

## Resolve runtime error for SAF.EventNotification under .NET 1.1

When you try to run Test.Client.SAF.EventNotification and Test.Server.SAF.EventNotification solution under VS.NET 2003/.NET Framework 1.1, you will get a runtime error that will prevent the calls between client and server.

The error message says that “System.DelegateSerializationHolder are not permitted to be deserialized...” This is due to a breaking change in .NET remoting in 1.1 version. If you are running the code using VS.NET 2002/.NET 1.0, you will not have this problem.

With the attempt to tighten up security, remoting in .NET 1.1 version does not allow the pass of delegate and objRef object across network boundary by default. Because SAF.EventNotification operates by passing around the delegates, this change in 1.1 will break the code.

Although disabled by default, passing delegate and object reference can be easily enabled by a line in the configuration file or few lines of code change (depending on which approach you use to register the remoting service).

SAF.EventNotification activates the .net remoting through code. You need to modify the following files to make it work under .NET 1.1

Modify the code for the server in file

**\SAF\Test.Server.SAF.EventNotification\TestConsole\Class1.cs**

```
BinaryServerFormatterSinkProvider serverProvider = new
BinaryServerFormatterSinkProvider();
serverProvider.TypeFilterLevel = TypeFilterLevel.Full;

IDictionary props = new Hashtable();
props["port"] = Int32.Parse(enc.GetPortNumber());

//HttpChannel channel = new HttpChannel(Int32.Parse(enc.GetPortNumber()));
HttpChannel channel = new HttpChannel(props,null,serverProvider);
```

Modify the code for client in file

**\SAF\SAF.EventNotification\EventClient.cs**

```
BinaryClientFormatterSinkProvider clientProvider = new
BinaryClientFormatterSinkProvider();

IDictionary props = new Hashtable();
props["port"] = 0;

//HttpChannel channel = new HttpChannel(0);
HttpChannel channel = new HttpChannel(props, clientProvider,null);
```

The commented code in the above code blocks represents the existing SAF code. You need to modify the code in the corresponding files as shown above. Notice that `TypeFilterLevel = TypeFilterLevel.Full` does the trick to fix the problems.

Achieving the same thing in configuration is even simpler. In the configuration file approach, you need to add additional formatter information on both client and server configuration file, as shown following

Client side:

```
<channel ref="http" port="0">
  <clientProviders>
    <formatter ref="binary" />
  </clientProviders>
  <serverProviders>
    <formatter ref="binary" typeFilterLevel="Full" />
  </serverProviders>
</channel>
```

Server side:

```
<channel ref="http" port="4000">
  <serverProviders>
    <formatter ref="binary" typeFilterLevel="Full" />
  </serverProviders>
</channel>
```

After make the code change, recompile both Test.Server.SAF.EventNotification solution and Test.Client.SAF.EventNotification solution. The demo should work normally.

If you have more questions about this issue, you can contact me at [xchen@xtremework.com](mailto:xchen@xtremework.com)