObject (implemented as MarshalByRefObject in the DataService.dll) implements the IDataObject interface, which provides a property 'Version' and a function to get an object of the class Data (implemented as sealed Serializable).

Now I have to extend the DataService.

As described in the .NET Remoting book of Ingo, I created a new interface IExtendedDataObject which inherits from IDataObject. The DataObject class now implements IExtendedDataObject, with the new property that returns a class ExtData (which inherits from Data (which is no longer declared as sealed)). I did not change anything at the VRemoteFactory.

So every thing works fine as long as I use the new addapted client with the new DataService (DataServer.exe). But if I try to run my new Client againts the old DataService (DataServer.exe) I get a

InvalidCastException and Remoting version compatibility

InvalidCastException (Return argument has invalid type), but I do not know why ??? I get this exception as soon as I call the IRemoteFactory.iDataObject in the client, but why ???

In a demo project every thing works!!!

Please help me!

PS: I read something about namespace clashes if the old version is compiled in .NET 1.0 and the new in 1.1. ????

Thanks

.