

Da die ConfigMgr PowerShell CmdLets im Standard nicht vom System Benutzer aus verwendet werden können und auch die geladenen Assemblies des Management Service das Laden der ConfigMgr CmdLets verhindern, habe ich ein Beispiel erstellt das zeigt wie man diese Hürde überwinden kann.

Das hier dargestellte Beispiel kann adaptiert werden, um die eigenen Szenarien durchzuführen und das, was bisher nicht direkt als Task in den Pipelines vorhanden ist mit eigenen Skripten zu verwirklichen. Voraussetzung: Die ConfigMgr Admin Console muss auf dem neo42 Management Service Server installiert sein.

1. Da der PowerShell Inline-Skript Task mit dem Computer Konto ausgeführt wird, muss der Management Service Server am ConfigManager Berechtigungen erhalten. In meinem Fall verwende ich die „Full Administrator“ Rolle.  
ConfigMgr Admin Console öffnen >
  - Administration
  - Security
  - Rechtsklick auf Administrative Users  
Add User or Group
  - User or Group name
  - Objekttypen
  - Computer anhaken
  - OK
  - Computernamen Angeben
  - OK
  - Add  
Full Administrator (oder die für Sie passende Rolle)
  - OK
2. Im ersten Schritt wird ein Verzeichnis benötigt, in dem das PowerShell Skript hinterlegt wird mit dem die ConfigMgr PowerShell CmdLets aufgerufen werden sollen.  
`New-Item -Path C:\neo42\ConfigMgrScripts -ItemType Directory`
3. Dort hinterlegen wir ein Skript mit dem Namen RenameCMApplication.ps1  
`New-Item -Path C:\neo42\ConfigMgrScripts\RenameCMApplication.ps1`

#### 4. Der Inhalt des Skripts RenameCMApplication.ps1.

```
param(
    $AppName,
    $ModelName,
    $SiteServer,
    $SiteCode
)

# Customizations
$initParams = @{}
#$initParams.Add("Verbose", $true) # Uncomment this line to enable verbose
logging
#$initParams.Add("ErrorAction", "Stop") # Uncomment this line to stop the script
on any errors

# Do not change anything below this line

# Import the ConfigurationManager.psd1 module
if((Get-Module ConfigurationManager) -eq $null) {
    Import-Module "$($ENV:SMS_ADMIN_UI_PATH)\..\ConfigurationManager.psd1"
}
@initParams

# Connect to the site's drive if it is not already present
if((Get-PSDrive -Name $SiteCode -PSProvider CMSite -ErrorAction
silentlyContinue) -eq $null) {
    New-PSDrive -Name $SiteCode -PSProvider CMSite -Root $SiteServer @initParams
}

# Set the current location to be the site code.
Set-Location "$($SiteCode):\\" @initParams

# Uncomment for Logging next to the Scriptfile
#Add-Content -Path "$PSScriptRoot\log.txt" -Value "$(Get-Date -Format
'[dd.MM.yyyy HH:mm:ss]'): Rename App with ModelID $($ModelName) to $($AppName)"
Set-CMApplication -ModelName $ModelName -NewName $AppName
```

#### 5. Der Pipeline wird nach dem „ConfigMgr Deploy“ Task der Task „Inline-Skript“ hinzugefügt mit folgendem Inhalt:

```
param(
    $AppName,
    $CI_UniqueID,
    $SiteServer,
    $SiteCode
)
$ModelName = "$(($CI_UniqueID -split ":")[0])/$(($CI_UniqueID -split ":")[1])"
Start-Process -PassThru -FilePath
"C:\windows\system32\windowsPowerShell\v1.0\powershell.exe" -ArgumentList "-
executionpolicy", "bypass", "-
File", "C:\neo42\ConfigMgrScripts\RenameCMApplication.ps1", "-
AppName", """"$AppName""", "-ModelName", "$ModelName", "-SiteServer", "$SiteServer", "-
SiteCode", "$SiteCode"
```

#### 6. Der Pipeline wird nach dem „ConfigMgr Deploy“ Task der Task „Inline-Skript“ hinzugefügt mit folgendem Inhalt, bitte dabei natürlich den eigenen SiteServer und SiteCode angeben:

```
-AppName "Prefix_<Run.Developer> <Run.Product> <Run.Version>" -CI_UniqueID
'<Run.ConfigMgrApplicationUniqueId>' -SiteServer "CM1.corp.contoso.com" -
SiteCode "CHQ"
```

[illegible]