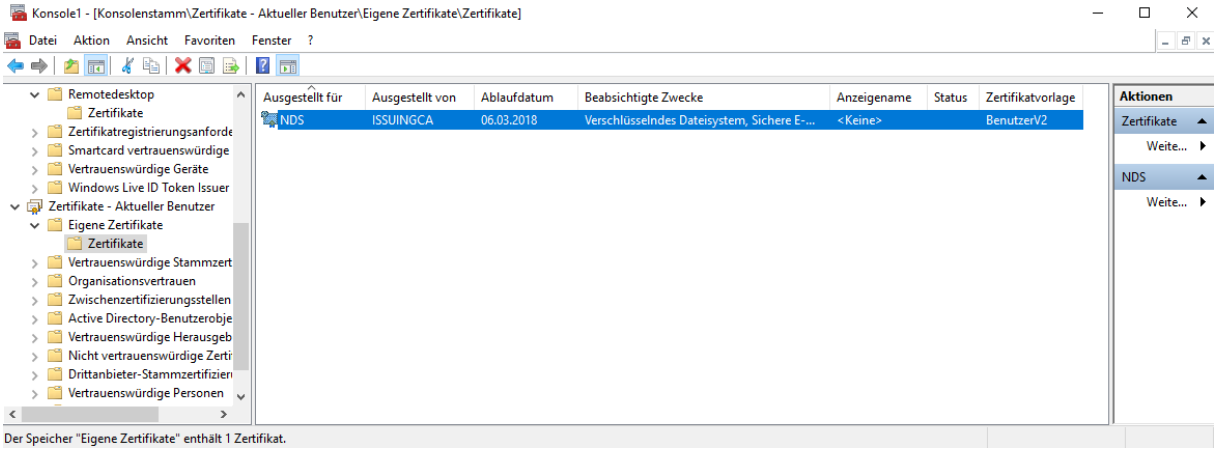
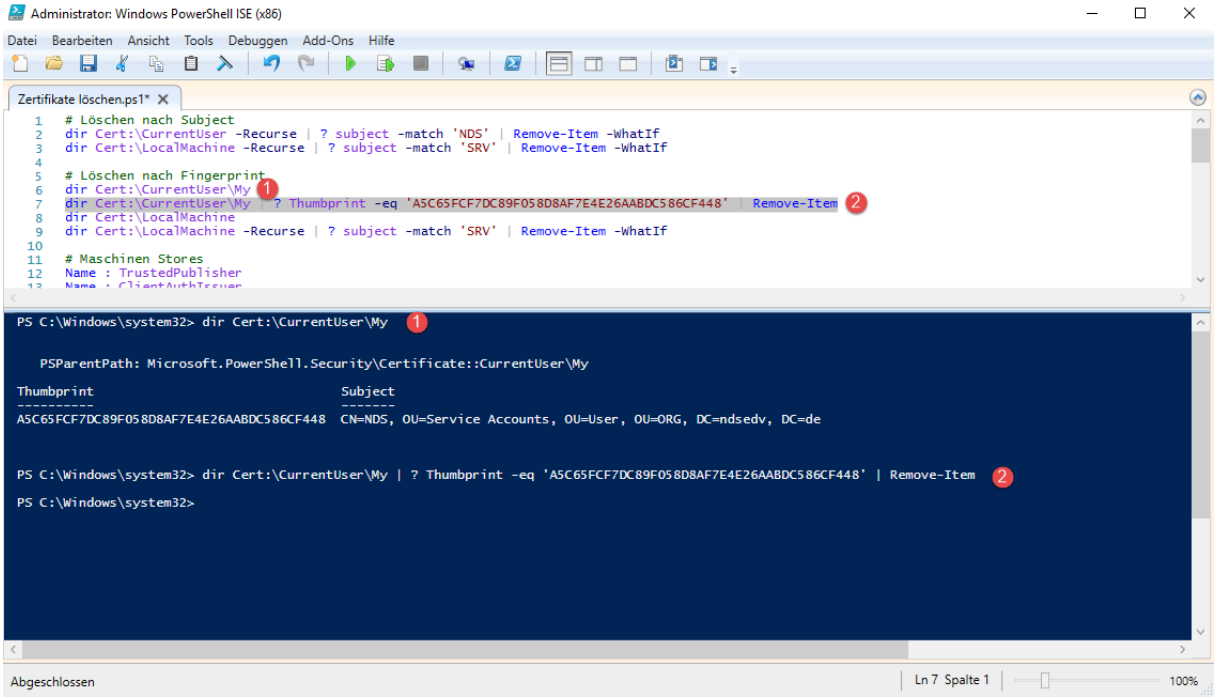
Löschen und importieren von Zertifikaten

Löschen eines Zertifikats mithilfe der Powershell. Das Ziel ist die Löschung des Benutzer-Zertifikats namens „NDS".



(1) Fingerabdruck ermitteln (2) Einfügen und Löschen



Hier ein paar Beispiele:

```
# Löschen nach Subject
dir Cert:\CurrentUser\My
dir Cert:\CurrentUser -Recurse | ? subject -match 'NDS' | Remove-Item -WhatIf
dir Cert:\LocalMachine -Recurse | ? subject -match 'SRV' | Remove-Item -WhatIf

# Löschen nach Fingerabdruck
dir Cert:\CurrentUser\My
dir Cert:\CurrentUser\My | ? Thumbprint -eq
'A5C65FCF7DC89F058D8AF7E4E26AABDC586CF448' | Remove-Item
dir Cert:\LocalMachine\My
dir Cert:\LocalMachine -Recurse | ? subject -match 'SRV' | Remove-Item -WhatIf

# Zertifikat Details auslesen
Set-Location Cert:\CurrentUser\My
Get-ChildItem | Format-Table Subject, Serialnumber, Thumbprint -AutoSize

# Fingerabdruck auslesen
$Thumbprint = (Get-ChildItem -Path Cert:\LocalMachine\My | Where-Object {$_.Subject
-match "NDS"}).Thumbprint;
Write-Host -Object "Mein Fingerabdruck lautet: $Thumbprint";
```

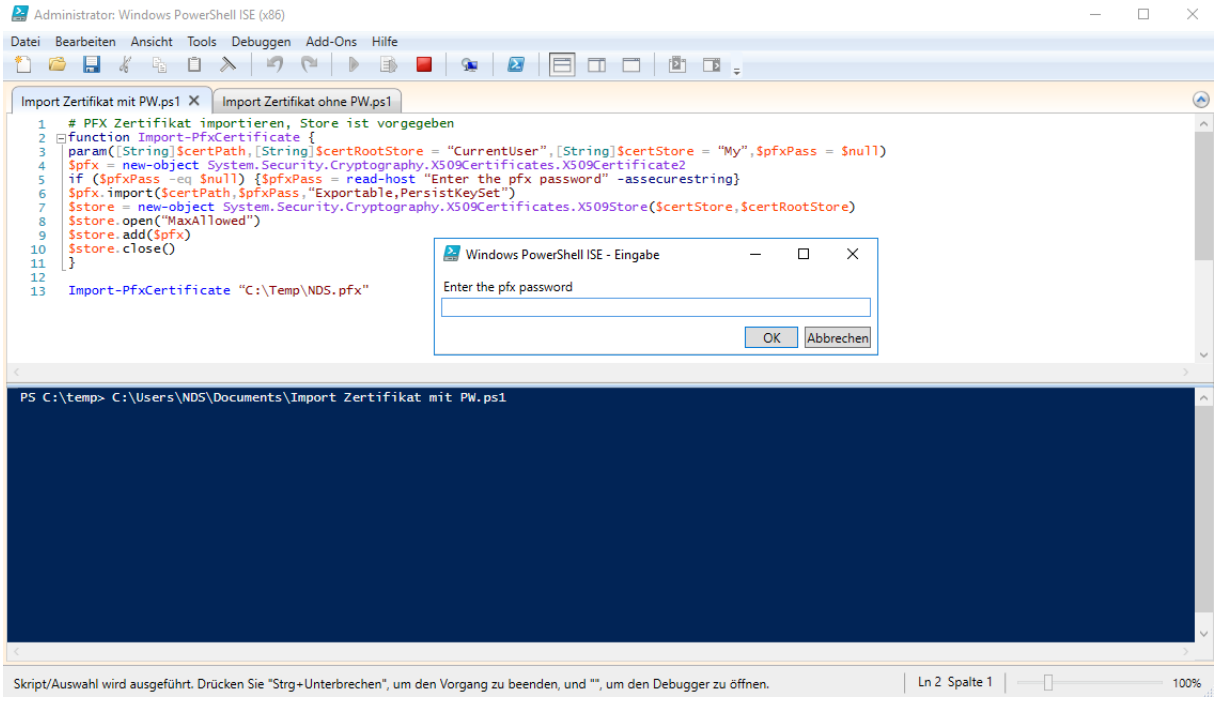
Erstellt von Jörn Walter www.der-windows-papst.de 06.03.2017

Weitere Beispiele:

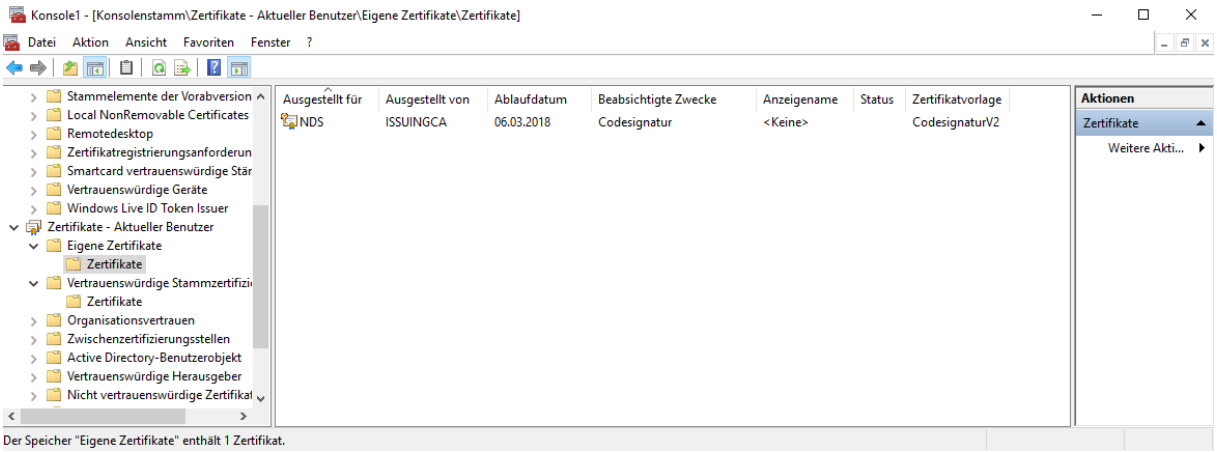
```
Get-ChildItem cert:LocalMachine\My
Get-ChildItem cert:LocalMachine\My | Select *

# Löschen nach Fingerprint
Get-ChildItem Cert:\LocalMachine\My\4191a97f0922d8a41930dfe230e65ee0e77c10e3 |
Remove-Item

# Löschen nach Subject
Get-ChildItem Cert:\LocalMachine\My |
Where-Object { $_.Subject -match 'SRV01.ndsedv.de' } |
Remove-Item

# Löschen nach Serialnumber
Get-ChildItem Cert:\LocalMachine\My |
Where-Object { $_.Serialnumber -match '1d0000002ac5eac60e9974a98500000000002a'
} |
Remove-Item

# Löschen nach Issuer
Get-ChildItem Cert:\LocalMachine\My |
Where-Object { $_.Issuer -match 'ISSUINGCA' } |
Remove-Item

Subject: SRV01.ndsedv.de
Serial: 1d0000002ac5eac60e9974a98500000000002a
Issuer: ISSUINGCA
Fingerprint: 4191a97f0922d8a41930dfe230e65ee0e77c10e3

# Löschen nach Fingerprint aus allen Stores
 Get-ChildItem cert: -Recurse |?{$_.Thumbprint -match
"4191a97f0922d8a41930dfe230e65ee0e77c10e3"}

Get-ChildItem cert: -Recurse |?{$_.Thumbprint -match
"4191a97f0922d8a41930dfe230e65ee0e77c10e3"} | Remove-Item

# Löschen nach Fingerprint aus der root (Vertrauenswürdige Stammzertifikatsstellen)
 Get-ChildItem cert:LocalMachine\root
Get-ChildItem cert:LocalMachine\root | Select *

Get-ChildItem cert: -Recurse |?{$_.Thumbprint -match
"4191a97f0922d8a41930dfe230e65ee0e77c10e3"} | Remove-Item
```

Löschen und importieren von Zertifikaten

PFX Zertifikat mithilfe einer Funktion importieren. Der Store wird vorgegeben in diesem Fall „CurrentUser" kann aber auch in „LocalMachine" geändert werden.



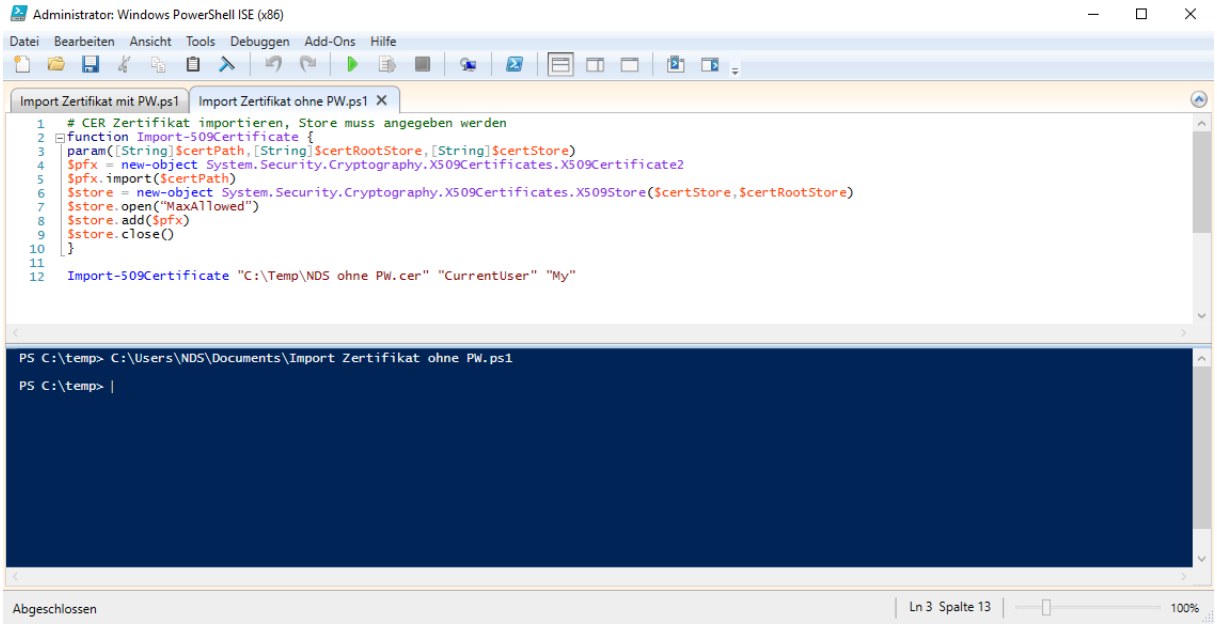
Passwort eingeben und OK klicken, Store aktualisieren.

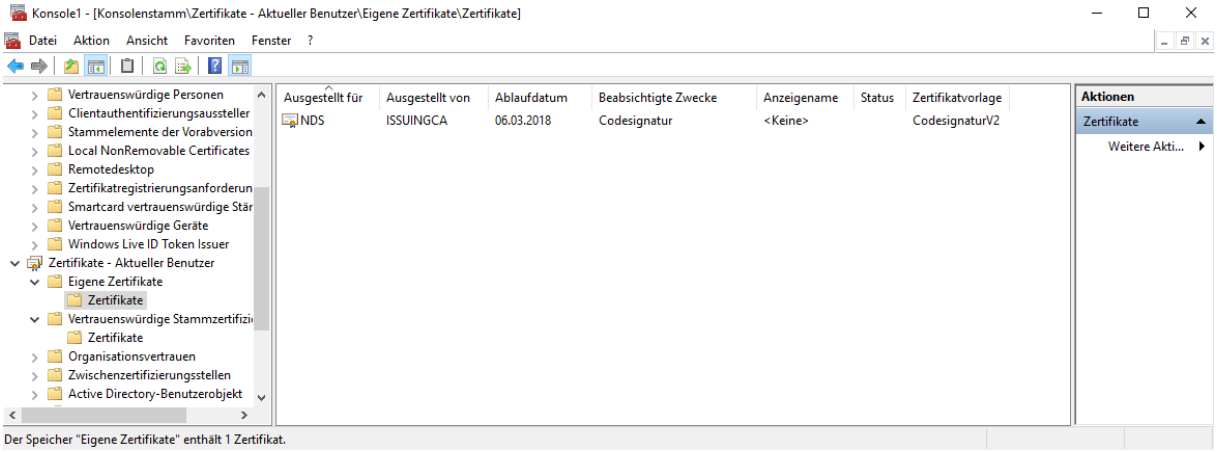



```
# PFX Zertifikat importieren, Store ist vorgegeben, Passwortabfrage
function Import-PfxCertificate {
param([String]$certPath,[String]$certRootStore = "CurrentUser",[String]$certStore =
"My",$pfxPass = $null)
$pfx = new-object System.Security.Cryptography.X509Certificates.X509Certificate2
if ($pfxPass -eq $null) {$pfxPass = read-host "Enter the pfx password" -
assecurestring}
$pfx.import($certPath,$pfxPass,"Exportable,PersistKeySet")
$store = new-object
System.Security.Cryptography.X509Certificates.X509Store($certStore,$certRootStore)
$store.open("MaxAllowed")
$store.add($pfx)
$store.close()
}

Import-PfxCertificate "C:\Temp\NDS.pfx"
```

Löschen und importieren von Zertifikaten

Zertifikat importieren; der Store muss der Funktion übergeben werden.






```
# CER Zertifikat importieren, Store muss angegeben werden
function Import-509Certificate {
param([String]$certPath,[String]$certRootStore,[String]$certStore)
$pfx = new-object System.Security.Cryptography.X509Certificates.X509Certificate2
$pfx.import($certPath)
$store = new-object
System.Security.Cryptography.X509Certificates.X509Store($certStore,$certRootStore)
$store.open("MaxAllowed")
$store.add($pfx)
$store.close()
}

Import-509Certificate "C:\Temp\NDS ohne PW.cer" "CurrentUser" "My"
Import-509Certificate "C:\Temp\NDS ohne PW.cer" "LocalMachine" "My"
```

Löschen und importieren von Zertifikaten

 # Maschinen Stores
Name : TrustedPublisher
Name : ClientAuthIssuer
Name : Remote Desktop
Name : Root
Name : TrustedDevices
Name : CA
Name : Windows Live ID Token Issuer
Name : REQUEST
Name : AuthRoot
Name : FlightRoot
Name : TrustedPeople
Name : Local NonRemovable Certificates
Name : My
Name : SmartCardRoot
Name : Trust
Name : Disallowed

 # User Stores
Name : TrustedPublisher
Name : ClientAuthIssuer
Name : Root
Name : UserDS
Name : CA
Name : ACRS
Name : REQUEST
Name : AuthRoot
Name : TrustedPeople
Name : My
Name : SmartCardRoot
Name : Trust
Name : Disallowed