Availability Group Failover Root Cause Analysis Tool

# What does the tool do?

Many factors can cause the failover of an availability group, such as user operation, unresponsive SQL Server, system wide issues hanging the SQL Server process, etc. The information contributing to the failover is logged in different places: Always On XEvent, System Health XEvent, SQL errorlog, system log, and windows cluster log. It is difficult for system administrators to understand the root cause of a failover. This tool provides the ability to automatically analysis log data and suggest possible root cause of a failover.

# What is the audience of this tool?

End users, DBA, system administrators and CSS engineers will find this tool helpful. As long as someone needs to know the root cause of a failover, this tool provides the analysis and diagnosis and presents possible root causes.

Before you can use the tool you need to provide the proper configuration. Then you can run the tool with various options.

# Configuration

After the file is unzipped, the following will be present in the folder:

* Tool executable FailoverDetector.exe
* Newtonsoft.Json.dll
* Configuration file Configuration.json
* Two empty folders named: Data and Results

The following shows the directory structure

${Executable\_Dir}

│   Configuration.json

│   FailoverDetector.exe

│   Newtonsoft.Json.dll

│

├───Data

└───Result

Now you need to configure the proper settings in Configuration.json. Open Configuration.json, and edit “Data Source Path” and “AG” (for Availability Group) topology based on your environment.

**Data Source Path** is the share location from where the analysis tool to access and copy all the necessary log files. The analysis tool will copy the log files from this location to the “Data” subdirectory under the analysis tool folder.

**AG** is availability group configuration information for the failover of the availability group to be analyzed. You will need to list the AG name and the instances (replicas) of this AG. The analysis tool will try to match the AG information with the copied log files and determine if the logs are complete or sufficient. The some files are missing such that the tool is unable to determine failover root cause or its ability to determine root cause will be constrained, the tool will report such information. For example, if a failover happened because the primary crashed, but the user does not provide the system log, the tool would be unable to determine the root cause and would report such information.

{

"Data Source Path" : "\\\\zechen-d1\\dbshare\\Data",

"Health Level": 3,

            "Instances": [

                "ze-2016-v1",

                "ze-2016-v2"

            ]

    ]

}

# Usage

## Collecting data:

Before you can use the tool, you need to gather the data and place them in a location that is accessible by the analysis tool.

Five categories of logs are required to do analysis:

* SQL error logs
* Always On XEvent
* System Health XEvent

The above three categories of logs can be found from the LOG folder of the SQL instance installation path. For example, you may have an SQL Server 2017 instance named "SQL17RTM01". If the installation path is on C drive, you will find the above three categories of logs under "C:\Program Files\Microsoft SQL Server\MSSQL14.SQL17RTM01\MSSQL\Log", and copy all AlwaysOn XEvent, System Health XEvent and SQL errorlog files.

* System log
* Windows cluster log

The above two categories of logs can be found.  You can use similar commands below to collect these two categories of logs.

Get-ClusterLog -Node $server.ComputerNamePhysicalNetBIOS -Destination $server.ErrorLogPath

Get-Eventlog -ComputerName $server.ComputerNamePhysicalNetBIOS -LogName System | Export-CSV -Path $SystemLogExportPath

Here is more specific information on each category of logs:

SQL Error log: Files whose name starts with ERRORLOG or has ERRORLOG as part of it.

Always On XEvent: Files whose name starts with AlwaysOn\_health.

System health XEvent: Files whose name starts with System\_health.

## Organizing data

The log files collected in the previous step should be placed in a folder with correct subfolder structure. This folder should be the folder path specified in the “Data Source Path” section of the configuration file “Configuration.json”.

The following is an example of the subfolder structure, assuming there are two replicas (instances) in the AG. One is “ze-2016-v1” and another one is “ze-2016-v2”.

│   └───Data

│       │

│       │

│       ├───ze-2016-v1

│       │       AlwaysOn\_health\_0\_131532583880140000.xel

│       │       AlwaysOn\_health\_0\_131532634780070000.xel

│       │       AlwaysOn\_health\_0\_131532680183890000.xel

│       │       AlwaysOn\_health\_0\_131532701164380000.xel

│       │       ERRORLOG

│       │       ERRORLOG.1

│       │       ERRORLOG.2

│       │       ERRORLOG.3

│       │     SQLDUMPER\_ERRORLOG.log

│       │       system\_health\_0\_131532583879830000.xel

│       │       system\_health\_0\_131532634779760000.xel

│       │       system\_health\_0\_131532680183430000.xel

│       │       system\_health\_0\_131532701164070000.xel

│       │       ze-2016-v1\_system.log

│       │       ze-2016-v1\_cluster.log

│       │

│       └───ze-2016-v2

│               AlwaysOn\_health\_0\_131532571347210000.xel

│               AlwaysOn\_health\_0\_131532578226200000.xel

│               AlwaysOn\_health\_0\_131532586348180000.xel

│               AlwaysOn\_health\_0\_131532725682240000.xel

│               ERRORLOG

│               ERRORLOG.1

│               ERRORLOG.2

│               ERRORLOG.3

│               ERRORLOG.4

│               ERRORLOG.5

│               ERRORLOG.6

│               system\_health\_0\_131532571346430000.xel

│               system\_health\_0\_131532578225950000.xel

│               system\_health\_0\_131532586347860000.xel

│               system\_health\_0\_131532725681930000.xel

│               ze-2016-v2\_system.csv

│               ze-2016-v2\_cluster.log

## Using the tool

**Default Mode (no parameter specified)**

In default mode, no parameter needs to be specified when running the tool. The analysis tool will load the configuration file, copy log data from data source path to local workspace data folder, examine the files with the configuration settings, perform the analysis, and write the analysis result to a Json report file.

User should make sure the execution credential running tool has access to share folder defined in Configuration.json. When tool runs into access denial issues or the directory does not exist, the tool will exit with proper error message.

After the tool copies the log files to the local tool directory, it will scan the local data folder and compare with the AG configuration in Configuration.json. If it finds log data is not complete, the tool will show which files are missing. The tool will also alert the user that the failover root cause might not be properly identified due to missing files.

If analysis is successful, the tool will write result report as Json format in Result directory. Analysis result will be presented on the console, unless the “--Show” parameter is specified when running the tool. See below for details.

**“--Analyze”**

When “--Analyze” is specified as a parameter, the tool will load configuration file without copying log data. It assumes the log files have already been copied over. It does everything as default mode except copying log data. This option is useful if you already have the data in the local tool execution subdirectories and want to rerun the analysis.

**“--Show”**

This tool running with show parameter will display analysis report on console after analyzing log data, in addition to persisting results to a Json file

# Currently supported failover root causes

The tool in current form can analyze and report failover reason if a failover is due to the following causes.

## Orderly service changes

* Stopping SQL service: Admin or other process gracefully shuts down SQL Serverhosting the AG primary. In this case, Windows notifies WSFC to initiate a failover and picks one secondary as new primary, if automatic failover is enabled
* Shutting down Windows Server: The Windows Server which hosts the SQL Server that has the primary replica is shut down or restarted. In this case, Windows notifies WSFC to initiate a failover and picks one secondary as new primary, if automatic failover is enabled
* Manual failover: An administrator performs a manual failover through Management Studio or through a direct Transact-SQL statement “ALTER AVAILABILITY GROUP <AG name> FAILOVER” or “ALTER AVAILABILITY GROUP <AG name> FAILOVER WITH DATA\_LOSS” on the secondary replica that will become the primary replica.

## SQL Service Internal Health Issue

* Too much dumps from SQL Server: There are more than 100 SQL Server dumps on the primary replica since the last SQL Server restart, and there is at least one dump in the last 10 seconds. In this case, sp\_server\_diagnostics running in SQL Server determines that SQL Server system component is in error state and notifies failover WSFC to initiate a failover, if automatic failover is enabled.
* Memory scribbler: SQL Server has write access violation and the write address is more than 64KB. If there are more than three such memory corruptions since SQL Server started,, sp\_server\_diagnostics running in SQL Server determines that SQL Server system component is in error state and notifies failover WSFC to initiate a failover, if automatic failover is enabled.
* Sick Spin-lock: After an access violation, a spin-lock is marked as sick if it backs off more than three times, which is the threshold. In this case, sp\_server\_diagnostics running in SQL Server determines that SQL Server system component is in error state and notifies failover WSFC to initiate a failover, if automatic failover is enabled.
* Out of Memory: If no memory has been freed in 2 minutes, sp\_server\_diagnostics running in SQL Server determines that SQL Server system component is in error state and notifies failover WSFC to initiate a failover, if automatic failover is enabled.
* Unresolved deadlock: If sp\_server\_diagnostics detects unresolved deadlock from query processing component, it determines that SQL Server system component is in error state and notifies failover WSFC to initiate a failover, if automatic failover is enabled. Deadlocked scheduler: If sp\_server\_diagnostics detects deadlocked schedulers from query processing component, it determines that SQL Server system component is inerror state and notifies failover WSFC to initiate a failover, if automatic failover is enabled.

## System Wide Issues

* Unexpected crash: SQL Server service was shut down unexpectedly. In this scenario, SQL Server crashed without error message or exception thrown. Resources host service (rhs.exe) does not detect lease check from SQL Server about availability group lease. This results an AG lease timeout signal to WSFC and WSFC will initiate a failover if automatic failover is enabled.
* Long dump: SQL Server is creating a dump file. During the process, threads handle AG lease are frozen and exceed lease timeout. Resources host service (rhs.exe) can not detect lease check from SQL Server for availability group lease. This results an AG lease timeout signal to WSFC and WSFC will initiate a failover if automatic failover is enabled.Quorum loss: AG resource is brought offline because quorum is lost. This could be due to connectivity issue, but we do not have further evidence to conclude more detail answer.
* Network interface failure: Network interface used to communicate between cluster nodes fails. Primary and secondary replicas cannot communicate. WSFC will initiate quorum vote and determine a new primary to fail over to.
* Cluster disk failure: Network interface used to communicate between cluster nodes failed. It results nodes between primary and secondary cannot communicate. Cluster will initiate quorum vote and determine a new primary failover to.
* Cluster Service halted: Cluster service on primary was halted, resulting in primary unable to communicate with other nodes in the cluster. Cluster will initiate quorum vote and determine a new primary to fail over to.
* Node offline: When primary node is frozen or loses power, WSFC loses connection from and to the primary. Failover cluster decide to initiate failover and pick a primary from other possible nodes.

In some situations, the analysis tool can only provide a best guess or estimate on the failover reason.

## Best estimate

* High CPU utilization: System wide performance issue causes SQL Server service unable respond to AG lease handler. It is possible that other process or SQL Server is taking 100% CPU resource for a long time.
* High I/O: System wide performance issue causes SQL Server service unable respond to AG lease handler. It is possible that other process or SQL Server has high I/O requests for a long time.