

## Re: Load Testing Errors

**Source:** <http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.general/2004-12/0004.html>

---

**From:** Ian Griffiths [C# MVP] ([ian-interact-sw\\_at\\_nospam.nospam](mailto:ian-interact-sw_at_nospam.nospam))

**Date:** 11/30/04

Date: Tue, 30 Nov 2004 10:18:33 -0000

This is not normal for a heavily loaded application – ASP.NET can quite happily put up with continuous heavy loads, as can SQL Server. These problems are indicative of programming errors.

There are many reasons an application might fail under load when it normally works, but the two main ones you run into most often are:

- \* multithreading problems
- \* resource starvation issues

I notice your exception is occurring inside a DataView. So this makes me wonder if perhaps it is the first one – are you sharing a single DataSet instance in your application?

I've seen problems when using a DataSet to cache information at the application level. You can do this, but you need to make sure that only one thread at a time uses the DataSet in question. (This is rather tricky if you're binding to the DataSet, as you appear to be doing in this case – you would need to enforce sequential access to the DataSet around your call to DataBind.)

An easy mistake to make with the DataSet is to think "I'm only reading data from this DataSet, so I don't need multi-threading protection." Unfortunately, anything that uses views onto the DataSet (e.g. data binding) does in fact modify the internal index cache of the DataTables in the DataSet, even if you're only reading data. I've seen code fail with NullReferenceException errors for exactly this reason when reading from a DataSet on multiple threads in the past, which is why I mention this.

So if you are sharing data in static (or Shared) fields or are using the Application state, then it could well be a multithreading thing.

Alternatively, there may be some resource that you're running out of when the system is under load, and you're not detecting this condition correctly, or are not taking the correct steps to avoid the problem. However, I'm not quite sure what could be happening that that would result in the errors you have posted, so I can't offer a useful hypothesis.

## microsoft.public.dotnet.general: Re: Load Testing Errors

Another possibility is that there's something about the way you're using the database that means you're just getting unexpected results when lots of stuff is happening concurrently. The errors suggest that you're expecting to see data but that it's missing. Perhaps you're getting null values back from your request when you were expecting non-null values. It is conceivable that you might see such a problem if you failed to use a transaction when one was necessary – a database request that works fine when the system is not under load can return inconsistent or unexpected results under heavy load if it doesn't ensure that its work is isolated through transactions.

(In other words, just because your request happens to work on an idle system doesn't mean that the request is bug-free.)

But whatever the problem is, this is not 'normal' – .NET web servers don't simply start throwing random exceptions under load. ASP.NET is a lot more robust than that!

--

Ian Griffiths - <http://www.interact-sw.co.uk/iangblog/>

DevelopMentor - <http://www.develop.com/>

"Shabam" wrote:

```
>A web application of mine developed using C# + MS SQL runs fine normally.
> However when I stress test it with a load testing software (using about 60
> simultaneous users) some instances start erroring out. I see two
> different
> errors. One is a "Object reference not set to an instance of an object."
> error, which appears to always contain the same information, and the other
> is a "There is no row at position X.", where X is a number.
>
> Is this an indication of bad coding or is this just a normal consequence
> of
> overloading a web application? How can the above two errors happen when
> the
> server is being overloaded when normally the application works fine?
>
>
> ERROR #1:
>
> Server Error in '/' Application.
> -----
> ----
>
> Object reference not set to an instance of an object.
> Description: An unhandled exception occurred during the execution of the
> current web request. Please review the stack trace for more information
> about the error and where it originated in the code.
>
> Exception Details: System.NullReferenceException: Object reference not set
> to an instance of an object.
>
> Source Error:
>
> An unhandled exception was generated during the execution of the current
> web
> request. Information regarding the origin and location of the exception
> can
> be identified using the exception stack trace below.
```

## microsoft.public.dotnet.general: Re: Load Testing Errors

```
>
> Stack Trace:
>
>
> [NullReferenceException: Object reference not set to an instance of an
> object.]
>   System.Data.DataView.GetRecord(Int32 recordIndex) +22
>   System.Data.DataView.IsOriginalVersion(Int32 index) +9
>   System.Data.DataRowView.GetColumnValue(DataColumn column) +23
>   System.Data.DataColumnPropertyDescriptor.GetValue(Object component) +25
>   System.Web.UI.DataBinder.GetPropertyValues(Object container, String
> propName) +72
>   System.Web.UI.DataBinder.GetPropertyValues(Object container, String
> propName, String format) +11
>   System.Web.UI.WebControls.ListControl.OnDataBinding(EventArgs e) +403
>   System.Web.UI.Control.DataBind() +26
>   FN.advancedsearch.populateListcontrols()
>   FN.advancedsearch.Page_Load(Object sender, EventArgs e)
>   System.Web.UI.Control.OnLoad(EventArgs e) +67
>   System.Web.UI.Control.LoadRecursive() +35
>   System.Web.UI.Page.ProcessRequestMain() +750
>
>
> ERROR #2:
>
> Server Error in '/' Application.
> -----
> ----
>
> There is no row at position 5.
> Description: An unhandled exception occurred during the execution of the
> current web request. Please review the stack trace for more information
> about the error and where it originated in the code.
>
> Exception Details: System.IndexOutOfRangeException: There is no row at
> position 5.
>
> Source Error:
>
> An unhandled exception was generated during the execution of the current
> web
> request. Information regarding the origin and location of the exception
> can
> be identified using the exception stack trace below.
>
> Stack Trace:
>
>
> [IndexOutOfRangeException: There is no row at position 5.]
>   System.Data.DataView.GetRecord(Int32 recordIndex) +60
>   System.Data.DataView.IsOriginalVersion(Int32 index) +9
>   System.Data.DataRowView.GetColumnValue(DataColumn column) +23
>   System.Data.DataColumnPropertyDescriptor.GetValue(Object component) +25
>   System.Web.UI.DataBinder.GetPropertyValues(Object container, String
> propName) +72
>   System.Web.UI.DataBinder.GetPropertyValues(Object container, String
> propName, String format) +11
>   System.Web.UI.WebControls.ListControl.OnDataBinding(EventArgs e) +403
>   System.Web.UI.Control.DataBind() +26
>   FN.advancedsearch.populateListcontrols()
>   FN.advancedsearch.Page_Load(Object sender, EventArgs e)
>   System.Web.UI.Control.OnLoad(EventArgs e) +67
```

microsoft.public.dotnet.general: Re: Load Testing Errors

```
> System.Web.UI.Control.LoadRecursive() +35  
> System.Web.UI.Page.ProcessRequestMain() +750  
>  
>
```