

Returning DataTables in WCF/.NET

Asked 16 years, 4 months ago Modified 8 years ago Viewed 58k times



51

I have a WCF service from which I want to return a DataTable. I know that this is often a highly-debated topic, as far as whether or not returning DataTables is a good practice. Let's put that aside for a moment.



When I create a DataTable from scratch, as below, there are no problems whatsoever. The table is created, populated, and returned to the client, and all is well:



```
[DataContract]
public DataTable GetTbl()
{
    DataTable tbl = new DataTable("testTbl");
    for(int i=0;i<100;i++)
    {
        tbl.Columns.Add(i);
        tbl.Rows.Add(new string[]{"testValue"});
    }
    return tbl;
}
```

However, as soon as I go out and hit the database to create the table, as below, I get a CommunicationException "The underlying connection was closed: The connection was closed unexpectedly."

```
[DataContract]
public DataTable GetTbl()
{
    DataTable tbl = new DataTable("testTbl");
    //Populate table with SQL query

    return tbl;
}
```

The table is being populated correctly on the server side. It is significantly smaller than the test table that I looped through and returned, and the query is small and fast - there is no issue here with timeouts or large data transfer. The same exact functions and DataContracts/ServiceContracts/BehaviorContracts are being used.

Why would the way that the table is being populated have any bearing on the table returning successfully?

c# .net wcf web-services datatable

Share

Improve this question

Follow

edited Jul 25, 2015 at 21:48



Peter Mortensen

31.6k ● 22 ● 109 ● 133

asked Aug 15, 2008 at 20:26



goric

11.8k ● 9 ● 55 ● 71

8 Answers

Sorted by: Highest score (default) ▾



85



For anyone having similar problems, I have solved my issue. It was several-fold.

- As Darren suggested and Paul backed up, the Max..Size properties in the configuration needed to be enlarged. The SvcTraceViewer utility helped in determining this, but it still does not always give the most helpful error messages.
- It also appears that when the Service Reference is updated on the client side, the configuration will sometimes not update properly (e.g. Changing config values on the server will not always properly update on the client. I had to go in and change the Max..Size properties multiple times on both the client and server sides in the course of my debugging)
- For a DataTable to be serializable, it needs to be given a name. The default constructor does not give the table a name, so:

```
return new DataTable();
```

will not be serializable, while:

```
return new DataTable("someName");
```

will name the table whatever is passed as the parameter.

Note that a table can be given a name at any time by assigning a string to the `TableName` property of the `DataTable`.

```
var table = new DataTable();  
table.TableName = "someName";
```

Hopefully that will help someone.

Share

Improve this answer

Follow

edited Jun 23, 2015 at 15:28



saluce

13.3k ● 3 ● 51 ● 68

answered Sep 3, 2008 at 19:14



goric


11.8k ● 9 ● 55 ● 71

1 Very strange with the name thingy. I had same problem. Setting a name on the DataTable solved it... Weird. – [MartinHN](#) Aug 1, 2009 at 17:52

8 +1 for table name. I looked at the HTTP return in Fiddler, and VS uses that table name to generate a custom datatype. If the name is null, I guess it tries to use a blank name for that

custom datatype and just errors out. – [akatakritos](#) Dec 21, 2011 at 18:37

- 1 Great question and great answer. I remember having this problem with something else before, I can't remember, I think it was just classic web services. I don't want to open a can of worms, but I don't feel guilty using DataTables across the pipe because if I am only using it for display information, I don't think it matters at all. When you start passing DataTables and DataRows as parameters, that is when it becomes a major @\$\$#(*@%_ up. Objects are always safer, but if your data changes often because of a skitzo decision maker - then I don't see the harm just for display purposes. – [dyslexicanaboko](#) Sep 1, 2012 at 0:05

Holy crap on a cringle... TableName... I've been circling WCF issues for hours (days?) and I think this is the root cause. He's a vote for an almost 10 year old answer. – [WernerCD](#) May 30, 2017 at 1:10 



The best way to diagnose these kinds of WCF errors (the ones that really don't tell you much) is to enable tracing. In your web.config file, add the following:

15



```
<system.diagnostics>
  <sources>
    <source name="System.ServiceModel"
      switchValue="Information"
      propagateActivity="true">
      <listeners>
        <add name="ServiceModelTraceListener"
          type="System.Diagnostics.XmlWriterTraceListener, System,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
          initializeData="wcf-traces.svclog"/>
      </listeners>
    </source>
  </sources>
</system.diagnostics>
```

You can then open the resulting file in the SvcTraceViewer.exe utility which comes in the .NET Framework SDK (or with Visual Studio). On my machine, it can be found at %PROGRAMFILES%\Microsoft SDKs\Windows\v6.0A\Bin\SvcTraceViewer.exe.

Just look for an error message (in bold red) and that will tell you specifically what your problem is.

Share Improve this answer Follow

answered Aug 23, 2008 at 18:56



[Chris Gillum](#)

15k ● 5 ● 50 ● 66

You can find the log file `wcf-traces.svclog` in the `Debug` folder of your project.

– [Daniel Bonetti](#) Dec 20, 2017 at 19:04



Other than setting maximum values for all binding attributes.

6

Make sure each table you are passing/returning from webservice must have a table name, meaning the `table.tablename` property should not be blank.



Share

edited Jul 25, 2015 at 21:52

answered Dec 21, 2010 at 13:33



Improve this answer



Peter Mortensen

31.6k ● 22 ● 109 ● 133



aditya pathak

61 ● 1 ● 1



Follow



I added the DataTable to a data set and returned the table like so...

5

```
DataTable result = new DataTable("result");
```

```
//linq to populate the table
```

```
Dataset ds = new DataSet();
ds.Tables.Add(result);
return ds.Tables[0];
```



Hope it helps :)

Share Improve this answer Follow

answered Oct 24, 2011 at 8:29



Jani5e

163 ● 2 ● 8

In my case DataTable created as result of Database Query was working fine but it was always having problem with DataTable created in my code. It got solved when I added this table to newly created dataset. It is quite bizzare. I believe it is some bug in Microsoft's serialisation.

– Softec Mar 14, 2017 at 15:12



The attribute you want isOperationContract (on interface) / Operation Behavior (on method):

4

```
[ServiceContract]
public interface ITableProvider
{
    [OperationContract]
    DataTable GetTbl();
}

[OperationBehavior]
public DataTable GetTbl(){
    DataTable tbl = new DataTable("testTbl");
    //Populate table with SQL query

    return tbl;
}
```



Also, in the... I think service configuration... you want to specify that errors can be sent. You might be hitting an error that is something like the message size is too big, etc. You can fix that by fudging with the reader quotas and such.

By default `wsHttpBinding` has a receive size quota of like 65 KB, so if the serialized data table's XML is more than that, it would throw an error (and I'm 95% sure the data table is more than 65 KB with data in it).

You can change the settings for the reader quotas and such in the `web.config` / `app.config` or you can set it on a binding instance in code. But yeah, that's probably what your problem is, if you haven't changed it by default.

[WSHttpBindingBase Members](#) - Look at `ReaderQuotas` property as well as the `MaxReceivedMessageSize` property.

Share

edited Jul 25, 2015 at 21:51

answered Aug 15, 2008 at 20:35

Improve this answer



Peter Mortensen

31.6k ● 22 ● 109 ● 133



Darren Kopp

77.6k ● 9 ● 80 ● 93

Follow



You probably blew your quota - the datatable is larger than the allowed maximum packet size for your connection.

2



You probably need to set **MaxReceivedMessageSize** and **MaxBufferSize** to higher values on your connection.



Share Improve this answer Follow

answered Oct 24, 2008 at 13:33



Sam

29k ● 50 ● 170 ● 250



There are 3 reason for failed return type as `datatable` in WCF services

1



- You have to specify data table name like:

```
MyTable=new DataTable("tableName");
```

- When you are adding reference on client side of WCF service select reusable dll `system.data`
- Specify attribute on `datatable` member variable like

```
[DataMember]  
public DataTable MyTable{ get; set; }
```

Share

edited Dec 19, 2016 at 4:49

answered Dec 2, 2016 at 5:14

Improve this answer

Follow



Mukesh Methaniya

772 ● 1 ● 7 ● 13



0



I think Darren is most likely correct - the default values provided for WCF are laughably small and if you bump into them you end up with errors that can be difficult to track down. They seem to appear as soon as you attempt to do anything beyond a simple test case. I wasted more time than I'd like to admit debugging problems that turned out to be related to the various configuration (size) settings on both the client and server. I think I ended up modifying almost all of them, ex. MaxBufferPoolSize, MaxBufferSize, MaxConnections, MaxReceivedMessageSize, etc.

Having said that, the SvcTraceViewer utility also mentioned is great. I did run into a few cases where it wasn't as helpful as I would have liked, but overall it's a nice tool for analyzing the communications flow and errors.

Share Improve this answer Follow

answered Sep 2, 2008 at 14:18



Paul Mrozowski

6,734 ● 9 ● 37 ● 49