

RE: WCF hosted in a webservice

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.webservices/2008-04/msg00054.htm>

- *From:* wdudek <wdudek@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 11 Apr 2008 07:42:01 -0700
-

Steven,

Thanks, I don't need to get rid of the parameter, just wanted to understand what it was so I can explain to the people writing the client apps why they got it.

Thanks again.

"Steven Cheng [MSFT]" wrote:

Hi Bill,

As for the additional "xxxSpecified" parameter, I think it is a webservice specific behavior for value type parameter/return value. For value type parameter in webservice, it support marking it as "nullable". However, since value type always contains a value, the .NET proxy generation will add an additional parameter for you to indicate whether this value has been set or not. You can try replace the "bool" type with another reference type such as string or a custom class .e .g.

```
=====  
[OperationContract]
```

```
MyResult CanRunRef(string AppName, string Username, string Function);  
.....  
}
```

```
[DataContract]  
public class MyResult  
{  
[DataMember]  
public bool CanRun;  
}
```

```
=====
```

Then, the generated proxy method won't have that additional output

RE: WCF hosted in a webservice

parameter. For your scenario, I think you do not quite need to use it if the return value is always a valid/expected value. For more information about Nullable types in webservice, here is another forum thread mentioned this:

#Consuming Webservice which returns NULL-values
<http://bytes.com/forum/thread375805.html>

Sincerely,

Steven Cheng

Microsoft MSDN Online Support Lead

Delighting our customers is our #1 priority. We welcome your comments and suggestions about how we can improve the support we provide to you. Please feel free to let my manager know what you think of the level of service provided. You can send feedback directly to my manager at:
msdnmg@xxxxxxxxxxxxxxxx

=====
Get notification to my posts through email? Please refer to
<http://msdn.microsoft.com/subscriptions/managednewsgroups/default.aspx#notifications>.

Note: The MSDN Managed Newsgroup support offering is for non-urgent issues where an initial response from the community or a Microsoft Support Engineer within 1 business day is acceptable. Please note that each follow up response may take approximately 2 business days as the support professional working with you may need further investigation to reach the most efficient resolution. The offering is not appropriate for situations that require urgent, real-time or phone-based interactions or complex project analysis and dump analysis issues. Issues of this nature are best handled working with a dedicated Microsoft Support Engineer by contacting Microsoft Customer Support Services (CSS) at
<http://msdn.microsoft.com/subscriptions/support/default.aspx>.

=====
This posting is provided "AS IS" with no warranties, and confers no rights.

Thread-Topic: WCF hosted in a webservice
thread-index: AcibJIPR/116RF3rRpymhXG08ZjkVQ==
X-WBNR-Posting-Host: 199.106.94.142
From: =?Utf-8?B?d2R1ZGVr?= <wdudek@xxxxxxxxxxxxxxxx>
Subject: WCF hosted in a webservice
Date: Thu, 10 Apr 2008 09:04:11 -0700

I have a web service hosting a WCF library, which works fine but produced a strange signature when interacting with vs 2005/2.0 clients. My method takes

RE: WCF hosted in a webservice

RE: WCF hosted in a webservice

3 strings and returns a bool as below.

```
bool CanRun(string appName, string userName, string function);
```

However the auto generated proxy in 2005 looks like this

```
void CanRun(string appName, string userName, string function, out bool  
result, out bool resultSpecified);
```

I understand the new result parameter, but what is the resultSpecified parameter that was created? It looks like it is always the same as the result parameter, although admittedly I haven't tested this very thoroughly yet? I'm pushing for WCF in our organization and don't want to get caught not having an explanation of what is occurring here.

Thanks,
Bill