

LotusScript und PowerShell – das beste aus beiden Sprachen kombinieren (2022 Edition)

Geht nicht? Geht ja wohl!

Déjà vu zu 2021?

Domino und PowerShell können gute Freunde sein

Auch als Domino Admin (oder auch als Benutzer) kann man PowerShell sehr sinnvoll integrieren – ich zeige euch, wie einfach das mit LotusScript und PowerShell gehen kann.

Andreas @ DominoCamp '21

PowerShell und Domino können gute Freunde sein

22.06.2021

Über mich

- 2002 – erste Notes Dev Erfahrung (R5)
- Ab 2004 dazu Administration
- 2008 kam MS Exchange dazu
- Seit 2013 Fokus Microsoft Cloud
 - Modern Workplace Architekt
 - Cloud Security
- ❤️ PowerShell ❤️
- Seit 2021 selbständig

andreas.haehnel@blackmagiccloud.de



[@nisnaju](https://twitter.com/nisnaju)



<https://m365evangelists.com>

<https://blackmagiccloud.de>



<https://github.com/cloudkobold>

Wieso ist das interessant?

Wieso ist das interessant?

- >99% aller IT Landschaften sind inhomogen
 - OS: Windows XP – Windows 11 / macOS / Linux
 - Handy: Android / iOS / BlackBerry
 - Verzeichnisdienste: Domino Directory / Active Directory / Azure Active Directory / Bind / AD LDS / “beliebiges” IDM System / ...
- Administratoren müssen alle Systeme (+Versionen) beherrschen
 - PROBLEM: Fehleranfälligkeit
 - PROBLEM: Einarbeitung neuer Kollegen
 - PROBLEM: Es gibt keine Software, die dieses Problem löst

Wieso ist das interessant?

- Darum: selbst ist der Mann / die Frau!
- Alle immer gleichen Arbeitsabläufe / Prozesse standardisieren
- Ein Administrations-Cockpit erstellen, das “alles” kann
- Fall 1: Ich administrierte viel Microsoft-lastig – geht das aus Notes heraus?
- Fall 2: Ich bin ein gottgleicher PowerShell Scripter – wie automatisiere ich meine Notes Admin Aufgaben?

Fall 1: aus Notes heraus ein Microsoft System administrieren

Fall 1: Voraussetzungen

- Notes Client natürlich
- Notes ID mit entsprechenden Rechten auf der jeweiligen (Admin-)DB
- AD Account (oder Systemadmin) mit entsprechenden Rechten
- Domino Designer
- Aktuelle PowerShell (mindestens 5.1)
- LotusScript Kenntnisse
- PowerShell Kenntnisse

Fall 1: HowTo

- Grundwissen “wie funktioniert ein PowerShell Script”
 - In erster Linie eine Textdatei
 - Diese hat die Endung .ps1
 - Und diese muss dann ausgeführt werden

Fall 1: HowTo

```
Dim s as New NotesSession  
[...]
```

```
ExchOServiceUser = „DominoService@myTenant.onmicrosoft.com“  
ExchOServicePW = „UncrackablePassword1“
```

```
[...]
```

```
Set FSOPS = CreateObject("Scripting.FileSystemObject")  
Set PSScript = FSOPS.CreateTextFile(fileNamePS, True, False)
```

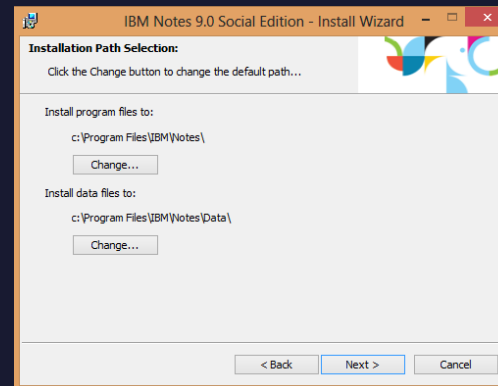
```
PSScript.WriteLine {Import-Module ExchangeOnlineManagement}  
PSScript.WriteLine {Set-Mailbox john -ProhibitSendQuota 48GB}
```

```
cmd = {C:\windows\systow64\windowsPowerShell\v1.0\powershell.exe -command "& } & fileNamePS & {"}  
retval = Shell(cmd,1)
```

Fall 2: aus PowerShell heraus Notes administrieren

Fall 2: Voraussetzungen

- ID mit entsprechenden Rechten
- Aktuelle PowerShell (mindestens 5.1)
- Notes Client (Single User Installation)

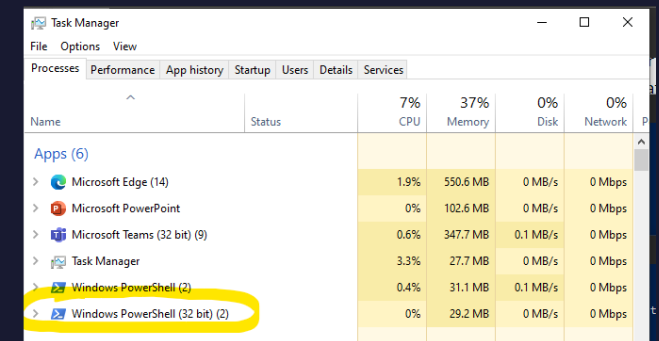


Fall 2: Voraussetzungen

- Der Installer von Notes installiert ein OLE Objekt auf dem Client
 - Windows spezifisch
 - OLE = object linking and embedding (ein Standard, den Microsoft ins Leben gerufen hat)
 - Über OLE kann ich Daten aus Programmen verknüpfen, die normalerweise keine Schnittstelle haben

Fall 2: HowTo

- Grundwissen “wie funktioniert ein OLE Objekt”
 - Anmeldung mit Notes ID & Kennwort
 - Das eigene Programm / Script agiert als Notes Client
 - Das eigene Programm / Script agiert dann mit den Rechten dieser Notes ID
- PowerShell muss als 32-Bit Prozess laufen
 - Wichtig, falls ein Script mal per Task Scheduler regelmäßig laufen soll
 - ISE / VSCode gibt es auch als 32 bit Version
 - %SystemRoot%\syswow64\WindowsPowerShell\v1.0\powershell.exe



Name	Status	CPU	Memory	Disk	Network
Apps (6)					
Microsoft Edge (14)		1.9%	550.6 MB	0 MB/s	0 Mbps
Microsoft PowerPoint		0%	102.6 MB	0 MB/s	0 Mbps
Microsoft Teams (32 bit) (9)		0.6%	347.7 MB	0.1 MB/s	0 Mbps
Task Manager		3.3%	27.7 MB	0 MB/s	0 Mbps
Windows PowerShell (2)		0.4%	31.1 MB	0.1 MB/s	0 Mbps
Windows PowerShell (32 bit) (2)		0%	29.2 MB	0 MB/s	0 Mbps

Fall 2: HowTo

```
$DominoServer = "MyWonderfulDomino/MyGreatCompany"
```

```
$NotesSession = New-Object -ComObject Lotus.NotesSession
```

```
$NotesSession.Initialize("")
```

```
#oder: $NotesSession.Initialize("mySecretPassword")
```

```
$notesAddressbook = `
$NotesSession.GetDatabase($DominoServer, "names.nsf", 0)
```

Genug Theorie – nun zur Praxis

Die Demoumgebung (Teil 1/2)

- Demos laufen auf Notes 11 mit Domino 10
- PowerShell Version 5.1
- Nicht Best-Practice (hier nur der Einfachheit halber):
 - Der Pfad auf der Festplatte muss existieren
 - Lokale Adminrechte um UAC zu umgehen
 - PowerShell ExecutionPolicy Unrestricted
 - Benutzerparameter wie Initialkennwort und OU sind „hard-coded“

Die Demoumgebung (Teil 2/2)

- PowerShell Modul **ExchangeOnlineManagement**:

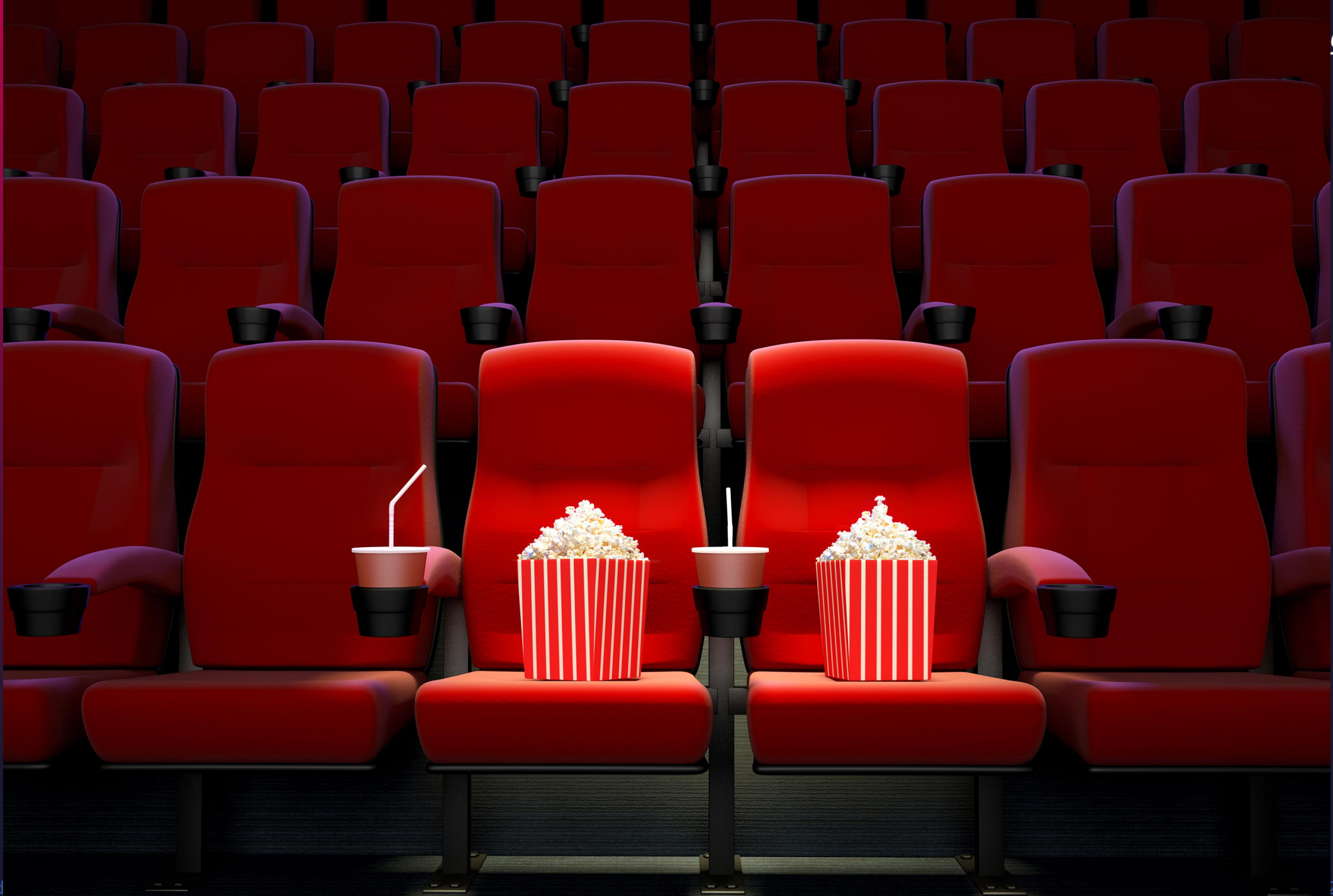
<https://www.powershellgallery.com/packages/ExchangeOnlineManagement>

```
Install-Module -Name ExchangeOnlineManagement -RequiredVersion 2.0.5
```

- PowerShell Modul **ActiveDirectory** (RSAT):

[Download Remote Server Administration Tools for Windows 10 from Official Microsoft Download Center](#)

- PowerShell Modul Exchange 2016 (als Remote PS)



Und jetzt – wie geht es weiter?

- Erstellt euer eigenes Admin-Cockpit und spart euch viel Zeit
 - Exchange
 - AD / AAD
 - Server DSC
 - MS Teams Einstellungen
 - Komplette Benutzerverwaltungsprozesse
- Ich unterstütze gerne (auch punktuell) – einfach melden!

andreas.haehnel@blackmagiccloud.de