

## PowerShell mit ABAP nutzen

PowerShell ist eine sehr leistungsfähige Skriptsprache. Sie wird im Standard seit Windows 7 ausgeliefert. Aktuell ist Windows 10 mit PowerShell 5 ausgestattet. Der SAP GUI für Windows bietet eine [COM-Schnittstelle](#) die von ABAP aus verwendet werden kann. Mittels dieser Schnittstelle, und einer Bibliothek von [SAPIEN](#), besteht die Möglichkeit das PowerShell auf dem Frontend-Server von ABAP genutzt werden kann. In diesem Beitrag soll kurz eine Realisierung mit einigen Anwendungsfällen umrissen werden.

Als erstes beschaffen wir uns die notwendigen Bibliotheken von [SAPIEN](#). Einfach im Bereich My Account, nach dem Login, Download auswählen und die Bibliotheken herunterladen.



Nach der Installation und Registrierung können diese sofort mit ABAP verwendet werden. Für die Registrierung einfach eine Konsole im Administratormodus öffnen und die folgenden Befehle ausführen.

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\regasm.exe "C:\Program Files (x86)\SAPIEN Technologies, Inc\ActiveXPowerShell V3\ActiveXPoshV3.dll" /codebase
```

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\regasm.exe "C:\Program Files\SAPIEN Technologies, Inc\ActiveXPowerShell V3\ActiveXPoshV3.dll"
```

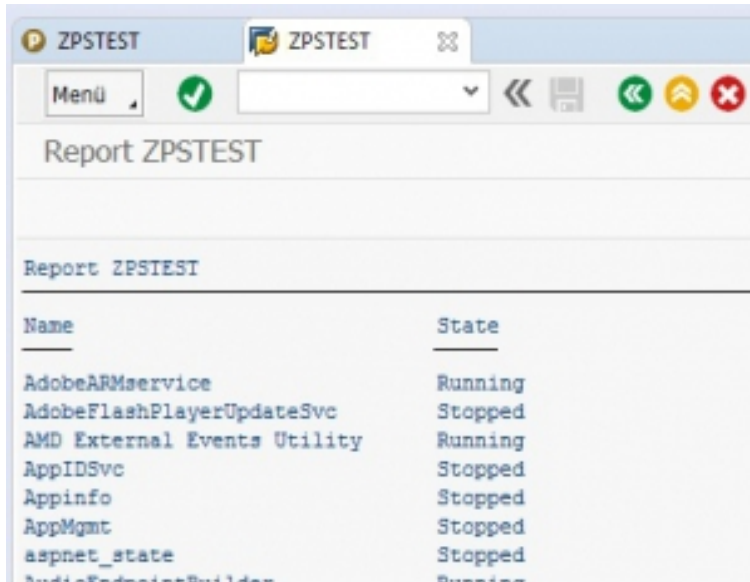
/codebase

Hier als Beispiel der Aufruf aller installierten Services mit ihrem Status.

[crayon-5b093461dfc7c388012467/]

In der Variablen cmd werden die PowerShell-Befehle übergeben, in diesem Fall die Ausgabe einer formatierten Tabelle mit Name und Status aller Services über das Windows Management Instrumentarium (WMI):

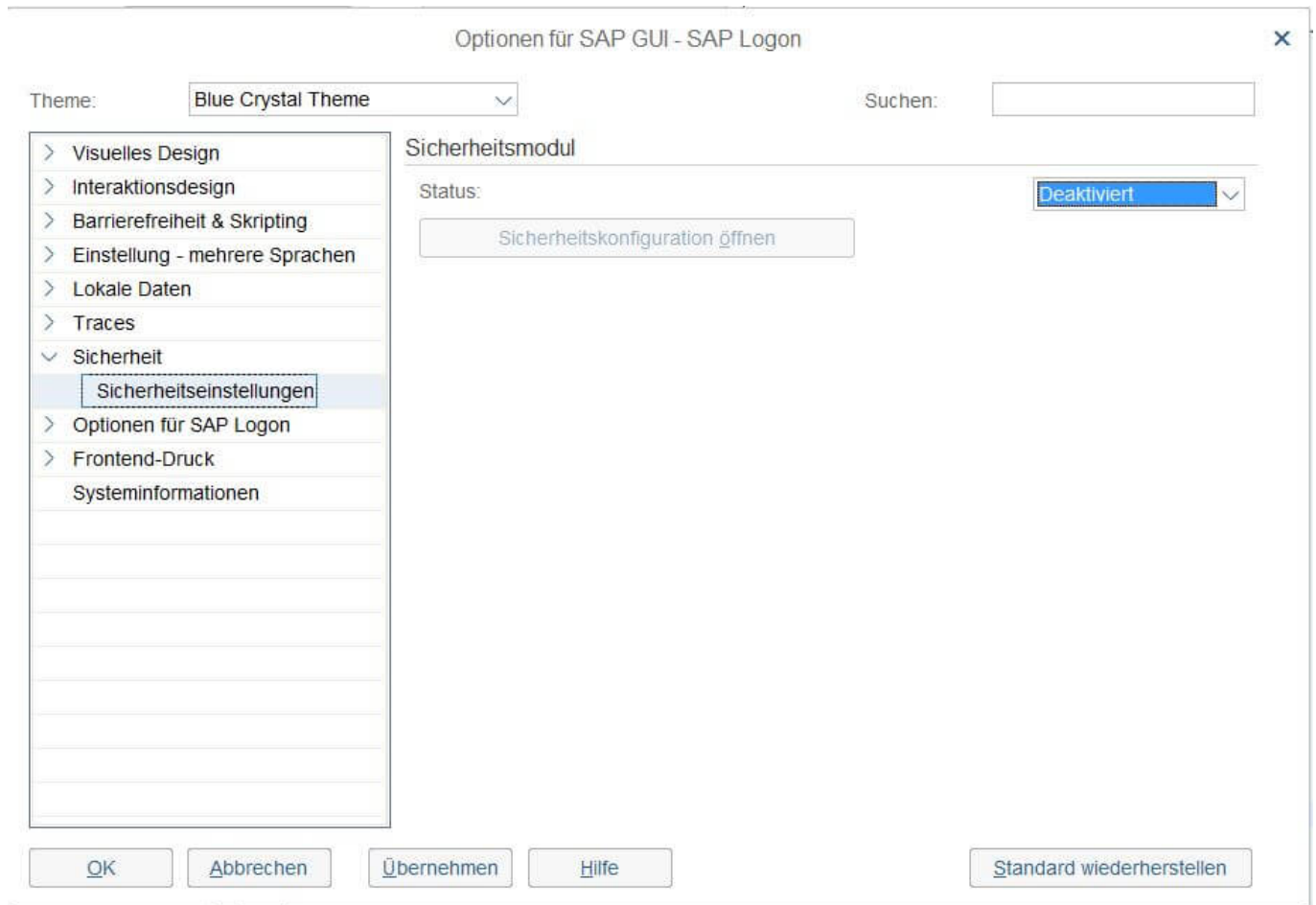
```
Get-WmiObject -class Win32_Service | Format-Table -property Name,State
```



The screenshot shows a SAP report window titled "Report ZPSTEST". The report content is a table with two columns: "Name" and "State". The table lists several Windows services and their current status.

Name	State
AdobeARMservice	Running
AdobeFlashPlayerUpdateSvc	Stopped
AMD External Events Utility	Running
AppIDSvc	Stopped
Appinfo	Stopped
AppMgmt	Stopped
aspnet_state	Stopped
AudioEndpointBuilder	Running

Sollte der Aufruf nicht funktionieren, kann es sein, dass eine Standard-Sicherheitsregel den Aufruf blockiert. Zum Experimentieren können diese über den SAP Logon deaktiviert werden.



Um den kompletten Umfang der Bibliothek einfach nutzen zu können, hier eine entsprechende Klasse.

```
[crayon-5b093461dfc89023447315/]
```

Hier ein Beispielprogramm zum Anzeigen einer Tabelle im TableGrid von PowerShell. Die Daten werden einfach mittels eines Select gelesen und mit dem JSON-Serialisierer umgewandelt. Dann wird PowerShell geladen und initialisiert. Die JSON-Daten werden dem GridView nach einer Konvertierung über die Pipeline übergeben.

```
[crayon-5b093461dfc9a965744776/]
```

Und so sieht es dann aus.

ABAP Editor: Report ZPOSH\_EXAMPLE2 anzeigen

Repository Browser  
Repository Infosystem  
Lokale Objekte

Filter  
Kriterien hinzufügen

MANDT	CARRID	CONNID	FLDATE	PRICE	CURRENCY	PLANETYPE	SEATSMAX	SEATSOCC	PAYMENTSUM	SEATSMAXB
500	AA	17	2017-09-14	422,94	USD	747-400	385	375	195.360,31	31
500	AA	17	2017-10-16	422,94	USD	747-400	385	369	191.600,35	31
500	AA	17	2017-11-17	422,94	USD	747-400	385	371	192.091,06	31
500	AA	17	2017-12-19	422,94	USD	747-400	385	371	192.446,32	31
500	AA	17	2018-01-20	422,94	USD	747-400	385	365	190.729,29	31
500	AA	17	2018-02-21	422,94	USD	747-400	385	370	190.653,10	31
500	AA	17	2018-03-25	422,94	USD	747-400	385	372	193.626,46	31
500	AA	17	2018-04-26	422,94	USD	747-400	385	374	192.450,59	31
500	AA	17	2018-05-28	422,94	USD	747-400	385	157	82.477,55	31
500	AA	17	2018-06-29	422,94	USD	747-400	385	192	98.376,06	31
500	AA	17	2018-07-31	422,94	USD	747-400	385	30	16.570,81	31
500	AA	17	2018-09-01	422,94	USD	747-400	385	43	22.373,56	31
500	AA	17	2018-10-03	422,94	USD	747-400	385	30	15.860,27	31
500	AA	64	2017-09-16	422,94	USD	A340-600	330	319	172.343,86	30
500	AA	64	2017-10-18	422,94	USD	A340-600	330	318	169.835,87	30
500	AA	64	2017-11-19	422,94	USD	A340-600	330	313	166.452,36	30
500	AA	64	2017-12-21	422,94	USD	A340-600	330	311	167.649,46	30
500	AA	64	2018-01-22	422,94	USD	A340-600	330	317	169.252,37	30
500	AA	64	2018-02-23	422,94	USD	A340-600	330	321	170.212,39	30
500	AA	64	2018-03-27	422,94	USD	A340-600	330	317	171.202,01	30
500	AA	64	2018-04-28	422,94	USD	A340-600	330	318	170.038,96	30
500	AA	64	2018-05-30	422,94	USD	A340-600	330	200	106.356,73	30
500	AA	64	2018-07-01	422,94	USD	A340-600	330	3	1.247,67	30

Report ZPOSH\_EXAMPLE2 aktiv

```

1  "--Begin
2  -----
3  REPORT zposh_example2.
4
5  DATA:
6    lt_sflight      TYPE STANDARD TABLE OF sflight,
7    lv_sflight_json TYPE string,
8    lo_posh         TYPE REF TO z_cl_activexposhv3,
9    lv_pscode      TYPE string,
10   lv_result       TYPE string
11
12
13 SELECT * INTO CORRESPONDING FIELDS OF TABLE lt_sflight FROM SFLIGHT.
14 lv_sflight_json = /ui2/cl_json=>serialize( data = lt_sflight ).
15
16 CREATE OBJECT lo_posh.
17 CHECK lo_posh->load_lib( ) = lo_posh->mc_true.
18 CHECK lo_posh->get_is_powershell_installed( ) = lo_posh->mc_true.
19 CHECK lo_posh->init( lv_load_profiles = lo_posh->mc_false ) = 0.
20
21 lo_posh->set_outputmode( lo_posh->mc_outputbuffer ).
22 lo_posh->set_outputwidth( 132 ).
23 lo_posh->clear_output( ).
24
25 lv_pscode =
26   '( "" && lv_sflight_json && "" | ConvertFrom-Json | Out-GridView'
27   .
28 lo_posh->execute( lv_pscode ).
29 lv_result = lo_posh->get_outputstring( ).
30
31 lo_posh->free_lib( ).
32
33 "--End

```

ABAP | Ze 1 Sp 1 | | | N...

Auf diese Art und Weise lassen sich nun sehr einfach Integrationsszenarien zwischen PowerShell und ABAP realisieren.

Hier ein Beispiel zum Anzeigen von .NET-Dialogen. Das Beispiel 1 öffnet einen Dialog, im Beispiel 2 wird das Ergebnis des 1. Dialoges angezeigt und im Beispiel 3 wird ein Datei-Öffnen-Dialog angezeigt und mit Auswahl eines Bildes wird dieses in einem eigenen Dialog angezeigt.

[crayon-5b093461dfca1328723570/]

In diesem Fall sind die PowerShell Sourcen direkt in den ABAP-Quellcode eingebettet. Diese können jedoch auch als Include-Entwicklungsobjekt gespeichert und mit der Methode READ\_INCL\_AS\_STRING eingelesen und der Methode EXECUTE übergeben werden. Mit diesem Ansatz kann nun die gesamte .NET-Welt in ABAP integriert werden. Jede .NET-Bibliothek kann so genutzt werden und darüber hinaus können auch [VB# resp. VB.NET](#) und C# direkt integriert werden.