

Exchange Management Shell

Windows vs. Unix

Al jaren is er een “strijd” tussen Windows en Unix beheerders. Windows beheerders staan nog wel eens bekend om hun klik- capaciteit. Grapjes over Microsoft Certified Click Engineer, klik-klak- klaar, next-next-finish en meer. De gehele Windows omgeving staat erom bekend dat hij geconfigureerd dient te worden met een of andere grafische Interface. Windows gebruikers zouden dan ook muizen verslijten per dozijn. Met de komst van Exchange 2007 en de Exchange Management Shell (EMS) is daar duidelijk verandering ingekomen. Windows Exchange Engineers kunnen nu eindelijk wedijveren met hun Unix “concullega’s”. Met de Exchange Management Shell is het nu mogelijk de gehele Exchange installatie en configuratie te doen met een commandoprompt. Het is zelfs onmogelijk om de installatie volledig uit te voeren zonder deze Shell. Met een van te voren goed doordacht script verloopt de installatie zelfs stukken sneller dan wanneer dit met de interface wordt uitgevoerd.

Erwin Jansen, www.c2ict.com

De Exchange Management Shell is gebouwd op Microsoft Windows Powershell technologie. Hiermee kunnen administratieve taken binnen Microsoft Exchange 2007 eenvoudig via een commandoprompt worden uitgevoerd. Sinds Microsoft Exchange 2007 Service pack 1 overlappen de Exchange Management Shell en de Exchange Management Console elkaar. Voordien was het onmogelijk de Exchange Management Shell niet te gebruiken.

Help

Om een beheerder op weg te helpen is de ingebouwde helpfunctie onontbeerlijk. De helpfunctie geeft uitgebreid weer waar een rol of taak voor dient en wat de mogelijke opties en variabelen zijn. Door op de TAB-toets te drukken is het mogelijk om door de beschikbare functies en opties heen te bladeren. Op deze manier wordt snel een bepaalde cmdlet aangevuld.

get-help *	Alle onderwerpen
get-help get-*	Alle 'get' onderwerpen

Cmdlet

Er bestaan voor Exchange ruim 360 Cmdlets. Deze Cmdlets (spreek uit als “Command-lets”) zijn een verzameling van commando’s voor een breed scala aan administratieve taken binnen Microsoft Exchange 2007. Het gaat te ver om in dit artikel alle Cmdlets te bespreken. Een selectie van deze commando’s zullen de revue passeren om het werk van een beheerder eenvoudiger te maken. Een

aantal van deze cmdlets zullen hergebruikt worden om de functionaliteit van Windows Powershell en de Exchange Management Shell toe te lichten. Het behoeft enige gewenning voordat je er gemakkelijk en adequaat mee om kan gaan. Is de ervaring er enigszins, dan gaat er een wereld open voor de beheerder en waant hij zich op weg in een wereld die voornamelijk gedomineerd werd door Unix beheerders. Nu moet dit niet zo letterlijk genomen worden. Echte Windows experts gebruiken al lang een command line interface of scripts om hun dagelijkse werkzaamheden te vereenvoudigen. Met de komst van Powershell en de Exchange Management Shell wordt dat meer dan eens bevestigd.

```
[PS] C:\>Get-Exchange
```

Name	Site	ServerRole	Edition	AdminDisplayVersion
----	----	-----	-----	-----
PACIFIC	SALTWATER	Mailbox,...	Standard...	Version 8.1...
SVR-HUB-01		HubTrans...	Standard...	Version 8.1...
SVR-HUB-02		HubTrans...	Standard...	Version 8.1...
SVR-CLIENTACCESS-01		ClientAc...	Standard...	Version 8.1...
SVR-CLIENTACCESS-02		ClientAc...	Standard...	Version 8.1...
SVR-MAILBOX-01		Mailbox	Standard...	Version 8.1...
SVR-MAILBOX-02		Mailbox	Standard...	Version 8.1...

Opmaak

Door de grote brei aan informatie die weergegeven wordt in je EMS is het niet altijd even duidelijk wat je precies nodig hebt.

```
[PS] C:\>Get-Mailbox
```

Name	Alias	ServerName	ProhibitSendQuota
----	----	-----	-----
Administrator	Administrator	pacific	unlimited
Jan Jansen	janj	pacific	unlimited
Piet Pietersen	pietp	pacific	unlimited
Frans Fransen	fransf	pacific	unlimited
Klaas Klaassen	klaask	pacific	unlimited

```
[PS] C:\>Get-Mailbox -Identity janj
```

Name	Alias	ServerName	ProhibitSendQuota
----	----	-----	-----
Jan Jansen	janj	pacific	unlimited

Door opties mee te geven aan een cmdlet kun je in dit bijvoorbeeld een identiteit meegeven voor een user mailbox. In het bovenstaande geval kun je specifiekere informatie weergeven van de user "janj." In de Commandline interface kan niet alle informatie naast elkaar worden weergegeven. Wil je een andere opmaak meer informatie te kunnen zien, dan kun je het "Format-List" cmdlet of "| fl" mee te geven.

```
[PS] C:\>Get-Mailbox -Identity janj | Format-List
```

```
Database : PACIFIC\First Storage Group\Mailbox Data base
DeletedItemFlags : DatabaseDefault
UseDatabaseRetentionDefaults : True
RetainDeletedItemsUntilBackup : False
DeliverToMailboxAndForward : False
RetentionHoldEnabled : False
EndDateForRetentionHold :
StartDateForRetentionHold :
ManagedFolderMailboxPolicy :
ExchangeGuid : 0792ff08-f36a-4d7e-8312-54ecba926679
```

```

ExchangeSecurityDescriptor      : System.Security.AccessControl.RawSecurityDescriptor
ExchangeUserAccountControl     : None
ExternalOofOptions             : External
ForwardingAddress              :
RetainDeletedItemsFor          : 14.00:00:00
IsMailboxEnabled               : True
Languages                      : {}
OfflineAddressBook             :
.
.
.
WhenChanged                    : 10-10-08 10:33:57
WhenCreated                    : 10-10-08 10:33:57

```

Je ziet meteen een enorme hoeveelheid aan informatie over de mailbox.

Nt: default wordt de opmaak Format-Table (ft) meegegeven.

Pipelining

Zodra je aan de slag gaat met EMS kom je in aanraking met Pipes en Pipelining. In de Exchange Management Shell kunnen meerder Cmdlets “achter elkaar” geschreven worden. Dit gebeurt met behulp van een ‘Pipe |’. Hierdoor wordt het resultaat uit de eerste cmdlet doorgestuurd, verwerkt of ingevoerd in het volgende cmdlet.

Nt. Op je toetsenbord wil deze karakter er nog wel eens uitzien als ‘|’. In je EMS wordt deze weergegeven als ‘|’, in je Editor wordt deze weergegeven als ‘|’

```
[PS] C:\>Get-Mailbox -organizationalunit Test | Set-Mailbox -IssueWarningQuota 1GB
```

Hiermee worden voor alle mailboxen in de Organizationalunit “Test” de IssueWarningQuota op 1GB gezet.

Simpel tot complex

Je kunt nu eenvoudige tot complexe cmdlets bouwen. Vooral “one-liners” zijn erg interessant. Met 1 regel code zoveel mogelijk opdrachten proberen uit te voeren.

```
[PS] C:\> Get-Mailbox -organizationalunit Test | Set-Mailbox -IssueWarningQuota 1GB –
ProhibitSendQuota 2GB –ProhibitSendReceiveQuota 3GB
```

Zet limieten voor alle users met een mailbox in Test.

Nt. Door middel van een ‘;’ kun je meerdere cmdlets achter elkaar uitvoeren. Het bovenstaande cmdlet kan dus ook als volgt worden weergegeven

```
[PS] C:\> Get-Mailbox -organizationalunit Test | Set-Mailbox -IssueWarningQuota 1GB ; Get-Mailbox
-organizationalunit Test | Set-Mailbox –ProhibitSendQuota 2GB ; Get-Mailbox -organizationalunit
Test | Set-Mailbox –ProhibitSendReceiveQuota 3GB
```

Array

Een array is een data structuur voor de opslag van een verzameling van gegevens van hetzelfde type. Windows PowerShell ondersteunt data-elementen, zoals string, int (32-bit integer) of long (64-bit integer), bool (boolean), een byte en andere. NET object types.

Variabelen

De Exchange management Shell kent twee typen variabelen. Ten eerste de “gewone” variabelen (Variables) en de Shell variabelen (Shell Variables). Een variabele is een object om informatie in op te slaan. In tegenstelling tot veel andere programmeertalen, hoef je een variabele niet eerst te declareren.

Shell Variables zijn een set van variabelen die automatisch gemaakt of gedeclareerd worden door de Exchange Management Shell. Deze variabelen worden bijgehouden gedurende de sessie als onderdeel van het systeem en zijn beschikbaar voor alle commando's, scripts en toepassingen die erop draaien in deze sessie. De variabele '\$_' bevat het huidige pijplijnobject, dat gebruikt wordt in scripts, filters of sets van cmdlets.

```
Get-Mailbox -organizationalunit Test | foreach-object { if ($_.IssueWarningQuota -eq "1GB" ) {Set-Mailbox $_ -IssueWarningQuota 900MB} }
```

De bovenstaande cmdlet houdt in: Verander voor alle gebruikers met een mailbox in de OU 'Test' de IssueWarningQuota op 900MB als deze gelijk is aan 1GB.

Doordat de waarden vanuit de Get-Mailbox cmdlet in een array gezet worden kunnen er meerdere bewerkingen uitgevoerd worden.

Hulpmiddel

Voor een Exchange 2007 project bij een klant moesten er een 20-tal Storage groups en databases worden aangemaakt. De Storage groups hadden quota's voor de grootte. Dit kun je doen met de Exchange Management Console, maar waarom niet met de Exchange Management Shell. Wel zo gemakkelijk en minder kans op fouten.

```
#Storage group en Database
MD N:\LOGS01
MD G:\SG01
new-StorageGroup -Server 'MAILSERVER' -Name 'SG01' -LogFolderPath 'N:\LOGSG01' -SystemFolderPath 'G:\SG01'
new-mailboxdatabase -StorageGroup 'MAILSERVER\SG01' -Name 'MBXSTORE01' -EdbFilePath 'G:\SG01\MBXSTORE01.edb'
Set-MailboxDatabase MBXSTORE01 -IssueWarningQuota 900MB -ProhibitSendQuota 1GB

# Public Folder
MD N:\ LOGSGPF
MD G:\SGPF
New-StorageGroup -Server 'MAILSERVER' -Name 'PFSG00' -LogFolderPath 'N:\LOGSGPF' -SystemFolderPath 'G:\SGPF'
New-PublicFolderDatabase -StorageGroup 'MAILSERVER\PFSTORE00' -Name 'PFSTORE00' -EdbFilePath
'G:\SGPF\PFSTORE00.edb'

# MailboxDatabase PublicFolderDatabase
Set-MailboxDatabase MBXSTORE01 -PublicFolderDatabase MAILSERVER\PFSG00\PFSTORE00
```

Het bovenstaande script is uitgewerkt voor het aanmaken van 20 storage groups, 20 databases en een Publicfolder

Last but not least.

Het Exchange 2007 project omvatte tevens een “rebranding” De klant kreeg een nieuwe naam. Voor het omzetten van alle email adressen is er gebruikgemaakt van een eigen ontwikkeld script. Er kon geen gebruik gemaakt worden van een policy om de email adressen om te zetten, door dat de naam niet voldeed aan een standaard.

```
#CREATE export file as CSV file with SamAccountname and primary adress.

get-mailbox | select Samaccountname,PrimarySmtAddress | export-csv .\output.csv

#Replace in output.csv "old-email-address.com" with "new-email-address.com"
# and write it to file: 'input.csv'

cat '\output.csv' | foreach { $_ -replace ' old-email-address.com', ' new-email-address.com ' } > .\input.csv

#Set new Primary email addresses

import-csv ".\input.csv"|foreach {set-mailbox -id $_.samAccountName -primarysmtpaddress $_.PrimarySMTPAddress}
```

Om te voorkomen dat een email adres verkeerd werd omgezet met een policy is er een script gebruikt.

Voor een overzicht van alle EMS cmdlets kun je terecht op de website van Microsoft.

[http://technet.microsoft.com/en-us/library/bb123703\(EXCHG.80\).aspx](http://technet.microsoft.com/en-us/library/bb123703(EXCHG.80).aspx)

Tip: Speedup Powershell en Exchange Management Shell

Schrijf het volgende script naar een .ps1 bestand. Run het script (met Administrator Rights) via Powershell. Sluit vervolgens alle Shells en start een nieuwe EMS of Powershell. Je zult merken dat de laadtijd van Powershell of de Exchange Management Shell aanzienlijk korter is dan voorheen.

Script

```
Set-Alias ngen (Join-Path
([System.Runtime.InteropServices.RuntimeEnvironment]::GetRuntimeDirectory()) ngen.exe)
[AppDomain]::CurrentDomain.GetAssemblies() |
sort {Split-path $_.location -leaf} |
%{
    $Name = (Split-Path $_.location -leaf)
    if ([System.Runtime.InteropServices.RuntimeEnvironment]::FromGlobalAccessCache($_))
    {
        Write-Host "Already GACed: $Name"
    }else
    {
        Write-Host -ForegroundColor Yellow "NGENing      : $Name"
        ngen $_.location | %{"`t$"}
    }
}
```