SQL Server Basic Report

<https://www.ourtechideas.com/blog/server-information/>

The below listed all SQL Server Information can be retrieved by a TSQL script.

But before that, we need to Enable xp\_cmdshell with sp\_configure.



**Turns on advanced options and enable configure xp\_cmdshell using below TSQL query.**

-- this turns on advanced options and is needed to configure xp\_cmdshell

sp\_configure 'show advanced options', '1'

RECONFIGURE

GO

-- this enables xp\_cmdshell

sp\_configure 'xp\_cmdshell', '1'

RECONFIGURE

GO

**Run the below TSQL Script and get the SQL Server Information.**

Use master

go

SET NOCOUNT ON;

DECLARE @SQLServerStartupMode [int]

,@SQLAgentStartupMode [int]

,@LoadID [int]

,@Position [int]

,@LoginMode [int]

,@SQLServerAuditLevel [int]

,@SQLServerStartupType [char] (12)

,@SQLAgentStartupType [char] (12)

,@SQLServerServiceAccount [varchar] (64)

,@SQLAgentServiceAccount [varchar] (64)

,@SQLServerRegistryKeyPath [varchar] (256)

,@SQLAgentRegistryKeyPath [varchar] (256)

,@InstanceName [nvarchar] (128)

,@FullInstanceName [nvarchar] (128)

,@SystemInstanceName [nvarchar] (128)

,@ErrorLogDirectory [nvarchar] (128)

,@Domain [nvarchar] (64)

,@IPLine [nvarchar] (256)

,@IpAddress [nvarchar] (16)

,@ActiveNode [nvarchar] (128)

,@AuthenticationMode [varchar] (64)

,@PortNumber [varchar] (8)

,@PageFile [varchar] (124)

,@ClusterNodes [nvarchar] (32)

,@BinariesPath [nvarchar] (128)

,@RegistryKeyPath [nvarchar] (256)

,@RegistryPath1 [nvarchar] (256)

,@RegistryPath2 [nvarchar] (256)

,@RegistryPath3 [nvarchar] (256)

,@SQLServerInstallationLocation [nvarchar] (512)

IF OBJECT\_ID('[Tempdb].[dbo].[#\_IPCONFIG\_OUTPUT]') IS NOT NULL

DROP TABLE [dbo].[#\_IPCONFIG\_OUTPUT]

IF OBJECT\_ID('[Tempdb].[dbo].[#\_PAGE\_FILE\_DETAILS]') IS NOT NULL

DROP TABLE [dbo].[#\_PAGE\_FILE\_DETAILS]

IF OBJECT\_ID('[Tempdb].[dbo].[#\_XPMSVER]') IS NOT NULL

DROP TABLE [dbo].[#\_XPMSVER]

IF EXISTS (

SELECT \*

FROM [tempdb].[sys].[objects]

WHERE [name] = '##\_SERVER\_CONFIG\_INFO'

AND [type] IN (N'U')

)

DROP TABLE [dbo].[##\_SERVER\_CONFIG\_INFO]

CREATE TABLE [dbo].[#\_PAGE\_FILE\_DETAILS] ([data] [varchar](500)

)

CREATE TABLE [dbo].[#\_IPCONFIG\_OUTPUT] ([IPConfigCommandOutput] [nvarchar](256)

)

CREATE TABLE [dbo].[#\_XPMSVER] (

[IDX] [int] NULL

,[C\_NAME] [varchar](100) NULL

,[INT\_VALUE] [float] NULL

,[C\_VALUE] [varchar](128) NULL

) ON [PRIMARY]

CREATE TABLE [dbo].[##\_SERVER\_CONFIG\_INFO] (

[Domain] [nvarchar](64) NULL

,[SQLServerName] [varchar](64) NULL

,[InstanceName] [nvarchar](128) NULL

,[ComputerNamePhysicalNetBIOS] [nvarchar](128) NULL

,[IsClustered] [varchar](13) NULL

,[ClusterNodes] [nvarchar](32) NULL

,[ActiveNode] [nvarchar](128) NULL

,[HostIPAddress] [nvarchar](16) NULL

,[PortNumber] [varchar](8) NULL

,[IsIntegratedSecurityOnly] [varchar](64) NULL

,[AuditLevel] [varchar](38) NOT NULL

,[ProductVersion] [varchar](100) NULL

,[ProductLevel] [varchar](100) NULL

,[ResourceVersion] [varchar](100) NULL

,[ResourceLastUpdateDateTime] [varchar](100) NOT NULL

,[EngineEdition] [varchar](64) NULL

,[BuildClrVersion] [varchar](100) NOT NULL

,[Collation] [varchar](100) NULL

,[CollationID] [varchar](100) NULL

,[ComparisonStyle] [varchar](100) NULL

,[IsFullTextInstalled] [varchar](26) NULL

,[SQLCharset] [varchar](100) NOT NULL

,[SQLCharsetName] [varchar](100) NOT NULL

,[SQLSortOrderID] [varchar](100) NOT NULL

,[SQLSortOrderName] [varchar](100) NOT NULL

,[Platform] [varchar](128) NULL

,[FileDescription] [varchar](128) NULL

,[WindowsVersion] [varchar](128) NULL

,[ProcessorCount] [float] NULL

,[ProcessorType] [varchar](128) NULL

,[PhysicalMemory] [float] NULL

,[ServerPageFile] [varchar](124) NULL

,[SQLInstallationLocation] [nvarchar](512) NULL

,[BinariesPath] [nvarchar](128) NULL

,[ErrorLogsLocation] [nvarchar](128) NULL

,[MSSQLServerServiceStartupUser] [varchar](64) NULL

,[MSSQLAgentServiceStartupUser] [varchar](64) NULL

,[MSSQLServerServiceStartupType] [char](12) NULL

,[MSSQLAgentServiceStartupType] [char](12) NULL

,[InstanceLastStartDate] [datetime] NULL

,[LoadID] [int]

) ON [PRIMARY]

------ Finding SQL Server and Agent Service Account Information ------

IF SERVERPROPERTY('InstanceName') IS NULL -- Default Instance

BEGIN --default instance

SET @SQLServerRegistryKeyPath =

'SYSTEM\CurrentControlSET\SERVICES\MSSQLSERVER'

SET @SQLAgentRegistryKeyPath =

'SYSTEM\CurrentControlSET\SERVICES\SQLSERVERAGENT'

END

ELSE

BEGIN --Named Instance

SET @SQLServerRegistryKeyPath =

'SYSTEM\CurrentControlSET\SERVICES\MSSQL$' + CAST(

SERVERPROPERTY('InstanceName') AS [sysname])

SET @SQLAgentRegistryKeyPath =

'SYSTEM\CurrentControlSET\SERVICES\SQLAgent$' + CAST(

SERVERPROPERTY('InstanceName') AS [sysname])

END

EXEC [master]..[xp\_regread] 'HKEY\_LOCAL\_MACHINE'

,@SQLServerRegistryKeyPath

,@value\_name = 'Start'

,@value = @SQLServerStartupMode OUTPUT

EXEC [master]..[xp\_regread] 'HKEY\_LOCAL\_MACHINE'

,@SQLAgentRegistryKeyPath

,@value\_name = 'Start'

,@value = @SQLAgentStartupMode OUTPUT

SET @SQLServerStartupType = (

SELECT 'Start Up Mode' = CASE

WHEN @SQLServerStartupMode = 2

THEN 'Automatic'

WHEN @SQLServerStartupMode = 3

THEN 'Manual'

WHEN @SQLServerStartupMode = 4

THEN 'Disabled'

END

)

SET @SQLAgentStartupType = (

SELECT 'Start Up Mode' = CASE

WHEN @SQLAgentStartupMode = 2

THEN 'Automatic'

WHEN @SQLAgentStartupMode = 3

THEN 'Manual'

WHEN @SQLAgentStartupMode = 4

THEN 'Disabled'

END

)

EXEC [master]..[xp\_regread] 'HKEY\_LOCAL\_MACHINE'

,@SQLServerRegistryKeyPath

,@value\_name = 'ObjectName'

,@value = @SQLServerServiceAccount OUTPUT

EXEC [master]..[xp\_regread] 'HKEY\_LOCAL\_MACHINE'

,@SQLAgentRegistryKeyPath

,@value\_name = 'ObjectName'

,@value = @SQLAgentServiceAccount OUTPUT

SET @InstanceName = COALESCE(CONVERT([nvarchar](100),

SERVERPROPERTY('InstanceName')), 'MSSQLSERVER');

IF @InstanceName != 'MSSQLSERVER'

BEGIN

SET @InstanceName = @InstanceName

END

SET @FullInstanceName = COALESCE(CONVERT([nvarchar](100),

SERVERPROPERTY('InstanceName')), 'MSSQLSERVER');

IF @FullInstanceName != 'MSSQLSERVER'

BEGIN

SET @FullInstanceName = 'MSSQL$' + @FullInstanceName

END

EXEC [master]..[xp\_regread] N'HKEY\_LOCAL\_MACHINE'

,

N'Software\Microsoft\Microsoft SQL Server\Instance Names\SQL'

,@InstanceName

,@SystemInstanceName OUTPUT;

SET @RegistryKeyPath = N'SYSTEM\CurrentControlSET\Services\' +

@FullInstanceName;

SET @RegistryPath1 =

N'Software\Microsoft\Microsoft SQL Server\' +

@SystemInstanceName + '\MSSQLServer\Parameters';

SET @RegistryPath2 =

N'Software\Microsoft\Microsoft SQL Server\' +

@SystemInstanceName +

'\MSSQLServer\supersocketnetlib\TCP\IP1';

SET @RegistryPath3 =

N'SYSTEM\ControlSET001\Services\Tcpip\Parameters\';

IF @RegistryPath1 IS NULL

BEGIN

SET @InstanceName = COALESCE(CONVERT([nvarchar](100),

SERVERPROPERTY('InstanceName')),

'MSSQLSERVER');

END

EXEC [master]..[xp\_regread] N'HKEY\_LOCAL\_MACHINE'

,

N'Software\Microsoft\Microsoft SQL Server\Instance Names\SQL'

,@InstanceName

,@SystemInstanceName OUTPUT;

EXEC [master]..[xp\_regread] N'HKEY\_LOCAL\_MACHINE'

,@RegistryKeyPath

,@value\_name = 'ImagePath'

,@value = @BinariesPath OUTPUT

EXEC [master]..[xp\_regread] N'HKEY\_LOCAL\_MACHINE'

,@RegistryPath1

,@value\_name = 'SQLArg1'

,@value = @ErrorLogDirectory OUTPUT

EXEC [master]..[xp\_regread] N'HKEY\_LOCAL\_MACHINE'

,@RegistryPath3

,@value\_name = 'Domain'

,@value = @Domain OUTPUT

SELECT @ClusterNodes = COALESCE(@ClusterNodes + ', ', '') +

[Nodename]

FROM [sys].[dm\_os\_cluster\_nodes]

IF @ClusterNodes IS NULL

BEGIN

SET @ClusterNodes = 'Not Clustered'

END

SET @InstanceName = CONVERT([varchar](25), SERVERPROPERTY(

'InstanceName'))

EXEC [master]..[xp\_instance\_regread] N'HKEY\_LOCAL\_MACHINE'

,N'Software\Microsoft\MSSQLServer\MSSQLServer'

,N'AuditLevel'

,@SQLServerAuditLevel OUTPUT

EXEC [master]..[xp\_instance\_regread] N'HKEY\_LOCAL\_MACHINE'

,N'SOFTWARE\Microsoft\MSSQLServer\Setup'

,N'SQLPath'

,@SQLServerInstallationLocation OUTPUT

------ Finding IP Address ------

INSERT #\_IPCONFIG\_OUTPUT

EXEC [master]..[xp\_cmdshell] 'ipconfig'

IF LEFT(CAST(SERVERPROPERTY('ProductVersion') AS [sysname]), 5) =

'10.50'

BEGIN

SELECT @IPLine = [IPConfigCommandOutput]

FROM #\_IPCONFIG\_OUTPUT

WHERE UPPER([IPConfigCommandOutput]) LIKE '%IPv4 Address%'

IF (ISNULL(@IPLine, '\*\*\*') != '\*\*\*')

BEGIN

SET @Position = CharIndex(':', @IPLine, 1);

SET @IPAddress = RTRIM(LTRIM(SUBSTRING(@IPLine,

@Position + 1, LEN(@IPLine) - @Position))

)

END

END

ELSE

BEGIN

SELECT @IPLine = [IPConfigCommandOutput]

FROM #\_IPCONFIG\_OUTPUT

WHERE UPPER([IPConfigCommandOutput]) LIKE '%IP Address%'

IF (ISNULL(@IPLine, '\*\*\*') != '\*\*\*')

BEGIN

SET @Position = CharIndex(':', @IPLine, 1);

SET @IPAddress = RTRIM(LTRIM(SUBSTRING(@IPLine,

@Position + 1, LEN(@IPLine) - @Position))

)

END

END

------ Finding Port Information ------

IF @InstanceName IS NULL

BEGIN

SET @RegistryKeyPath =

'Software\Microsoft\MSSQLServer\MSSQLServer\SuperSocketNetLib\Tcp\'

END

ELSE

BEGIN

SET @RegistryKeyPath =

'Software\Microsoft\Microsoft SQL Server\' +

@InstanceName + '\MSSQLServer\SuperSocketNetLib\Tcp\'

END

EXEC [master]..[xp\_regread] 'HKEY\_LOCAL\_MACHINE'

,@RegistryKeyPath

,@value\_name = 'tcpPort'

,@value = @PortNumber OUTPUT -- Port Number

------ Finding Authentication Mode ------

EXEC [master]..[xp\_instance\_regread] N'HKEY\_LOCAL\_MACHINE'

,N'Software\Microsoft\MSSQLServer\MSSQLServer'

,@value\_name = N'LoginMode'

,@value = @LoginMode OUTPUT

SET @AuthenticationMode = (

SELECT 'AuTHENtication Mode' = CASE

WHEN @LoginMode = 1

THEN 'Windows Authentication'

WHEN @LoginMode = 2

THEN 'Mixed Mode Authentication'

END

)

------ Finding Active Node ------

EXEC [master]..[xp\_regread] @rootkey = 'HKEY\_LOCAL\_MACHINE'

,@RegistryKeyPath =

'SYSTEM\CurrentControlSet\Control\ComputerName\ComputerName'

,@value\_name = 'ComputerName'

,@value = @ActiveNode OUTPUT

INSERT INTO [#\_PAGE\_FILE\_DETAILS]

EXEC [master]..[xp\_cmdshell] 'wmic pagefile list /format:list'

SELECT @PageFile = RTRIM(LTRIM([data]))

FROM #\_PAGE\_FILE\_DETAILS

WHERE [data] LIKE 'AllocatedBaseSize%'

INSERT INTO [#\_XPMSVER]

EXEC ('master.dbo.xp\_msver')

----------

CREATE TABLE #Server\_Info\_SST (

id INT IDENTITY(1, 1)

,Field VARCHAR(50)

,Detaila VARCHAR(MAX)

)

INSERT INTO #Server\_Info\_SST (

Field

,Detaila

)

VALUES (

'Domain'

,UPPER(@Domain)

)

,(

'SQLServerName'

,CONVERT([varchar](64), SERVERPROPERTY('ServerName'))

)

,(

'InstanceName'

,@FullInstanceName

)

,(

'ComputerNamePhysicalNetBIOS'

,@ActiveNode

)

,(

'IsClustered'

,(

CASE

WHEN CONVERT([varchar](100), SERVERPROPERTY(

'IsClustered')) = 1

THEN 'Clustered'

WHEN SERVERPROPERTY('IsClustered') = 0

THEN 'Not Clustered'

WHEN SERVERPROPERTY('IsClustered') = NULL

THEN 'Error'

END

)

)

,(

'ClusterNodes'

,@ClusterNodes

)

,(

'ActiveNode'

,@ActiveNode

)

,(

'HostIPAddress'

,@IPAddress

)

,(

'PortNumber'

,@PortNumber

)

,(

'IsIntegratedSecurityOnly'

,@AuthenticationMode

)

,(

'AuditLevel'

,(

CASE

WHEN @SQLServerAuditLevel = 0

THEN 'None.'

WHEN @SQLServerAuditLevel = 1

THEN 'Successful Logins Only'

WHEN @SQLServerAuditLevel = 2

THEN 'Failed Logins Only'

WHEN @SQLServerAuditLevel = 3

THEN 'Both Failed and Successful Logins Only'

ELSE 'N/A'

END

)

)

,(

'ProductVersion'

,CONVERT([varchar](100), SERVERPROPERTY('ProductVersion')

)

)

,(

'ProductLevel'

,CONVERT([varchar](100), SERVERPROPERTY('ProductLevel'))

)

,(

'ResourceVersion'

,ISNULL(CONVERT([varchar](100), SERVERPROPERTY(

'ResourceVersion')), CONVERT([varchar](100),

SERVERPROPERTY('ProductVersion')))

)

,(

'ResourceLastUpdateDateTime'

,ISNULL(CONVERT([varchar](100), SERVERPROPERTY(

'ResourceLastUpdateDateTime')),

'Information Not Available')

)

,(

'EngineEdition'

,CAST(SERVERPROPERTY('Edition') AS [varchar](64))

)

,(

'BuildClrVersion'

,ISNULL(CONVERT([varchar](100), SERVERPROPERTY(

'BuildClrVersion')), 'NOT Applicable')

)

,(

'Collation'

,CONVERT([varchar](100), SERVERPROPERTY('Collation'))

)

,(

'CollationID'

,CONVERT([varchar](100), SERVERPROPERTY('CollationID'))

)

,(

'ComparisonStyle'

,CONVERT([varchar](100), SERVERPROPERTY('ComparisonStyle'

))

)

,(

'IsFullTextInstalled'

,(

CASE

WHEN CONVERT([varchar](100), SERVERPROPERTY(

'IsFullTextInstalled')) = 1

THEN 'Full-text is installed'

WHEN SERVERPROPERTY('IsFullTextInstalled') = 0

THEN 'Full-text is not installed'

WHEN SERVERPROPERTY('IsFullTextInstalled') = NULL

THEN 'Error'

END

)

)

,(

'SQLCharset'

,ISNULL(CONVERT([varchar](100), SERVERPROPERTY(

'SqlCharSet')), 'No Information')

)

,(

'SQLCharsetName'

,ISNULL(CONVERT([varchar](100), SERVERPROPERTY(

'SqlCharSetName')), 'No Information')

)

,(

'SQLSortOrderID'

,ISNULL(CONVERT([varchar](100), SERVERPROPERTY(

'SqlSortOrder')), 'No Information')

)

,(

'SQLSortOrderName'

,ISNULL(CONVERT([varchar](100), SERVERPROPERTY(

'SqlSortOrderName')), 'No Information')

)

,(

'Platform'

,(

SELECT C\_VALUE

FROM [#\_XPMSVER]

WHERE [C\_NAME] = 'Platform'

)

)

,(

'FileDescription'

,(

SELECT C\_VALUE

FROM [#\_XPMSVER]

WHERE [C\_NAME] = 'FileDescription'

)

)

,(

'WindowsVersion'

,CAST((

SELECT C\_VALUE

FROM [#\_XPMSVER]

WHERE [C\_NAME] = 'WindowsVersion'

) AS VARCHAR(50))

)

,(

'ProcessorCount'

,CAST((

SELECT INT\_VALUE

FROM [#\_XPMSVER]

WHERE [C\_NAME] = 'ProcessorCount'

) AS VARCHAR(50))

)

,(

'ProcessorType'

,(

SELECT ISNULL(C\_VALUE, CAST(INT\_VALUE AS VARCHAR(9)))

FROM #\_XPMSVER

WHERE [C\_NAME] = 'ProcessorType'

)

)

,(

'PhysicalMemory'

,CAST((

SELECT INT\_VALUE

FROM [#\_XPMSVER]

WHERE [C\_NAME] = 'PhysicalMemory'

) AS VARCHAR(50))

)

,(

'ServerPageFile'

,@PageFile

)

,(

'SQLInstallationLocation'

,@SQLServerInstallationLocation

)

,(

'BinariesPath'

,@BinariesPath

)

,(

'ErrorLogsLocation'

,@ErrorLogDirectory

)

,(

'MSSQLServerServiceStartupUser'

,@SQLServerServiceAccount

)

,(

'MSSQLAgentServiceStartupUser'

,@SQLAgentServiceAccount

)

,(

'MSSQLServerServiceStartupType'

,@SQLServerStartupType

)

,(

'MSSQLAgentServiceStartupType'

,@SQLAgentStartupType

)

,(

'[InstanceLastStartDate]'

,CONVERT(VARCHAR(20), (

CONVERT(DATETIME, (

SELECT [login\_time]

FROM [master]..[sysprocesses]

WHERE [spid] = 1

))

))

) --INTO SQLServerInfo

SELECT \*

FROM #Server\_Info\_SST

-- Dropping temporary table

IF OBJECT\_ID('[Tempdb].[dbo].[#\_IPCONFIG\_OUTPUT]') IS NOT NULL

DROP TABLE [dbo].[#\_IPCONFIG\_OUTPUT]

IF OBJECT\_ID('[Tempdb].[dbo].[#\_PAGE\_FILE\_DETAILS]') IS NOT NULL

DROP TABLE [dbo].[#\_PAGE\_FILE\_DETAILS]

IF OBJECT\_ID('[Tempdb].[dbo].[#\_XPMSVER]') IS NOT NULL

DROP TABLE [dbo].[#\_XPMSVER]

DROP TABLE #Server\_Info\_SST

GO