**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.

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

retrieve server information  
**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

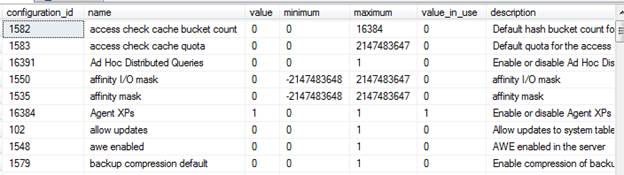
Go

RECONFIGURE with OVERRIDE

Go

SP\_CONFIGURE

Go

  
**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](http://www.webopedia.com/TERM/S/system_administrator.html) 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

  
**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();

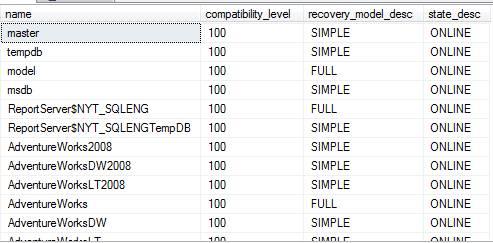
list all the trace flags that are enabled on the current sql server connection  
**Fig 1.4**

**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

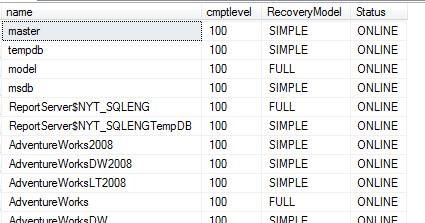
SELECT name,compatibility\_level,recovery\_model\_desc,state\_desc FROM sys.databases

  
**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,

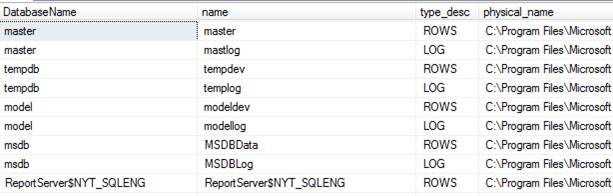
DATABASEPROPERTYEX(name,'Status') as Status FROM sysdatabases

  
**Fig 1.6**

**T-SQL Statement 6**

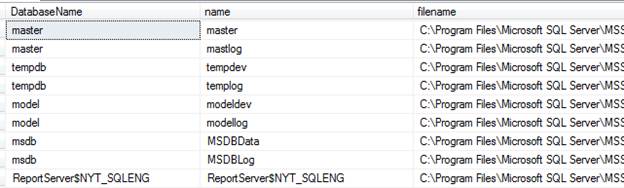
The next level of information related to database that is needed is the location of the database. The following T-SQL Statement provides the logical name and the physical location of the data/log files of all the databases available in the current SQL Server instance. Referg Fig 1.7

SELECT db\_name(database\_id) as DatabaseName,name,type\_desc,physical\_name FROM sys.master\_files

  
**Fig 1.7**

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

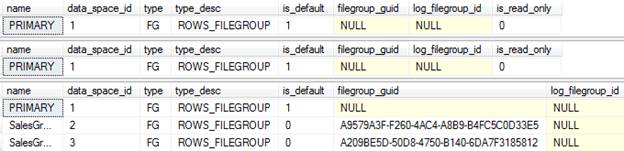
SELECT db\_name(dbid) as DatabaseName,name,filename FROM master.dbo.sysaltfiles

  
**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'

  
**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

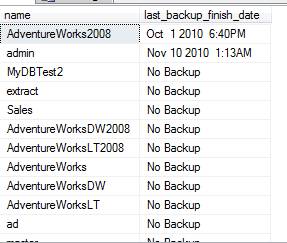
FROM sys.databasesdb

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

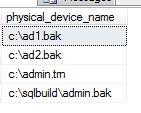
ORDER BY 2 DESC

  
**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 Distinctphysical\_device\_name FROM msdb.dbo.backupmediafamily

  
**Fig 1.11**

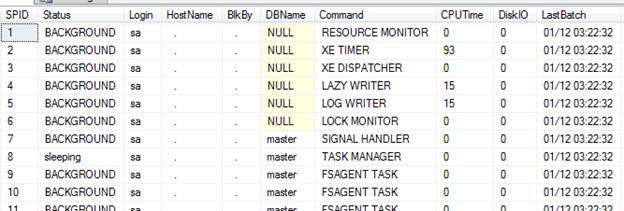
**Process Level Transact-SQL Statements a SQL Server DBA should know**

**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



Check the database size

SELECT sys.databases.name,

CONVERT(VARCHAR,SUM(size)\*8/1024)+' MB'AS[Total disk space]

FROMsys.databases

JOINsys.master\_files

ONsys.databases.database\_id=sys.master\_files.database\_id

GROUPBY sys.databases.name

ORDERBY sys.databases.name;

Insert data script

set nocount on

declare @i int

set @i=1

while(@i<=10)

begin

insert test01 values('test'+cast(@i as varchar(10)))

set @i=@i+1

end