Instance-Level Configuration Queries Part 1

Glenn Berry
Glenn@SQLskills.com
http://www.SQLskills.com/blogs/Glenn

pluralsight
hardcore developer training

Instance-Level Configuration Queries

- **Initial queries to collect hardware and instance configuration information**
  - These can be run in the context of any database on the instance
  - These are not database specific
- **Very high percentage of SQL Server instances have instance-level configuration issues**
  - Most instance-level configuration issues are relatively easy to correct
- **My Pluralsight course *SQL Server 2012: Installation and Configuration* covers best practice instance-level configuration**
  - http://bit.ly/1nUiv7m

Getting the Product Version

- **This query is actually a T-SQL function: @@VERSION**
  - MSDN link: http://bit.ly/1gR4xtJ
- **It is very important to know the exact version, edition, and build information for a SQL Server instance**
  - Major SQL Server version (SQL Server 2005-2014)
  - SQL Server Edition (Standard or Enterprise)
    - Many product features are only available in Enterprise Edition
    - There are different license limits between Standard and Enterprise Edition
  - SQL Server build number
    - Tells you the release branch (RTM, SP1, SP2, etc.)
    - Tells you the Cumulative Update (CU) level
  - Windows version information is also returned from this query
    - It is useful to know the major version of Windows that your server is using

When Was SQL Server Installed?

- **This query is actually a security catalog view**
  - sys.server_principals
  - MSDN link: http://bit.ly/1epCebh
- **Knowing when SQL Server was installed is very valuable**
  - It gives you a sense for how old the server hardware might be
  - It gives you an idea of how old the SQL Server instance is

Getting Useful Server Properties

- **This query is actually a T-SQL metadata function**
  - SERVERPROPERTY
  - MSDN link: http://bit.ly/1eeDgq6
- **Returns property information about the SQL Server instance**
  - Many different properties are available
    - MachineName
    - InstanceName
    - ComputerNamePhysicalNetBIOS
    - Edition
    - ProductVersion
    - Collation
    - IsIntegratedSecurityOnly
    - IsHadrEnabled
    - IsXTPSupported

Getting SQL Server Agent Job Information

- **This query returns information about SQL Server Agent jobs from the msdb system database**
  - msdb.dbo.sysjobs and msdb.dbo.syscategories
- **High-level information about each SQL Server Agent job**
  - Name
  - Description
  - Job owner
  - Date created
  - Whether the job is enabled
  - Whether there is an e-mail operator for the job
  - The job category name

Getting SQL Server Agent Alert Information

- **This query returns information about SQL Server Agent alerts from the msdb system database**
  - msdb.dbo.sysalerts
- **SQL Server Agent Alerts are not the same as Agent Jobs…**
  - They are fired when certain errors or events occur
  - You can read more about SQL Server Agent Alerts here:
    - http://bit.ly/1n9wAK9
- **Query returns high-level information about your Agent Alerts**
  - Name
  - Message_id and severity
  - Whether the alert is enabled and has a notification operator
  - The delay between notifications (in seconds)
  - The alert occurrence count
  - The last occurrence date and time

Global Trace Flag Information

- **This query is a DBCC command**
  - DBCC TRACESTATUS (-1)
  - MSDN link: http://bit.ly/1hEuqNe
- **Knowing which global trace flags are enabled is very useful**
  - Helps detect some third-party and Microsoft applications
  - Helps you determine whether the instance might be operating in an unusual fashion due to specific trace flags
  - Books Online Trace Flag Reference
    - http://bit.ly/1hwGAh5
- **Two common trace flags that are usually a good idea to enable**
  - TF 3226
  - TF 1118

Getting Information About Windows

- **This is a DMV query (added in SQL Server 2008 R2 SP1) that returns high-level information about Windows**
  - sys.dm_os_windows_info
  - MSDN link: http://bit.ly/1eeJNkR
- **Returns major OS version, service pack, SKU, and language**

SQL Server Services Information

- **This is a DMV query (added in SQL Server 2008 R2 SP1) that returns information about your SQL Server Services**
  - sys.dm_server_services
  - MSDN link: http://bit.ly/OLQQoS
- **Gives you information about:**
  - Service name
  - Process ID
  - Startup type description
  - Current service status
  - Last startup time
  - Whether the service is clustered
  - What node the service is running on

SQL Server NUMA Node Information

- **This DMV query gives you information about your NUMA nodes**
  - sys.dm_os_nodes
  - MSDN link: http://bit.ly/1jxeNf9
- **Lets you determine how many NUMA nodes are visible to the instance**
  - This is useful for both physical and virtual machines
- **Lets you see how many schedulers are available per NUMA node**

Getting Hardware Information

- **This is a very handy DMV for getting quite a bit of useful hardware information**
  - sys.dm_os_sys_info
  - MSDN link: http://bit.ly/1jxfGnN
- **Gives you your CPU socket count and total logical processor count**
  - Cannot tell the difference between physical and logical cores
- **Tells you how much physical RAM is in the server**
- **Tells you the last SQL Server start time**
  - This is very important information
- **The virtual machine type description is deceiving**
  - It will show HYPERVISOR if you have Hyper-V running on your machine, even if your SQL Server instance is NOT virtualized
  - Books Online is currently somewhat confusing about this

Getting the Server Model Number

- **This query uses xp_readerrorlog to look for the manufacturer of the server you are running on**
  - It also usually returns the model number of the server
  - This query might take some time if your SQL Server error log is very large
- **The first parameter is the error log to read from**
  - A value of zero is the most recent error log
- **The second parameter is the error log type**
  - 1 is the SQL Server error log
  - 2 is the SQL Server Agent error log
- **Knowing the server manufacturer and model number is useful**
  - Helps you determine the age and capabilities of the server
  - How many processors, how much memory can it hold, how many PCI-E slots it has, etc.

Getting the Processor Description

- **This query uses xp_instance_regread to get the processor description**
  - This is sort of a cumbersome way to get this information
  - It would be better if sys.dm_os_sys_info had this information
- **Usually gives you the manufacturer and model number of your processor(s), along with the rated base clock speed**
  - It does not tell you the current clock speed of your processor
  - You cannot use this to determine whether power management is in effect

Summary

- **DMV/DMO queries can help you detect most configuration issues**
  - They can help you find instance-level settings that may be incorrect
- **They can help you determine what type of hardware you are using**
  - How old it is, what its capabilities are

What is Next?

- **Module 3: Instance-Level Configuration Queries Part 2**
  - SQL Server error log properties
  - Cluster properties
  - Current cluster node
  - AlwaysOn AG cluster information
  - Instance configuration information
  - Buffer Pool Extension (BPE) information
  - BPE usage information
  - TCP listener states
  - Memory dump information
  - Database filenames and paths
  - Volume information