

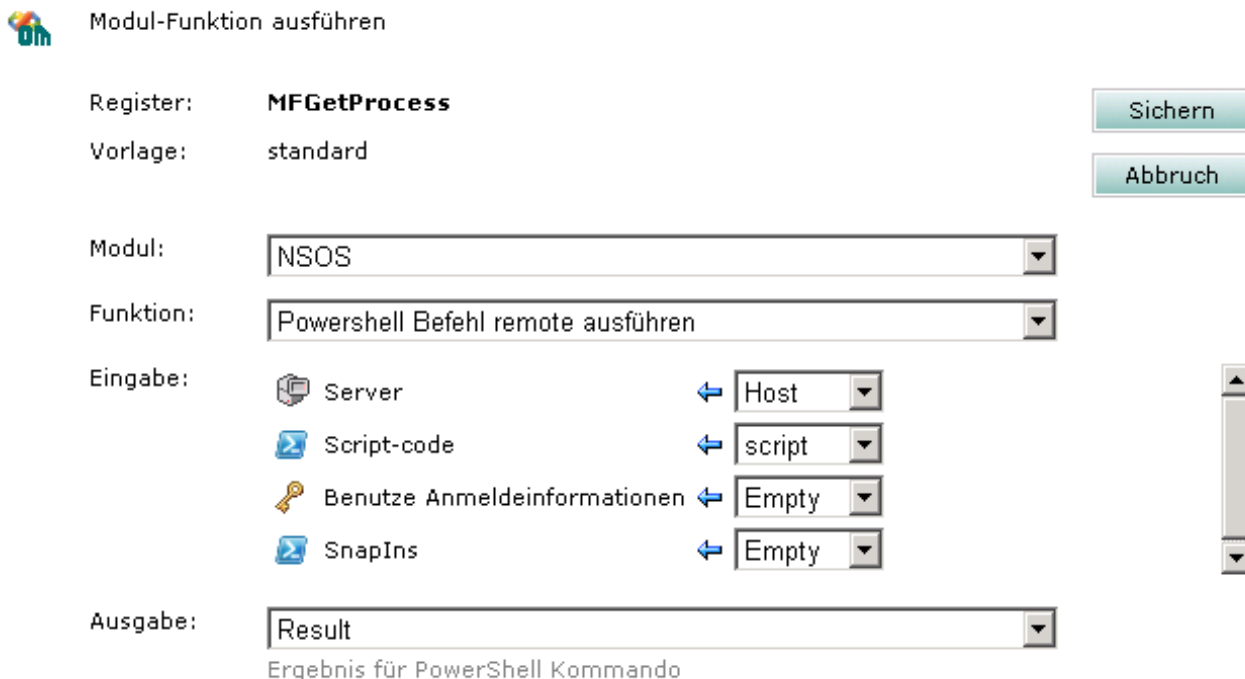
Der Operations Manager unterstützt den Einsatz der Powershell durch die folgenden Methoden:

1. Berechnung von Registern im PreProzess

Neben den VBScript Registern können auch Powershell-Register verwendet werden, um Inhalte von Registern zu bestimmen. Hierfür wird das HDBCom (COM+ hierarchische Datenbank) als \$HDB im Registerscript zur Verfügung gestellt. Wie auch beim VBScript wird der Registerinhalt durch Zuweisung auf die Variable {Registername} am Ende bestimmt. Die Powershell-Register sind zur Berechnung bzw. Transformation von Eingangs- in Ausgangsdaten gedacht. Der Sicherheitskontext ist der des Web-Servers bzw. des Kontos unter dem Application-Pool. Grundsätzlich wird ein verlassen der Maschine übers Netzwerk in Powershell-Registern daher nicht unterstützt!

2. Modulfunktion 'NSOS.OS.RunShellRemote' (Powershell Befehl remote ausführen)

Über die Modulfunktion kann ein Registerwert zur Laufzeit des PreProzess mit dem Inhalt eines Powershell-Scripts gefüllt werden bzw. ein Script ausgeführt werden. Das Register wird nach Ausführung mit dem Console-Output des Scripts gefüllt. Zur Weiterverarbeitung wird dies häufig in ein VBScript-Register eingefügt, daher werden vor dem Einfügen die Zeilenumbrüche durch Pipes '|' ersetzt. In Modulfunktionen kann grundsätzlich beliebiger Code auf dem Zielsystem zur Ausführung kommen. Der Kontext ist entweder der unter dem Zielhost in der HDB hinterlegt oder alternative Anmeldeinformationen, die als Objekt mit hinterlegten Anmeldeinformationen als HDB-Pfad angegeben werden können:



Modul-Funktion ausführen

Register: **MFGetProcess** Sichern

Vorlage: standard Abbruch

Modul: NSOS

Funktion: Powershell Befehl remote ausführen

Eingabe:

- Server: Host
- Script-code: script
- Benutze Anmeldeinformationen: Empty
- SnapIns: Empty

Ausgabe: Result

Ergebnis für PowerShell Kommando

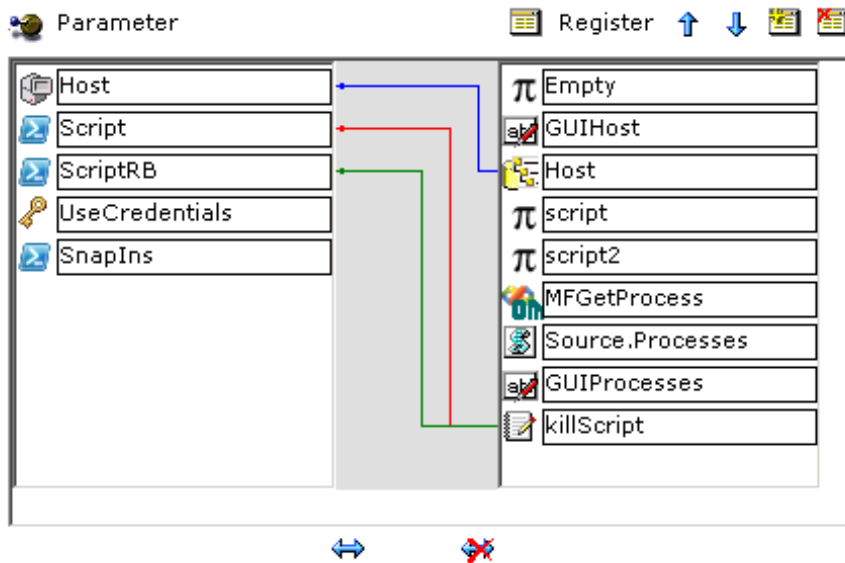
Bitte beachten: Sofern die Ergebnisvariable des ausgeführten Powershell-Scripts keine Zeichenkette enthält, kann es vorkommen, dass die Modulfunktion ein leeres Ergebnis zurückliefert. In diesem Fall muss der Ergebniswert nach String gecastet werden.

Beispiel:

```
$result.ToString()
```

3. Ausführen eines Powershell-Script in einer Elementar-Operation

Die Elementaroperation 'NSOS.OS.OSExecShellRemote' (Powershell Befehl remote ausführen) ist in den Parametern identisch mit der Modulfunktion:



Um in einem solchen Script die Funktionen der HDBCom nutzen zu können, muss diese wie folgt instanziiert werden:

```
$HDB=New-Object -Com "HDBCom.ComHDB"
$HDB.Hostname = "{RequestDBServer}"
$HDB.Database = "{RequestDatabase}"
```

Als Script wird immer ein Powershell-Script-Text angegeben, wie z.B.:

```
$Result = get-process | select name, id
foreach ($Line in $Result)
{
  $Out += [string] ($Line.Name) + "~" + [string] ($Line.Id) + ","
}
$Out = $Out.Trim(",")
$Out
```

Dieses Script berechnet z.B. die Quelle einer Prozess-Auflistung, die dann in einer Listbox angezeigt werden kann.

Probleme mit Powershell

Damit Powershell-Befehle remote ausgeführt werden können, muss auf dem Zielhost der Befehl:

```
WINRM quickconfig
```

ausgeführt worden sein. Dieser Befehl konfiguriert das System so, dass per Powershell remote darauf zugegriffen werden kann. Für die Ausführung von Scripten muss ggfs. noch

Set-ExecutionPolicy -ExecutionPolicy unrestricted

ausgeführt werden.

Testen ob remote Powershell funktional

Mit dem folgenden Befehl aus der Powershell des Dispatcher kann überprüft werden, ob das Zielsystem remote Powershell unterstützt:

```
invoke-command -Computer RemoteServerName -ScriptBlock { Get-Process }
```

Dies sollte im angemeldeten Kontext des Dienstes unter dem Host-Objekt funktionieren und die Liste der Prozesse vom Remote-System zurück liefern.

Ausführen von PS/PS1-Dateien mit Scripten

Möchte man ein Script mit Parametern zur Ausführung bringen, ist wie folgt vorzugehen:

```
invoke-command -ComputerName RemoteServerName -FilePath "C:\temp\ps\doit.ps1" -ArgumentList  
"Name2"  
-Credential $OMCredentials
```

In der Variablen \$OMCredentials stellt der OM das jeweilige Credential-Objekt für die Umgebung bereit. Das obige Script kann also auf irgend einem Host (auch Dispatcher lokal mit .) zur Ausführung gebracht werden. Im \$OMCredentials Objekt befinden sich dann die Anmeldeinformationen des lokalen bzw. Ausführungshost oder die des angegebenen Objektes für die Operation.

Powershell Scripten remote ausführen (double-hop)

Bei der remote Ausführung von Powershell-Scripten gilt immer auch die von CMD und VBScript bekannte folgende Problematik: <http://blogs.msdn.com/b/clustering/archive/2009/06/25/9803001.aspx>

Beispielantrag

Anbei ein Beispielantrag, der die Prozesse einer remote-Maschine per Modulfunktion ausliest und dann einen ausgewählten Prozess per EO remote beendet. Es sind zwei Script-Register für die Modulfunktion vorhanden. Einmal wird die Quelle für die Listbox direkt per Powershell berechnet (Register: 'script') und zum zweiten in einem expliziten VBScript (Register 'script2' und 'Source.Processes') aufbereitet.