Server Level Transact-SQL Statements a SQL Server DBA should know

T-SQL Statement 1

The following T-SQL statement retrieves information such as Hostname, Current instance name, Edition, Server type, ServicePack and version number from current SQL Server connection. 'Edition' will give information on a 32 bit or 64 bit architecture and 'Productlevel' gives information about what service pack your SQL Server is on. It also displays if the current SQL Server is a clustered server. Refer Fig 1.1

```
SELECT
          SERVERPROPERTY('MachineName') as Host,
          SERVERPROPERTY('InstanceName') as Instance,
          SERVERPROPERTY('Edition') as Edition, /*shows 32 bit or 64 bit*/
          SERVERPROPERTY('ProductLevel') as ProductLevel, /* RTM or SP1 etc*/
          Case SERVERPROPERTY('IsClustered') when 1 then 'CLUSTERED' else
      'STANDALONE' end as ServerType,
          @@VERSION as VersionNumber
```

Host	Instance	Edition	ProductLevel	ServerType	VersionNumber
POWERPC	NYT_SQLENG	Enterprise Edition	RTM	STANDALONE	Microsoft SQL Server 2008 R2 (RTM) - 10.50.1600...

Fig 1.1

T-SQL Statement 2

Server level configuration controls some of the features and performance of SQL Server. It is also important for a SQL Server DBA to know the server level configuration information. The following SQL Statement will give all of the information related to Server level configuration. Refer Fig 1.2

```
SELECT * from sys.configurations order by NAME
```

If you are using SQL Server 2000, you can execute the following command instead.

```
SP_CONFIGURE 'show advanced options',1
Go
RECONFIGURE with OVERRIDE
Go
SP_CONFIGURE
Go
```

configuration_id	name	value	minimum	maximum	value_in_use	description
1582	access check cache bucket count	0	0	16384	0	Default hash bucket count fo
1583	access check cache quota	0	0	2147483647	0	Default quota for the access
16391	Ad Hoc Distributed Queries	0	0	1	0	Enable or disable Ad Hoc Dis
1550	affinity I/O mask	0	-2147483648	2147483647	0	affinity I/O mask
1535	affinity mask	0	-2147483648	2147483647	0	affinity mask
16384	Agent XPs	1	0	1	1	Enable or disable Agent XPs
102	allow updates	0	0	1	0	Allow updates to system table
1548	awe enabled	0	0	1	0	AWE enabled in the server
1579	backup compression default	0	0	1	0	Enable compression of backu

Fig 1.2

T-SQL Statement 3

Security is a very important aspect that a DBA should know about. It is also important to know which login has a sysadmin or security admin server level role. The following SQL Command will show information related to the security admin server role and system admin server role. Refer Fig 1.3

```
SELECT l.name, l.denylogin, l.isntname, l.isntgroup, l.isntuser
  FROM master.dbo.syslogins l
WHERE l.sysadmin = 1 OR l.securityadmin = 1
```

name	denylogin	isntname	isntgroup	isntuser
sa	0	0	0	0
NT AUTHORITY\SYSTEM	0	1	0	1
NT SERVICE\MSSQL$NYT_SQLENG	0	1	1	0
PowerPC\MAK	0	1	0	1
NT SERVICE\SQLAgent$NYT_SQLENG	0	1	1	0

Fig 1.3

T-SQL Statement 4

Another important bit of information that you need to know as a DBA is all of the traces that are enabled. The following T-SQL statement will list all of the trace flags that are enabled gloabally on the server. Refer Fig 1.4

```
DBCC TRACESTATUS(-1);
```

The following T-SQL statement will list all the trace flags that are enabled on the current sql server connection. Refer Fig 1.4

```
DBCC TRACESTATUS();
```

TraceFlag	Status	Global	Session
3605	1	0	1

Database Level Transact-SQL Statements a SQL Server DBA should know

T-SQL Statement 5

Getting Database level information is equally as important as Server level information. The following T-SQL statement gives information on the database names, their compatibility level and also the recovery model and their current status. The result from this T-SQL Statement will help you to determine if there is any compatibility level update necessary. When upgrading from an older version to new version, the compatibility level of the database may not be in the desired level. The following statement will help you to list all of the database names with compatibilty level. It also lists the online/offline status of the database as well as helping the DBA to see if any update to recovery model is necessary. Refer Fig 1.5

```
SELECT name,compatibility_level,recovery_model_desc,state_desc  FROM
sys.databases
```

name	compatibility_level	recovery_model_desc	state_desc
master	100	SIMPLE	ONLINE
tempdb	100	SIMPLE	ONLINE
model	100	FULL	ONLINE
msdb	100	SIMPLE	ONLINE
ReportServer$NYT_SQLENG	100	FULL	ONLINE
ReportServer$NYT_SQLENGTempDB	100	SIMPLE	ONLINE
AdventureWorks2008	100	SIMPLE	ONLINE
AdventureWorksDW2008	100	SIMPLE	ONLINE
AdventureWorksLT2008	100	SIMPLE	ONLINE
AdventureWorks	100	FULL	ONLINE
AdventureWorksDW	100	SIMPLE	ONLINE

Fig 1.5

If you are using SQL Server 2000, you could execute the following T-SQL Statement. Refer Fig 1.6

```
SELECT name,cmptlevel,DATABASEPROPERTYEX(name,'Recovery')AS RecoveryModel,
DATABASEPROPERTYEX(name,'Status') as Status FROM sysdatabases
```

name	cmptlevel	RecoveryModel	Status
master	100	SIMPLE	ONLINE
tempdb	100	SIMPLE	ONLINE
model	100	FULL	ONLINE
msdb	100	SIMPLE	ONLINE
ReportServer$NYT_SQLENG	100	FULL	ONLINE
ReportServer$NYT_SQLENGTempDB	100	SIMPLE	ONLINE
AdventureWorks2008	100	SIMPLE	ONLINE
AdventureWorksDW2008	100	SIMPLE	ONLINE
AdventureWorksLT2008	100	SIMPLE	ONLINE
AdventureWorks	100	FULL	ONLINE
AdventureWorksDW	100	SIMPLE	ONLINE

T-SQL Statement 6

The next level of information related to database that is needed is the location of the database. The following T-SQL Statement provides the logical name and the physical location of the data/log files of all the databases available in the current SQL Server instance. Referg Fig 1.7

```
SELECT db_name(database_id) as DatabaseName,name,type_desc,physical_name FROM
sys.master_files
```

DatabaseName	name	type_desc	physical_name
master	master	ROWS	C:\Program Files\Microsoft
master	mastlog	LOG	C:\Program Files\Microsoft
tempdb	tempdev	ROWS	C:\Program Files\Microsoft
tempdb	templog	LOG	C:\Program Files\Microsoft
model	modeldev	ROWS	C:\Program Files\Microsoft
model	modellog	LOG	C:\Program Files\Microsoft
msdb	MSDBData	ROWS	C:\Program Files\Microsoft
msdb	MSDBLog	LOG	C:\Program Files\Microsoft
ReportServer$NYT_SQLENG	ReportServer$NYT_SQLENG	ROWS	C:\Program Files\Microsoft

Fig 1.7

If you are using SQL Server 2000, you could execute the following T-SQL Statement. Refer Fig 1.8

```
SELECT db_name(dbid) as DatabaseName,name,filename FROM
master.dbo.sysaltfiles
```

DatabaseName	name	filename
master	master	C:\Program Files\Microsoft SQL Server\MSS
master	mastlog	C:\Program Files\Microsoft SQL Server\MSS
tempdb	tempdev	C:\Program Files\Microsoft SQL Server\MSS
tempdb	templog	C:\Program Files\Microsoft SQL Server\MSS
model	modeldev	C:\Program Files\Microsoft SQL Server\MSS
model	modellog	C:\Program Files\Microsoft SQL Server\MSS
msdb	MSDBData	C:\Program Files\Microsoft SQL Server\MSS
msdb	MSDBLog	C:\Program Files\Microsoft SQL Server\MSS
ReportServer$NYT_SQLENG	ReportServer$NYT_SQLENG	C:\Program Files\Microsoft SQL Server\MSS

Fig 1.8

T-SQL Statement 7

A database may contain filegroups other than the primary file group. The following T-SQL Statement gets executed in each database on the server and displays the file groups related results. Refer Fig 1.9

```
EXEC master.dbo.sp_MSforeachdb @command1 = 'USE [?] SELECT * FROM
sys.filegroups'
```

name	data_space_id	type	type_desc	is_default	filegroup_guid	log_filegroup_id	is_read_only
PRIMARY	1	FG	ROWS_FILEGROUP	1	NULL	NULL	0

name	data_space_id	type	type_desc	is_default	filegroup_guid	log_filegroup_id	is_read_only
PRIMARY	1	FG	ROWS_FILEGROUP	1	NULL	NULL	0

name	data_space_id	type	type_desc	is_default	filegroup_guid	log_filegroup_id
PRIMARY	1	FG	ROWS_FILEGROUP	1	NULL	NULL
SalesGr...	2	FG	ROWS_FILEGROUP	0	A9579A3F-F260-4AC4-A8B9-B4FC5C0D33E5	NULL
SalesGr...	3	FG	ROWS_FILEGROUP	0	A209BE5D-50D8-4750-B140-6DA7F3185812	NULL

Fig 1.9

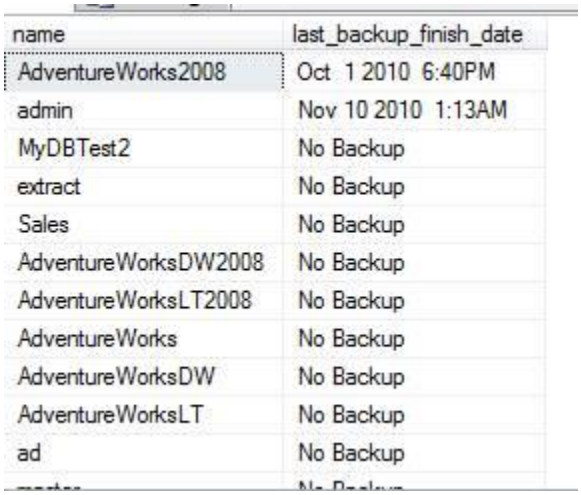
Backup Level Transact-SQL Statements a SQL Server DBA should know

T-SQL Statement 8

Backup of a database is bread and butter for database administrators. The following T-SQL Statement lists all of the databases in the server and the last day the backup happened. This will help the database administrators to check the backup jobs and also to make sure backups are happening for all the databases. Refer Fig 1.10

```
SELECT db.name,
case when MAX(b.backup_finish_date) is NULL then 'No Backup' else
convert(varchar(100),
```

```
        MAX(b.backup_finish_date)) end AS last_backup_finish_date
FROM sys.databases db
LEFT OUTER JOIN msdb.dbo.backupset b ON db.name = b.database_name AND b.type
= 'D'
        WHERE db.database_id NOT IN (2)
GROUP BY db.name
ORDER BY 2 DESC
```



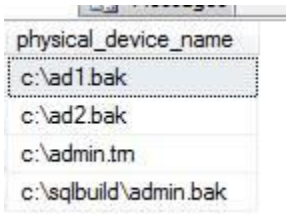
name	last_backup_finish_date
AdventureWorks2008	Oct 1 2010 6:40PM
admin	Nov 10 2010 1:13AM
MyDBTest2	No Backup
extract	No Backup
Sales	No Backup
AdventureWorksDW2008	No Backup
AdventureWorksLT2008	No Backup
AdventureWorks	No Backup
AdventureWorksDW	No Backup
AdventureWorksLT	No Backup
ad	No Backup

Fig 1.10

T-SQL Statement 9

The next level of information that is important for a SQL Server database administrator to know is the location of all the backup files. You don't want the backups to go to the local drive or to an OS drive. The following T-SQL statement gets all the information related to the current backup location from the msdb database. Refer Fig 1.11

```
SELECT Distinct physical_device_name FROM msdb.dbo.backupmediafamily
```



physical_device_name
c:\ad1.bak
c:\ad2.bak
c:\admin.tm
c:\sqlbuild\admin.bak

Fig 1.11

Process Level Transact-SQL Statements a SQL Server DBA should know

T-SQL Statement 10

Last but not least, is the information related to current processes and connection related information. From the beginning, SQL Server database administrators used sp_who and

sp_who2 to check the current users, process and session information. These statements also provided information related to cpu, memory and blocking information related to the sessions. Refer Fig 1.12. Also, search the internet for sp_who3. You can find many articles related to sp_who3.

```
sp_who
sp_who2
```

SPID	Status	Login	HostName	BlkBy	DBName	Command	CPUTime	DiskIO	LastBatch
1	BACKGROUND	sa	.	.	NULL	RESOURCE MONITOR	0	0	01/12 03:22:32
2	BACKGROUND	sa	.	.	NULL	XE TIMER	93	0	01/12 03:22:32
3	BACKGROUND	sa	.	.	NULL	XE DISPATCHER	0	0	01/12 03:22:32
4	BACKGROUND	sa	.	.	NULL	LAZY WRITER	15	0	01/12 03:22:32
5	BACKGROUND	sa	.	.	NULL	LOG WRITER	15	0	01/12 03:22:32
6	BACKGROUND	sa	.	.	NULL	LOCK MONITOR	0	0	01/12 03:22:32
7	BACKGROUND	sa	.	.	master	SIGNAL HANDLER	0	0	01/12 03:22:32
8	sleeping	sa	.	.	master	TASK MANAGER	0	0	01/12 03:22:32
9	BACKGROUND	sa	.	.	master	FSAGENT TASK	0	0	01/12 03:22:32
10	BACKGROUND	sa	.	.	master	FSAGENT TASK	0	0	01/12 03:22:32
11	BACKGROUND	sa	.	.	master	FSAGENT TASK	0	0	01/12 03:22:32