

Aspnet_wp.exe (PID: 1332) was recycled because it was suspected to be in a deadlocked state

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.adonet/2004-03/1718.html>

From: Tang Thim Sing (*tangts_at_silverglobe.com*)

Date: 03/18/04

Date: Fri, 19 Mar 2004 04:07:00 +0800

Hi

We have been tracking the source of this problem and we are pretty sure that the deadlock was not caused by our application. Most interestingly, we have also found from monitoring our web page requests from the log captured by the Internet Services Manager (in WINNT/system32/logfiles) that; on a few occasions when this error was reported in the event viewer, there were no users using the system.

I'm also aware that this problem is also reported on the Microsoft support site, being a false alarm due to slow write to the client. But in our case, during those times when there were no users???

We are also up against the timeout problem due to connection pool maxed out. We are seeing a pattern between these 2 problems. New connection pools are created each time the Aspnet_wp.exe is recycled. When this happens several times, the number of connection pools goes up. The inactive pools would not disappear until we restart the server. We noticed that there is a higher chance of the timeout problem when the number of connection pools is high (7-8).

We are also aware that the textbook answer to this timeout problem is to ensure that the connections are explicitly closed. While we are tidying up our system to ensure this, we have also experimented with the connection lifetime value. By setting a very small value (5 minutes) in a controlled environment, we found that all the connections in the pool are destroyed after 5 minutes or so after we logged off our system, even after having requested those pages that we know we have not explicitly closed the connections. This implies that the connections are being returned to the pool.

Regards

Tang Thim Sing