SQLDBA Core Håndbog

(Mikael Veistrup-Vetlov)

Indhold

[System Opsætning 4](#_Toc76548703)

[Activating the Server Core System 4](#_Toc76548704)

[Server Configurator 4](#_Toc76548705)

[Configure Local Security Policy on Windows Server Core 4](#_Toc76548706)

[Adding the Server Core System to a Domain 6](#_Toc76548707)

[Netdom join <computername> /domain:<domain> /OU:<ou path> /userd:<domain>\<username> /passwordD:\* 6](#_Toc76548708)

[PowerShell kommandoer 6](#_Toc76548709)

[Diskopsætning: 6](#_Toc76548710)

[Network Shares 6](#_Toc76548711)

[VMI option (v1) 6](#_Toc76548712)

[SmbShare (v2) # kræver PS v3 7](#_Toc76548713)

[Lokal Gruppe Håndtering 7](#_Toc76548714)

[Get-Acl and Set-Acl Folder Permissions 8](#_Toc76548715)

[ACL in Registry 8](#_Toc76548716)

[ACL ver 2: NTFSSecurity 9](#_Toc76548717)

[ACL Praktisk: 10](#_Toc76548718)

[Sql from PowerShell 11](#_Toc76548719)

[SQLSERVER: Invoke-Sqlcmd 11](#_Toc76548720)

[SMO 11](#_Toc76548721)

[Legacy 11](#_Toc76548722)

[NuGet provider is required to continue 12](#_Toc76548723)

[SQLCMD 12](#_Toc76548724)

[invoke-sqlcmd 12](#_Toc76548725)

[Comparing Invoke-Sqlcmd and the sqlcmd Utility 13](#_Toc76548726)

[cd sqlserver: 16](#_Toc76548727)

[System 16](#_Toc76548728)

[Find FSMO roller 16](#_Toc76548729)

[GPO update force 16](#_Toc76548730)

[WMI 17](#_Toc76548731)

[Get Windows version: 17](#_Toc76548732)

[Cores & logical processors 17](#_Toc76548733)

[Pc name, domain & memory 17](#_Toc76548734)

[Communication to Domain Controllers 19](#_Toc76548735)

[Andre kommandoer 19](#_Toc76548736)

[Virker på Core Server 20](#_Toc76548737)

[Virker ikke på Core Server 21](#_Toc76548738)

[Opdater GPO på maskine efter GPO ændring. 28](#_Toc76548739)

[Vis Ip info alle kort 28](#_Toc76548740)

[Find logged on & active users on server 28](#_Toc76548741)

[# connect remote 28](#_Toc76548742)

[# Import-module 28](#_Toc76548743)

[# Se installerede features 28](#_Toc76548744)

[# Tilføje feature 29](#_Toc76548745)

[# Se updates 29](#_Toc76548746)

[# Rename-computer 29](#_Toc76548747)

[# Genstart 29](#_Toc76548748)

[# Luk computer 29](#_Toc76548749)

[# Se IP adresse -remote 29](#_Toc76548750)

[# Sæt ny IP 29](#_Toc76548751)

[# Sætte dns-server 29](#_Toc76548752)

[# Tilføje computer til domain 29](#_Toc76548753)

[# Se netværksstatistik 29](#_Toc76548754)

[# Test netværk (kombination af ping og nslookup) 30](#_Toc76548755)

[# Port check 30](#_Toc76548756)

[# Genskabe forbindelsen til domainet for klient UDEN at skulle melde den ind og ud af domainet 30](#_Toc76548757)

[Test-ComputerSecureChannel -Credential dksund\adm-XXX -repair 30](#_Toc76548758)

[# Genskabe domain forbindelse på klient remote 30](#_Toc76548759)

[# Se Error logs remote 30](#_Toc76548760)

[# Services 30](#_Toc76548761)

[Remote SQL Server Configuration Management (From Gui machine with SQL) 30](#_Toc76548762)

[Windows 2019 core AppCompatibility (.mmc på core) 31](#_Toc76548763)

[# SQL Services 31](#_Toc76548764)

[# Se services, som er stopped 32](#_Toc76548765)

[#Services på remote pc 32](#_Toc76548766)

[# Firewall 32](#_Toc76548767)

[# Sætte remote managemnet via GUI 33](#_Toc76548768)

[# Active Directory 33](#_Toc76548769)

[# Sætte nyt password på user-account 33](#_Toc76548770)

[# Oprette AD-gruppe 33](#_Toc76548771)

[# ISE Snippets 34](#_Toc76548772)

[# DNS 34](#_Toc76548773)

[# Tilføje ny dns zone 34](#_Toc76548774)

[# Tilføje host-record 34](#_Toc76548775)

[# File shares 34](#_Toc76548776)

[# Nyt File share 34](#_Toc76548777)

[# remote desktop permissions 34](#_Toc76548778)

# System Opsætning

## Activating the Server Core System

Activating a Server Core system can be achieved in two easy steps. First, you enter a product key, and then you activate the server. The syntax to enter a product key is as follows:

slmgr.vbs –ipk<productkey>

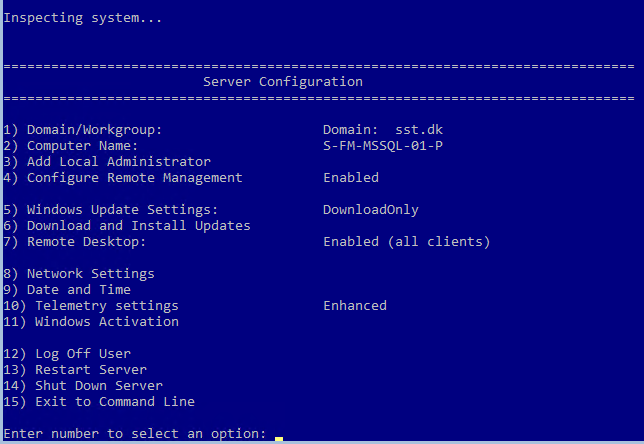
After the product key has been successfully entered, you activate the server by typing in the following command.

slmgr.vbs –ato

<eller> via Sconfig (11)

## Server Configurator

Sconfig



# Configure Local Security Policy on Windows Server Core

Setting the account policy and local security on a Windows Server Core system, you must first create a security template on a full Windows Server installation and subsequently apply these settings to the Windows Server Core system:

On the reference server (ie Windows Server full installation)

1. From the Start menu, enter secpol.msc in the Start Search box and hit Enter to launch the Local Security Policy snap-in on another system.

2. Configure the security policies according to your needs, then right-click the Security Settings and click Export policy to save this as a security template.

You can use an editor to remove the unwanted text (leave the headings) and save as a .inf file

NB! Leave text after [Version]

[Unicode]

Unicode=yes

[System Access]

[Event Audit]

[Registry Values]

[Privilege Rights]

SeBatchLogonRight = \*S-1-5-21-2812754299-3050866767-1751104339-1287,\*S-1-5-32-559,\*S-1-5-32-551,\*S-1-5-32-544

[Version]

signature="$CHICAGO$"

Revision=1

On the Server Core server

1.Copy the newly created security template from the reference server to the Server Core system. (must be on local disc not mapped drive)

2.Run the below command to apply the security policy to the server Core system:

secedit /configure /cfg <Policy File Name> /db secedit.sdb

3.Check the result:

notepad %windir%\security\logs\scesrv.log

notepad c:\windows\security\logs\scesrv.log

Result like:

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

9. oktober 2017 16:47:45

----Configuration engine was initialized successfully.----

----Reading Configuration Template info...

----Configure User Rights...

Configure S-1-5-21-2812754299-3050866767-1751104339-1287.

add SeBatchLogonRight.

Configure S-1-5-32-559.

Configure S-1-5-32-551.

Configure S-1-5-32-544.

User Rights configuration was completed successfully.

----Configure Group Membership...

Group Membership configuration was completed successfully.

----Configure 64-bit Registry Keys...

Configuration of Registry Keys was completed successfully.

----Configure 32-bit Registry Keys...

----Configure File Security...

File Security configuration was completed successfully.

----Configure General Service Settings...

General Service configuration was completed successfully.

----Configure available attachment engines...

Configuration of attachment engines was completed successfully.

----Configure Security Policy...

Configure password information.

System Access configuration was completed successfully.

Configuration of Registry Values was completed successfully.

Audit/Log configuration was completed successfully.

----Configure available attachment engines...

Configuration of attachment engines was completed successfully.

----Un-initialize configuration engine...

scite c:\windows\security\logs\scesrv.log

copy P:\SQLDBA\ps1\2016\s-mssql02-p\secpol2.inf

secedit /configure /cfg secpol2.inf /db secedit.sdb

shutdown -r -t 10

## Adding the Server Core System to a Domain

## Netdom join <computername> /domain:<domain> /OU:<ou path> /userd:<domain>\<username> /passwordD:\*

<eller>

Add-Computer -domainname <domain> -OUPath "OU=OU,=Domain,DC=com"

# PowerShell kommandoer

## Diskopsætning:

Get-volume viser hvilke diske og drev, der er defineret / tilknyttet

Get-WmiObject -Class Win32\_volume -Filter "DriveLetter = 'z:'" viser opsætningen af en disk.

$rc=Get-WmiObject -Class Win32\_volume -Filter "DriveLetter = 'r:'"

$rc.DriveType viser drevtype, 5=cd-rom, 3=harddisk

Ændring af drevbogstav: Finder objektet med DriveLetter = z: og giver nyt drevbogstav

Set-WmiInstance -InputObject ( Get-WmiObject -Class Win32\_volume -Filter "DriveLetter = 'z:'" ) -Arguments @{

DriveLetter='d:'}

## Network Shares

### VMI option (v1)

IF (!(TEST-PATH C:\NewShare)) { # if folder NOT exist

                NEW-ITEM C:\NewShare -type Directory # Create folder ( -force hvis flere niveauer & opret.)

}

$Shares=[WMICLASS]”WIN32\_Share” # Use wmi class WIN32\_Share

$shares|get-member # check which methods & properties of vmi-class

$Shares.Create(“<name of folder>”,”<share name>”,<share type>) #

#Share types:

#Value Meaning

#0 (0x0) Disk Drive

#1 (0x1) Print Queue

#2 (0x2) Device

#3 (0x3) IPC

#2147483648 (0x80000000) Disk Drive Admin

#2147483649 (0x80000001) Print Queue Admin

#2147483650 (0x80000002) Device Admin

#2147483651 (0x80000003) IPC Admin

If (!(GET-WMIOBJECT Win32\_Share -filter “name=’ MiveTemp’”) { # Check if share exist

$Shares.Create(“U:\MiveTemp”,” MiveTemp”,0) # Create disk share MiveTemp on U:\MiveTemp

}

# remove share:

Get-WmiObject -Class Win32\_Share -ComputerName . -Filter "Name='TempShare'").InvokeMethod("Delete",$null)''

net SHARE share=d:\share /GRANT:EVERYONE`,FULL /REMARK:" # Note ` before ,FULL

### SmbShare (v2) # kræver PS v3

NEW-ITEM <sharepath> -type Directory #Opretter folder Mvvshare på drev u:

Get-SmbShare #lister navn, scopename, path , Description (alle shares)

New-SMBShare -name <share> -path <sharepath> -FullAccess <group> -Change sst.dk\everyone –Confirm #Opretter Share med privilegier format: domain\gruppe

Get-SmbShareAccess <share> #rettighedsvisning til Mvvshare

Revoke-SmbShareAccess <share> -AccountName everyone #Fjerner rettighed på share for user

Grant-SmbShareAccess <share> -AccountName <group> -AccessRight Full -Force #Full, Change, Read, Custom

Remove-SmbShare < share> #Fjerner share

Get-SmbServerConfiguration #?

Set-SmbServerConfiguration #?

New-SMBShare -name Mvvshare -path u:\MvvShare -FullAccess sst.dk\adm\_stgn -Change sst.dk\svd\_sqle\_fskmsql -Force

Grant-SmbShareAccess mivetemp -AccountName sst.dk\svd\_sqlefmsq21p$ change -Force

## Lokal Gruppe Håndtering

Get-LocalGroupMember -Group Administrators

Add-LocalGroupMember -Group Administrators -Member [mive@dksund.dk](mailto:mive@dksund.dk)

Remove-LocalGroupMember -Group Administrators -Member mive@dksund.dk

## Get-Acl and Set-Acl Folder Permissions

To quote the PowerShell documentation "Get-Acl gets the security descriptor for a resource, such as a file or registry key." while "Set-Acl changes the security descriptor of a specified resource, such as a file or a registry key." In other words; if you want Folder\_A to have the exact same permissions as Folder\_B, then you simply copy the Access Control List (ACL) of Folder\_B and "paste" it onto Folder\_A.

$Acl = Get-Acl "C:\Folder\_B"

Set-Acl "C:\Folder\_A" $Acl

So far, so good.

Changing the ACL

Okay, so you want to change the ACL. Here's some sample code for how to do that:

New-Item -type directory -path C:\MyFolder

$Acl = Get-Acl "C:\MyFolder"

$Ar = New-Object system.security.accesscontrol.filesystemaccessrule("username","FullControl","Allow")

$Acl.SetAccessRule($Ar)

Set-Acl "C:\MyFolder" $Acl

So, first we create a new folder. We then copy the ACL of that folder. We then create a new AccessRule that gives "username" full control. We then add this AccessRule to the ACL, and finally we reapply the new, altered ACL to the folder.

If we wanted to we could also have used $Acl.RemoveAccessRule($Ar) or possibly $Acl.RemoveAccessRuleAll() as well.

inheritance and propagation flags):

$TSPath = "C:tempa"

$Acl = Get-Acl $TSPath

$networkService = New-Object System.Security.Principal.SecurityIdentifier([System.Security.Principal.WellKnownSidType]::NetworkServiceSid, $null);

$rights = [System.Security.AccessControl.FileSystemRights]::FullControl

$inheritance = [int]([System.Security.AccessControl.InheritanceFlags]::ContainerInherit) + [int]([System.Security.AccessControl.InheritanceFlags]::ObjectInherit)

$propagation = [System.Security.AccessControl.PropagationFlags]::None

$accessControl = [System.Security.AccessControl.AccessControlType]::Allow

$AccessRule = New-Object System.Security.AccessControl.FileSystemAccessRule($networkService, $rights, $inheritance, $propagation , $accessControl)

$Acl.SetAccessRule($AccessRule)

Set-Acl $TSPath $Acl

## ACL in Registry

Granting / denying privileges on the registry is fairly easy with PowerShell. Here's a bit that adds read/write permissions for sources on a specific log to a specific principle:

# Set parameters:

param(

[string]$Principle = 'Builtin\Remote Desktop Users',

[string]$LogName = 'Application'

)

# Compose Key:

$LogPath = 'HKLM:\SYSTEM\CurrentControlSet\services\eventlog\'+$LogName;

if(Test-Path $LogPath)

{

$acl = Get-Acl $LogPath

$ace = New-Object System.Security.AccessControl.RegistryAccessRule $Principle,'WriteKey, ReadKey','allow'

$acl.AddAccessRule($ace)

#Set-Acl $LogPath $acl

}else{Write-Error "Cannot acesss log $LogName"}

## ACL ver 2: NTFSSecurity

P:\anjet2\ps1\2016\is-ntfs.ps1 Checker om NTFSSecurity skal installers, og gør det.

Herefter

Get-Item D:\Data | Get-NTFSAccess Læser acl for 1 fil/folder via pipe

Get-NTFSAccess -Path D:\Data Læser acl for 1 fil/folder via parm

If you want to display only permissions that have been added explicitly and hide all the inherited permissions, use the ExcludeInherited switch:

dir | Get-NTFSAccess –ExcludeInherited

If you want to display only the permissions assigned to a certain user, use the Account parameter:

dir | Get-NTFSAccess -Account raandree9\randr\_000

Add-NTFSAccess -Path C:\Data -Account 'NT AUTHORITY\Authenticated Users' -AccessRights Read

Add-NTFSAccess -Path C:\Data -Account 'BUILTIN\Administrators', 'raandree9\Editors' -AccessRights FullControl

Add-NTFSAccess .\Data -Account raandree1\install -AccessRights Modify -AppliesTo ThisFolderOnly

Get-ChildItem -Path d:\ -R |Get-NTFSAccess -Account raandree0\randr\_000 –ExcludeInherited | Remove-NTFSAccess

dir -R | Get-NTFSAccess -Account BUILTIN\Administrators -ExcludeExplicit | Where-Object InheritedFrom -eq 'D:\Data'

dir -Recurse | Get-NTFSOrphanedAccess | Remove-NTFSAccess remove orphaned access

## ACL Praktisk:

$us="ssisshare0"

$uf="u:\$us"

$u0=”DKSUND\L-ORG-MSSQL-Sysadmin”

$r1="HKLM:\SOFTWARE\Wow6432Node\Microsoft\WBEM\CIMOM"

$User=”dksund\mive”

If(!(Test-Path $uf)){NEW-ITEM –path $uf -type Directory} #Test om folder IKKE findes, og opret den

$rc=Get-Smbshare $us #Test om Share IKKE finds, og opret den

if($rc.name.count -eq 0){New-SMBShare -name $us -path $uf -FullAccess "$u0" –Confirm} Else{Write-Host “Findes allerede”}

Function Set-Ssispriv {

Param ($UserGrp)

Grant-SmbShareAccess $us -AccountName "$UserGrp" -AccessRight Full –Force # Giv share full acces til gruppe

$Acl = Get-Acl "$uf" # Giv Folder acl full acces til gruppe

$Ar = New-Object system.security.accesscontrol.filesystemaccessrule("$UserGrp","FullControl","Allow")

$Acl.SetAccessRule($Ar)

Set-Acl "$uf" $Acl

$Acl = Get-Acl $r1 # Giv Reg acl full acces til gruppe

$Ar = New-Object System.Security.AccessControl.RegistryAccessRule ("$UserGrp","FullControl","Allow")

$Acl.SetAccessRule($Ar)

$acl |Set-Acl –Path $r1

}

Set-Ssispriv $user

Get-Acl $uf|fl #Viser acl privilegier

Get-Acl $r1|fl

Function Slet-Ssispriv {

Revoke-SmbShareAccess $us -AccountName $User

Param ($UserGrp)

$Acl = Get-Acl "$uf"

$AcR = $Acl.Access | ?{ $\_.IsInherited -eq $false -and $\_.IdentityReference -eq $UserGrp }

$acl.RemoveAccessRuleAll($AcR)

Set-Acl "$uf" $Acl

$Acl = Get-Acl $r1

$AcR = $Acl.Access | ?{ $\_.IsInherited -eq $false -and $\_.IdentityReference -eq $UserGrp }

$acl.RemoveAccessRuleAll($AcR)

$acl |Set-Acl –Path $r1

}

Set-Ssispriv $user

# Sql from PowerShell

## SQLSERVER: Invoke-Sqlcmd

Fra powershell køres

Invoke-Sqlcmd # uden parametre for at klargøre modulerne.

Hvis dette fejler skal de først installeres, se nedenstående.

P:\anjet2\ps1\2016\is-sql.ps1 Checker om SQLserver skal installers, og gør det.

Så kan man på powershell vis få listet forlderstrukturen som Get-PSdrive

Sqlserver: # hopper til sqlserver: -drev

Get-Childitem # eller alias dir, ls eller lignende viser indhold af nuværende folder/struktur.

Cd sql # skifter til sql strukturen / folderen

$r=get-childitem; cd $r.PSChildName

Cd <hostnavn> # skifter til server strukturen / folderen

Cd <instansnavn> # skifter til Instans strukturen / folderen

Nu kan man se områder på sql instansen, som man har adgang til

Efter cd databases kan man liste databaser på serveren

Man kan også lave sqlkald

invoke-sqlcmd -query "Select Serverproperty('Productversion')" #viser sql version i tal

eller mere avancerede

Invoke-SQLCMD -ServerInstance "srv-freud-db01.sst.dk\Freud\_preprod" -Query "ALTER DATABASE <dbnavn> MODIFY …….. "

## SMO

### Legacy

There are two SQL Server PowerShell modules; SqlServer and SQLPS. The SQLPS module is included with the SQL Server installation (for backwards compatibility), but is no longer being updated. The most up-to-date PowerShell module is the SqlServer module. The SqlServer module contains updated versions of the cmdlets in SQLPS, and also includes new cmdlets to support the latest SQL features. Previous versions of the SqlServer module were included with SQL Server Management Studio (SSMS), but only with the 16.x versions of SSMS. To use PowerShell with SSMS 17.0 and later, the SqlServer module must be installed from the PowerShell Gallery.

To install the SqlServer module from the PowerShell Gallery, start a PowerShell session and use the following commands. If you run into problems installing, see the Install-Module documentation and Install-Module reference.

To install the SqlServer module: Run Powershell as administrator:

>Install-Module -Name SqlServer

If there are previous versions of the SqlServer module on the computer, you may be able to use Update-Module, or provide the -AllowClobber parameter:

>Install-Module -Name SqlServer -AllowClobber

If you are not able to run the PowerShell session as administrator, you can install for the current user:

>Install-Module -Name SqlServer -Scope CurrentUser

When updated versions of the SqlServer module are available, you can update the version using Update-Module:

>Update-Module -Name SqlServer

To view the versions of the module installed:

>Get-Module SqlServer -ListAvailable

To use a specific version of the module, you can import it with a specific version number similar to the following:

>Import-Module SqlServer -Version 21.0.17178

### NuGet provider is required to continue

Vælg Y for at installere

## SQLCMD

1>SELECT name FROM sys.databases

2>go

1>exit

SELECT DISTINCT

local\_tcp\_port

FROM sys.dm\_exec\_connections

WHERE local\_tcp\_port IS NOT NULL

## invoke-sqlcmd

Since all of the steps I required weren't listed in one place, I thought I'd list them here in case it helps anyone:

1.Install SQL Server 2008 R2 Management Objects using Web PI (I'm not sure about versions prior to 2008 R2... if you have more info, please let us know in the comments)

2.Install 'Windows PowerShell Extensions for SQL Server' from the Microsoft® SQL Server® 2008 R2 Feature Pack page (it's about halfway down the page). Make sure you pick the correct flavor for your instance (32 or 64 bit).

3.Run these two commands before calling invoke-sqlcmd in your script:

Add-PSSnapin SqlServerCmdletSnapin100

Add-PSSnapin SqlServerProviderSnapin100

UPDATE: for SQL Server 2012, they have switched to a single module. Run this instead:

Import-Module SqlPs

The Invoke-Sqlcmd cmdlet lets you run your sqlcmd script files in a Windows PowerShell environment. Much of what you can do with sqlcmd can also be done using Invoke-Sqlcmd.

This is an example of calling Invoke-Sqlcmd to execute a simple query, similar to specifying sqlcmd with the -Q and -S options:

Invoke-Sqlcmd -Query "SELECT GETDATE() AS TimeOfQuery;" -ServerInstance "MyComputer\MyInstance"

This is an example of calling Invoke-Sqlcmd, specifying an input file and piping the output to a file This is similar to specifying sqlcmd with the -i and -o options:

Invoke-Sqlcmd -InputFile "C:\MyFolder\TestSQLCmd.sql" | Out-File -filePath "C:\MyFolder\TestSQLCmd.rpt"

This is an example of using a Windows PowerShell array to pass multiple sqlcmd scripting variables to Invoke-Sqlcmd. The "$" characters identifying the sqlcmd scripting variables in the SELECT statement have been escaped by using the PowerShell back-tick "`" escape character:

$MyArray = "MyVar1 = 'String1'", "MyVar2 = 'String2'"

Invoke-Sqlcmd -Query "SELECT `$(MyVar1) AS Var1, `$(MyVar2) AS Var2;" -Variable $MyArray

This is an example of using the SQL Server provider for Windows PowerShell to navigate to an instance of the Database Engine, and then using the Windows PowerShell Get-Item cmdlet to retrieve the SMO Server object for the instance and passing it to Invoke-Sqlcmd:

Set-Location SQLSERVER:\SQL\MyComputer\MyInstance

Invoke-Sqlcmd -Query "SELECT GETDATE() AS TimeOfQuery;" -ServerInstance (Get-Item .)

The -Query parameter is positional and does not have to be named. If the first string that is passed to Invoke-Sqlcmd: is unnamed, it is treated as the -Query parameter.

Invoke-Sqlcmd "SELECT GETDATE() AS TimeOfQuery;" -ServerInstance "MyComputer\MyInstance"

For example, assume that the default database for your Windows account in the default instance of the local computer is master. Then, the following commands would return master:

Set-Location SQLSERVER:\SQL

Invoke-Sqlcmd "SELECT DB\_NAME() AS DatabaseName;"

The following commands would return AdventureWorks2008R2:

Set-Location SQLSERVER:\SQL\MyComputer\DEFAULT\Databases\AdventureWorks2008R2\Tables\Person.Person

Invoke-Sqlcmd "SELECT DB\_NAME() AS DatabaseName;"

Invoke-Sqlcmd provides a warning when it uses the path database context. You can use the -SuppressProviderContextWarning parameter to turn off the warning message. You can use the -IgnoreProviderContext parameter to tell Invoke-Sqlcmd to always use the default database for the login.

## Comparing Invoke-Sqlcmd and the sqlcmd Utility

Invoke-Sqlcmd can be used to run many of the scripts that can be run using the sqlcmd utility. However, Invoke-Sqlcmd runs in a Windows PowerShell environment which is different than the command prompt environment that sqlcmd is run in. The behavior of Invoke-Sqlcmd has been modified to work in a Windows PowerShell environment.

Not all of the sqlcmd commands are implemented in Invoke-Sqlcmd. Commands that are not implemented include the following: :!!, :connect, :error, :out, :ed, :list, :listvar, :reset, :perftrace, and :serverlist.

Invoke-Sqlcmd does not initialize the sqlcmd environment or scripting variables such as SQLCMDDBNAME or SQLCMDWORKSTATION.

Invoke-Sqlcmd does not display messages, such as the output of PRINT statements, unless you specify the Windows PowerShell -Verbose common parameter. For example:

Invoke-Sqlcmd -Query "PRINT N'abc';" -Verbose

invoke-sqlcmd -query "SELECT name FROM sys.databases"

Not all of the sqlcmd parameters are needed in a PowerShell environment. For example, Windows PowerShell formats all output from cmdlets, so the sqlcmd parameters specifying formatting options are not implemented in Invoke-Sqlcmd.

The following table shows the relationship between the Invoke-Sqlcmd parameters and sqlcmd options:

|  |  |  |
| --- | --- | --- |
| **Description** | **sqlcmd option** | **Invoke-Sqlcmd parameter** |
| Server and instance name. | -S | -ServerInstance |
| The initial database to use. | -d | -Database |
| Run the specified query and exit. | -Q | -Query |
| SQL Server Authentication login ID. | -U | -Username |
| SQL Server Authentication password. | -P | -Password |
| Variable definition. | -v | -Variable |
| Query timeout interval. | -t | -QueryTimeout |
| Stop running on an error | -b | -AbortOnError |
| Dedicated Administrator Connection. | -A | -DedicatedAdministratorConnection |
| Disable interactive commands, startup script, and environment variables. | -X | -DisableCommands |
| Disable variable substitution. | -x | -DisableVariables |
| Minimum severity level to report. | -V | -SeverityLevel |
| Minimum error level to report. | -m | -ErrorLevel |
| Login timeout interval. | -l | -ConnectionTimeout |
| Hostname. | -H | -HbostName |
| Change password and exit. | -Z | -NewPassword |
| Input file containing a query | -i | -InputFile |
| Maximum length of character output. | -w | -MaxCharLength |
| Maximum length of binary output. | -w | -MaxBinaryLength |
| Connect using SSL encryption. | No parameter | -EncryptConnection |
| Display errors | No parameter | -OutputSqlErrors |
| Output messages to stderr. | -r | No parameter |
| Use client's regional settings | -R | No parameter |
| Run the specified query and remain running. | -q | No parameter |
| Code page to use for output data. | -f | No parameter |
| Change a password and remain running | -z | No parameter |
| Packet size | -a | No parameter |
| Column separator | -s | No parameter |
| Control output headers | -h | No parameter |
| Specify control characters | -k | No parameter |
| Fixed length display width | -Y | No parameter |
| Variable length display width | -y | No parameter |
| Echo input | -e | No parameter |
| Enable quoted identifiers | -I | No parameter |
| Remove trailing spaces | -W | No parameter |
| List instances | -L | No parameter |
| Format output as Unicode | -u | No parameter |
| Print statistics | -p | No parameter |
| Command end | -c | No parameter |
| Connect using Windows Authentication | -E | No parameter |

## cd sqlserver:

Brug Get-Psdrive for at checke om sqlserver: drive er klargjort.

Brug Invoke-Sqlcmd hvis det ikke er klargjort.

SQLSERVER:\SQL\<ComputerName>\<InstanceName>\Databases\<DatabaseName> The specified database in the specified instance on the specified computer.

SELECT suser\_sname(owner\_sid), \* FROM sys.databases -- viser db owner

Set-Location "SQLSERVER:\SQL\(local)\DEFAULT"

Set-Location "SQLSERVER:\SQL\(local)\devl"

# System

## Find FSMO roller

Get-ADDomain | Select-Object InfrastructureMaster, RIDMaster, PDCEmulator

Get-ADForest | Select-Object DomainNamingMaster, SchemaMaster

#Eller

Get-ADDomainController -Filter \* |

     Select-Object Name, Domain, Forest, OperationMasterRoles |

     Where-Object {$\_.OperationMasterRoles} |

     Format-Table –AutoSize

#Eller

$a=Get-ADDomain

$b= Get-ADForest

Write-host “InfrastructureMaster:” $a.InfrastructureMaster

Write-host “RIDMaster:” $a.RIDMaster

Write-host “PDCEmulator:” $a.PDCEmulator

Write-host “DomainNamingMaster:” $b.DomainNamingMaster

Write-host “SchemaMaster:” $b.SchemaMaster

Write-host “GlobalCatalogs:”

Foreach($g in $b.GlobalCatalogs){write-Host $g}

Tilføj domainets PDC-emulator til kommandoen:

”Get-ADPrincipalGroupMembership adm\_mive -Server s-ad-dc-02p.dksund.dk”

## GPO update force

Hvis man har flyttet en server til rette OU I AD kan man på serveren i powershell skrive:

>gpupdate /force

Hvorefter den vil tilegne sig den nye OU opsætning.

# WMI

## Get Windows version:

gwmi win32\_operatingsystem | % caption

(Get-ItemProperty -Path c:\windows\system32\hal.dll).VersionInfo.FileVersion

[Environment]::OSVersion

(Get-CimInstance Win32\_OperatingSystem).Version

systeminfo /fo csv | ConvertFrom-Csv | select OS\*, System\*, Hotfix\* | Format-List

## Cores & logical processors

Get-WmiObject -class Win32\_processor | ft systemname,Name,DeviceID,NumberOfCores,NumberOfLogicalProces

sors, Addresswidth

udvidet:

Get-WmiObject –class Win32\_processor | select \*

## Pc name, domain & memory

Get-WmiObject -Class Win32\_ComputerSystem

get-ciminstance -class "cim\_physicalmemory" | % {$\_.Capacity}

Get-ADTrust -Filter \* (viser hvilke domæner der er domain trust mellem)

Restart pc

(Get-WmiObject -Class Win32\_OperatingSystem -ComputerName .).Win32Shutdown(2)

#Eller

Reststart-computer

Get information about the BIOS of the current computer:

Get-WmiObject -Class Win32\_BIOS -ComputerName .

Get information about the current server:

systeminfo.exe

List installed hotfixes -- QFEs, or Windows Update files:

Get-WmiObject -Class Win32\_QuickFixEngineering -ComputerName .

#eller

wmic qfe list brief /format:Table

#eller den udvidede

wmic qfe list

Get-WmiObject -Class Win32\_Product -property name, version, vendor

Get-WmiObject -Class Win32\_Product| select name, version, vendor|ft

Remove installed Hotfix (found above)

wusa /uninstall /kb:4570012

Get the username of the person currently logged on to a computer:

Get-WmiObject -Class Win32\_ComputerSystem -Property UserName -ComputerName .

Find just the names of installed applications on the current computer:

Get-WmiObject -Class Win32\_Product -ComputerName . | Format-Wide -Column 1

Eller

Get-AppxPackage -AllUsers | Select Name, PackageFullName

Get IP addresses assigned to the current computer:

Get-WmiObject -Class Win32\_NetworkAdapterConfiguration -Filter IPEnabled=TRUE -ComputerName . | Format-Table -Property IPAddress

Get a more detailed IP configuration report for the current machine:

Get-WmiObject -Class Win32\_NetworkAdapterConfiguration -Filter IPEnabled=TRUE -ComputerName . | Select-Object -Property [a-z]\* -ExcludeProperty IPX\*,WINS\*

Find network cards with DHCP enabled on the current computer:

Get-WmiObject -Class Win32\_NetworkAdapterConfiguration -Filter "DHCPEnabled=true" -ComputerName .

Enable DHCP on all network adapters on the current computer:

Get-WmiObject -Class Win32\_NetworkAdapterConfiguration -Filter IPEnabled=true -ComputerName . | ForEach-Object -Process {$\_.EnableDHCP()}

Enter into a remote PowerShell session -- you must have remote management enabled:

enter-pssession TARGETMACHINE

Use the PowerShell invoke command to run a script on a remote server:

invoke-command -computername machine1, machine2 -filepath c:\Script\script.ps1

Search / søg multiple files recursive for specific string

Get-ChildItem \*.ps1 -recurse | Select-String -pattern "System.Windows.Forms" | group path | select name

List services not started by LocalSystem

Gwmi Win32\_Service | where {$\_.StartName -ne "LocalSystem" } | select Name,StartMode,Startname

## Communication to Domain Controllers

The following table lists the port requirements for establishing DC to DC communication in all versions of Windows Sever beginning with Windows Server 2003.

Additional ports are required for communication between a read-only domain controller (RODC) and a writeable DC.

|  |  |  |
| --- | --- | --- |
| **Protocol and Port** | **AD and AD DS Usage** | **Type of traffic** |
| TCP and UDP 389 | Directory, Replication, User and Computer Authentication, Group Policy, Trusts | LDAP |
| TCP 636 | Directory, Replication, User and Computer Authentication, Group Policy, Trusts | LDAP SSL |
| TCP 3268 | Directory, Replication, User and Computer Authentication, Group Policy, Trusts | LDAP GC |
| TCP 3269 | Directory, Replication, User and Computer Authentication, Group Policy, Trusts | LDAP GC SSL |
| TCP and UDP 88 | User and Computer Authentication, Forest Level Trusts | Kerberos |
| TCP and UDP 53 | User and Computer Authentication, Name Resolution, Trusts | DNS |
| TCP and UDP 445 | Replication, User and Computer Authentication, Group Policy, Trusts | SMB,CIFS,SMB2, DFSN, LSARPC, NbtSS, NetLogonR, SamR, SrvSvc |
| TCP 25 | Replication | SMTP |
| TCP 135 | Replication | RPC, EPM |
| TCP Dynamic | Replication, User and Computer Authentication, Group Policy, Trusts | RPC, DCOM, EPM, DRSUAPI, NetLogonR, SamR, FRS |
| TCP 5722 | File Replication | RPC, DFSR (SYSVOL) |
| UDP 123 | Windows Time, Trusts | Windows Time |
| TCP and UDP 464 | Replication, User and Computer Authentication, Trusts | Kerberos change/set password |
| UDP Dynamic | Group Policy | DCOM, RPC, EPM |
| UDP 138 | DFS, Group Policy | DFSN, NetLogon, NetBIOS Datagram Service |
| TCP 9389 | AD DS Web Services | SOAP |
| UDP 67 and UDP 2535 | DHCP Note DHCP is not a core AD DS service but it is often present in many AD DS deployments. | DHCP, MADCAP |
| UDP 137 | note   |  | | --- | | User and Computer Authentication, | | NetLogon, NetBIOS Name Resolution |
| TCP 139 | User and Computer Authentication, Replication | DFSN, NetBIOS Session Service, NetLogon |

# Andre kommandoer

### Virker på Core Server

Joblisten: taskmgr.exe

**Server Core Roles and Feature Installations**

The typical Windows server roles can be configured on a Server Core installation. The following server roles are currently supported on a Server Core installation:

* Active Directory Certificate Services
* Active Directory Domain Services (AD DS)
* Active Directory Lightweight Directory Services (AD LDS)
* Active Directory Rights Management Server
* DHCP Server
* DNS Server
* File and Storage Services
* Hyper-V
* Print and Document Services
* Remote Access Server
* Remote Desktop Services (partial support)
  + Connection Broker
  + Licensing
  + Virtualization Host only
* Volume Activation Services
* Web Server (IIS)
* Windows Deployment Services
* Windows Software Update Services

The following are some of the features that are also supported on a Server Core installation:

* .NET Framework 4.5
* BITS Compact Server
* BitLocker Drive Encryption
* BranchCache
* Failover Clustering
* Group Policy Management
* IP Address Management (IPAM) Server
* Multipath I/O
* Network Load Balancing
* Remote Server Administration Tools (partial)
* Simple Network Management Protocol (SNMP)
* Telnet Client
* Windows PowerShell 2.0
* Windows Server Backup

The following command lists all the potential server roles and associated features:

Dism /online /get-features /format:table

**Using Powershell to Install a Server Role on a Server Core Installation**

In addition to OCSetup, Powershell can be used to install roles and features, as follows:

1. Run PowerShell by executing powershell.exe in the command prompt.
2. Run import-module servermanager to enable Server Manager features within the PowerShell session.
3. Use Get-WindowsFeature to identify installed and available roles and features.
4. Use Add-WindowsFeature to add the required role or feature. Table 3.4 outlines an example of common server role installations using Add-WindowsFeature.

#### Server Role Installation Command Lines with PowerShell

| **Server Role** | **Command** |
| --- | --- |
| DNS Server role | Add-WindowsFeature DNS |
| DHCP Server role | Add-WindowsFeature DHCP |
| File Server role | Add-WindowsFeature File-Services |
| Print Server role | Add-WindowsFeature Print-Server |
| Active Directory Lightweight Directory Server role | Add-WindowsFeature ADLDS |
| Windows Server Update Services role | Add-WindowsFeature UpdateServices |
| Web Server (IIS) role | Add-WindowsFeature Web-Server |
| Remote Access role | Add-WindowsFeature RemoteAccess |
| Hyper-V role | Add-WindowsFeature Hyper-V |

### Virker ikke på Core Server

System Konfigurationen: msconfig.exe

Computer Management: compmgmt.msc

Component Services: Dcomcnfg.exe

Disk Management: diskmgmt.msc

Services: services.msc

Enhedshåndtering: devmgmt.msc

System Properties: sysdm.cpl

Performance Options: SystemPropertiesPerformance.exe

|  |  |
| --- | --- |
| To Access… | Run Command |
| Accessibility Controls | access.cpl |
| Accessibility Wizard | accwiz |
| Add Hardware Wizard | hdwwiz.cpl |
| Add/Remove Programs | appwiz.cpl |
| Administrative Tools control | admintools |
| Adobe Acrobat (if installed) | acrobat |
| Adobe Designer (if installed) | formdesigner |
| Adobe Distiller (if installed) | acrodist |
| Adobe ImageReady (if installed) | imageready |
| Adobe Photoshop (if installed) | photoshop |
| Automatic Updates | wuaucpl.cpl |
| Bluetooth Transfer Wizard | fsquirt |
| Calculator | calc |
| Certificate Manager | certmgr.msc |
| Character Map | charmap |
| Check Disk Utility | chkdsk |
| Clipboard Viewer | clipbrd |
| Command Prompt | cmd |
| Component Services | dcomcnfg |
| Computer Management | compmgmt.msc |
| Control Panel | control |
| Date and Time Properties | timedate.cpl |
| DDE Shares | ddeshare |
| Device Manager | devmgmt.msc |
| Direct X Control Panel (if installed)\* | directx.cpl |
| Direct X Troubleshooter | dxdiag |
| Disk Cleanup Utility | cleanmgr |
| Disk Defragment | dfrg.msc |
| Disk Management | diskmgmt.msc |
| Disk Partition Manager | diskpart |
| Display Properties control | desktop |
| Display Properties | desk.cpl |
| Display Properties (w/Appearance Tab Preselected) control | color |
| Dr. Watson System Troubleshooting Utility | drwtsn32 |
| Driver Verifier Utility | verifier |
| Event Viewer | eventvwr.msc |
| Files and Settings Transfer Tool | migwiz |
| File Signature Verification Tool | sigverif |
| Findfast | findfast.cpl |
| Firefox (if installed) | firefox |
| Folders Properties | folders |
| Fonts control | fonts |
| Fonts Folder | fonts |
| Free Cell Card Game | freecell |
| Game Controllers | joy.cpl |
| Group Policy Editor (XP Prof) | gpedit.msc |
| Hearts Card Game | mshearts |
| Help and Support | helpctr |
| HyperTerminal | hypertrm |
| Iexpress Wizard | iexpress |
| Indexing Service | ciadv.msc |
| Internet Connection Wizard | icwconn1 |
| Internet Explorer | iexplore |
| Internet Properties | inetcpl.cpl |
| Internet Setup Wizard | inetwiz |
| IP Configuration (Display Connection Configuration) | ipconfig /all |
| IP Configuration (Display DNS Cache Contents) | ipconfig /displaydns |
| IP Configuration (Delete DNS Cache Contents) | ipconfig /flushdns |
| IP Configuration (Release All Connections) | ipconfig /release |
| IP Configuration (Renew All Connections) | ipconfig /renew |
| IP Configuration (Refreshes DHCP & Re-Registers DNS) | ipconfig /registerdns |
| IP Configuration (Display DHCP Class ID) | ipconfig /showclassid |
| IP Configuration (Modifies DHCP Class ID) | ipconfig /setclassid |
| Java Control Panel (if installed) | jpicpl32.cpl |
| Java Control Panel (if installed) | javaws |
| Keyboard Properties control | keyboard |
| Local Security Settings | secpol.msc |
| Local Users and Groups | lusrmgr.msc |
| Logs You Out Of Windows | logoff |
| Malicious Software Removal Tool | mrt |
| Microsoft Access (if installed) | msaccess |
| Microsoft Chat | winchat |
| Microsoft Excel (if installed) | excel |
| Microsoft Frontpage (if installed) | frontpg |
| Microsoft Movie Maker | moviemk |
| Microsoft Paint | mspaint |
| Microsoft Powerpoint (if installed) | powerpnt |
| Microsoft Word (if installed) | winword |
| Microsoft Syncronization Tool | mobsync |
| Minesweeper Game | winmine |
| Mouse Properties control | mouse |
| Mouse Properties | main.cpl |
| Nero (if installed) | nero |
| Netmeeting | conf |
| Network Connections control | netconnections |
| Network Connections | ncpa.cpl |
| Network Setup Wizard | netsetup.cpl |
| Notepad | notepad |
| Nview Desktop Manager (if installed) | nvtuicpl.cpl |
| Object Packager | packager |
| ODBC Data Source Administrator | odbccp32.cpl |
| On Screen Keyboard | osk |
| Opens AC3 Filter (if installed) | ac3filter.cpl |
| Outlook Express | msimn |
| Paint | pbrush |
| Password Properties | password.cpl |
| Performance Monitor | perfmon.msc |
| Performance Monitor | perfmon |
| Phone and Modem Options | telephon.cpl |
| Phone Dialer | dialer |
| Pinball Game | pinball |
| Power Configuration | powercfg.cpl |
| Printers and Faxes control | printers |
| Printers Folder | printers |
| Private Character Editor | eudcedit |
| Quicktime (If Installed) | QuickTime.cpl |
| Quicktime Player (if installed) | quicktimeplayer |
| Real Player (if installed) | realplay |
| Regional Settings | intl.cpl |
| Registry Editor | regedit |
| Registry Editor | regedit32 |
| Remote Access Phonebook | rasphone |
| Remote Desktop | mstsc |
| Removable Storage | ntmsmgr.msc |
| Removable Storage Operator Requests | ntmsoprq.msc |
| Resultant Set of Policy (XP Prof) | rsop.msc |
| Scanners and Cameras | sticpl.cpl |
| Scheduled Tasks control | schedtasks |
| Security Center | wscui.cpl |
| Services | services.msc |
| Shared Folders | fsmgmt.msc |
| Shuts Down Windows | shutdown |
| Sounds and Audio | mmsys.cpl |
| Spider Solitare Card Game | spider |
| SQL Client Configuration | cliconfg |
| System Configuration Editor | sysedit |
| System Configuration Utility | msconfig |
| System File Checker Utility (Scan Immediately) | sfc /scannow |
| System File Checker Utility (Scan Once At The Next Boot) | sfc /scanonce |
| System File Checker Utility (Scan On Every Boot) | sfc /scanboot |
| System File Checker Utility (Return Scan Setting To Default) | sfc /revert |
| System File Checker Utility (Purge File Cache) | sfc /purgecache |
| System File Checker Utility (Sets Cache Size to size x) | sfc /cachesize=x |
| System Information | msinfo32 |
| System Properties | sysdm.cpl |
| Task Manager | taskmgr |
| TCP Tester | tcptest |
|  |  |
| Control panel tool | Command |
| ----------------------------------------------------------------- |  |
| Accessibility Options | control access.cpl |
| Add New Hardware | control sysdm.cpl add new hardware |
| Add/Remove Programs | control appwiz.cpl |
| Date/Time Properties | control timedate.cpl |
| Display Properties | control desk.cpl |
| FindFast | control findfast.cpl |
| Fonts Folder | control fonts |
| Internet Properties | control inetcpl.cpl |
| Joystick Properties | control joy.cpl |
| Keyboard Properties | control main.cpl keyboard |
| Microsoft Exchange | control mlcfg32.cpl |
|  | (or Windows Messaging) |
| Microsoft Mail Post Office | control wgpocpl.cpl |
| Modem Properties | control modem.cpl |
| Mouse Properties | control main.cpl |
| Multimedia Properties | control mmsys.cpl |
| Network Properties | control netcpl.cpl |
|  | NOTE: In Windows NT 4.0, Network |
|  | properties is Ncpa.cpl, not Netcpl.cpl |
| Password Properties | control password.cpl |
| PC Card | control main.cpl pc card (PCMCIA) |
| Power Management (Windows 95) | control main.cpl power |
| Power Management (Windows 98) | control powercfg.cpl |
| Printers Folder | control printers |
| Regional Settings | control intl.cpl |
| Scanners and Cameras | control sticpl.cpl |
| Sound Properties | control mmsys.cpl sounds |
| System Properties | control sysdm.cpl |

Windows substitutes the name of the tool you want to run for %1%. For example:

"rundll32.exe shell32.dll,Control\_RunDLL appwiz.cpl".

To run the Users tool in Control Panel, type control Ncpa.cpl users, and then press ENTER.

**Task: View system information**

* Msinfo32
* Set
* Systeminfo

**Task: View user information**

* Whoami

**Task: Manage users and groups**

* Net accounts
* Net group
* Net localgroup
* Net user

**Task: View or change computer name**

* Hostname
* Netdom renamecomputer

**Task: Join or leave a domain**

* Netdom join

**Task: Log off or shut down**

* Logoff
* Shutdown

**Task: Configure networking**

* Ipconfig
* Netsh interface
* Netsh routing
* Route

**Task: Configure Windows Firewall**

* Netsh advfirewall

**Task: Configure Internet Protocol security (IPsec)**

* Netsh ipsec
* Scregedit.wsf

**Task: Activate Windows**

* Slmgr.vbs

**Task: Manage services**

* Net continue
* Net pause
* Net start
* Net stop
* Sc
* Tasklist

**Task: Manage processes**

* Taskkill
* Tasklist
* Taskmgr

**Task: Manage tasks**

* At
* Schtasks

**Task: Collect and analyze performance data**

* Logman
* Relog
* Typeperf

**Task: View events and manage event logs**

* Wevtutil

**Task: Manage disks and storage**

* Compact
* Defrag
* Diskpart
* Diskraid
* Mountvol

**Task: Manage Volume Shadow Copy Service (VSS)**

* Vssadmin
* Cacl

**Task: Manage file systems and file permissions**

* Convert
* Fsutil
* Icacls
* Takeown

**Task: Manage files**

* Openfiles
* Sigverif

**Task: Manage shares and share permissions**

* Net share

**Task: Manage the registry**

* Reg
* Regedit

**Task: Install and manage drivers**

* Driverquery
* Pnputil
* Sc

**Task: Install and manage updates**

* Pkgmgr
* Scregedit.wsf
* Systeminfo
* Wuauclt
* Wusa

**Task: Install roles and features**

* Oclist
* Ocsetup

**Task: Install applications**

* Msiexec

**Task: Manage Group Policy**

* Gpresult
* Gpupdate
* Secedit

**Task: Manage certificates**

* Certreq
* Certutil

**Task: Manage Terminal Services (Remote Desktop for Administration)**

* Change
* Logoff
* Msg
* Mstsc
* Qappsrv
* Qprocess
* Query
* Qwinsta
* Reset session
* Rwinsta
* Shadow
* Tscon
* Tsdiscon
* Tskill

### Opdater GPO på maskine efter GPO ændring.

GPO virker ofte på OU niveau i AD, så server skal flyttes til korrekt ou for at få korrekt gpo.

gpupdate (/force)

### Vis Ip info alle kort

ipconfig /all

Frigiv IPv4-adressen for det angivne netværkskort.

ipconfig /release

Forny IPv4-adressen for det angivne netværkskort.

ipconfig /renew

Opdater alle DHCP-rettigheder, og registrer DNS-navne igen.

ipconfig /registerdns

### Find logged on & active users on server

query user /server:srv-sas-term02.ssi.ad

### # connect remote

enter-pssession s-inf-file-04-p.dksund.dk -credential XXX

enter-pssession s-exc-hyb-01p.dksund.dk

### # Import-module

Add-PSSnapin \*exch\*

### # Se installerede features

Get-WindowsFeature

### # Tilføje feature

install-windowsfeature -name WindowsPowerShellWebAccess -includemanagementtools -restart

install-windowsfeature -name WindowsPowerShellWebAccess Net-Framework-Core -source d:\source\sxs

### # Se updates

get-hotfix

### # Rename-computer

rename-computer s-inf-file-04-p –newname abcDC2

### # Genstart

Restart-Computer

### # Luk computer

Stop-Computer evt <fqdn>

### # Se IP adresse -remote

Get-NetIPConfiguration evt. <fqdn>

<eller>

netsh interface ipv4 show interfaces

### # Sæt ny IP

New-NetIPAddress -InterfaceAlias Ethernet -IPAddress <IP> -Prefixlength 24 -DefaultGateway <IP>

<eller>

Get-NETIPInterface

Set-NetIPAddress –InterfaceIndex 12 –IPv4Address 192.168.115.10 –PrefixLength 24 –DefaultGateway 192.168.115.1

Set-DNSClientServerAddress –InterfaceIndex 12 –ServerAddresses "DNSIP1","DNSIP

### # Sætte dns-server

Set-DNSClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses <ip> <ip>

Eksempel: Set-DnsClientServerAddress -InterfaceIndex 12ipconfig /all -ServerAddresses ("195.80.246.106","127.0.0.1")

### # Tilføje computer til domain

Add-Computer -DomainName dksund.dk

### # Se netværksstatistik

Get-NetAdapterStatistics

### # Test netværk (kombination af ping og nslookup)

Test-NetConnection

Test-NetConnection 8.8.8.8

Test-NetConnection srv-ad-dc04 -traceroute

### # Port check

Test-NetConnection srv-ad-dc04 -port 389

Test-NetConnection s-exc-hyb-01p.dksund.dk -port 25

Test-NetConnection -ComputerName s-mssqlasis05-t.dksund.dk -Port 1433 -InformationLevel Detailed

### # Genskabe forbindelsen til domainet for klient UDEN at skulle melde den ind og ud af domainet

### Test-ComputerSecureChannel -Credential dksund\adm-XXX -repair

### # Genskabe domain forbindelse på klient remote

Test-ComputerSecureChannel -Server WXU004872 dksund\adm-XXX -repair

### # Se Error logs remote

Get-EventLog -LogName System -EntryType Error

### # Services

Stop-services

Start-Service

Restart-service

Set-services

Get-services

## Remote SQL Server Configuration Management (From Gui machine with SQL)

* Right-click the Start menu button, or start Server Manager and then click Computer Management (Local).
* On the Action menu, click Connect to another computer.
* In the Select Computer dialog box, in the Another computer text box, type the name of the computer you want to manage, and then click OK.
* Computer Management displays the services running on the remote computer. The top-level node changes to Computer Management <remotecomputer>.
* In the console tree, expand Services and Applications, and then expand SQL Server Configuration Manager to manage the remote computer's services.

**To save a link to SQL Server Configuration Manager for another computer**

1. On the **Start** menu, click **Run**.
2. In the **Open** box, type **mmc -a** (type **mmc /32 -a** on a 64-bit computer) to open the Microsoft Management Console in author mode.
3. On the **File** menu, click **Add/Remove Snap-in**.
4. In the **Add/Remove Snap-in** window, click **Add**.
5. In the **Add Standalone Snap-in** window, click **Computer Management** and then click **Add**.
6. In the **Computer Management** window, click **Another computer**, type the name of the remote computer you wish to manage, and then click **Finish**.
7. In the **Add Standalone Snap-in** window, click **Close**.
8. In the **Add/Remove Snap-in** window, click **OK**.
9. Expand **Computer Management (**<computer name>**)**, and **Services and Applications**.
10. Right-click **SQL Server Configuration Manager**, and then click **New Window from here**.
11. On the **Window** menu, click **Console Root**, to switch back to the first window, and delete the window.
12. On the **File** menu, click **Save As**, and save the file in the desired folder, with an appropriate name with the **.msc** file extension. Close the Microsoft Management Console.
13. To open SQL Server Configuration Manager on the target computer, double-click the file. If desired, save a link to the file on the desktop or in the **Start** menu.

## Windows 2019 core AppCompatibility (.mmc på core)

# I Powershell tilføjes evnen

copy P:\SQLDBA\WindowsFeatureOnDemand\ -r c:\install

$mountResult=Mount-DiskImage -passthru –ImagePath C:\Install\WindowsFeatureOnDemand\SW\_DVD9\_NTRL\_Win\_Svr\_FOD\_App\_Compat\_2019\_64Bit\_MultiLang\_X21-91284.ISO

$mountResult

Add-WindowsCapability -Online -Name ServerCore.AppCompatibility~~~~0.0.1.0 -Source e: -LimitAccess

Restart-Computer

# Start .mmc som SQL Configuration Manager

.\Windows\SysWOW64\SQLServerManager15.msc

# I Powershell fjernes evnen

Remove-WindowsCapability -Online -Name ServerCore.AppCompatibility~~~~0.0.1.0

Restart-Computer

### # SQL Services

For at stoppe SQL Service for alle SQL Instanser på aktuel server kan du i PowerShell:

Get-Service -displayname 'Sql Server Agent (\*' | ForEach-Object {Stop-Service $\_.Name}

Get-Service -displayname 'Sql \*' | ForEach-Object {Stop-Service $\_.Name}

# Eller

Stop-Service -displayname Sql\*

#Dette er for at sikre at SQL Agenten bliver stoppet først

Du kan altid checke status ved:

Get-Service -displayname 'Sql \*'

Og du kan starte SQL Services igen ved:

Get-Service -displayname 'Sql Server (\*' | ForEach-Object {Start-Service $\_.Name}

Get-Service -displayname 'Sql \*' | ForEach-Object {Start-Service $\_.Name}

# Eller

Start-Service -displayname Sql\*

#Dette er for at sikre at SQL Engine bliver startet først

En bedre mulig er dog at genstarte serveren når du er færdig

# Problematiske services

Service kan ikke stoppes, status Stopping

Get-Process | Where {$\_.processname -like "\*check\*"}

tasklist /m check\_mk\_agent.exe

taskkill /PID 1484 /F

Stop-Process -id 1484 -force

### # Se services, som er stopped

Get-Service | where-Object {$\_.Status -eq "Stopped"}

### #Services på remote pc

Get-Service -comp s-mssql61-t \*sql\*

Get-Service -comp s-mssql61-t \*sql\*|Set-Service -status Running

### # Firewall

# Søg om local firewall port er åben 1: Søg i Netstat

/\* virker i powershell \*/

$f=netstat -an

$f.count #viser hvor mange linier der er fundet.

foreach($fw in $f) {if($fw.contains("443") ){$fw}}

/\* Opretter temporær local funktion til at søge I firewall: Virker i powershell \*/

Function Get-FireWallRule{

Param ($Name, $Direction, $Enabled, $Protocol, $profile, $action, $grouping)

$Rules=(New-object –comObject HNetCfg.FwPolicy2).rules

If ($name) {$rules= $rules | where-object {$\_.name -like $name}}

If ($direction) {$rules= $rules | where-object {$\_.direction -eq $direction}}

If ($Enabled) {$rules= $rules | where-object {$\_.Enabled -eq $Enabled}}

If ($protocol) {$rules= $rules | where-object {$\_.protocol -eq $protocol}}

If ($profile) {$rules= $rules | where-object {$\_.Profiles -bAND $profile}}

If ($Action) {$rules= $rules | where-object {$\_.Action -eq $Action}}

If ($Grouping) {$rules= $rules | where-object {$\_.Grouping -like $Grouping}}

$rules

}

Get-FirewallRule | where { $\_.LocalPorts -in 80,443}

# Tilføje port åbning

New-NetFirewallRule -DisplayName "Tillad inbound port 80" -Direction Inbound -Localport 80 -Protocol TCP -Action Allow

# Søg firewall regel

Get-NetFirewallRule | where { $\_.Enabled -eq 'True' -and $\_.Direction -eq 'Inbound' -and $\_.DisplayName -match 'SQL Server Analysis Services'}

# Eller

p:\SQLDBA\ps1\2016\get-fwrule.ps1 #lister firewall regler og viser sql & Check\_MK

# Eller

$f1=Get-NetFirewallRule | where { $\_.Enabled -eq 'True' -and $\_.Direction -eq 'Inbound' -and $\_.DisplayName -Like 'SQL\*Analysis\*'}

# Find firewall port

$f2=$f1 | Get-NetFirewallPortFilter

Write-Host “Firewall rule displayname: “ $f1.DisplayName

Write-Host “Firewall rule LocalPort: “ $f2.LocalPort

# Naviger I Firewall regler

$rules=Get-NetFirewallRule -Enabled True -Direction Inbound | Where {$\_.DisplayName -match 'Remote'}

$rules[0]|Get-NetFirewallPortFilter

$rules[0]|Get-NetFirewallAddressFilter

$rules[0]|Get-NetFirewallApplicationFilter

$rules[0]|Get-NetFirewallInterfaceFilter

$rules[0]|Get-NetFirewallProfile

### # Sætte remote managemnet via GUI

Enable-NetFirewallRule -DisplayGroup "Remote Administration"

Get-NetFirewallRule | Where-Object {$\_.DisplayGroup -like '\*wmi\*'}

Enable-NetFirewallRule -DisplayGroup 'Windows Management Instrumentation (WMI)'

-Enabled "True"??

## # Active Directory

### # Sætte nyt password på user-account

$newpwd = CovertTo-SecureString -String "kodeordet" -AsPlainText -Force

Set-ADAccountPassword <user account> -NewPasssword $newpwd -reset -PassThru | Set-ADuser -ChangePasswordAtLogon $False

search-ADAccount -PasswordNeverExpire

### # Oprette AD-gruppe

New-ADgroup -name xxx -SamAccountName xxx -GroupCategory Security -GroupScope Global -Path "OU=FS,OU=T2Groups,OU=Tier2,DC=dksund,DC=dk"

### # ISE Snippets

# Lægge meget brugt ps-kode, som lægges ind i ISE

# Kræver PS policy sat til Unrestricted

New-IseSnippet -Force -Title "Password\_String" -Description "Secure Password String" -Text "'$newpwd = CovertTo-SecureString -String "kodeordet" -AsPlainText -Force"

## # DNS

### # Tilføje ny dns zone

Add-DnsServerPrimaryZone -Name "etellerandet.xx.dk" -ReplicationScope "Forest" -Passthru

### # Tilføje host-record

Add-DnsServerResourceRecordA -name "nogetrelevant" -ZoneName "sst.dk" -AllowUpdatesAny -IPv4Address "192.168.31.12" -TimeToLive 01:00:00

## # File shares

### # Nyt File share

New-SmbShare -Name <indsæt navn> -Path D:\data\<indsæt navn> -FullAccess dksund\Authenticated users

### # remote desktop permissions

# Local Users & Groups \Groups\Remote Desktop Users Properties

# Local Security Policy

# på maskinen Secpol.msc

# Security Settings\Local Policies\User Rights Assignment\"Allow log on through RDP" (tidligere "Allow log on through Terminal Services"

# Opret AD gruppe, og tilføj gruppen her.

Who is logged on to a computer?

We often need to know who is currently logged on to a remote computer.

With native Windows (up to and including XP) commands only:

NBTSTAT -a remotecomputer | FIND "<03>" | FIND /I /V "remotecomputer"

The first name in the list usually is the logged on user (try playing with the NET NAME command to learn more about the names displayed by NBTSTAT).

This is the fastest way to find the logged on user name, and the results that you do get are correct, but NBTSTAT won't always return a user name, even when a user is logged on.

Using WMIC (Windows XP Professional and later):

WMIC /Node:remotecomputer ComputerSystem Get UserName

This is arguably the most reliable (native) command to find out who is logged on.

With the help of SysInternals' PSTools:

PSLOGGEDON -L \\remotecomputer

or:

PSEXEC \\remotecomputer NET CONFIG WORKSTATION | FIND /I " name "

or:

PSEXEC \\remotecomputer NET NAME

or for Windows XP only:

PSEXEC \\remotecomputer NETSH DIAG SHOW COMPUTER /V | FIND /i "username"

Using REG.EXE (Windows 2000 and later):

FOR /F %%A IN ('REG Query \\remotecomputer\HKU ˆ| FINDSTR /R /B /C:"HKEY\_USERS\\S-1-5-[0-9][0-9]-[0-9-]\*$"') DO (

FOR /F "tokens=3 delims=\" %%B IN ('REG Query "\\remotecomputer\%%A\Volatile Environment"') DO (

SET LoggedinUser=%%B

)

)

or for Windows 7:

FOR /F %%A IN ('REG Query \\remotecomputer\HKU /K /F "S-1-5-21-" ˆ| FINDSTR /R /B /C:"HKEY\_USERS\\S-1-5-[0-9][0-9]-[0-9-]\*$"') DO (') DO (

FOR /F "tokens=2\*" %%B IN ('REG Query "\\remotecomputer\%%~A\Volatile Environment" /V "UserName" ˆ| FIND /V ":"') DO (

SET LoggedinUser=%%C

)

)

NETSH and WMIC are for XP or later, and are the most reliable of all commands shown here.

WMIC requires WMI enabled remote computers and Windows XP on the administrator's computer; NETSH requires Windows XP on the local and remote computers.

PSLOGGEDON is a more accurate solution than NBTSTAT, but it will return the last logged on user if no one is currently logged on.

The NET and NBTSTAT commands show more or less identical results, but the NBTSTAT command is much faster.

The REG command is accurate, but may need to be modified depending on the version used.

More information on REG versions can be found on my REG Query page.

For Windows NT 4 and 2000: use NBTSTAT (fast, but it won't always return the user name!), and only switch to REG if NBTSTAT doesn't return a user name (modify the REG command for Windows NT 4).

For Windows XP and later: if you want to search lots of computers for logged on users, I recommend you try NBTSTAT first (fast, but it won't always return the user name!), and only switch to NETSH, REG or WMIC (accurate) if NBTSTAT doesn't return a user name.

What is this colleague's login name?

My colleagues often forget to mention their logon account name when calling the helpdesk, and the helpdesk doesn't always ask either. I suppose they expect me to know all 1500+ accounts by heart.

With (native) Windows Server 2003 commands only:

DSQUERY USER -name \*lastname\* | DSGET USER -samid –display

What is the full name for this login name?

With the native NET command:

NET USER loginname /DOMAIN | FIND /I " name "

With (native) Windows Server 2003 commands:

DSQUERY USER -samid \*loginname\* | DSGET USER -samid –display

What groups is this user a member of?

In Windows NT 4 and later, users usually are members of global groups. These global groups in turn are members of (domain) local groups. Access permissions are given to (domain) local groups.

To check if a user has access to a resource, we need to check group membership recursively.

With (native) Windows Server 2003 commands:

DSQUERY USER -samid loginname | DSGET USER -memberof –expand

What permissions does a user have on this directory?

One could use the previous command to check what permissions a user has on a certain directory.

However, sometimes SHOWACLS from the Windows Server 2003 Resource Kit Tools is a better alternative:

CD /D d:\directory2check

SHOWACLS /U:domain\userid

When did someone last change his password?

With the native NET command:

NET USER loginname /DOMAIN | FIND /I "Password last set"

How do I reset someone's password?

With the native NET command:

NET USER loginname newpassword /DOMAIN

With (native) Windows Server 2003 commands:

DSQUERY USER -samid loginname | DSMOD USER -pwd newpassword

Is someone's account locked?

With the native NET command:

NET USER loginname /DOMAIN | FIND /I "Account active"

The account is either locked ("Locked") or active ("Yes").

How to unlock a locked account

With the native NET command:

NET USER loginname /DOMAIN /ACTIVE:YES

or, if the password needs to be reset as well:

NET USER loginname newpassword /DOMAIN /ACTIVE:YES

Make sure a local user's password never expires

With WMIC (Windows XP Professional or later):

WMIC.EXE /Node:remotecomputer Path Win32\_UserAccount Where Name="user" Set PasswordExpires="FALSE"

Make sure a local user's password will expire

With WMIC (Windows XP Professional or later):

WMIC.EXE /Node:remotecomputer Path Win32\_UserAccount Where Name="user" Set PasswordExpires="TRUE"

List all domains and workgroups in the network

With the native NET command:

NET VIEW /DOMAIN

List all computers in the network

With the native NET command:

NET VIEW

or, to list the names only:

FOR /F "skip=3 delims=\ " %%A IN ('NET VIEW') DO ECHO.%%A

delims is a backslash, followed by a tab and a space.

List all domain controllers

With native Windows 2000 commands:

NETDOM QUERY /D:MyDomain DC

NETDOM is part of the support tools found in the \SUPPORT directory of the Windows 2000 installation CDROM.

With (native) Windows Server 2003 commands (Active Directory only):

DSQUERY Server

or, if you prefer host names only (tip by Jim Christian Flatin):

DSQUERY Server -o rdn

Find the primary domain controller

With native Windows 2000 commands:

NETDOM QUERY /D:MyDomain PDC

or, to find the FSMO with (native) Windows Server 2003 commands (Active Directory only):

NETDOM QUERY /D:mydomain.com FSMO

NETDOM is part of the support tools found in the \SUPPORT directory of the Windows 2000 installation CDROM.

List all member servers

With native Windows 2000 commands:

NETDOM QUERY /D:MyDomain SERVER

NETDOM is part of the support tools found in the \SUPPORT directory of the Windows 2000 installation CDROM.

List all workstations

With native Windows 2000 commands:

NETDOM QUERY /D:MyDomain WORKSTATION

NETDOM is part of the support tools found in the \SUPPORT directory of the Windows 2000 installation CDROM.

Delete a computer account

With native Windows 2000 commands:

NETDOM /DOMAIN:MyDomain MEMBER \\computer2Bdeleted /DELETE

NETDOM is part of the support tools found in the \SUPPORT directory of the Windows 2000 installation CDROM.

"I need an up-to-date list of disk space usage for all servers, on my desk in 5 minutes"

Sounds familiar?

With (native) Windows XP Professional or Windows Server 2003 commands:

FOR /F %%A IN (servers.txt) DO (

WMIC /Node:%%A LogicalDisk Where DriveType="3" Get DeviceID,FileSystem,FreeSpace,Size /Format:csv | MORE /E +2 >> SRVSPACE.CSV

)

The only prerequisites are:

1.SRVSPACE.CSV should not exist or be empty,

2.a list of server names in a file named SERVERS.TXT, one server name on each line,

3.and WMIC.EXE, which is native in Windows XP Professional and later.

The CSV file format is ServerName,DeviceID,FileSystem,FreeSpace,Size (one line for each harddisk partition on each server).

If you have a strict server naming convention, SERVERS.TXT itself can be generated with the NET command:

FOR /F "delims=\ " %%A IN ('NET VIEW ^| FINDSTR /R /B /C:"\\\\SRV\-"') DO (>>SERVERS.TXT ECHO.%%A)

Notes: (1) assuming server names start with "SRV-"; modify to match your own naming convention.

(2) delims is a backslash, followed by a tab and a space.

List all drivers on any PC

With (native) Windows XP Professional or Windows Server 2003 commands:

DRIVERQUERY /V /FO CSV > %ComputerName%.csv

Or, for remote computers:

DRIVERQUERY /S remote\_PC /V /FO CSV > remote\_PC.csv

List all printers on any PC

With (native) Windows XP+ commands:

WMIC /Node:remote\_PC Path Win32\_Printer Get DeviceID

List all local administrators

With (native) Windows NT 4+ commands:

NET LOCALGROUP Administrators

Or, to remove header and footer lines:

FOR /F "delims=[]" %%A IN ('NET LOCALGROUP Administrators ˆ| FIND /N "----"') DO SET HeaderLines=%%A

FOR /F "tokens=\*" %%A IN ('NET LOCALGROUP Administrators') DO SET FooterLine=%%A

NET LOCALGROUP Administrators | MORE /E +%HeaderLines% | FIND /V "%FooterLine%"

Locate rogue DHCP servers

Never had an "illegal" router wreaking havoc on your network yet...?

With a (native) Windows Server 2003 command:

DHCPLOC -p local\_IP\_address [ valid\_DHCP\_server1 [ valid\_DHCP\_server2 [ .. ] ] ]

DHCPLOC.EXE is native in Windows Server 2003, and will run in Windows XP if copied/installed.

I didn't test this in Windows Server 2003 yet, but in Windows XP you need to press "d" to start the discovery, or "q" to quit.

Disable Windows Firewall for domain only

Disable the firewall only when the computer (e.g. a laptop) is connected to the domain:

NETSH Firewall Set OpMode Mode = DISABLE Profile = DOMAIN

Completely disable Windows Firewall (not recommended)

Disable the firewall completely (not recommended unless an alternative enterprise firewall is used that requires you to do so):

SC [ \\Remote\_computer ] Stop SharedAccess

SC [ \\Remote\_computer ] Config SharedAccess start= disabled

Is IP v4 supported on this computer?

Check if IP v4 is supported on the local computer:

PING 127.0.0.1 | FIND "TTL=" >NUL 2>&1

IF ERRORLEVEL 1 (ECHO IP v4 NOT supported) ELSE (IP v4 supported)

or:

WMIC Path Win32\_PingStatus WHERE "Address='127.0.0.1'" Get StatusCode /Format:Value | FINDSTR /X "StatusCode=0" >NUL 2>&1

IF ERRORLEVEL 1 (ECHO IP v4 NOT supported) ELSE (IP v4 supported)

The WMIC command is faster, but requires Windows XP Professional or later.

Is IP v6 supported on this computer?

Check if IP v6 is supported on the local computer:

PING ::1 | FINDSTR /R /C:"::1:[ˆ$]" >NUL 2>&1

IF ERRORLEVEL 1 (ECHO IP v6 NOT supported) ELSE (IP v6 supported)

or:

WMIC Path Win32\_PingStatus WHERE "Address='::1'" Get StatusCode >NUL 2>&1

IF ERRORLEVEL 1 (ECHO IP v6 NOT supported) ELSE (IP v6 supported)

The WMIC command is faster, but requires Windows XP Professional or later.

Which updates were installed on this compter?

Windows 7 and 8:

DISM /Online /Get-Packages

or:

WMIC QFE List

DISM will return far more details than WMIC.

Windows 2000 and XP:

QFECHECK /V