ConfigMgr mit PowerShell automatisieren…

Roger Zander

Experts Live Café

Roger Zander

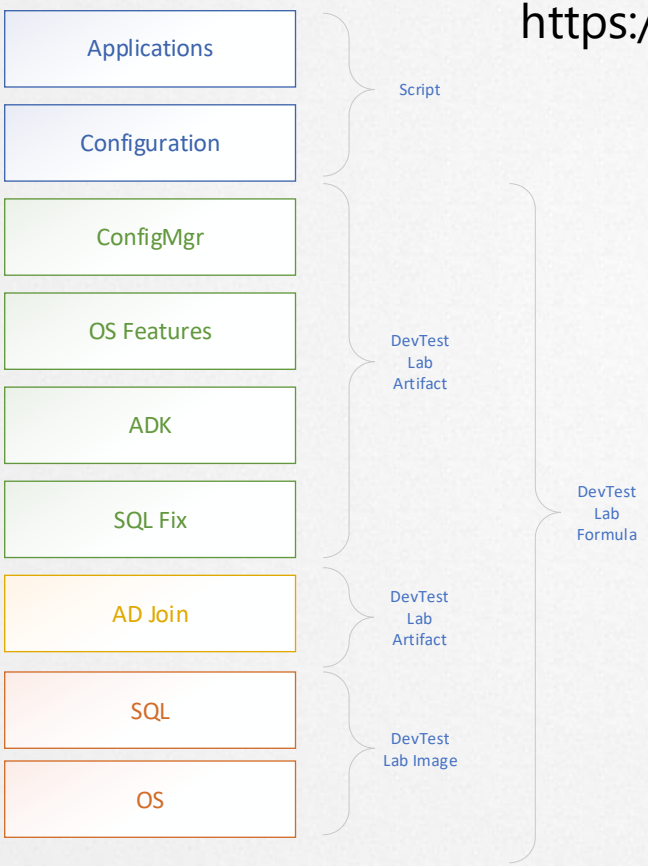
itnetX, Microsoft MVP
@roger_zander
http://rzander.azurewebsites.net/

Agenda

1. ConfigMgr Setup automatisieren
   - **Azure DevTest Lab**
2. ConfigMgr Tasks mit PowerShell
   - **Collections, UpdateGroups etc.**

ConfigMgr Setup… siehe Azure DevTest Lab !

https://github.com/Azure/azure-devtestlab



ADK Setup…

```
$proc = (Start-Process -FilePath "adksetup.exe" -
ArgumentList "/features OptionId.DeploymentTools
OptionID.UserStateMigrationTool
OptionID.ImagingAndConfigurationDesigner
OptionID.WindowsPreinstallationEnvironment /quiet
/norestart /ceip off /log $env:temp\win_adk.log" -PassThru)

$proc.WaitForExit()
$ExitCode = $proc.ExitCode
```

Windows Features…

```
Add-WindowsFeature RDC
Add-WindowsFeature BITS-IIS-Ext
Add-WindowsFeature Web-Metabase
Add-WindowsFeature Web-WMI
Add-WindowsFeature Web-ISAPI-Ext
Add-WindowsFeature NET-Framework-Core
Add-WindowsFeature Web-Windows-Auth
Add-WindowsFeature NET-HTTP-Activation
Add-WindowsFeature NET-Non-HTTP-Activ
Add-WindowsFeature Web-Asp-Net
Add-WindowsFeature Web-Static-Content
Add-WindowsFeature Web-Stat-Compression
Add-WindowsFeature NET-Framework-45-ASPNET
```

https://docs.microsoft.com/en-us/sccm/core/plan-design/configs/site-and-site-system-prerequisites

IIS Configuration (optional)…

```
Import-Module WebAdministration

Set-WebConfigurationProperty -Filter 'system.webServer/security/requestFiltering' -Location 'Default Web Site' -Name
AllowDoubleEscaping -Value $true
Set-WebConfigurationProperty -Filter 'system.webServer/security/requestFiltering' -Name AllowDoubleEscaping -Value $true
Set-WebConfigurationProperty -Filter 'system.webServer/security/requestFiltering' -Location 'Default Web Site' -Name
allowHighBitCharacters -Value $true

foreach($obj in Get-WebConfiguration -Filter 'system.webServer/security/requestFiltering/fileextensions' -Location 'Default Web
Site') { remove-WebConfigurationProperty -Filter $obj.ItemxPath -Name Collection -PSPath 'MACHINE/WEBROOT/APPHOST/Default Web
Site’ }

foreach($obj in Get-WebConfiguration -Filter 'system.webServer/security/requestFiltering/hiddensegments' -Location 'Default Web
Site') { remove-WebConfigurationProperty -Filter $obj.ItemxPath -Name Collection -PSPath 'MACHINE/WEBROOT/APPHOST/Default Web
Site’ }

foreach($obj in Get-WebConfiguration -Filter 'system.webServer/security/requestFiltering/verbs' -Location 'Default Web Site') {
remove-WebConfigurationProperty -Filter $obj.ItemxPath -Name Collection -PSPath 'MACHINE/WEBROOT/APPHOST/Default Web Site’ }

foreach($obj in Get-WebConfigurationProperty -Filter 'system.webServer/security/requestFiltering/fileextensions' -Name Collection)
{ remove-WebConfigurationProperty -Filter $obj.ItemxPath -Name Collection }

foreach($obj in Get-WebConfiguration -Filter 'system.webServer/security/requestFiltering/hiddensegments') { remove-
WebConfigurationProperty -Filter $obj.ItemxPath -Name Collection -PSPath 'MACHINE/WEBROOT/APPHOST’ }

foreach($obj in Get-WebConfiguration -Filter 'system.webServer/security/requestFiltering/verbs') { remove-WebConfigurationProperty
-Filter $obj.ItemxPath -Name Collection -PSPath 'MACHINE/WEBROOT/APPHOST' }
```

ConfigMgr… 1. sccmsetup.ini

```
if(!(Test-Path c:\sccmsetup.ini))
{
$hostname = [System.Net.Dns]::GetHostByName(($env:computerName)).Hostname;
'[Identification]' | out-file -filepath C:\sccmsetup.ini
'Action=InstallPrimarySite' | out-file -filepath C:\sccmsetup.ini -append
'[Options]' | out-file -filepath C:\sccmsetup.ini -append
'ProductID="EVAL"' | out-file -filepath C:\sccmsetup.ini -append
'PrerequisiteComp=0' | out-file -filepath C:\sccmsetup.ini -append
'PrerequisitePath="C:\SCCMDownloads"' | out-file -filepath C:\sccmsetup.ini -append
'SiteCode=TST' | out-file -filepath C:\sccmsetup.ini -append
'SiteName="Test Site"' | out-file -filepath C:\sccmsetup.ini -append
'SMSInstallDir="C:\Microsoft Configuration Manager"' | out-file -filepath C:\sccmsetup.ini -append
"SDKServer=$($hostname)" | out-file -filepath C:\sccmsetup.ini -append
'AdminConsole=1' | out-file -filepath C:\sccmsetup.ini -append
'JoinCEIP=0' | out-file -filepath C:\sccmsetup.ini -append
'RoleCommunicationProtocol=HTTPorHTTPS' | out-file -filepath C:\sccmsetup.ini -append
'ClientsUsePKICertificate=0' | out-file -filepath C:\sccmsetup.ini -append
'AddServerLanguages=' | out-file -filepath C:\sccmsetup.ini -append
'AddClientLanguages=DEU' | out-file -filepath C:\sccmsetup.ini -append
'MobileDeviceLanguage=0' | out-file -filepath C:\sccmsetup.ini -append
"ManagementPoint=$($hostname)" | out-file -filepath C:\sccmsetup.ini -append
'ManagementPointProtocol=HTTP' | out-file -filepath C:\sccmsetup.ini -append
"DistributionPoint=$($hostname)" | out-file -filepath C:\sccmsetup.ini -append
'DistributionPointProtocol=HTTP' | out-file -filepath C:\sccmsetup.ini -append
'DistributionPointInstallIIS=0' | out-file -filepath C:\sccmsetup.ini -append
'[SQLConfigOptions]' | out-file -filepath C:\sccmsetup.ini -append
"SQLServerName=$($hostname)" | out-file -filepath C:\sccmsetup.ini -append
'DatabaseName=CM_TST' | out-file -filepath C:\sccmsetup.ini -append
'SQLSSBPort=4022' | out-file -filepath C:\sccmsetup.ini -append
'[HierarchyExpansionOption]' | out-file -filepath C:\sccmsetup.ini -append
}
```

ConfigMgr… 2. setup.exe

```
$proc = (Start-Process -FilePath
"$($env:temp)\SMSSETUP\BIN\x64\Setup.exe" -ArgumentList
"/script `"c:\sccmsetup.ini`"" -PassThru)

$proc.WaitForExit()
$ExitCode = $proc.ExitCode
```

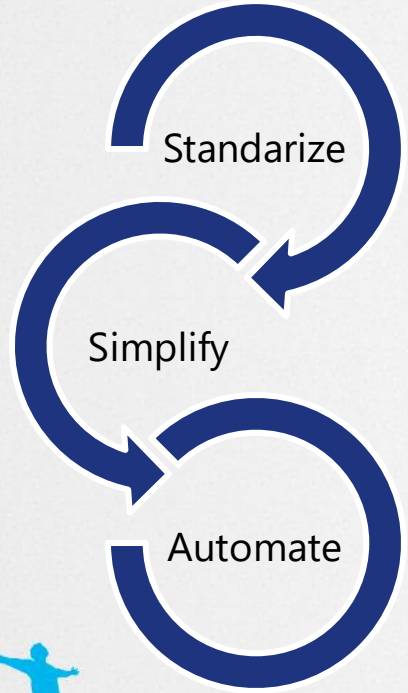
ConfigMgr… 3. SQL Memory

```
#Get Memory of Machine
$mem = Get-WmiObject -Class Win32_ComputerSystem

#Max Mem = 1/2 of full memory, Min Mem = 1/2 of Max Mem
$max = [math]::truncate($($mem.TotalPhysicalMemory/1MB/2))
$min = [math]::truncate($max / 2)

#Store settings in SQL
[System.Reflection.Assembly]::LoadWithPartialName('Microsoft.SqlServer.SMO') | out-null
$s = New-Object ('Microsoft.SqlServer.Management.Smo.Server') "localhost"
$s.Configuration.MinServerMemory.ConfigValue = $min
$s.Configuration.MaxServerMemory.ConfigValue = $max
$s.Configuration.Alter()
```

ConfigMgr… 4. WSUS

```
import-module servermanager
add-windowsfeature UpdateServices-RSAT
add-windowsfeature UpdateServices-API
add-windowsfeature UpdateServices-UI
add-windowsfeature UpdateServices-Services
add-windowsfeature UpdateServices-DB
add-windowsfeature Web-Asp-Net
#Postinstall
&'c:\Program Files\Update Services\Tools\wsusutil.exe'
postinstall SQL_INSTANCE_NAME=localhost CONTENT_DIR=C:\WSUS
```

Denn sie tun nicht was sie wissen…



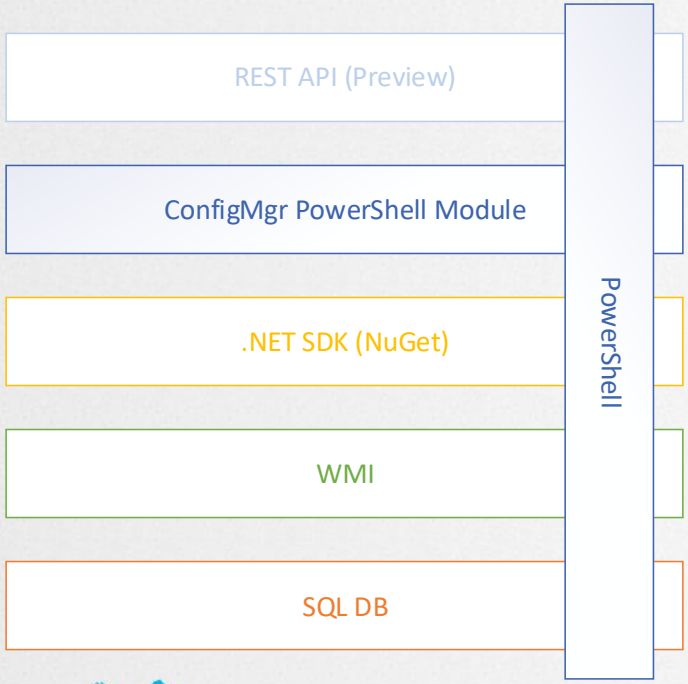
Beispiel 1:

1. Standardisieren
   - SW wird Benutzer zugewiesen
2. Vereinfachen
   - Alle Benutzer sehen alles
3. Automatisieren
   - RuckZuck for ConfigMgr !!!

Beispiel 2:

1. Standardisieren
   - SW wird Benutzer zugewiesen
2. Vereinfachen
   - SW Zuweisung basierend auf Organisation
3. Automatisieren
   - PS:UserCollection pro SW erstellen; AD Gruppe von SW-Tester der Collection hinzufügen…

Configure ConfigMgr with PowerShell




```
#Import SCCM PowerShell Module
import-module (Join-Path $(Split-Path
$env:SMS_ADMIN_UI_PATH)
ConfigurationManager.psd1)

Set-Location ((Get-PSDrive -
PSProvider CMSite).Name + ':')
```

PowerShell Module is part of ConfigMgr CB;
no need to update the Module manually…

https://docs.microsoft.com/en-us/powershell/sccm/overview?view=sccm-ps

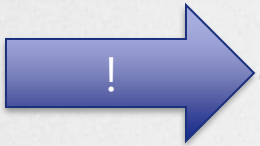
Configure ConfigMgr with PowerShell

```
#List all Functions and Cmdlets
Get-Command -Module ConfigurationManager

#Update Help
update-help -Module ConfigurationManager -Force
```

Beispiel: Folders

```
New-Item -Path .\DeviceCollection -Name "Workstation Mgmt"

New-Item -Path '.\DeviceCollection\Workstation Mgmt' -Name "WKS_Update Management"
New-Item -Path '.\DeviceCollection\Workstation Mgmt' -Name "WKS_Software Management"
New-Item -Path '.\DeviceCollection\Workstation Mgmt' -Name "WKS_Settings Management"
New-Item -Path '.\DeviceCollection\Workstation Mgmt' -Name "WKS_OS Deployment"

New-Item -Path .\Application -Name "Workstation Mgmt"
New-Item -Path .\Package -Name "Workstation Mgmt"
New-Item -Path .\TaskSequence -Name "Workstation Mgmt"
New-Item -Path .\BootImage -Name "Workstation Mgmt"
New-Item -Path .\OperatingSystemInstaller -Name "Workstation Mgmt"
New-Item -Path .\OperatingSystemImage -Name "Workstation Mgmt"
New-Item -Path .\DriverPackage -Name "Workstation Mgmt"
New-Item -Path .\Driver -Name "Workstation Mgmt"
New-Item -Path .\ConfigurationItem -Name "Workstation Mgmt"
New-Item -Path .\ConfigurationBaseline -Name "Workstation Mgmt"
```

Beispiel: Collections




```
#All managed Workstations Collection
$dateMin = get-date -year 2016 -month 1 -day 1
$Sched = New-CMSchedule -RecurCount 1 -RecurInterval Days -Start (new-object datetime
(Get-Random -min $dateMin.ticks -max (Get-Date).ticks)).tostring()

$CollExclude = New-CMDeviceCollection -Name "All non-managed Workstations" -
LimitingCollectionId SMS00001 -Comment "All non-managed Workstations" -RefreshType None

$Coll = New-CMDeviceCollection -Name "All managed Workstations" -LimitingCollectionId
SMS00001 -Comment "All managed Workstations" -RefreshType Both -RefreshSchedule $Sched

Add-CMDeviceCollectionQueryMembershipRule -Collection $Coll -RuleName "All managed
Workstations" -QueryExpression 'select * from SMS_R_System where
SMS_R_System.OperatingSystemNameandVersion like "Microsoft Windows NT Workstation %"'

Add-CMDeviceCollectionExcludeMembershipRule -Collection $Coll -ExcludeCollection
$CollExclude

Invoke-CMDeviceCollectionUpdate -InputObject $Coll

Move-CMObject -FolderPath '.\DeviceCollection\Workstation Mgmt'  -InputObject $Coll
Move-CMObject -FolderPath '.\DeviceCollection\Workstation Mgmt'  -InputObject $CollExclude
```

Beispiel: Objekte exportieren

```
Get-Command export* -Module ConfigurationManager
```

Collections:

```
Get-CMDeviceCollection | ForEach { Export-CMDeviceCollection -Name $_.Name -ExportFilePath
"c:\Temp\$($_.Name).mof‟ -force }
```

Applications:

```
Get-CMApplication | ForEach { Export-CMApplication -Name $_.LocalizedDisplayName  -Path
"c:\temp\$($_.LocalizedDisplayName).zip" -OmitContent -IgnoreRelated -Force }
```

Boundaries (https://github.com/rzander/jaindb/tree/master/examples/ConfigMgr%20Backup):

```
Get-CMBoundary | ForEach-Object {
    $object = New-Object PSObject
    $id = "bip-" + $_.BoundaryID
    $bg = $_ | Select-Object $_.PropertyNames
    $object | Add-Member -MemberType NoteProperty -Name "Boundary" -Value $bg
    $object | convertto-json
}
```

AutoDoc: http://blog.cyberadvisors.com/export-all-those-sccm-settings-with-this-easy-script

Beispiel: Update Groups

https://rzander.azurewebsites.net/softwareupdate-deployment-with-sccm/

```
#Security Scope
New-CMSecurityScope -Name "Workstation Mgmt" -Description "Workstation Management"

#********** Update Groups **********
$UpdGrp = New-CMSoftwareUpdateGroup -Name "WKS_Test" -Description "WKS Phase 1"
Add-CMObjectSecurityScope -InputObject $UpdGrp -Name "Workstation Mgmt"
$UpdGrp = New-CMSoftwareUpdateGroup -Name "WKS_Pilot" -Description "WKS Phase 2"
Add-CMObjectSecurityScope -InputObject $UpdGrp -Name "Workstation Mgmt"
$UpdGrp = New-CMSoftwareUpdateGroup -Name "WKS_Wave1" -Description "WKS Phase 3"
Add-CMObjectSecurityScope -InputObject $UpdGrp -Name "Workstation Mgmt"
$UpdGrp = New-CMSoftwareUpdateGroup -Name "WKS_Wave2" -Description "WKS Phase 4"
Add-CMObjectSecurityScope -InputObject $UpdGrp -Name "Workstation Mgmt"
$UpdGrp = New-CMSoftwareUpdateGroup -Name "WKS_Rollup" -Description "WKS Phase 5"
Add-CMObjectSecurityScope -InputObject $UpdGrp -Name "Workstation Mgmt"
```

<Next session 00:00 – 00:00 uur>

<Title next session>

Mirko Colemberg

Experts Live Café