

# *Eine kleine Einführung in die Citrix PowerShell-Commandlets*

und wie man sich auf dieser Basis einfache eigene  
Administrationstools erstellen kann

# Kontakt

Andreas Nick

[Info@nick-it.de](mailto:Info@nick-it.de)

@NickInformation

Nick Inform

Dribusch 2

30539 Han

+49 511 165 810 190

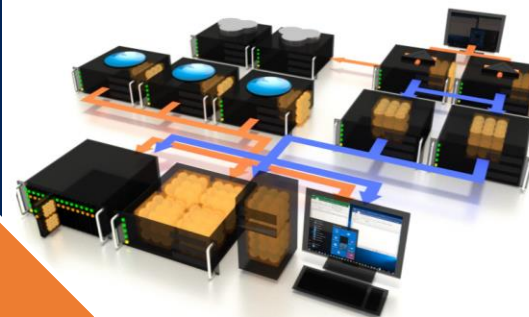
[www.nick-it.de](http://www.nick-it.de)

[www.software-virtualisierung.de](http://www.software-virtualisierung.de)

[www.AndreasNick.com](http://www.AndreasNick.com)

Und viel PowerShell !

Andreas Nick



Softwarevirtualisierung mit

**App-V 5**

Ein Praxisbuch

- App-V Infrastruktur und Hochverfügbarkeit
- Client und Sequencer umfassend erklärt
- App-V mit SCCM und mit Citrix XenDesktop
- Tools und Troubleshooting

# Warum eigene Tools?

- Schneller
- Für einen speziellen Anwendungsfall
- Neue Funktionen
- Erleichterung der Administration
- Dedizierte Administration

The screenshot displays the AppBot V1.3 interface, which includes several windows:

- AppBot V1.3 - Streaming Profiler to App-V**: Shows a file path and progress bar.
- NITCxBatcher**: Displays 'Found Limited Hotfixes: 20 Hotfixes: 7'.
- Delete Terminalserver**: A window for managing terminal servers.
- AppVClientGUI powered by Nick-IT**: The main dashboard showing a list of packages and their status.

The **AppVClientGUI** dashboard includes the following information:

- Last refresh**: HH:MM:SS
- Packages**: A list of installed packages with their status bars.

Package	Status
dotPeekSetup-1.1.1.33	Progress bar
Firefox57	Green bar
Gimp2.10	Green bar

The **Tag Tool** window shows a table of delivery groups and their associated servers:

Tag	Delivery Group	Server
ADM_NoLogon		
ADM_OutofOrder		
DSM_NoLogon		
RBT_FRI0400-14		
RBT_MON0400-14		
RBT_SUN0400-14		
RBT_TUE0400-14		
RBT_WED0400-14		
WG_InstallRunning		
WG_Reboot		
WG_xXAxxxx		

The **Tag Tool** window also displays a table of delivery groups and their associated servers:

Name	Betriebssystem	DT	PA	ADM	RBT	DSM	WG	JOB	OTH
SURANVDA01.uram.local	Windows 2016			NoLogon	SUN0400-14		InstallR		
				OutofOrder	TUE0400-14		xxAxxxx		
					WED0400-14				
					FRI0400-14				
					MON0400-14				
					SUN0400-14				
					TUE0400-14				
					WED0400-14				
uc1-001.uram.local					SUN0400-14		xxAxxxx		
uc1-002.uram.local					SUN0400-14		xxAxxxx		
uc1-003.uram.local					SUN0400-14		xxAxxxx		
uc1-004.uram.local					SUN0400-14		xxAxxxx		
uc1-005.uram.local					SUN0400-14		xxAxxxx		
uc1-006.uram.local					SUN0400-14		xxAxxxx		
uc1-007.uram.local					SUN0400-14		xxAxxxx		
uc1-008.uram.local					SUN0400-14		xxAxxxx		
uc1-009.uram.local					SUN0400-14		xxAxxxx		



# Installation der Citrix Commandlets

- Auf dem XenDesktop Datenträger
- XA717\x64\Citrix Desktop Delivery Controller\...
- Trust\_PowerShell !!!

	Orchestration_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	Broker_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	AppLibrary_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	Storefront_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	MachineCreation_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	ADIdentity_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	Monitor_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	Configuration_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	Analytics_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	Host_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	DelegatedAdmin_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	Trust_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	ConfigurationLogging_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	EnvTest_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket
	UserProfileManager_PowerShellSnapIn_x64.msi	Z:\packages\Citrix\XA717\x64\Citrix Desktop Deliver...	Typ: Windows Installer-Paket

# Einbinden der Citrix Commandlets

- Import der Citrix Commandlets
- PowerShell-Snapins sind Binärdateien (.dll), die Cmdlets und Provider implementieren. In diesem Fall für Citrix

```
Add-PSSnapIn Citrix.*
```

```
Add-PSSnapIn Citrix.Broker.*
```

- Wie werden die gefunden? Über den PSModulepath

```
PS C:\Users\Administrator> $env:PSModulePath
C:\Users\Administrator\Documents\WindowsPowerShell\Modules;C:\Program Files\WindowsPowerShell\Modules;C:\Windows\system32\WindowsPowerShell\v1.0\Modules;C:\Program Files (x86)\Microsoft SQL Server\110\Tools\PowerShell\Modules\;C:\Program Files\Citrix\Telemetry Service\C:\Program Files\Citrix\Broker\Modules\C:\Program Files\Citrix\XenDesktopPoshSdk\Module\itrix.XenDesktop.Admin.V1\
PS C:\Users\Administrator>
PS C:\Users\Administrator>
```

# Dokumentation und Hilfe

## Get-Command Citrix\*

```
PS C:\Users\Administrator> Get-Command -Module Citrix*
```

CommandType	Name	Version	Source
-----	----	-----	-----
Function	Get-CtxIcon	1.0	Citrix.Common.Commands
Cmdlet	Add-AcctADAccount	7.17.0.20	Citrix.ADIdentity.Admin.V2
Cmdlet	Add-AcctIdentityPoolScope	7.17.0.20	Citrix.ADIdentity.Admin.V2
Cmdlet	Add-AdminPermission	7.17.0.20	Citrix.DelegatedAdmin.Admin.V1
Cmdlet	Add-AdminRight	7.17.0.20	Citrix.DelegatedAdmin.Admin.V1
Cmdlet	Add-AppLibAppDiskScope	7.17.0.21	Citrix.AppLibrary.Admin.V1
Cmdlet	Add-AppLibIsolationGroupPackage	7.17.0.21	Citrix.AppLibrary.Admin.V1
Cmdlet	Add-AppLibTaskMetadata	7.17.0.21	Citrix.AppLibraryv.Admin.V1

Beispiel:

```
Get-Help Get-BrokerMachine
```

```
Get-Help Get-BrokerMachine -example
```

Im Web docs.citrix.com:

<https://docs.citrix.com/de-de/xenapp-and-xendesktop/7/cds-reference-wrapper-rho/cds-sdk-wrapper-rho/cds-sdk-cmdlet-help.html>

# Admin Address

- Jeder Citrix Posh Befehl kennt den Parameter „-AdminAddress“
- Mit den Citrix PowerShell Modulen somit auch remote einsetzbar

-AdminAddress <DDC-FQDN>:Port

## Beispiel

-AdminAddress suranddc01.uran.local:443

# Einfache Befehle

- Einfache Befehle
- Get-BrokerCatalog
- Get-BrokerMachine
- Get-BrokerApplication
- Get-BrokerDesktopGroup
- (Get-BrokerDesktopGroup).name
- Get-BrokerSession
- Disconnect-BrokerSession
- Stop-BrokerSession



# Etwas komplexer

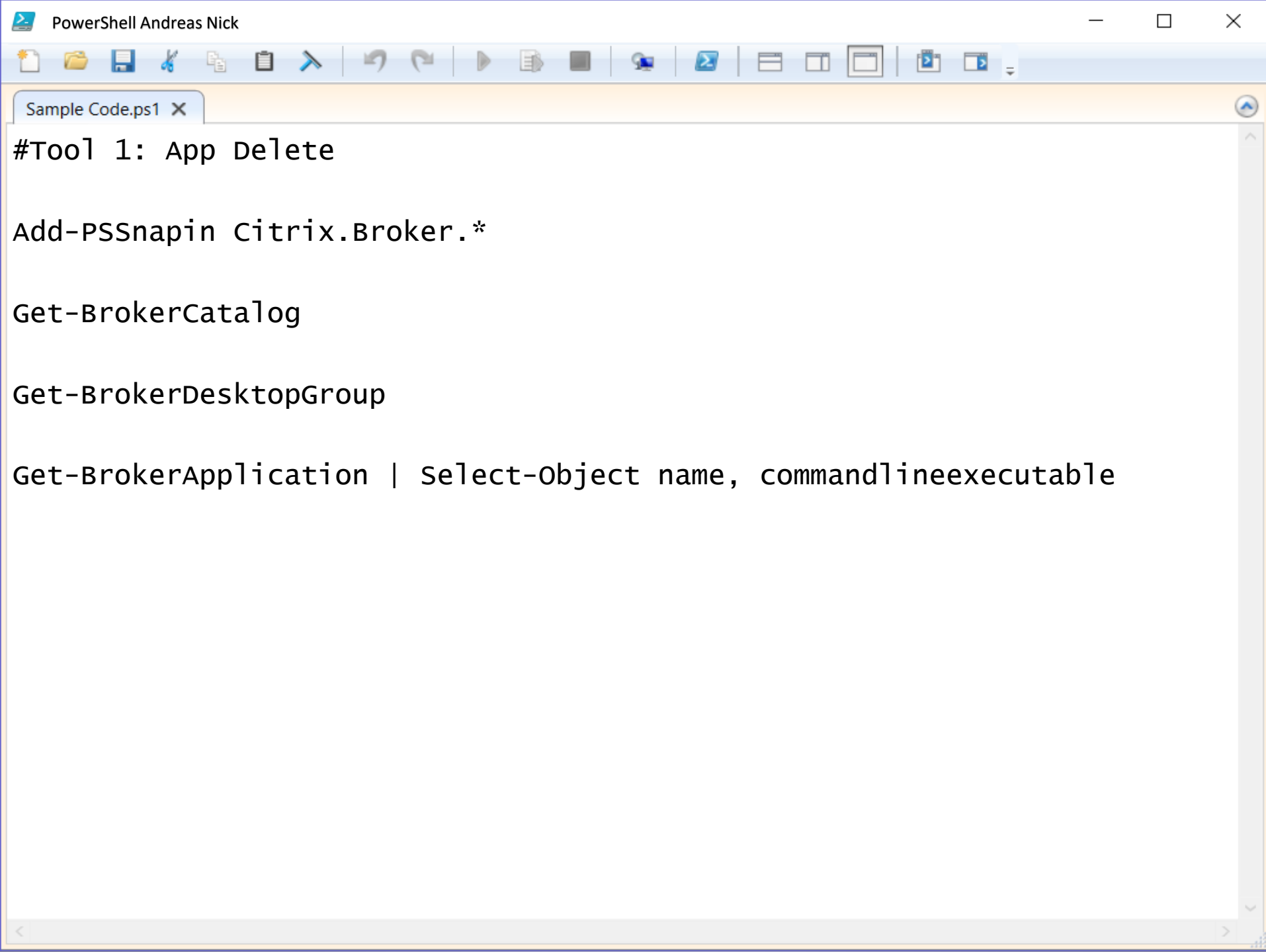
- Anwendungen einer Delivery Group auflisten

```
$UUID = (Get-BrokerDesktopGroup -Name  
$name ).UUID
```

```
$apps = Get-BrokerApplication -  
AssociatedDesktopGroupUUID $UUID |  
Select-Object -Property Name, IconUid
```



# Einfache Beispiele One-Liner (Demo)



```
#Tool 1: App Delete

Add-PSSnapin Citrix.Broker.*

Get-BrokerCatalog

Get-BrokerDesktopGroup

Get-BrokerApplication | select-object name, commandlineexecutable
```

Sample Code.ps1 X

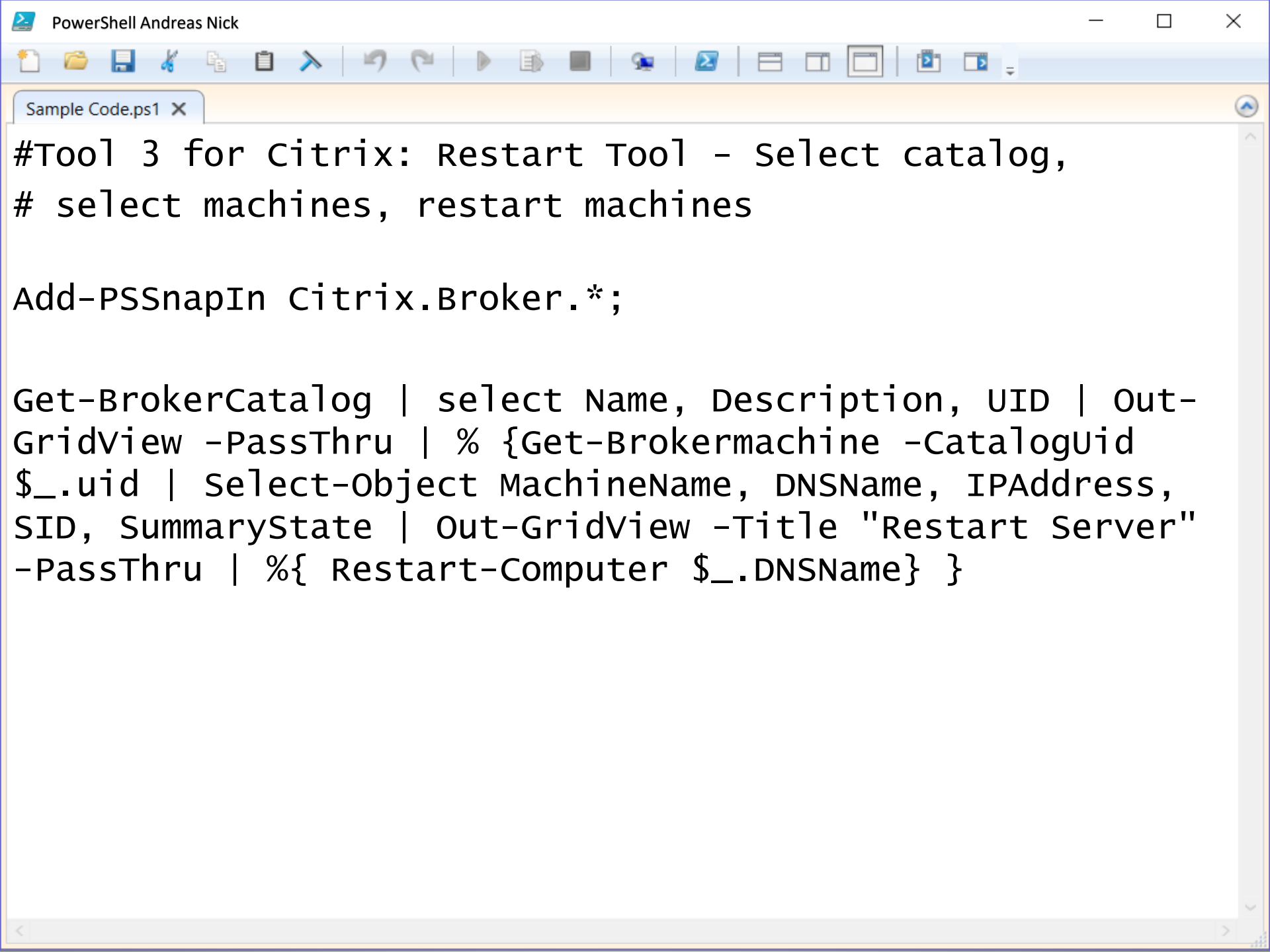
```
#Tool 2: #Machines Update GPO and Tags
```

```
Add-PSSnapin Citrix.Broker.*
```

```
Get-BrokerMachine
```

```
Get-BrokerMachine | select MachineName, IPAddress,  
SummaryState | Out-GridView
```

```
get-brokmachine | select MachineName, DNSName,  
IPAddress, SummaryState | Out-GridView -PassThru -Title  
"Machine update Tool" | % { Invoke-Command -ComputerName  
$_.DNSName {gpupdate /force;Restart-Service  
BrokerAgent;}}
```



Sample Code.ps1 X

```
#Tool 3 for Citrix: Restart Tool - select catalog,  
# select machines, restart machines
```

```
Add-PSSnapIn Citrix.Broker.*;
```

```
Get-BrokerCatalog | select Name, Description, UID | Out-  
GridView -PassThru | % {Get-Brokmachine -CatalogUid  
$_.uid | Select-Object MachineName, DNSName, IPAddress,  
SID, SummaryState | Out-GridView -Title "Restart Server"  
-PassThru | %{ Restart-Computer $_.DNSName} }
```



Sample Code.ps1 X

```
# Tool 4 for Citrix: Kill User Session
```

```
Get-PSSnapIn Citrix.Broker.*; Get-BrokerSession | Out-GridView -PassThru | Stop-BrokerSession
```

```
Get-PSSnapIn Citrix.Broker.*; Get-BrokerSession -Username  
"domain\UsernameToLogoff" -DesktopGroupName  
„DeliveryGroup“ | Stop-BrokerSession
```

# Eingabemöglichkeiten

- Show-Command
- Show-Command Write-Host
- Eingabe einlesen und ausführen

```
Function x1 {param($Eingabetext) Write-Host
$Eingabetext};Show-Command x1 -PassThru | % {Invoke-
Expression $_ }
```

```
function x1 {param($Eingabetext,[switch] $Schalter)
Write-Host $Eingabetext $Schalter};Show-Command x1 -
PassThru | % {Invoke-Expression $_ }
```

Sample Code.ps1 X

#Man kann hier sehr leicht spagetticode erzeugen  
„ONE-LINER mit GUI zur DateeEingabe“

```
Get-PSSnapIn Citrix.Broker.*; function Get-delGroup  
{param($DeliveryGroup) ((Get-BrokerDesktopGroup -  
Name $DeliveryGroup ).UUID) }; Show-Command Get-delGroup -  
PassThru | % {Invoke-Expression " $_" } | % {Get-  
BrokerApplication -AssociatedDesktopGroupUUID $_  
}| Select-Object -Property Name, IconUid | out-gridview
```

```
Get-PSSnapIn Citrix.Broker.*;  
function Get-delGroup {param($DeliveryGroup) ((Get-  
BrokerDesktopGroup -Name $DeliveryGroup ).UUID) }  
Show-Command Get-delGroup -PassThru | % {  
Invoke-Expression " $_" } | % {  
Get-BrokerApplication -AssociatedDesktopGroupUUID $_  
}| Select-Object -Property Name, IconUid | out-gridview
```



# Mit WPF

Windows Presentation Foundation

# Nutzen der XAML in Powershell

- Übersetzen der XAML in XML
- Exportieren der XAML Variablen
- Füllen der Binding-Variablen
- UI anzeigen
- Oder: ISE-Steroids und Visual Studio

```
$xaml = @"
<Window
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  MinWidth="400"
  MinHeight="400"
  Width="400"
  Height="400"
  ResizeMode="CanResizeWithGrip"
  Title="Update GPO and Citrix GPO"
  Topmost="True">
  <Grid Margin="10,40,10,10">
    <Grid.ColumnDefinitions>
      <ColumnDefinition width="Auto"/>
      <ColumnDefinition width="*/>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
      <RowDefinition Height="Auto"/>
      <RowDefinition Height="*/>
      <RowDefinition Height="50"/>
    </Grid.RowDefinitions>
    <TextBlock Grid.Column="1" Margin="10">
      <TextBlock Grid.Column="0" Grid.Row="1">
        <DataGrid Name="ComboService" Grid.Column="0" Grid.Row="1">
          <StackPanel Orientation="Horizontal">
            <Button Name="ButOk" MinWidth="80">OK</Button>
            <Button Name="ButCancel" MinWidth="80">Cancel</Button>
          </StackPanel>
        </DataGrid>
      </TextBlock>
    </TextBlock>
  </Grid>
</Window>
"@
```



# Erstellung der Datenbindungen

- get-brookermachine
- Datenbindung (Get-Brookermachine)
- Befehlsausführung (Schalter)
- Get-Brookermachine liefert ein Array mit PowerShell Objekten
- `get-brookermachine | select MachineName, DNSName, IPAddress, SummaryState`

# Aktion

## Binden von Schaltfunktionen

```
$window = Convert-XAMLtoWindow -XAML $xaml -NamedElement
'ButCancel', 'ButOk', 'ComboService' -PassThru
```

```
function Update-Server{
param ($DnsName)
    Invoke-Command -ComputerName $DNSName {
        gpupdate /force
        Restart-Service BrokerAgent -Verbose
    } -Verbose | Out-Host
}

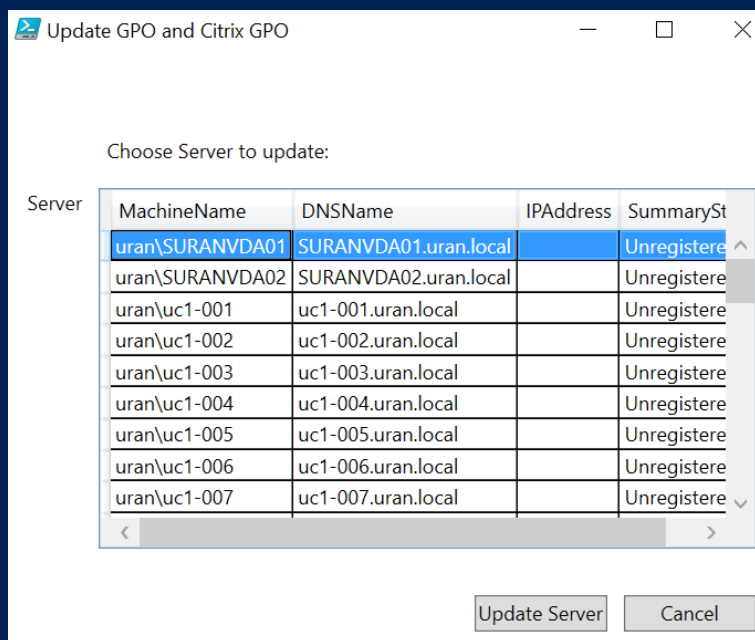
#Own Code here
$window.ButOk.add_Click{
    #Action
    $window.ComboService.SelectedItem | % {Write-Host $("GPUUpdate "+$_.DNSName) -ForegroundColor Yellow }
    $window.ComboService.SelectedItem | % { Update-Server -DNSName $_.DNSName }
    Write-Host "Set ItemSource" -ForegroundColor Yellow
    $window.ComboService.ItemsSource = Get-BrokerMachine | select MachineName, DNSName, IPAddress, SummaryState
}

$window.ButCancel.add_Click{
    $window.DialogResult = $false
}
```



Demo

# Resultat: Ein einfaches schnelles Tool



```
PS C:\Users\Administrator\Desktop> .\GPOUpdate.ps1 -ServerName SURANVDA01.uran.local
Updating policy...
```

```
Computer Policy update has completed successfully.
```

```
User Policy update has completed successfully.
```

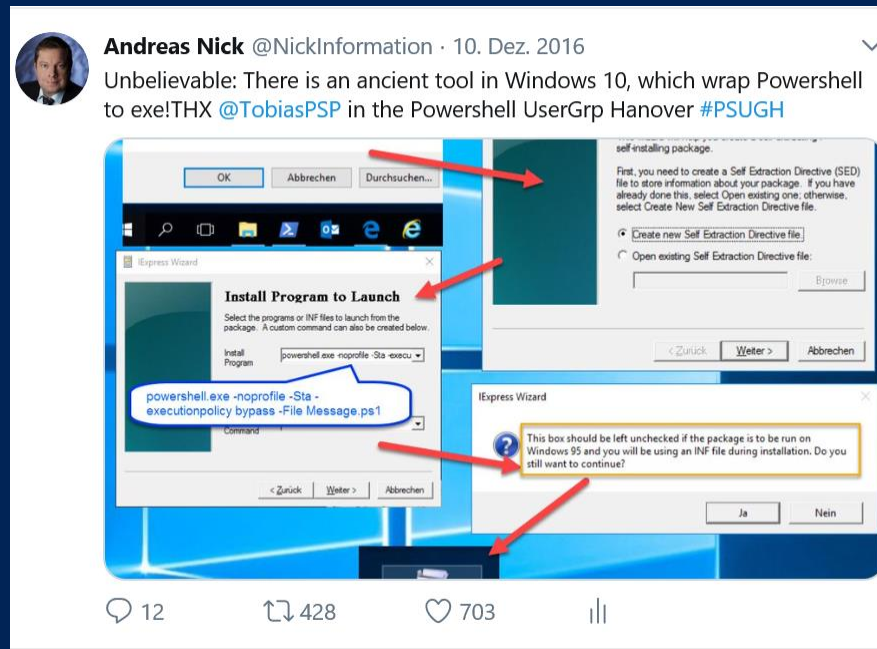
```
VERBOSE: Performing the operation "Restart-Service" on target "Citrix Desktop Service (BrokerAgent)".
Set ItemSource
```

# Bonus:

## Mit Bordmitteln PowerShell 2 Exe?

Geht  
nicht?

Geht!

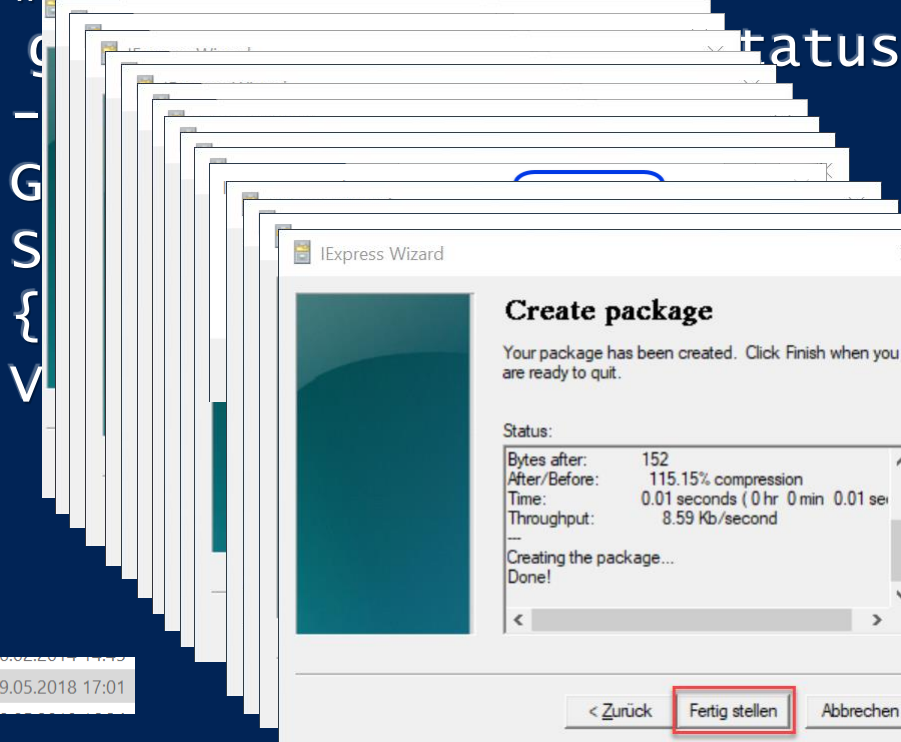




# Mit iexpress! Aus PowerShell eine Exe

- Skript erstellen
- Powershell als Admin und „iexpress“ starten
- Den Workflow durchlaufen
- Exe starten

```
#restarterTool.ps1
```





Fragen?