

# APPLICATION NOTE

## DOKUMENTATION

## OFFICEMASTER POWERSHELL COMMANDLETS

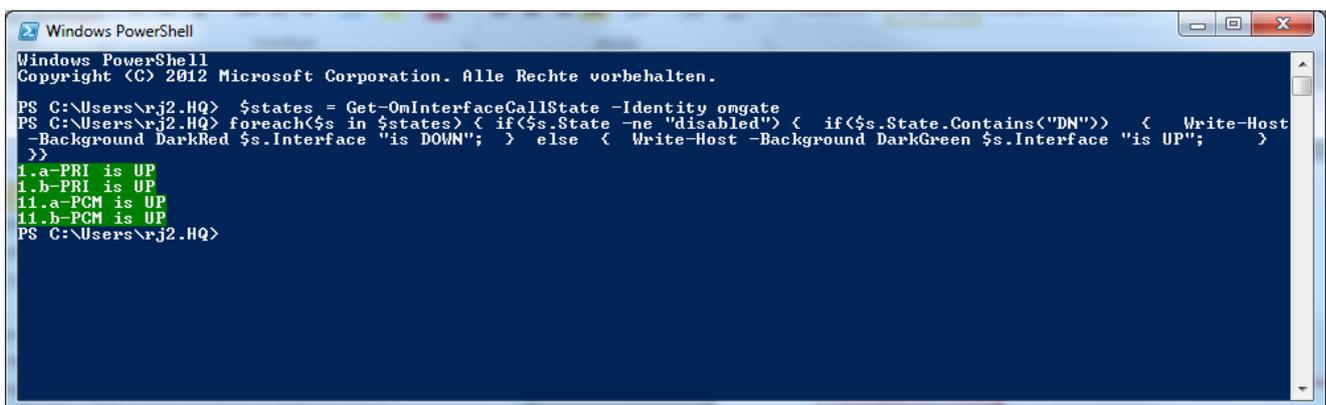
### REV 1.0

---

No. 2016-03

Revision 2.1

---



```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\rj2.HQ> $states = Get-OmInterfaceCallState -Identity omgate
PS C:\Users\rj2.HQ> foreach($s in $states) { if($s.State -ne "disabled") { if($s.State.Contains("DN")) { Write-Host
-Background DarkRed $s.Interface "is DOWN"; } else { Write-Host -Background DarkGreen $s.Interface "is UP"; }
} }
1.a-PRI is UP
1.b-PRI is UP
11.a-PCM is UP
11.b-PCM is UP
PS C:\Users\rj2.HQ>
```

Application Note | Ferrari electronic

---

## I. Revision History

Revision	Date	Author	Changes
2.1	30.03.2016	Ralf Jänike	HOWTO 0MQ statt SSH
2.0	14.10.2015	Ralf Jänike	ALLE Kommandos kurz erwähnt
1.3	07.07.2015	Ralf Jänike	Restart, Backup, Restore
1.2	01.07.2015	Ralf Jänike	SSH Cmdlets
1.1.2	12.06.2015	Ralf Jänike	Timeout Parameter
1.1.1	08.06.2015	Ralf Jänike	Mehr Datentyp Informationen
1.1	05.06.2015	Ralf Jänike	Bilder
1.1	02.06.2015	Ralf Jänike	Datentypen Dokumentation
1.0	01.06.2015	Ralf Jänike	Initial Release
	25.06.2015	Nils Kuchler	Redaktion
	01.07.2015	Johann Deutinger	Redaktion

# Einleitung

Diese Dokumentation beschreibt Funktions- und Anwendungshinweise der OfficeMaster PowerShell Commandlets. Diese Beschreibungen geben die technische Grundlage zur Erarbeitung weiterer Dokumentationen.

---

## Haftungsausschluss

Die in diesem Dokument zusammengefassten Informationen wurden nach bestem Wissen und Gewissen zusammengetragen. Für etwaige Fehler, sowie Änderungen, die dem technischen Fortschritt dienen, wird keine Haftung übernommen.

---

## Beabsichtigte Leserschaft

Dieses Dokument richtet sich an Erstverwender/Betatester der dokumentierten Software, sowie an interne Mitarbeiter zur technischen Dokumentation des Produktes.

---

## Symbole

Folgende Symbole werden im Dokument verwendet und haben folgende Bedeutung.



### Warnung

Warnungen sollen auf unbedingte Kenntnisnahme hinweisen, um die korrekte Funktion der Software zu gewährleisten.



### Hinweis

Hinweise informieren den Benutzer über Besonderheiten im Gebrauch der Software.



### Anmerkung

Anmerkungen geben zusätzliche Informationen zum Gebrauch oder der Funktionsweise der Software.

---

## Copyright und rechtliche Hinweise

Copyright © 2015 von Ferrari electronic AG. Alle Rechte vorbehalten. Kein Teil dieser Dokumentation oder der Software darf ohne schriftliche Genehmigung der Ferrari electronic AG auf irgendeinem Wege kopiert werden. Alle in dieser Dokumentation genannten Warenzeichen sind registrierte Warenzeichen der jeweiligen Warenzeicheninhaber. Änderungen der Software und der Dokumentation, auch ohne vorherige Ankündigung, vorbehalten. Die in diesem Dokument enthaltenen Informationen wurden mit größter Sorgfalt zusammengestellt. Dennoch können fehlerhafte Angaben nicht völlig ausgeschlossen werden. Die Ferrari electronic AG haftet nicht für eventuelle Fehler und deren Folgen.

# INHALT

---

Einleitung .....	1
Inhalt.....	2
1. Allgemein .....	3
1.1 Windows PowerShell mit OfficeMaster Commandlets	3
1.2 Einsatzgebiete	3
2. Installation.....	4
2.1 Installationsvorgang	4
2.2 Installationskomponenten	4
2.3 Deinstallation	5
3. Verwendung.....	6
3.1 Allgemeine Cmdlet-Argumente/Parameter	6
3.2 Allgemeine Get-Cmdlet-Argumente/Parameter (ZeroMQ)	7
3.3 Commandlets zur Überwachung von OfficeMaster Gate	9
3.3 Commandlets zum Überprüfen der OfficeMaster Gate Konfiguration	13
3.4 Commandlets zur Wartung eines OfficeMaster Gate	14
3.5 Commandlets	17
3.6 Info Commandlets	30
Allgemeine Info Cmdlet-Argumente/Parameter	30
3.7 Lizenz Commandlets	32
Allgemeine Lizenz Cmdlet-Argumente/Parameter	32
3.8 Dateilogging Commandlets	34
Allgemeine Logging Cmdlet-Argumente/Parameter	34
3.9 Versionsinformationen Commandlets	36
4. OmgUtils Typen.....	37
4.1 FileSystemInfo	37
4.2 InterfaceState	37
4.3 InterfaceState.ChannelState	38
4.4 NameValue	39
5. Anwendungsbeispiele .....	40
5.1 Benutze ZeroMQ Verbindung zu OMG, statt SSH	40
5.2 Verwendung des Beispielskripts Monitor-OMGates.ps1	42
Abkürzungen.....	43

# 1. Allgemein

---

## 1.1 Windows PowerShell mit OfficeMaster Commandlets

Mit der Windows PowerShell können Sie Anwendungen über die Befehlszeile verwalten. Die Shell verfügt über eine Befehlszeilenumgebung, produktspezifische Befehle und eine komplette Skriptsprache. Die Windows PowerShell wurde 2006 eingeführt, und in der Befehlszeilenschnittstelle zur Verwaltung von Microsoft Server Produkten integriert.

OfficeMaster PowerShell Commandlets stellen als Erweiterungsmodul die Integration von Verwaltungs- und Überwachungsfunktionen von OfficeMaster Produkten, insbesondere OfficeMaster Gate (OMG), bereit.

Die folgenden Abschnitte enthalten eine Liste von Cmdlets sowie Beschreibungen der Cmdlets. Diese Informationen stehen Ihnen auch direkt über die Befehlszeile zur Verfügung. Geben Sie einfach den folgenden Befehl an der Eingabeaufforderung der PowerShell:

```
Get-Help <cmdlet name> -Full
```

Wenn Sie beispielsweise Hilfeinformationen zum Cmdlet **Get-OmInterfaceCallStates** von der Befehlszeile aus abrufen möchten, geben Sie folgenden Befehl ein:

```
Get-Help Get-OmInterfaceCallStates -Full
```

---

## 1.2 Einsatzgebiete

Das Haupteinsatzgebiet der Software ist das automatisierte Überwachen und die codegesteuerte Verwaltung (z.B. Konfiguration, Backup, Wiederherstellung) von OfficeMaster Gate.

## 2. Installation

### 2.1 Installationsvorgang

Für die Installation der OM-Cmdlets steht keine spezialisierte Setupdatei zur Verfügung. Die Cmdlets sind Bestandteil der Produkte *OfficeMaster-SIP2Lync*, *OfficeMaster Management Pack* und *OfficeMaster-SBA*. Für Computer, auf denen keines dieser Produkte installiert ist, stellt Ferrari electronic ein ZIP-Paket zur Verfügung, dessen Inhalt Sie in ein spezielles Verzeichnis entpacken müssen.



#### Installationsvoraussetzungen

- Microsoft Windows Server 2008 64Bit oder höher
- Microsoft Windows 7 64Bit oder höher
- Microsoft .Net-Framework 4.0 Full oder höher
- Microsoft Windows PowerShell 3.0 oder höher

Das ZIP-Paket mit den Cmdlets beziehen Sie aus dem Abschnitt *OfficeMaster Hardware* des Bereichs *Downloadcenter* der Ferrari electronic Webseite.

Da es sich bei den Cmdlets um Komponenten handelt auf die automatisiert und ohne Benutzerkontext zugegriffen werden soll, entpacken Sie den Inhalt des Paketes in folgendes Verzeichnis:

```
%Windir%\System32\WindowsPowerShell\v1.0\Modules\OfficeMaster
```



#### Hinweis

Wird der Inhalt des ZIP-Paketes in ein anderes Verzeichnis entpackt, müssen Sie vor der Verwendung in Windows PowerShell den Befehl **Import-Module** mit Angabe des Verzeichnispfades ausführen bzw. den Pfad in die Umgebungsvariable *PSModulePath* einfügen.

### 2.2 Installationskomponenten

Die OM Cmdlets bestehen aus folgenden Komponenten:

Datei	Beschreibung
OmPowerShellCmdlets.dll	Basis-DLL der Cmdlets mit produktunspezifischen Funktionen
OmgPowerShellCmdlets.dll	Basis-DLL der Cmdlets mit Funktionen für OfficeMaster Gate
OmPowerShellUtils.dll	DLL zur Bereitstellung von produktunspezifischen Typen und Funktionen
OmPowerShellCmdlets.dll-help.xml	Datei mit Hilfe- und Beschreibungstexten für die OM Cmdlets
OmgPowerShellCmdlets.dll-help.xml	Datei mit Hilfe- und Beschreibungstexten für die OMG Cmdlets
OfficeMaster.psd1	PowerShell Modul Manifest zum Laden der Cmdlets in die PowerShell Sitzung

Independentsoft.Sip.dll, Independentsoft.Sip.xml	DLL zur Bereitstellung von SIP Funktionen
Jscape.Ssh.dll	DLL zur Bereitstellung der verschlüsselten Kommunikation via SSH mit der Hardware
FFUMS.Logger.dll	DLL zur Bereitstellung von Logfunktionen
ZeroMQ.dll, i386/libzmq.dll, i386/libsodium.dll, amd64/libzmq.dll, amd64/libsodium.dll	DLL zur Bereitstellung der verschlüsselten Kommunikation via ZeroMQ mit der Hardware
foam_cs.dll, foam.dll, foam64.dll, snfs64.dll	DLL zur Bereitstellung proprietärer Verarbeitungsfunktion und Zugriffsmechanismen auf die Hardware
Monitor-OMGates.ps1, gateways_sample.txt	PowerShell Skript mit beispielhafter Verwendung der OMG Cmdlets

**Tabelle 1: Installationskomponenten**

---

### 2.3 Deinstallation

Die Deinstallation der Cmdlets erfolgt über das Programme und Features Modul der Windows Systemsteuerung bei der Deinstallation eines der unter 2.1. genannten OfficeMaster Produkte bzw. durch Löschen des Verzeichnisses, in das Sie die Dateien entpackten.

## 3. Verwendung

### 3.1 Allgemeine Cmdlet-Argumente/Parameter

Für die Verwendung der OMG Cmdlets starten Sie eine Windows PowerShell Konsole. Stellen Sie sicher, dass die OfficeMaster PowerShell Commandlets in dieser PowerShell Sitzung zur Verfügung stehen, indem Sie folgenden Befehl ausführen:

```
Get-Command -Module OfficeMaster
```

Wird daraufhin keine Liste der Cmdlets angezeigt, stellen Sie sicher, dass sie installiert sind und führen folgenden Befehl aus:

```
Import-Module OfficeMaster
```

Bis auf wenige Ausnahmen (z.B. **Get-OmAllBoxes**) kennen alle Cmdlets im Modul die gleichen Befehlszeilenargumente, die beim Aufruf übergeben werden müssen/können.

#### Identity

Gibt die Identität des OMG an, gegen welches der Befehl ausgeführt wird. *Identity* kann dabei die fünfstellige Seriennummer, die IPv4-Adresse oder der auflösbare DNS-Name des OMG sein. Wird „LOCAL“ angegeben, wird der Wert für Identity vom Registryeintrag *MGBSerial* unter *HKLM\Software\Ferrari\SBA* angenommen.

Erforderlich: ja  
Standardwert: <keiner>

#### UserName

Gibt den Namen des Benutzers an, der die Rechte hat, den Befehl am OMG auszuführen. Die Verbindung zum OMG wird über SSH aufgebaut, wofür eine Authentifizierung erforderlich ist. Wenn dieses Argument nicht angegeben wird, wird der Wert aus der PowerShell-Sitzungsvariablen *OmgUserName* verwendet. Ist die Variable nicht gesetzt, wird der Wert vom Registryeintrag *sshusername* unter *HKLM\Software\Ferrari\SBA* angenommen. Ist der Wert leer, wird der Standardwert angenommen. Dieses Argument sollte immer dann angegeben werden, wenn die Authentifizierungsinformationen im OMG nicht dem Auslieferungszustand entsprechen.

Erforderlich: nein  
Standardwert: root

#### UserKeyFile

Gibt den Pfad zur Datei an, die den privaten SSH-Schlüssel des Benutzers enthält, der die Rechte hat den Befehl am OMG auszuführen. Die Verbindung zum OMG wird über SSH aufgebaut, wofür eine Authentifizierung erforderlich ist. Wenn dieses Argument nicht angegeben wird, wird der Wert aus der PowerShell-Sitzungsvariablen *OmgUserKeyFile* verwendet. Ist die Variable nicht gesetzt, wird der Wert vom Registryeintrag *sshuserkeyfile* unter *HKLM\Software\Ferrari\SBA* angenommen. Ist der Wert leer bzw. verweist auf eine nicht existente Datei, wird der Wert von *UserPassword* für die Authentifizierung verwendet.

Erforderlich: Nein  
Standardwert: <keiner>

*UserKeyfile* und *UserPassword* können separat und in Kombination verwendet werden. Ist die Schlüssel-Datei passwortgeschützt

wird der Wert von *UserPassword* zur Entschlüsselung genutzt.

Am OfficeMaster Gate muss der öffentliche SSH-Schlüssel des Benutzers bekannt sein, damit die Anmeldung erfolgreich ist. Um den öffentlichen Schlüssel im OMG zu hinterlegen, führen Sie das Cmdlet **Import-OmSshKey** aus.

### UserPassword

Gibt das Passwort des Benutzers an, der die Rechte hat, den Befehl am OMG auszuführen. Die Verbindung zum OMG wird über SSH aufgebaut, wofür eine Authentifizierung erforderlich ist. Wenn dieses Argument nicht angegeben wird, wird der Wert aus der PowerShell-Sitzungsvariablen *OmgUserPassword* verwendet. Ist die Variable nicht gesetzt, wird der Wert vom Registryeintrag *sshuserpw* unter *HKLM\Software\Ferrari\SBA* angenommen. Ist der Wert leer und kein privater SSH-Schlüssel mit dem Argument *UserKeyFile* angegeben, wird der Standardwert angenommen. Dieses Argument sollte immer dann angegeben werden, wenn die Authentifizierungsinformationen im OMG nicht dem Auslieferungszustand entsprechen.

Erforderlich:	nein
Standardwert:	omc

*UserKeyfile* und *UserPassword* können separat und in Kombination verwendet werden.

### ShellPrompt

Gibt das Zeichen/die Zeichenfolge an, an der die Eingabeaufforderung der SSH-Shell Sitzung im OMG erkannt werden kann. Wenn dieses Argument nicht angegeben wird, wird der Wert aus der PowerShell-Sitzungsvariablen *OmgShellPrompt* verwendet. Ist die Variable nicht gesetzt, wird der Wert vom Registryeintrag *sshshellprompt* unter *HKLM\Software\Ferrari\SBA* angenommen. Ist der Wert leer, wird der Standardwert angenommen. Dieses Argument sollte immer dann angegeben werden, wenn die Shell-Einstellungen im OMG nicht dem Auslieferungszustand entsprechen.

Erforderlich:	nein
Standardwert:	#

### Timeout

Gibt die Zeit in Millisekunden an, für die versucht wird, eine Verbindung (z.B. für die SSH Sitzung) zum OMG herzustellen. Wird die Verbindung in der angegebenen Zeit nicht erfolgreich hergestellt, werden davon abhängige Prozesse nicht erfolgreich ausgeführt. Wenn dieses Argument nicht angegeben wird, wird der Wert aus der PowerShell-Sitzungsvariablen *OmgTimeout* verwendet. Ist die Variable nicht gesetzt, wird der Wert vom Registryeintrag *connecttimeout* unter *HKLM\Software\Ferrari\SBA* angenommen. Ist der Wert leer, wird der Standardwert angenommen.

Erforderlich:	nein
Standardwert:	5000

---

## 3.2 Allgemeine Get-Cmdlet-Argumente/Parameter (ZeroMQ)

Die Cmdlets, welche keine Schreib-Aktionen auf OfficeMaster Gate durchführen, Konfigurationsdateien lesen oder den Neustart eines oder mehrerer Dienste erfordern, unterstützen eine weitere Methode der Authentifizierung und Abfrage: Secure ZeroMQ. Diese Methode hat den entscheidenden Vorteil, dass sie um ein Vielfaches schneller das Ergebnis liefert. Die Verschlüsselte Kommunikation über ZeroMQ steht in der Regel nur Cmdlets mit dem Präfix *Get* zur Verfügung, die einen aktuellen Status des OMG abfragen.

Wenn diese Methode der Authentifizierung verwendet wird, ist es nicht erforderlich Benutzername, Passwort oder SSH-Schlüsseldatei anzugeben.

Für die erfolgreiche Authentifizierung müssen den Cmdlets dabei 3 Schlüssel zur Verfügung stehen, welche als Argumente übergeben, in der Registry gespeichert oder als Variablen in der PowerShell Sitzung definiert sind. Kann die Verbindung aufgrund fehlender Schlüssel nicht über ZeroMQ hergestellt werden, wird sie stattdessen über SSH (Benutzername, Passwort erforderlich) hergestellt.

Die gültigen Schlüssel-Werte für die ZeroMQ-Authentifizierung können aus der Web-Administrationsoberfläche des OMG abgeschrieben oder über das Cmdlet **Export-OmZmqKeys** erhalten werden.



#### Hinweis

Die vollständige technische Unterstützung der Verbindung von Cmdlets zu OMG über ZeroMQ ist nur mit folgenden Versionsnummern gewährleistet:

Datei	Min. Versionsnummer
OmPowerShellCmdlets.dll	2.4.0.0
OmgPowerShellCmdlets.dll	2.4.0.0
OmPowerShellUtils.dll	2.4.0.0
OMG UserFS	4.0.280
OMG Modules	4.0.280

#### ZmqOmgPublicKey

Gibt den öffentlichen ZeroMQ Schlüssel des OfficeMaster Gate an. Die Verbindung zum OMG wird verschlüsselt über ZeroMQ aufgebaut, wofür eine Authentifizierung erforderlich ist. Wenn dieses Argument nicht angegeben wird, wird der Wert aus der PowerShell-Sitzungsvariablen *OmgZmqOmgPublicKey* verwendet. Ist die Variable nicht gesetzt, wird der Wert vom Registryeintrag *zmqpublickey* unter *HKLM\Software\Ferrari\SBA* angenommen. Ist der Wert leer, wird kein Standardwert angenommen und die Verbindung zum OMG über SSH aufgebaut.

Erforderlich:           nein

Standardwert:         <keiner>

#### ZmqUserPublicKey

Gibt den öffentlichen ZeroMQ Schlüssel des Benutzers an. Die Verbindung zum OMG wird verschlüsselt über ZeroMQ aufgebaut, wofür eine Authentifizierung erforderlich ist. Wenn dieses Argument nicht angegeben wird, wird der Wert aus der PowerShell-Sitzungsvariablen *OmgZmqUserPublicKey* verwendet. Ist die Variable nicht gesetzt, wird der Wert vom Registryeintrag *zmquserpublickey* unter *HKLM\Software\Ferrari\SBA* angenommen. Ist der Wert leer, wird kein Standardwert angenommen und die Verbindung zum OMG über SSH aufgebaut.

Erforderlich: nein  
Standardwert: <keiner>

### ZmqUserPrivateKey

Gibt den privaten ZeroMQ Schlüssel des Benutzers an. Die Verbindung zum OMG wird verschlüsselt über ZeroMQ aufgebaut, wofür eine Authentifizierung erforderlich ist. Wenn dieses Argument nicht angegeben wird, wird der Wert aus der PowerShell-Sitzungsvariablen *OmgZmqUserPrivateKey* verwendet. Ist die Variable nicht gesetzt, wird der Wert vom Registryeintrag *zmquserprivatekey* unter *HKLM\Software\Ferrari\SBA* angenommen. Ist der Wert leer, wird kein Standardwert angenommen und die Verbindung zum OMG über SSH aufgebaut.

Erforderlich: nein  
Standardwert: <keiner>

### PreferSSH

Wenn dieser Wert auf *\$True* gesetzt ist, wird die Verbindung trotz zur Verfügung stehender Schlüssel nicht über ZeroMQ sondern über SSH zum OMG aufgebaut.

Erforderlich: nein  
Standardwert: *\$False*

---

## 3.3 Commandlets zur Überwachung von OfficeMaster Gate

Die wichtigsten Cmdlets zur Überwachung des Status der OMG sind **Get-OmFileSystemInfo** und **Get-OmInterfaceCallState**. Die daraus abzulesenden Daten geben Ihnen Auskunft über den aktuellen Zustand der Hardware und werden herangezogen, um beispielsweise Szenarien abzubilden, die Benachrichtigungen auslösen, wenn ein bestimmter Status erreicht wird.

---

### Get-OmFileSystemInfo

Die Rückgabewerte dieses Cmdlet geben an, welche Menge Daten im flüchtigen und nicht flüchtigen Speicher der Hardware gespeichert sind. Die Werte werden in absoluten Zahlen und Vergleichswerten (Prozent) angegeben. Hohe Auslastungswerte sind ein Indiz dafür, dass eine manuelle Wartung der Hardware erforderlich ist.

#### Beispiel

```
Get-OmFileSystemInfo -Identity omgate
```

Gibt eine Liste von *OmgUtils.FileSystemInfo* Objekten zurück. Jedem Objekt der Liste können Sie die Mengenangaben eines bestimmten Teils des Dateisystems der OMG ablesen.

#### Ausgabe

```
PercentUsed : 2%  
MountPoint  : /
```

```

Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\rj2.HQ> $fsi = Get-OmFileSystemInfo -Identity omgate
PS C:\Users\rj2.HQ> foreach($info in $fsi) { if([System.Convert]::ToInt32($info.PercentUsed.Substring(0,$info.PercentUsed
d.IndexOf("<\">")) -gt 10) { Write-Host -BackgroundColor DarkRed "Free space on filesystem " $info.MountPoint "is running
low"; [System.Media.SystemSounds]::Exclamation.Play(); } }
Free space on filesystem /var/log/omgweblog is running low
PS C:\Users\rj2.HQ>

```

```

FileSystem : /dev/md0
Used      : 2.9G
Available : 210G
Size     : 224G

PercentUsed : 1%
MountPoint  : /tmp
FileSystem  : tmpfs
Used       : 716K
Available  : 931M
Size      : 931M

PercentUsed : 14%
MountPoint  : /var/log/omgweblog
FileSystem  : /var/log/omgweblog.img
Used       : 13M
Available  : 79M
Size      : 97M

```

Bei dieser Form der Ausgabe können Sie einzelne Werte direkt adressieren, wodurch eine automatisierte Verarbeitung der Daten möglich ist:

```

$fsi = Get-OmFileSystemInfo -Identity omgate
foreach($info in $fsi) {
if([System.Convert]::ToInt32($info.PercentUsed.Substring(0,$info.PercentUsed.IndexOf("%"))) -gt 89) {
Write-Host -BackgroundColor DarkRed "Free space on filesystem " $info.MountPoint "is running low";
[System.Media.SystemSounds]::Exclamation.Play(); } }

```

Detaillierte Hilfeinformationen erhalten Sie über die Befehlszeile:

```
Get-Help Get-OmFileSystemInfo -Full
```

## Get-OmInterfaceCallState

Den Rückgabewerten dieses Cmdlet können Sie entnehmen, welche der Schnittstellen (ISDN, Analog, Virtuell) der OMG als aktiv konfiguriert sind, wie der Status jedes einzelnen B- und D-Kanal ist und ob darüber aktuell Verbindungen zu anderen Gegenstellen aktiv sind. Die Status können beispielsweise bei der Diagnose von Verbindungs- oder Auslastungsproblemen hilfreich sein.

### Beispiel

```
Get-OmInterfaceCallState -Identity omgate
```

Gibt eine Liste von *OmgUtils.InterfaceState* Objekten zurück. Jedes Objekt der Liste enthält den Status einer Schnittstelle (D-Kanal), die Status der zugehörigen B-Kanäle und die Menge der aktuell aktiven Verbindungen.

### Ausgabe

```
Interface   : 1.a-PRI
State       : L1UPL2UP
ChannelStates : {1.a-PRI-B04=stopped, 1.a-PRI-B03=stopped,1.a-PRI-B01=stopped, 1.a-PRI-B02=stopped...}
Channels    : {1.a/B30/S2M/2xG704IDT_E1, 1.a/B29/S2M/2xG704IDT_E1, 1.a/B28/S2M/2xG704IDT_E1,
1.a/B27/S2M/2xG704IDT_E1...}
CallCount   : 5

Interface   : 1.b-PRI
State       : L1UPL2UP
ChannelStates : {1.b-PRI-B28=stopped, 1.b-PRI-B29=stopped,1.b-PRI-B14=stopped, 1.b-PRI-B17=stopped...}
Channels    : {1.b/B30/S2M/2xG704IDT_E1, 1.b/B29/S2M/2xG704IDT_E1, 1.b/B28/S2M/2xG704IDT_E1,
1.b/B27/S2M/2xG704IDT_E1...}
CallCount   : 0

Interface   : 3.a-PRI
State       : disabled
ChannelStates :
Channels    : {3.a/B30/S2M/2xG704IDT_E1, 3.a/B29/S2M/2xG704IDT_E1, 3.a/B28/S2M/2xG704IDT_E1,
3.a/B27/S2M/2xG704IDT_E1...}
CallCount   : 0

Interface   : 3.b-PRI
State       : disabled
ChannelStates :
Channels    : {3.b/B30/S2M/2xG704IDT_E1, 3.b/B29/S2M/2xG704IDT_E1, 3.b/B28/S2M/2xG704IDT_E1,
3.b/B27/S2M/2xG704IDT_E1...}
CallCount   : 0

Interface   : 5.a-FXS
```

```

State      : disabled
ChannelStates :
Channels    : {5.a/B1/FXS/4xFXS_SIL}
CallCount   : 0

Interface   : 11.a-PCM
State       : L1UPL2UP
ChannelStates : {11.a-PCM-B20=stopped, 11.a-PCM-B02=stopped, 11.a-PCM-B03=stopped, 11.a-PCM-
B05=stopped...}
Channels    : {11.a/B30/PCM/loop, 11.a/B29/PCM/loop, 11.a/B28/PCM/loop, 11.a/B27/PCM/loop...}
CallCount   : 0

Interface   : 11.b-PCM
State       : L1UPL2UP
ChannelStates : {11.b-PCM-B01=stopped, 11.b-PCM-B02=stopped, 11.b-PCM-B03=stopped, 11.b-PCM-
B04=stopped...}
Channels    : {11.b/B30/PCM/loop, 11.b/B29/PCM/loop, 11.b/B28/PCM/loop, 11.b/B27/PCM/loop...}
CallCount   : 0

```

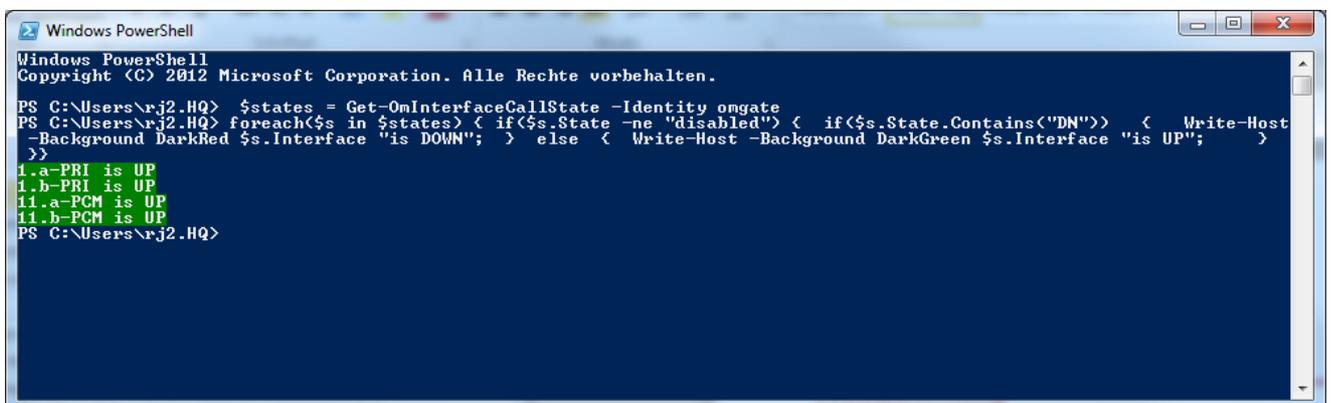
Bei dieser Form der Ausgabe können Sie einzelne Werte direkt adressieren, wodurch eine automatisierte Verarbeitung der Daten möglich ist:

```

$states = Get-OmInterfaceCallState -Identity omgate

foreach($s in $states) {
    if($s.State -ne "disabled") {
        if($s.State.Contains("DN")) {
            Write-Host -Background DarkRed $s.Interface "is DOWN"; }
        else {
            Write-Host -Background DarkGreen $s.Interface "is UP"; }
        }
}

```



```

Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\rj2.HQ> $states = Get-OmInterfaceCallState -Identity omgate
PS C:\Users\rj2.HQ> foreach($s in $states) { if($s.State -ne "disabled") { if($s.State.Contains("DN")) { Write-Host
-Background DarkRed $s.Interface "is DOWN"; } else { Write-Host -Background DarkGreen $s.Interface "is UP"; } } }
1.a-PRI is UP
1.b-PRI is UP
11.a-PCM is UP
11.b-PCM is UP
PS C:\Users\rj2.HQ>

```

Detaillierte Hilfeinformationen erhalten Sie über die Befehlszeile:

```
Get-Help Get-OmInterfaceCallState -Full
```

### 3.3 Commandlets zum Überprüfen der OfficeMaster Gate Konfiguration

Abseits von einer komplett eingerichteten komplexen Telefonie-Umgebung steht Ihnen das Cmdlet **Test-OmOutboundCall** für einen Schnelltest der Konfiguration der Wählregeln im OMG als besonders einfach gehaltene SIP Gegenstelle zur Verfügung.

#### Test-OmOutboundCall

Dieses Cmdlet initiiert eine simple SIP Verbindung zum OMG. Audiodaten werden nicht übertragen. Führen Sie den Befehl aus, und es wird bei der angegebenen Zielrufnummer kein eingehender Ruf signalisiert, ziehen Sie unbedingt die Diagnosemöglichkeiten des OMG zur Analyse des Vorgangs heran (z.B. syslog).

#### Argumente/Parameter

- **LocalIp**: Gibt die IPv4 Adresse der Netzwerkschnittstelle auf dem Computer an, auf dem der Befehl ausgeführt wird. Dieses Argument muss angegeben werden, wenn der Computer über mehrere Netzwerkschnittstellen verfügt, damit die SIP-Signalisierung die korrekten Gegenstellenwerte enthält.

Erforderlich:           nein, außer der Computer besitzt mehrere Netzwerkschnittstellen

Standardwert:        <keiner>

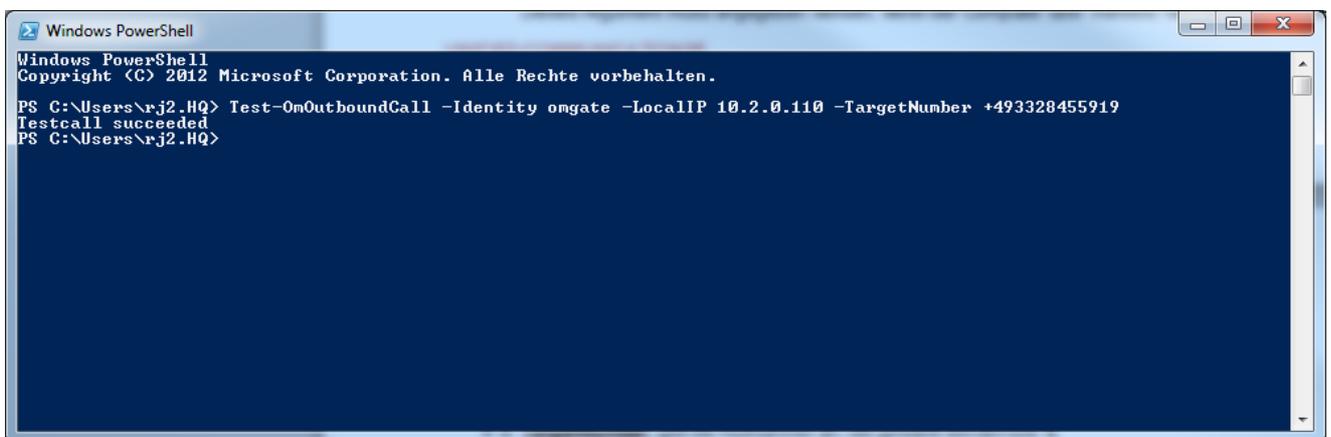
- **TargetNumber**: Gibt die Rufnummer an, die gewählt werden soll.

Erforderlich:        ja

Standardwert:        <keiner>

#### Beispiel

```
Test-OmOutboundCall -Identity omgate -LocalIp 10.2.3.4 -TargetNumber +49332845590
```



```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\rj2.HQ> Test-OmOutboundCall -Identity omgate -LocalIP 10.2.0.110 -TargetNumber +493328455919
Testcall succeeded
PS C:\Users\rj2.HQ>
```

Detaillierte Hilfeinformationen erhalten Sie über die Befehlszeile:

Get-Help Test-OmOutboundCall -Full

### 3.4 Commandlets zur Wartung eines OfficeMaster Gate

Die Wartung eines OfficeMaster Gate besteht im Wesentlichen daraus, die bestehende Konfiguration zu sichern bzw. wiederherzustellen und einen Neustart der Dienste bzw. des Betriebssystems durchzuführen.

Der optimale Weg die Wartung eines OMG führt über das OfficeMaster Gate Konfigurationsprogramm. Damit führen Sie dialoggesteuert die Sicherung bzw. Wiederherstellung aller Einstellungen, Lizenzen und Zertifikate und den Neustart des OMG durch.

#### Backup-OmGateConfiguration

Dieses Cmdlet ruft die aktuelle Konfiguration des OMG ab und speichert sie im angegebenen Pfad. Dies ist in Umgebungen nützlich, in denen sich die Einstellungen häufig ändern, wenn z.B. die Wahl- und Rufannahmeregeln im OMG durch automatisierte Prozesse (z.B. OfficeMaster Directory Service) verändert werden, damit Zwischenzustände nachvollziehbar und wiederherstellbar sind.

Der Rückgabewert "successful", zeigt an, dass die Einstellungen erfolgreich in der angegebenen Datei gespeichert wurden.

#### Argumente/Parameter

- **Destination:** Gibt den Pfad zur Datei an, in der die Einstellungen gespeichert werden. Der Verzeichnispfad muss existieren. Eine bereits vorhandene Datei mit gleichem Namen wird ohne Rückfrage überschrieben.

Erforderlich: ja

Standardwert: <keiner>

#### Beispiel

Backup-OmGateConfiguration -Identity omgate -Destination "c:\backup\omgate.cfg"



#### Anmerkung

Mit diesem Cmdlet wird ausschließlich die Konfiguration in einer lokalen Datei gespeichert. Lizenzen und Zertifikate werden nicht lokal gespeichert.

```

Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\rj2.HQ> Backup-OmGateConfiguration -Identity omgate -Destination "C:\temp\omgate$(Get-Date -Format yyyyMMddhhmm).cfg"
successful
PS C:\Users\rj2.HQ> Get-ChildItem -Path "c:\temp\*.cfg"

Verzeichnis: C:\temp

Mode                LastWriteTime         Length Name
----                -
-a---             07.07.2015   09:47         83379 omgate201507070947.cfg
-a---             07.07.2015   10:01         83379 omgate201507071001.cfg

PS C:\Users\rj2.HQ>

```

Detaillierte Hilfeinformationen erhalten Sie über die Befehlszeile:

```
Get-Help Backup-OmGateConfiguration -Full
```

## Restore-OmGateConfiguration

Dieses Cmdlet ersetzt die aktuelle Konfiguration des OMG durch die im angegebenen Pfad.

Der Rückgabewert "successful", zeigt an, dass die Einstellungen aus der angegebenen Datei erfolgreich im OMG gespeichert wurden.

### Argumente/Parameter

- **Destination:** Gibt den Pfad zur Datei an, in der die Einstellungen gespeichert sind. Die Datei muss existieren.

Erforderlich: ja

Standardwert: <keiner>

### Beispiel

```
Restore-OmGateConfiguration -Identity omgate -Destination "c:\backup\omgate.cfg"
```



#### Hinweis

Die Einstellungen in der gesicherten Konfigurationsdatei können Sie mit Hilfe des OfficeMaster Gate Konfigurationsprogramms (im Offline-Modus) betrachten und anpassen.



#### Warnung

Die Konfiguration wird ohne weitere Nachfrage überschrieben. Führen Sie dieses Cmdlet nur aus, wenn sichergestellt ist, dass die Konfiguration im angegebenen Pfad kein ungewolltes Verhalten des OMG herbeiführt.

Nachdem die Konfiguration erfolgreich übertragen wurde, ist ein Neustart des OMG erforderlich, um sicherzustellen, dass die neuen Einstellungen wirksam werden.

```

Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\rj2.HQ> Restart-OmGate -Identity 10.2.19.17 -Quick
Shutting down system logger: OK
Starting system logger: OK
OMG: Get kernel commandline OK
OMG: Running on a i386 OK
OMG: find number of channels OK
OMG: System without hardware just loop devices INFO
OMG: hostname OMGU93317 INFO
Shutting down system logger: OK
Starting system logger: OK
Stopping snmpd: OK
Starting snmpd: OK
Loop only
OMG: Start factory reset daemon OK
OMG: Start snfsserver OK
OMG: Start udpsnmpd OK
OMG: Start omgsnmp OK
Shutting down ntpd: OK
Hwlock set
Starting ntpd: OK
OMG: Start omgstart OK
PS C:\Users\rj2.HQ>

```

Detaillierte Hilfeinformationen erhalten Sie über die Befehlszeile:

```
Get-Help Restore-OmGateConfiguration -Full
```

## Restart-OmGate

Dieses Cmdlet initiiert den Neustart des Betriebssystems bzw. der Dienste des OMG.

Der Neustart der Dienste ist für die meisten Konfigurationsänderungen ausreichend. Wurden tiefgreifende Änderungen an der Konfiguration vorgenommen (z.B. Änderung der IP-Adresse, Datum/Uhrzeit) ist ein Neustart des Betriebssystems erforderlich.

### Argumente/Parameter

- Quick:** Wenn auf `$true` gesetzt, werden nur die OMG Dienste neu gestartet.
 

Erforderlich:	nein
Standardwert:	<code>\$false</code> , es wird ein Neustart des Betriebssystems des OMG durchgeführt.
- Wait:** Wenn `Quick` nicht bzw auf `$false` gesetzt ist und `Wait` auf `$true` gesetzt ist, wird bis zu 150 Sekunden gewartet, um sicherzustellen, dass der Neustart abgeschlossen ist.
 

Erforderlich:	nein
Standardwert:	<code>\$false</code>
- PowerOff:** Wenn auf `$true` gesetzt, wird das Betriebssystem des OMG komplett heruntergefahren und das Gerät ausgeschaltet. Ein physischer Zugriff darauf ist zwingend sicherzustellen, damit das Gerät nach den Wartungsarbeiten wieder eingeschaltet werden kann.
 

Erforderlich:	nein
Standardwert:	<code>\$false</code>

### Beispiel

### Restart-OmGate -Identity omgate



#### Warnung

Der Neustart wird ohne weitere Nachfrage durchgeführt. Führen Sie dieses Cmdlet nur aus, wenn sichergestellt ist, dass die Konfiguration des OMG korrekt ist und aktuell keine betriebsrelevanten Prozesse ausgeführt werden.

Der Neustart eines OMG sollte nicht durchgeführt werden, wenn nicht sichergestellt worden ist, dass ein physischer Zugriff darauf möglich ist.

Detaillierte Hilfeinformationen erhalten Sie über die Befehlszeile:

Get-Help Restart-OmGate -Full

## 3.5 Commandlets

Die im PowerShell Modul OfficeMaster enthaltenen Commandlets sind folgende (auf die Nennung der Standardargumente wird aus Gründen der Übersichtlichkeit verzichtet).

Detaillierte Hilfeinformationen erhalten Sie jeweils über die Befehlszeile:

Get-Help *Cmdlet* -Full

### Backup-OmGateConfiguration

Sichert die Konfiguration des OMG in einer lokalen Datei (siehe oben).

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

### Clear-OmCertificateRequest

Löscht die mit *Initialize-OmCertificateRequest* erstellte Zertifikatsanforderung.

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

### Close-OmSyslogFirewall

Entfernt Einstellungen aus der Windows Firewall, die es erlauben, dass der Syslogdienst TCP und UDP Nachrichten empfängt (Windows Administrator Rechte erforderlich)

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Nein
Unterstützt Standardargumente	Nein

---

### Export-OmZmqKeys

Dieses Cmdlet muss mindestens einmal ausgeführt werden, um die Authentifizierung des Benutzers mittels ZeroMQ-Schlüssel zu gewährleisten.

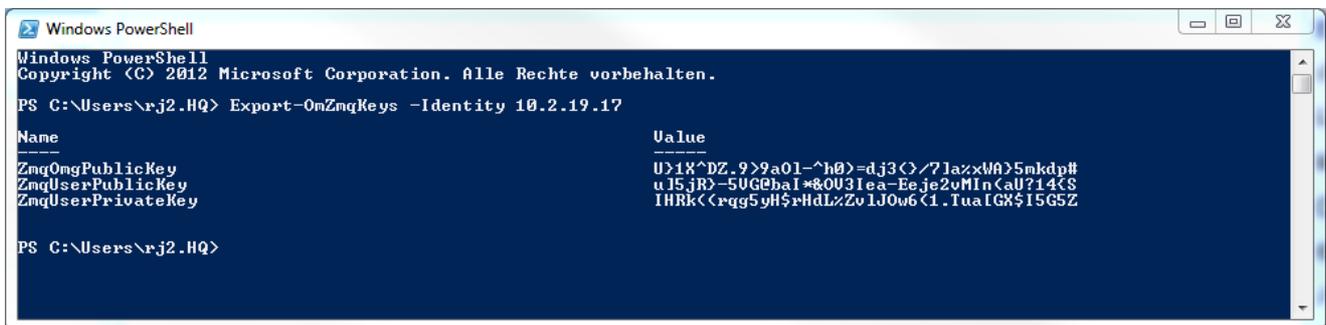
Als Rückgabewert erhalten Sie eine Liste von Paaren mit eindeutigen Bezeichnern und den entsprechenden ZeroMQ-Schlüssel als Wert.

#### Argumente/Parameter

- **Save:** optionaler Parameter, der angibt, ob die erhaltenen Werte für die weitere Verwendung in der Registry unter *HKLM\Software\Ferrari\SBA* gespeichert werden sollen (erfordert administrative Rechte zum Schreiben der Registry). Standardwert: \$False

#### Beispiel

```
Export-OmZmqKeys -Identity omgate
```



```

Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\rj2.HQ> Export-OmZmqKeys -Identity 10.2.19.17

Name                                     Value
----                                     -
ZmqOmgPublicKey                         U>1X^DZ.9>9a01-^h0)=dj3C/7IazxWA>5mkdp#
ZmqUserPublicKey                        u15jR>-5UG@baI*%0U3Iea-Eeje2vMInCaU?14<S
ZmqUserPrivateKey                       IHRk<<rqq5yH$rHdLzZv1J0w6<1.TuaIGX$15G5Z

PS C:\Users\rj2.HQ>

```

---

### Get-OmAllBoxes

Findet alle OfficeMaster Gate im eigenen Netzwerk und gibt deren Seriennummer und IPv4 Adresse zurück.

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Nein
Unterstützt Standardargumente	Nein

---

### Get-OmCertificateRequest

Liest die Zertifikatsanforderung vom OMG, die mit *Initialize-OmCertificateRequest* erstellt wurde. Die zurück gegebene Zeichenkette muss einer Zertifizierungsstelle angegeben werden, damit ein gültiges Zertifikat zur verschlüsselten Kommunikation

über TLS.

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

### Get-OmDateTime

Gibt aktuelles Datum und Uhrzeit des OMG zurück (als DateTime).

Unterstützt ZeroMQ:	Ja
Erfordert SSH Authentifizierung	Nur wenn nicht über ZeroMQ
Unterstützt Standardargumente	Ja

### Get-OmEepromInfos

Gibt Inhalt des EEPROM des OMG zurück.

Unterstützt ZeroMQ:	Ja
Erfordert SSH Authentifizierung	Nur wenn nicht über ZeroMQ
Unterstützt Standardargumente	Ja

### Get-OmEnvironmentVariable

Gibt den Wert der angegebenen Umgebungsvariable des OMG zurück.

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Nur wenn nicht über ZeroMQ
Unterstützt Standardargumente	Ja

### Argumente/Parameter

- **Variable:** Name der Variable, dessen Wert zurück gegeben werden soll.

### Get-OmGateBaseConfiguration

Liest die Basiskonfiguration des OMG aus und gibt sie zurück (PriType, NumberOfPri, NumberOfBri, Channel counts and IP Address)

Unterstützt ZeroMQ: Nein

Erfordert SSH Authentifizierung Ja

Unterstützt Standardargumente Ja

### Get-OmGateBaseDialplan

Gibt den Basiswählplan des OMG zurück, sofern dieser nicht durch das OfficeMaster Gate Konfigurationsprogramm geändert wurde

Unterstützt ZeroMQ: Nein

Erfordert SSH Authentifizierung Ja

Unterstützt Standardargumente Ja

### Get-OmInboundCallResult

Gibt das Ergebnis des Test für eingehende Rufe zurück. Ein Test wird über *Test-OmInboundCall* initialisiert.

Unterstützt ZeroMQ: Nein

Erfordert SSH Authentifizierung Ja

Unterstützt Standardargumente Ja

### Get-OmInterfaceCallState

Gibt den Status der Kommunikations-Schnittstellen und Rufe zurück (siehe oben).

Unterstützt ZeroMQ: Ja

Erfordert SSH Authentifizierung Nur wenn nicht über ZeroMQ

Unterstützt Standardargumente Ja

### Get-OmInterfaceChannels

Gibt die Namen der B-Kanäle der Kommunikations-Schnittstellen zurück.

Unterstützt ZeroMQ: Ja

Erfordert SSH Authentifizierung Nur wenn nicht über ZeroMQ

Unterstützt Standardargumente Ja

---

### Get-OmInterfaceState

Gibt den Status der Kommunikations-Schnittstellen zurück.

Unterstützt ZeroMQ: Ja

Erfordert SSH Authentifizierung Nur wenn nicht über ZeroMQ

Unterstützt Standardargumente Ja

---

### Get-OmIpAddress

Gibt die IPv4 Adresse eines OMG zurück. Dies kann nützlich sein, wenn das OMG die Netzwerkkonfiguration von einem DHCP Server bezieht und Sie nur die Seriennummer des OMG kennen.

Unterstützt ZeroMQ: Ja

Erfordert SSH Authentifizierung Nur wenn nicht über ZeroMQ

Unterstützt Standardargumente Ja

---

### Get-OmRealHardwareInfo

Gibt Informationen zum Prozessor des OMG zurück.

Unterstützt ZeroMQ: Ja

Erfordert SSH Authentifizierung Nur wenn nicht über ZeroMQ

Unterstützt Standardargumente Ja

---

### Get-OmTimeZoneNames

Gibt eine Sammlung von Namen und Identifikatoren von Zeitzonen zurück, wie sie für die Konfiguration von Datum und Uhrzeit des OMG weiter verwendet werden können.

Unterstützt ZeroMQ: Nein

Erfordert SSH Authentifizierung Nein

Unterstützt Standardargumente Nein

---

### Get-OmVoipConfiguration

Liest die VoIP spezifischen Konfigurationseinstellungen vom OMG und gibt sie zurück (VAD, PacketTime, EchoCanceller, EarlyMedia, EarlyProgress, SRTP, Codecs)

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

---

### Import-OmCertificate

Überträgt das angegebene Zertifikat zum OMG. Zuvor muss eine Zertifikatsanforderung mittels *Initialize-OmCertificateRequest* erstellt werden und diese Zertifikatsanforderung zur Zertifizierungsstelle übertragen werden, um das Zertifikat zu erhalten.

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

### Argumente/Parameter

- **CertificateFile:** Gibt den lokalen Pfad zur Zertifikatsdatei an (\*.cer,\*.pem,\*.p7b)

---

### Import-OmRootCertificate

Überträgt das angegebene Stammzertifikat (Rootcertificate) zum OMG

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

### Argumente/Parameter

- **CertificateFile:** Gibt den lokalen Pfad zur Zertifikatsdatei an (\*.cer,\*.pem,\*.p7b)

---

### Import-OmSshKey

Dieses Cmdlet muss mindestens einmal ausgeführt werden, um die Authentifizierung des Benutzers mittels SSH-Schlüssel zu gewährleisten. Dafür müssen sowohl öffentlicher als auch privater SSH-Schlüssel in Dateien vorliegen.

Der öffentliche Schlüssel aus der lokalen Datei, wird der Datei `.ssh/authenticated_keys` im Benutzer-Verzeichnis im OMG angefügt.

Der Rückgabewert „Success“ zeigt ein erfolgreiches Hinzufügen des SSH-Schlüssel an.

### Argumente/Parameter

- **KeyFile:** Gibt den lokalen Pfad zur Datei an, in dem der öffentliche Schlüssel des Benutzers gespeichert ist.
- **UserName:** Name des Benutzers, der sich einloggt (siehe oben). *UserName* ist für dieses Cmdlet ein erforderliches Argument
- **UserPassword:** Passwort des Benutzers, der sich einloggt (siehe oben). *UserPassword* ist für dieses Cmdlet ein erforderliches Argument

### Beispiel

```
Import-OmSshKey -Identity omgate -KeyFile 'c:\keys\sshkey.pub' -UserName root -UserPassword omc
```

```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\rj2.HQ> Import-OmSshKey -Identity omgate -KeyFile 'C:\Users\rj2.HQ\Documents\ssh public key' -UserName root -UserPassword omc
Success
PS C:\Users\rj2.HQ>
```

Detaillierte Hilfeinformationen erhalten Sie über die Befehlszeile:

```
Get-Help Import-OmSshKey -Full
```

Nachdem der öffentliche Schlüssel des Benutzers im OMG hinterlegt ist, können Sie weitere Aufrufe von Cmdlets mittels Benutzername und privatem SSH-Schlüssel durchführen. Den Benutzernamen und den Pfad zur Schlüsseldatei können Sie dabei sowohl als Cmdlet-Argumente angeben als auch in PowerShell-Variablen wie auch der Registry ablegen.

Die Angabe eines falschen privaten Schlüssel wird mit einer entsprechenden Fehlermeldung quittiert.

```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\rj2.HQ> Get-OmVersion -Identity omgate

Name                               Value
----                               -
Modules                             4.0-212.develop.e16
UserFS                               4.0-222.develop.e16

PS C:\Users\rj2.HQ> Get-OmProductInfo -Identity omgate -UserName root -UserKeyFile 'C:\Users\rj2.HQ\Documents\ssh private key.ppk'

LicenseName : OfficeMaster Gate UC
PartNo      : 22114
ProductCode : 139
Prefix      : OPR02
Product     :

PS C:\Users\rj2.HQ> Get-OmRealHardwareInfo -Identity omgate -UserName root -UserKeyFile 'C:\Users\rj2.HQ\Documents\ssh wrong key.ppk'
Get-OmRealHardwareInfo : cannot restore key pair.
In Zeile:1 Zeichen:1
+ Get-OmRealHardwareInfo -Identity omgate -UserName root -UserKeyFile 'C:\Users\rj...
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [Get-OmRealHardwareInfo], ArgumentException
+ FullyQualifiedErrorId : System.ArgumentException,OmgsPowerShell.GetOmRealHardwareInfo

PS C:\Users\rj2.HQ>
```

---

### Initialize-OmCertificateRequest

Erstellt eine Zertifikatsanforderung für verschlüsselte Kommunikation am OMG

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

#### Argumente/Parameter

- **CommonName:** Zertifikatselement: Common Name
- **EmailAddress:** Zertifikatselement: Email Adresse (optional)
- **CountryName:** Zertifikatselement: Länderkennzeichen (DE, US) (optional)
- **StateProvince:** Zertifikatselement: State or Province
- **Locality:** Zertifikatselement: Locality name
- **OrgName:** Zertifikatselement: Organisation Name
- **OrgUnit:** Zertifikatselement: Organisation Unit

---

### Invoke-OmCommand

Führt ein beliebiges Kommando über die gesicherte Verbindung am OMG aus und gibt den Rückgabewert des Kommandos zurück.

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

#### Argumente/Parameter

- **Command:** Kommando das ausgeführt werden soll.

---

### Open-OmSyslogFirewall

Stellt die Windows-Firewall so ein, dass TCP/UDP Kommunikation zum Syslogdienst möglich sind. (Administrator Rechte notwendig)

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Nein
Unterstützt Standardargumente	Nein

---

### Read-OmFile

Überträgt eine beliebige Datei vom OMG zum lokalen Computer.

Unterstützt ZeroMQ: Nein

Erfordert SSH Authentifizierung Ja

Unterstützt Standardargumente Ja

### Argumente/Parameter

- **LocalFile:** Dateipfad am lokalen Computer, wo die Datei erstellt werden soll.
- **RemoteFile:** Dateipfad am OMG, die auf den lokalen Computer geschrieben werden soll.

---

### Remove-OmFile

Löscht eine Datei am OMG.

Unterstützt ZeroMQ: Nein

Erfordert SSH Authentifizierung Ja

Unterstützt Standardargumente Ja

### Argumente/Parameter

- **RemoteFile:** Dateipfad am OMG, wo die Datei gelöscht werden soll

---

### Restart-OmGate

Startet das Betriebssystem des OfficeMater Gate neu (siehe oben).

Unterstützt ZeroMQ: Nein

Erfordert SSH Authentifizierung Ja

Unterstützt Standardargumente Ja

---

### Restore-OmGateConfiguration

Stellt die Konfiguration eines OMG aus einem Backup wieder her (siehe oben).

Unterstützt ZeroMQ: Nein

Erfordert SSH Authentifizierung Ja

Unterstützt Standardargumente Ja

---

### Set-OmDateTime

Stellt die aktuelle Uhrzeit und Datum am OMG ein. Es ist entweder Date und Time oder DateTime als Argument anzugeben. Ist DateTime angegeben hat dieses Vorrang vor Date und Time.

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

### Argumente/Parameter

- **Date:** Datum im Format MM/DD/YY
- **Time:** Uhrzeit im Format HH:MM:SS
- **DateTime:** Datum/Uhrzeit als DateTime Objekt, die gesetzt werden.
- **NtpServer:** Adresse eines NTP Server (z.B. poo.ntp.org), wenn Datum und Uhrzeit automatisch ermittelt werden soll. (optional)
- **Timezone:** Name der Zeitzone für die Uhrzeit, wie sie von *Get-OmTimeZoneNames* zurückgegeben wird.
- **PackagesPath:** Pfad zum Verzeichnis in dem sich die Konfigurationsdateivorlage master.cfg befindet. (optional)

---

### Set-OmGateBaseConfiguration

Erstellt eine grundsätzliche Basiskonfiguration aus einer Vorlage und den übergebenen Parametern und überträgt sie zum OMG.

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

### Argumente/Parameter

- **PriType:** Art des S2M Anschlusses (E1 oder T1)
- **NumberOfPri:** Anzahl der S2M-Anschlüsse die benutzt werden sollen (0-2)
- **NumberOfBri:** Anzahl der S0-Anschlüsse die benutzt werden sollen (0-4)
- **Pri1Channels:** Anzahl der B-Kanäle die am ersten S2M-Anschluss benutzt werden sollen (E1: 1-30, T1: 1-23)
- **Pri2Channels:** Anzahl der B-Kanäle die am zweiten S2M-Anschluss benutzt werden sollen (E1: 1-30, T1: 1-23)
- **SyslogIP:** Pfad zum Verzeichnis in dem sich die Konfigurationsdateivorlage master.cfg befindet. (optional)
- **UseDhcp:** Gibt an, dass OMG die Netzwerkkonfiguration über DHCP beziehen soll. Muss \$False gesetzt werden, wenn eine statische Netzwerkkonfiguration verwendet werden soll.
- **IpAddress:** IP Adresse des OMG

- **IpMask:** Subnetzmaske passend zur IP Adresse
- **IpGateway:** Adresse des Netzwerkgateway
- **IpDns1:** Adresse des primären DNS Server
- **IpDns2:** Adresse des sekundären DNS Server
- **PackagesPath:** Pfad zum Verzeichnis in dem sich die Konfigurationsdateivorlage master.cfg befindet.

---

### Set-OmGateBaseDialplan

Erstellt einen grundsätzlichen Wählplan auf Basis einer Vorlage und den übergebenen Parametern und überträgt sie zum OMG.

Unterstützt ZeroMQ:                   Nein

Erfordert SSH Authentifizierung   Ja

Unterstützt Standardargumente   Ja

### Argumente/Parameter

- **ExtensionLength:** Länge der Durchwahl
- **E164Prefix:** E.164 Nummernpräfix (ohne Erweiterung)
- **CountryCode:** Telefonie Landeskennziffer (z.B. 49 für Deutschland)
- **MediationServer:** Name (FQDN) des MediationServer
- **Analog1:** Rufnummer des ersten Analoganschluss
- **Analog2:** Rufnummer des zweiten Analoganschluss
- **Analog3:** Rufnummer des dritten Analoganschluss
- **Analog4:** Rufnummer des vierten Analoganschluss
- **TLS:** Gibt an ob TLS (verschlüsselte Kommunikation zum MediationServer) verwendet werden soll (Standard: \$False)
- **PackagesPath:** Pfad zum Verzeichnis in dem sich die Konfigurationsdateivorlage master.cfg befindet.

---

### Set-OmVoipConfiguration

Setzt VoIP Konfigurationsparameter des OMG

Unterstützt ZeroMQ:                   Nein

Erfordert SSH Authentifizierung   Ja

Unterstützt Standardargumente   Ja

### Argumente/Parameter

- **Vad:** Gibt an ob Spracherkennung (Voice activity detection) aktiv sein soll (on/off)

- **PacketTime:** Packet size (20ms)
- **EchoCanceller:** Gibt an ob Echounterdrückung aktiv sein soll (on/off)
- **EarlyMedia:** Gibt an ob EarlyMedia unterstützt sein soll (on/off)
- **EarlyProgress:** Gibt an ob EarlyProgress unterstützt sein soll (on/off)
- **Srtp:** Gibt an ob SRTP unterstützt sein soll (supported,disabled,required)
- **Codecs:** stellt zu verwendende Sprachcodecs (alaw, ulaw, beide) ein (a,u,au,ua)

---

### Stop-OmnInboundCall

Beendet einen Test für eingehende Rufe.

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

---

### Test-OmnInboundCall

Startet einen Test für eingehende Rufe. Der nächste Anruf der am OMG eingeht wird speziell behandelt.

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

---

### Test-OmnOutboundCall

Starten einen Test für ausgehende Rufe (siehe oben).

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja
Unterstützt Standardargumente	Ja

---

### Test-OmnRootCertificate

Testet die verschlüsselte Kommunikation zur angegebenen Gegenstelle mittels des eingestellten Root-Zertifikats.

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Ja

Unterstützt Standardargumente Ja

#### Argumente/Parameter

- **Host:** IPv4-Adresse oder FQDN eines Computers zu dem das OMG eine Verbindung aufbauen soll.
- **Port:** TCP/IP-Port zu dem das OMG eine Verbindung aufbauen soll

---

#### Write-OmFile

Überträgt eine beliebige Datei vom lokalen Computer auf das OMG.

Unterstützt ZeroMQ: Nein

Erfordert SSH Authentifizierung Ja

Unterstützt Standardargumente Ja

#### Argumente/Parameter

- **LocalFile:** Dateipfad am lokalen Computer, die auf das OMG geschrieben werden soll.
- **RemoteFile:** Dateipfad am OMG, wo die Datei erstellt werden soll

---

### 3.6 Info Commandlets

Folgende Cmdlets geben Informationen zum Zustand bzw. allgemeinen Status des OfficeMaster Gate.

Unterstützt ZeroMQ:	Ja
Erfordert SSH Authentifizierung	Nur wenn nicht über ZeroMQ
Unterstützt Standardargumente	Ja

---

### Allgemeine Info Cmdlet-Argumente/Parameter

Die Info Cmdlets kennen alle die gleichen Befehlszeilenargumente, die beim Aufruf übergeben werden können.

#### RawFormat

Legt fest in welchem Format die Rückgabewerte formatiert sind. Dadurch kann eine automatisierte Weiterverarbeitung vereinfacht werden. Gültige Werte sind: xml, json, txt, lua

Erforderlich:	Nein
Standardwert:	xml

#### RawOutput

Gibt an, ob die Rückgabewerte im angegebenen oder dem im Cmdlet präferierten Format (Standard) zurückgegeben werden.

Erforderlich:	Nein
Standardwert:	\$False

#### LuaFilter

Ein regulärer Ausdruck um Lua Format zum filtern der Rückgabewerte (nur verwendbar mit RawOutput=\$True)

Erforderlich:	Nein
Standardwert:	<keiner>

---

### Get-OmFileSystemInfo

Gibt Informationen über die Menge der gespeicherten Daten im Dateisystem des OMG zurück (siehe oben)

---

### Get-OmNetworkInfo

Gibt Informationen zu den Netzwerkschnittstellen des OMG zurück.

---

### Get-OmProductInfo

Gibt kaufmännische Informationen zum Produkt OMG zurück.

---

**Get-OmSystemInfo**

Gibt umfangreiche Informationen zum Produkt, Netzwerk, Hardware und Version des OMG zurück.

---

**Get-OmVersion**

Gibt die Versionsnummern der einzelnen Betriebssystem- (Firmware-) Elemente zurück.

---

### 3.7 Lizenz Commandlets

Folgende Cmdlets geben Informationen zur eingesetzten Lizenz am OfficeMaster Gate zurück und werden in der Regel ausschließlich von OfficeMaster Produkten verwendet, da sie ein tiefes technisches Verständnis der Produkte voraussetzen.

Unterstützt ZeroMQ:	Ja
Erfordert SSH Authentifizierung	Nur wenn nicht über ZeroMQ
Unterstützt Standardargumente	Ja

---

### Allgemeine Lizenz Cmdlet-Argumente/Parameter

Die Path Cmdlets kennen in der Regel alle die gleichen Befehlszeilenargumente, die beim Aufruf übergeben werden können.

#### Path

Pfad zum Lizenzverzeichnis im OMG. Sollte nur angegeben werden, wenn keine kumulierten Lizenzinformationen abgefragt werden sollen.

Erforderlich:	Nein
Standardwert:	/data/

#### File

Name einer Lizenzdatei (Wildcards möglich) im OMG. Sollte nur angegeben werden, wenn keine kumulierten Lizenzinformationen abgefragt werden sollen.

Erforderlich:	Nein
Standardwert:	<keiner>

#### TagModuleCode

Die Identifikation einer bestimmten Produktfunktion die durch Lizenzierung gesteuert werden kann.

Erforderlich:	Nein
Standardwert:	0

---

### Get-OmlsLicenseFree

Gibt an ob sich das OMG im Vorauslieferungszustand befindet.

---

### Get-OmLicenseDateLimit

Gibt die Datumsinformation zurück (als DateTime Objekt), zu der die Lizenz ihre Gültigkeit verliert (z.B. Testlizenz). Handelt es sich um eine Lizenz ohne Ablaufdatum wird DateTime.Max zurückgegeben.

---

### Get-OmLicenseEnabled

## UNIFIED COMMUNICATIONS

www.ferrari-electronic.de

Copyright © 2015 Ferrari electronic AG

**Ferrari**  
electronic

Gibt zurück, ob eine bestimmte Funktion lizenziert ist.

---

#### **Get-OmLicenseMaxLines**

Gibt zurück für wie viele Kanäle eine bestimmte Funktion lizenziert ist.

---

#### **Get-OmLicenseMaxUsers**

Gibt zurück für wie viele Benutzer eine bestimmte Funktion lizenziert ist.

---

#### **Get-OmLicenseSerialCode**

Gibt die kaufmännischen Seriennummern der Lizenzen zurück.

---

#### **Get-OmLicenseSerialNumber**

Gibt die kumulierte Seriennummer der Lizenz zurück.

---

#### **Get-OmLicenseTag**

Gibt durch Angabe exakter Abfrageparameter sehr spezifische Lizenzinformation zurück.

---

### 3.8 Dateilogging Commandlets

Um einen immer gleichen Aufbau einer Datei zu gewährleisten, die Loginformationen enthält, werden folgende Cmdlets zum Schreiben von Loginformationen im PowerShell Module OfficeMaster bereitgestellt.

Unterstützt ZeroMQ:	Nein
Erfordert SSH Authentifizierung	Nein
Unterstützt Standardargumente	Nein

---

### Allgemeine Logging Cmdlet-Argumente/Parameter

Die Logging Cmdlets kennen alle die gleichen Befehlszeilenargumente, die beim Aufruf übergeben werden müssen.

#### Path

Kompletter Pfad zum Verzeichnis, in dem die Logdatei geschrieben wird. Das Verzeichnis muss bereits existieren.

Erforderlich:	Ja
Standardwert:	<keiner>

#### File

Name der Datei die geschrieben wird. Die Angabe muss ohne Dateierweiterung erfolgen, da sie als Basis dient und automatisch um eine Tagesziffer (1-7) und die Erweiterung .log erweitert wird. Existiert die Datei bereits, wird sie nicht neu erstellt sondern fortgesetzt, es sei denn sie ist älter als 6 Tage.

Erforderlich:	Ja
Standardwert:	<keiner>

#### Message

Text der in die Logdatei geschrieben wird.

Erforderlich:	Ja
Standardwert:	<keiner>

---

### Initialize-OmLog

Eröffnet eine neue Logsitzung, und schreibt dafür einige Einleitungszeilen in die Logdatei. Das Rufen dieses Cmdlet ist optional für die Erstellung von Logdateien.

---

### Get-OmLogCompleteFilePath

Gibt den kompletten Dateipfad, inklusive der automatischen an den Dateinamen angehängten Merkmale, zurück, unter dem die Logdatei im Dateisystem zu finden ist.

---

### **Write-OmLogDebugMessage**

Schreibt eine Nachricht vom Typ *Debug* in die Logdatei.

---

### **Write-OmLogErrorMessage**

Schreibt eine Nachricht vom Typ *Exception* in die Logdatei.

---

### **Write-OmLogInfoMessage**

Schreibt eine Nachricht vom Typ *Info* in die Logdatei.

---

### **Write-OmLogMessage**

Schreibt eine Nachricht vom angegebenen Typ in die Logdatei.

#### **Argumente/Parameter**

- **LogLevel:** Typ der Nachricht, die in die Logdatei geschrieben wird. Gültige Werte sind: Debug, Verbose, Info, Warning, Exception
- 

### **Write-OmLogWarningMessage**

Schreibt eine Nachricht vom Typ *Warning* in die angegebene Logdatei.

---

### 3.9 Versionsinformations Commandlets

Die Einzelmodule der OfficeMaster PowerShell Moduls enthalten Cmdlets zum Abfragen der Cmdlet-Dateiversion und Dateipfad. Dies kann dabei hilfreich sein, zu identifizieren ob die Commandletdateien in der Version vorliegen, die eine bestimmte Funktion voraussetzt oder Konflikte bei Mehrfachinstallationen aufdecken.

Unterstützt ZeroMQ:                      Nein

Erfordert SSH Authentifizierung    Nein

Unterstützt Standardargumente    Nein

---

#### Get-OmCmdletAssemblyVersion

Gibt die Assembly-Versionsnummer der geladenen ompowershellcmdlet.dll zurück.

---

#### Get-OmCmdletFileVersion

Gibt die Datei-Versionsnummer der geladenen ompowershellcmdlet.dll zurück.

---

#### Get-OmCmdletInfo

Gibt Pfad und Versionsnummer der geladenen ompowershellcmdlet.dll zurück.

---

#### Get-OmCmdletPath

Gibt Pfad der geladenen ompowershellcmdlet.dll zurück.

---

#### Get-OmCmdletVersion

Gibt die Datei-Versionsnummer der geladenen ompowershellcmdlet.dll zurück.

---

#### Get-OmgCmdletAssemblyVersion

Gibt die Assembly-Versionsnummer der geladenen omgpowershellcmdlet.dll zurück.

---

#### Get-OmgCmdletFileVersion

Gibt die Datei-Versionsnummer der geladenen omgpowershellcmdlet.dll zurück.

---

#### Get-OmgCmdletInfo

Gibt Pfad und Versionsnummer der geladenen omgpowershellcmdlet.dll zurück.

---

#### Get-OmgCmdletPath

Gibt Pfad der geladenen omgpowershellcmdlet.dll zurück.

---

#### Get-OmgCmdletVersion

Gibt die Datei-Versionsnummer der geladenen omgpowershellcmdlet.dll zurück.

## 4. OmgUtils Typen

### 4.1 FileSystemInfo

Datentyp mit Auslastungs-Informationen zum OMG Dateisystem. Ein Objekt entspricht einem Dateisystem-Punkt. Dieser Datentyp findet Anwendung im Cmdlet **Get-OmFileSystemInfo**.

Membername	Datentyp	Funktion
MountPoint	String	Name des Dateisystem-Punkts
FileSystem	String	Quelle des Dateisystem-Punkts
Size	String	Absoluter Wert des gesamten Speicherplatzes
Used	String	Absoluter Wert der verwendeten Speicherplatz
Available	String	Absoluter Wert des verfügbaren Speicherplatz
PercentUsed	String	Relativer Wert des verwendeten Speicherplatzes in Bezug auf den gesamten Speicherplatz
ToString()	String	mount 'MountPoint' uses Used(PercentUsed) of Size (Available available)

Einheiten für Größenangaben des Speicherplatzes

- G: Gigabyte
- M: Megabyte
- K: Kilobyte
- %: Prozent

### 4.2 InterfaceState

Datentyp mit Informationen zu ISDN, Analog und virtuellen OMG Kommunikationsschnittstellen (D-Kanäle). Der Status von B-Kanälen wird nur für diejenigen angezeigt, die seit dem letzten Neustart des OMG verwendet wurden. Dieser Datentyp findet Anwendung in den Cmdlets **Get-OmInterfaceState**, **Get-OmInterfaceCallState**, **Get-OmInterfaceChannels**.

Membername	Datentyp	Funktion
Interface	String	Name der Schnittstelle (D-Kanal)
State	String	Status der Schnittstelle

Channels	List<String>	Liste mit Namen der B-Kanäle dieser Schnittstelle
ChannelStates	InterfaceState.ChannelState	Namen und Status der verwendeten B-Kanäle
CallCount	Int	Anzahl der aktuell aktiven Verbindungen
ToString()	String	Interface=State

Statuswerte:

- disabled: Die Schnittstelle ist in der Konfiguration deaktiviert
- L1UPL2UP bzw DN: Layerstatus, UP=gut, DN=nicht gut ← Wartungsbedarf

Aufbau/Syntax des Schnittstellen-Namens (z.B.: 1.a-PRI):

- Nummer der Schnittstelle (Nummer a/b/c/d)
- Typ der Schnittstelle:
  - PRI: S2M (30 B-Kanäle)
  - BRI: S0 (2 B-Kanäle)
  - FXS: Analog (1 Kanal)
  - PCM: virtuell (30 B-Kanäle)

Aufbau/Syntax der Namen der B-Kanalliste (1.a/B7/S2M/2xG704IDT\_E1):

- Nummer der Schnittstelle (Nummer a/b/c/d)
- Name des B-Kanal (durchnummeriert, 1 basiert)
- Typ der Schnittstelle (S2M/S0/FXS/PCM)
- Modus

### 4.3 InterfaceState.ChannelState

Datentyp mit Informationen zu OMG Kommunikationskanälen der Schnittstellen. Dieser Datentyp findet Anwendung im Cmdlet `Get-OmInterfaceCallState`.

Membername	Datentyp	Funktion
Channel	String	Name des Kanals
State	String	Status des Kanals
ToString()	String	Channel=State

Aufbau/Syntax des Kanalname (z.B.: 1.a-PRI-B07):

- Nummer der Schnittstelle (Nummer a/b/c/d)
- Typ der Schnittstelle (PRI/BRI/FXS/PCM)

- Name des B-Kanal (durchnummeriert, 1 basiert)

Kanalstatuswerte:

- stopped: Verbindung ist aktuell nicht aktiv
- end\_of\_fax: Fax Übertragung beendet
- end\_of\_sms: SMS Übertragung beendet
- disconnected: Verbindung ist aktuell nicht aktiv
- active: Verbindung ist aktuell aktiv

---

#### 4.4 NameValue

Einfacher Datentyp mit Name-Wert Zuordnungspaaren. Dieser Datentyp wird in diversen Cmdlets als Rückgabewert, meist in Form einer Liste, verwendet. (z.B.: **Get-OmAllBoxes**, **Get-OmVersion**)

Membername	Datentyp	Funktion
Name	String	Name des Objekts
Value	String	Wert des Objekts
ToString()	String	Name=Value

## 5. Anwendungsbeispiele

### 5.1 Benutze ZeroMQ Verbindung zu OMG, statt SSH

Die Unterstützung des Cmdlet Zugriffs via ZeroMQ ist erst mit Cmdlet-Version 2.4 und OMG-Version 280 (siehe 3.2) außerhalb von Laborumgebungen sichergestellt. Zur Feststellung welche Versionen tatsächlich verwendet werden sollten Sie die Cmdlets `Get-OmVersion` (siehe 3.6) und `Get-OmCmdletVersion` (siehe 3.8) aufrufen.

#### Beispiel

```
Get-OmVersion -Identity omgate
```

#### Ausgabe

Name	Value
Modules	4.0-272.develop.el6
UserFS	4.0-272.develop.el6

#### Beispiel

```
Get-OmCmdletVersion
```

#### Ausgabe

```
2.4.0.0
```

Sind die Versionsvoraussetzungen erfüllt, können Sie über das Cmdlet `Export-OmZmqKeys` (siehe 3.5) das OMG anweisen, die benötigten ZeroMQ Authentifizierungsschlüssel zu erstellen bzw. auszugeben. Das erhaltene Schlüsseltrio wird für alle Cmdlets, die über ZeroMQ mit diesem OMG kommunizieren, benötigt. Der vierte Schlüssel zur erfolgreichen Authentifizierung und Entschlüsselung von ZeroMQ Nachrichten bleibt geheim im OMG gespeichert. Das `Export-OmZmqKeys` Cmdlet kommuniziert selbstverständlich über SSH mit dem OMG.

#### Beispiel

```
Export-OmZmqKeys -Identity omgate
```

#### Ausgabe

Name	Value
ZmqOmgPublicKey	OwGjtV]%Nk(^y!:4C:=iA^a%68F/^=b9(M6kCHz/
ZmqUserPublicKey	0kSwk71bntsEQ*a!{tBo<fAeA9{y2#r0L%cx[AS5
ZmqUserPrivateKey	gxRz{]/#M..#)\$8E?x}DGs[+j<>9=ARQJS1Kyj

Da das Exportieren der Schlüssel im Regelfall nur einmalig notwendig ist, und Sie die Cmdlets immer mit dem selben OMG verwenden, sollten Sie das Cmdlet `Export-OmZmqKeys` mit dem Parameter `-Save` ausführen, damit die Schlüssel in der Systemregistrierung unter `HKLM\Software\Ferrari\SBA` abgespeichert werden (Computer-Administratorrechte notwendig).

Sollten Sie die Speicherung des ZeroMQ-Schlüsseltrios in der Registry nicht wünschen, müssen Sie sie alternativ speichern, da sie für jeden Cmdlet Aufruf wieder als Parameter übergeben werden müssen bzw. alternativ aus PS-Sitzungsvariablen gelesen

werden. (siehe 3.2).

Zum Speichern der ZeroMQ-Schlüssel in PS-Sitzungsvariablen verwenden Sie folgendes Cmdlets.

### Beispiel

```
$zkeys = Export-OmZmqKeys -Identity omgate
foreach($k in $zkeys) {Set-Variable -Name "Omg${$k.Name}" -Value $k.Value}
Get-Variable Om*
```

### Ausgabe

Name	Value
----	----
OmgZmqOmgPublicKey	OwGjtV]%Nk(^y!:4C:=iA^a%68F/^=b9(M6kCHz/
OmgZmqUserPrivateKey	gxRz{]/#M.:#)\$8E?x}DGs[+/j<>9=ARQJS1Kyj
OmgZmqUserPublicKey	0kSwk71bntsEQ*a!{tBo<fAeA9{y2#r0L%cx[AS5

Nach Ausführung des Beispiels (die ersten beiden Zeilen genügen) stehen das Schlüsseltrio allen in dieser PS-Sitzung nachfolgend aufgerufenen Cmdlets zur Verfügung. Wird die PS-Sitzung geschlossen, werden die Variablen gelöscht. Werden Cmdlets in einer anderen PS Sitzung gerufen stehen die Variablen dort nicht zur Verfügung.

Der positive Effekt, den die verschlüsselte Kommunikation via ZeroMQ gegenüber der verschlüsselten Kommunikation via SSH zum OMG hat, ist die Geschwindigkeit in der die Anfragen vom OMG beantwortet werden, die insbesondere bei Überwachungsszenarios eine große Rolle spielen kann.

Nachfolgendes Beispiel verdeutlicht den Effekt sehr anschaulich. Die Ausführung des Cmdlet `Get-OmInterfaceState` (zum selben OMG, in der selben PS-Sitzung) dauert über SSH ungefähr 180 mal länger als über ZeroMQ.

### Beispiel (ZeroMQ)

```
$start=Get-Date;Get-OmInterfaceState omgate;$end=Get-Date;$timespan=(New-Timespan -Start $start -End $end).TotalMilliseconds;Write-Host $timespan
```

### Ausgabe

```
24,4099
```

### Beispiel (SSH)

```
$start=Get-Date;Get-OmInterfaceState omgate -PreferSSH;$end=Get-Date;$timespan=(New-Timespan -Start $start -End $end).TotalMilliseconds;Write-Host $timespan
```

### Ausgabe

```
4347,6274
```

Die Werte sind als Beispiele zu deuten und unterscheiden sich von Ausführung zu Ausführung. Eine Vielzahl von Einflüssen wirkt sich auf die Ausführungsgeschwindigkeit der Cmdlets aus, jedoch ist die Verbindungsart (ZeroMQ oder SSH) ein sehr wesentlicher Faktor.



### Hinweis

Nicht alle Cmdlets unterstützen die ZeroMQ-Verbindung zum OMG. Insbesondere Cmdlets die Schreibaktionen oder Konfigurationsänderungen vornehmen erfordern die SSH Verbindung (siehe 3.2)

---

## 5.2 Verwendung des Beispielskripts Monitor-OMGates.ps1

Cmdletversion 2.4, Firmware 280,

Configprogramm zum Erstellen der Gateways.txt

Parameter (smtp, log, timeout...)

# Abkürzungen

OMG – OfficeMaster Gate  
Cmdlet – Commandlet  
OM-Cmdlets – OfficeMaster PowerShell Commandlets  
Registry – Windows Registrierungsdatenbank  
HKLM – HKEY\_LOCAL\_MACHINE  
PS - PowerShell