### 1) Are all of your SQL Server services running?

Obviously once you connect to your instance you know that the database service is up and running but you can use the [extended stored procedure xp\_servicecontrol](https://www.mssqltips.com/sqlservertip/2036/monitor-start-and-stop-sql-server-services-using-xpservicecontrol/) to check if any service is up and running. Below are a few examples of what you can check.

exec master.dbo.xp\_servicecontrol 'QUERYSTATE', 'MSSQLServer'

exec master.dbo.xp\_servicecontrol 'QUERYSTATE', 'SQLServerAgent'

exec master.dbo.xp\_servicecontrol 'QUERYSTATE', 'SQLBrowser'

**2) Check Jobs error**

SELECT J.[name]

,[step\_name]

,[message]

,[run\_status]

,[run\_date]

,[run\_time]

,[run\_duration]

FROM [msdb].[dbo].[sysjobhistory] JH

JOIN [msdb].[dbo].[sysjobs] J

ON JH.job\_id= J.job\_id

WHERE [run\_date] <= Format(getdate(),'yyyyMMdd') and [run\_status] = 0

order by 5 desc

### 3) Are you running out of space on any of your disks on your SQL Server?

You can used the extended stored procedure xp\_fixeddrives to get a quick look at the space left on your drives.

GB

SELECT DISTINCT

volume\_mount\_point [Disk Mount Point],

file\_system\_type [File System Type],

logical\_volume\_name as [Logical Drive Name],

CONVERT(DECIMAL(18,2),total\_bytes/1073741824.0) AS [Total Size in GB], ---1GB = 1073741824 bytes

CONVERT(DECIMAL(18,2),available\_bytes/1073741824.0) AS [Available Size in GB],

CAST(CAST(available\_bytes AS FLOAT)/ CAST(total\_bytes AS FLOAT) AS DECIMAL(18,2)) \* 100 AS [Space Free %]

FROM sys.master\_files

CROSS APPLY sys.dm\_os\_volume\_stats(database\_id, file\_id)

---------------------------------------------------

MB

SELECT DISTINCT dovs.logical\_volume\_name AS LogicalName,

dovs.volume\_mount\_point AS Drive,

CONVERT(INT,dovs.available\_bytes/1048576.0) AS FreeSpaceInMB

FROM sys.master\_files mf

CROSS APPLY sys.dm\_os\_volume\_stats(mf.database\_id, mf.FILE\_ID) dovs

ORDER BY FreeSpaceInMB ASC

-----------------------------------

EXEC MASTER..xp\_fixeddrives

**4) Check Databases Log Space**

DBCC SQLPERF(logspace)

**5) Backup should be checked everyday**

SELECT DB.name AS Database\_Name

,MAX(DB.recovery\_model\_desc) AS Recovery\_Model

,MAX(BS.backup\_start\_date) AS Last\_Backup

,MAX(CASE WHEN BS.type = 'D'

THEN BS.backup\_start\_date END)

AS Last\_Full\_backup

,SUM(CASE WHEN BS.type = 'D'

THEN 1 END)

AS Count\_Full\_backup

,MAX(CASE WHEN BS.type = 'L'

THEN BS.backup\_start\_date END)

AS Last\_Log\_backup

,SUM(CASE WHEN BS.type = 'L'

THEN 1 END)

AS Count\_Log\_backup

,MAX(CASE WHEN BS.type = 'I'

THEN BS.backup\_start\_date END)

AS Last\_Differential\_backup

,SUM(CASE WHEN BS.type = 'I'

THEN 1 END)

AS Count\_Differential\_backup

,MAX(CASE WHEN BS.type = 'F'

THEN BS.backup\_start\_date END)

AS LastFile

,SUM(CASE WHEN BS.type = 'F'

THEN 1 END)

AS CountFile

,MAX(CASE WHEN BS.type = 'G'

THEN BS.backup\_start\_date END)

AS LastFileDiff

,SUM(CASE WHEN BS.type = 'G'

THEN 1 END)

AS CountFileDiff

,MAX(CASE WHEN BS.type = 'P'

THEN BS.backup\_start\_date END)

AS LastPart

,SUM(CASE WHEN BS.type = 'P'

THEN 1 END)

AS CountPart

,MAX(CASE WHEN BS.type = 'Q'

THEN BS.backup\_start\_date END)

AS LastPartDiff

,SUM(CASE WHEN BS.type = 'Q'

THEN 1 END)

AS CountPartDiff

FROM sys.databases AS DB

LEFT JOIN

msdb.dbo.backupset AS BS

ON BS.database\_name = DB.name

WHERE ISNULL(BS.is\_damaged, 0) = 0-- exclude damaged backups

GROUP BY DB.name

ORDER BY Last\_Backup desc;

--Backup History detail--------------------

SELECT

CONVERT(CHAR(100), SERVERPROPERTY('Servername')) AS Server,

msdb.dbo.backupset.database\_name,

msdb.dbo.backupset.backup\_start\_date,

msdb.dbo.backupset.backup\_finish\_date,

msdb.dbo.backupset.expiration\_date,

msdb..backupset.type ,

CASE msdb..backupset.type

WHEN 'D' THEN 'Database'

WHEN 'I' THEN 'DIFF'

WHEN 'L' THEN 'Log'

END AS backup\_type,

msdb.dbo.backupset.backup\_size,

msdb.dbo.backupmediafamily.logical\_device\_name,

msdb.dbo.backupmediafamily.physical\_device\_name,

msdb.dbo.backupset.name AS backupset\_name,

msdb.dbo.backupset.description

FROM msdb.dbo.backupmediafamily

INNER JOIN msdb.dbo.backupset ON msdb.dbo.backupmediafamily.media\_set\_id = msdb.dbo.backupset.media\_set\_id

WHERE CONVERT(varchar(10), msdb.dbo.backupset.backup\_start\_date,103) = CONVERT(varchar(10),GETDATE(),103)

ORDER BY

msdb.dbo.backupset.database\_name,

msdb.dbo.backupset.backup\_finish\_date

**6) Are there any errors in your SQL Server Error Log?**

In order to check the [SQL Server Error Log](https://www.mssqltips.com/sqlservertip/1476/reading-the-sql-server-log-files-using-tsql/) we are going to use the undocumented extended stored procedure, xp\_readerrorlog. This query will look at the current log and go back a maximum of 2 days looking for any errors during that time frame.

DECLARE @ArchiveID INT

,@Filter1Text NVARCHAR(4000)

,@Filter2Text NVARCHAR(4000)

,@FirstEntry SMALLDATETIME

,@LastEntry SMALLDATETIME

SELECT @ArchiveID = 0

,@Filter1Text = ''

,@Filter2Text = ''

-- this will only take the logs from the current day,

--you can change the date ranges to suit your needs

,@FirstEntry = DATEADD(DAY, - 1, getdate())

,@LastEntry = getdate()

CREATE TABLE #ErrorLog (

[date] [datetime] NULL

,[processinfo] [varchar](2000) NOT NULL

,[text] [varchar](2000) NULL

) ON [PRIMARY]

INSERT INTO #ErrorLog

EXEC master.dbo.xp\_readerrorlog @ArchiveID

,1

,@Filter1Text

,@Filter2Text

,@FirstEntry

,@LastEntry

,N'asc'

SELECT \*

FROM (

SELECT [date]

,[processinfo]

,[text] AS [MessageText]

,LAG([text], 1, '') OVER (

ORDER BY [date]

) AS [error]

FROM #ErrorLog

) AS ErrTable

WHERE [error] LIKE 'Error%'

-- you can change the text to filter above.

DROP TABLE #ErrorLog

**6) Check Database Size**

---MB----------------

SELECT sys.databases.name,

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

FROM sys.databases

JOIN sys.master\_files

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

GROUP BY sys.databases.name

ORDER BY sys.databases.name

---MB or GB----------------

SELECT sys.databases.name, CASE WHEN CONVERT(VARCHAR,((SUM(size)\*8)/1024)/1024) = 0

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

FROM sys.databases JOIN sys.master\_files

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

GROUP BY sys.databases.name ORDER BY sys.databases.name

----------------------

### 7) Background Process Query

SELECT s.session\_id

,r.STATUS

,r.blocking\_session\_id AS 'blocked\_by'

,r.wait\_type

,r.wait\_resource

,CONVERT(VARCHAR, DATEADD(ms, r.wait\_time, 0), 8) AS 'wait\_time'

,r.cpu\_time

,r.logical\_reads

,r.reads

,r.writes

,CONVERT(varchar, (r.total\_elapsed\_time/1000 / 86400))+ 'd ' +

CONVERT(VARCHAR, DATEADD(ms, r.total\_elapsed\_time, 0), 8) AS 'elapsed\_time'

,CAST((

'<?query -- ' + CHAR(13) + CHAR(13) + Substring(st.TEXT, (r.statement\_start\_offset / 2) + 1, (

(

CASE r.statement\_end\_offset

WHEN - 1

THEN Datalength(st.TEXT)

ELSE r.statement\_end\_offset

END - r.statement\_start\_offset

) / 2

) + 1) + CHAR(13) + CHAR(13) + '--?>'

) AS XML) AS 'query\_text'

,COALESCE(QUOTENAME(DB\_NAME(st.dbid)) + N'.' + QUOTENAME(OBJECT\_SCHEMA\_NAME(st.objectid, st.dbid)) + N'.' +

QUOTENAME(OBJECT\_NAME(st.objectid, st.dbid)), '') AS 'stored\_proc'

--,qp.query\_plan AS 'xml\_plan' -- uncomment (1) if you want to see plan

,r.command

,s.login\_name

,s.host\_name

,s.program\_name

,s.host\_process\_id

,s.last\_request\_end\_time

,s.login\_time

,r.open\_transaction\_count

FROM sys.dm\_exec\_sessions AS s

INNER JOIN sys.dm\_exec\_requests AS r ON r.session\_id = s.session\_id

CROSS APPLY sys.dm\_exec\_sql\_text(r.sql\_handle) AS st

--OUTER APPLY sys.dm\_exec\_query\_plan(r.plan\_handle) AS qp -- uncomment (2) if you want to see plan

WHERE r.wait\_type NOT LIKE 'SP\_SERVER\_DIAGNOSTICS%'

OR r.session\_id != @@SPID

ORDER BY r.cpu\_time DESC

,r.STATUS

,r.blocking\_session\_id

,s.session\_id

### 8) LogShipping Monitor

### -----------------------------

### EXEC sp\_help\_log\_shipping\_monitor

### -------------------------

DECLARE @LowRPOWarning INT = 5

DECLARE @MediumRPOWarning INT = 10

DECLARE @HighRPOWarning INT = 15

;WITH LastRestores AS

(

SELECT

[d].[name] [Database],

bmf.physical\_device\_name [LastFileRestored],

bs.backup\_start\_date LastFileRestoredCreatedTime,

r.restore\_date [DateRestored],

RowNum = ROW\_NUMBER() OVER (PARTITION BY d.Name ORDER BY r.[restore\_date] DESC)

FROM master.sys.databases d

INNER JOIN msdb.dbo.[restorehistory] r ON r.[destination\_database\_name] = d.Name

INNER JOIN msdb..backupset bs ON [r].[backup\_set\_id] = [bs].[backup\_set\_id]

INNER JOIN msdb..backupmediafamily bmf ON [bs].[media\_set\_id] = [bmf].[media\_set\_id]

)

SELECT

CASE WHEN DATEDIFF(MINUTE,LastFileRestoredCreatedTime,GETDATE()) > @HighRPOWarning THEN 'RPO High Warning!'

WHEN DATEDIFF(MINUTE,LastFileRestoredCreatedTime,GETDATE()) > @MediumRPOWarning THEN 'RPO Medium Warning!'

WHEN DATEDIFF(MINUTE,LastFileRestoredCreatedTime,GETDATE()) > @LowRPOWarning THEN 'RPO Low Warning!'

ELSE 'RPO Good'

END [Status],

[Database],

[LastFileRestored],

[LastFileRestoredCreatedTime],

[DateRestored]

FROM [LastRestores]

### WHERE [RowNum] = 1

### 9) Check DB Mirroring

### select \* from sys.database\_mirroring where mirroring\_guid is not null

### ----------------------------------------------------------------

SELECT

SERVERPROPERTY('ServerName') AS Principal,

m.mirroring\_partner\_instance AS Mirror,

DB\_NAME(m.database\_id) AS DatabaseName,

SUM(f.size\*8/1024) AS DatabaseSize,

CASE m.mirroring\_safety\_level

WHEN 1 THEN 'HIGH PERFORMANCE'

WHEN 2 THEN 'HIGH SAFETY'

END AS 'OperatingMode',

RIGHT(m.mirroring\_partner\_name, CHARINDEX( ':', REVERSE(m.mirroring\_partner\_name) + ':' ) - 1 ) AS Port

FROM sys.database\_mirroring m

JOIN sys.master\_files f ON m.database\_id = f.database\_id

WHERE m.mirroring\_role\_desc = 'PRINCIPAL'

GROUP BY m.mirroring\_partner\_instance, m.database\_id, m.mirroring\_safety\_level, m.mirroring\_partner\_name

### 10) check the active connection

SELECT DB\_NAME(dbid) as "Database", COUNT(dbid) as "Number Of Open Connections",

loginame as "LoginName"

FROM sys.sysprocesses

WHERE dbid > 0

GROUP BY dbid, loginame