

Citrix PVS Store – Replikationsskript V2

Im kürzlich erschienenen Artikel habe ich die erste Fassung des Replikationsskriptes vorgestellt. In der aktuell überarbeiteten Version habe ich einerseits die Ausgaben ein wenig „verschönert“ und vor allem werden die Stores nun direkt aus dem PVS ausgelesen.

Voraussetzungen dafür sind:

- Citrix PVS PowerShell Module an den Standardpfaden (kann in einer Variable angepasst werden)
- Der ausführende Benutzer muss entsprechende PVS Rechte besitzen
- Die Stores sollten verständliche Namen (Auswahl) und Beschreibungen (Hilfetexte bei Auswahl) vorweisen.

Zuerst wird nun also der Store Array dynamisch aus dem PVS ausgelesen. Voraussetzung dazu ist ein Laden des entsprechenden PVS PowerShell Moduls:

```
# Script variables
$PVSMODULEPATH = "C:\Program Files\Citrix\Provisioning
Services Console\Citrix.PVS.SnapIn.dll"

# Load PVS PowerShell module
Import-Module $PVSMODULEPATH

# Get PVS vDisk stores from PVS configuration
# Create an array for later use
$Stores = Get-PvsStore

$StoreArray = $null
$StoreArray = @()
$StoreID = 0
ForEach ($Store in $Stores){
    $StoreID = $StoreID +1
```

```

$SplitChar = $Store.Path.IndexOf(":")
$StoreDisk = $Store.Path.Substring(0,$SplitChar)
$StorePath = $Store.Path.Substring($SplitChar+1)
$StoreName = $Store.StoreName

$StoreArray +=
[pscustomobject]@{StoreID=$StoreID;Store=$StoreName;StoreDisk=
$StoreDisk;StorePath=$StorePath;StoreDescription=$Store.Descri
ption}
}

```

Nun wird aus dem Array eine entsprechende PowerShell Auswahl generiert, wobei hier aktuell auch eine Limite von maximal 5 Stores eingebaut ist:

```

# Prompt for Store choice
$StoreTitle = "PVS vDisk stores"
$StoreMessage = "Chose the store to replicate:"
$CancelAll = New-Object
System.Management.Automation.Host.ChoiceDescription "&Cancel",
"Skip this operation and all subsequent operations."

```

```

# Generate choice output of each vDisk store to replicate
$StoreOptionCount = 0
ForEach ($Store2 in $StoreArray) {
    $StoreOptionCount = $StoreOptionCount + 1
    $ChoiceID = $Store2.StoreID
    $ChoiceStore = $Store2.Store
    $ChoiceHelp = $Store2.StoreDescription
    $ChoiceCmd = New-Object
System.Management.Automation.Host.ChoiceDescription
"&$ChoiceID $ChoiceStore", $ChoiceHelp
    New-Variable "StoreOption$ChoiceID" $ChoiceCmd
}

```

```

# Generate $StoreOptions variable for final choice output
depending on the amount of options
Switch ($StoreOptionCount) {
    0 {$StoreOptions =
[System.Management.Automation.Host.ChoiceDescription[]]($Cance
lAll)}
    1 {$StoreOptions =
[System.Management.Automation.Host.ChoiceDescription[]]($Cance

```

```

lAll, $StoreOption1)}
                2      {$StoreOptions      =
[System.Management.Automation.Host.ChoiceDescription[]]($Cance
lAll, $StoreOption1, $StoreOption2)}
                3      {$StoreOptions      =
[System.Management.Automation.Host.ChoiceDescription[]]($Cance
lAll, $StoreOption1, $StoreOption2, $StoreOption3)}
                4      {$StoreOptions      =
[System.Management.Automation.Host.ChoiceDescription[]]($Cance
lAll, $StoreOption1, $StoreOption2, $StoreOption3,
$StoreOption4)}
                5      {$StoreOptions      =
[System.Management.Automation.Host.ChoiceDescription[]]($Cance
lAll, $StoreOption1, $StoreOption2, $StoreOption3,
$StoreOption4, $StoreOption5)}
}

```

Zu guter Letzt werden anhand der Auswahl die Variablen definiert:

```

# Prepare variables after the user entry
If ($Store2Replicate -eq 0) {
    # Cancel all
    return; break
}
Else {
    $IDStore = $Store2Replicate-1
    $StoreDisk = $StoreArray[$IDStore].StoreDisk
    $StorePath = $StoreArray[$IDStore].StorePath
    $LocalStore = $StoreDisk + ":" + $StorePath + "\"
    $RemoteStore = "\\\" + $PVSRemote + "\" + $StoreDisk + "$"
+ $StorePath + "\"
    $LocalStorePath = $LocalStore + "*"
    $RemoteStorePath = $RemoteStore + "*"
}

```

Der restliche Teil ist soweit gleich geblieben und im ersten Artikel bereits beschrieben.

Viel Spass beim Nachbauen :-)

ReplicatePVSStoresV2.ps1_Herunterladen