

SQL SERVER TROUBLESHOOTING CHECKLIST

Instance Name: _____

Performed by: _____

Date: _____

1. Can You Connect?

Connect to the SQL Server and run:

```
SELECT * FROM sys.Databases
```

Check returned columns for expected values, especially user_access_desc, state_desc, and log_reuse_wait.

- Connected successfully?
- Anything notable in returned columns? _____

Connect with the dedicated admin connection if necessary: <http://BrentOzar.com/go/DAC>

2. Who is active?

Don't stop to figure everything out, first gather a few key pieces of information using Adam Machanic's sp_whoisactive stored procedure:

```
EXEC master.dbo.sp_whoisactive
```

- Did you run this successfully?
- How many rows did it return? _____
- Was blocking present? (See the blocking_session_id column) _____

More info: <http://BrentOzar.com/go/active>

3. What's in the SQL Server Error Log?

Check with SQL Server Management Studio using 'Management' → 'SQL Server Logs'.

Alternately, query the log with this command:

```
EXEC xp_readerrorlog @p1=1 /*Error Log Number*/,  
@p2=1 /*Error Log Type- 1= SQL Server*/
```

Check all logs since at least the last startup and just prior. (Use @p1=2, @p1=3, etc to access prior logs.)

- Are there any recent errors/login failures recorded? _____
- When was the last startup date and time? _____
- Was the last restart part of planned maintenance? _____



4. What's your Quick Assessment?

Check all which appear to be involved:

• Access failure (security related):

• Availability loss (multiple reboots):

• Performance problem:

• Notes: _____

Is this an access failure? Verify if this is a security issue, a network issue, an application tier problem, etc.

Otherwise keep on goin'.

5. What's the Windows Event Log Got to Say?

Are there events in the windows logs at the same time or just before the problem periods?

• System log: _____

• Application log: _____

• Security log: _____

Remember, particularly in the system log, informational events may help explain errors you're seeing. Don't filter for errors *all* the time.

This step can be time consuming. If you have someone who can help, divide and conquer: ask them to look through the event logs and summarize them.

6. Capture SQL Server Activity to a Table

Run sp_whoisactive in a loop to log activity to a table. This gathers data while you keep looking.

• Activity is being captured to the database and table named: _____

More info: <http://BrentOzar.com/go/activehistory>

7. SQL Server Overall Waits

Run a query to find the top three SQL Server waits in sys.dm_os_wait_stats by percent since the last restart:

- _____
- _____
- _____

Other waits of interest are:

- _____

Many waits reported are ignorable. How do you know which you can skip? Get a solid query for sys.dm_os_wait_stats for your version of SQL from Glenn Berry: <http://BrentOzar.com/go/glenn>
Once you open the file search for 'os_wait'.

8. Review Performance Counters

Look at performance counters (and, ideally, their history) to identify system performance patterns and problems.

- Performance counters checked?
- Notable counter info : _____

More info: <http://BrentOzar.com/go/perfmon>

9. Make Notes by Major Areas

- CPU: _____
- Memory: _____
- Network: _____
- Disk: _____

10. Identify Recent changes

Have there been recent changes in these areas?

- Query Changes
 - Application Tier: _____
 - Stored procedures: _____
- Schema changes: _____
- Index changes: _____
- Major data changes: _____
- Infrastructure changes
 - Maintenance changes: _____
 - SQL Configuration changes: _____
 - Windows Configuration changes: _____

Notes

References:

- Adam Machanic ([blog](#) | [twitter](#)) created sp_whoisactive. We think it's great and like to help you use it.
- Glenn Berry ([blog](#) | [twitter](#)) publishes great queries for DMVs. We like to help you get to them [here](#).