



**DOAG München 2012**

**ORACLE**



**[Oracle & Powershell]**

**Marco Patzwahl**

---

# MuniQSoft GmbH

- ◆ **Gegründet: 1998**
- ◆ **Tätigkeitsbereiche:**
  - ▶ **Oracle Support**  
Hotline: Mo-Fr 8.00 – 18.00 Uhr  
Erweiterung um eine Rufbereitschaft auch am Wochenende möglich
  - ▶ **Oracle IT Consulting & Services**
  - ▶ **Oracle Schulungen (SQL, PL/SQL, DBA, APEX, B&R, ...)**
  - ▶ **Software-Lösungen**
  - ▶ **Oracle Lizenzen**

MuniQSoft GmbH  
Grünwalder Weg 13 a  
D-82008 Unterhaching  
[www.muniqsoft.de](http://www.muniqsoft.de)  
+49 89 6228 6789-0



# Einführung

- ◆ **Powershell ist eine Skriptsprache für Windows-Rechner**
- ◆ **Ergebnisse werden als Objekt mit Eigenschaften dargestellt**
- ◆ **Powershell ist bereits integriert in**
  - ▶ Win 2008
  - ▶ Win 7
- ◆ **und kann nachinstalliert werden für**
  - ▶ **ältere Windows Systeme (KB968930)**
  - ▶ **z.B. auch für Win 2003 !**



## Update für Windows XP (KB968930)

### Kurzbeschreibung

Das Windows Management Framework-Kernpaket stellt aktualisierte Verwaltungsfunktionen für IT-Spezialisten zur Verfügung.

# Einführung (f)

- ◆ **Die folgenden Seiten sollen Ihnen einen kurzen Überblick über die Funktionen und Möglichkeiten der Powershell bieten**
- ◆ **Der Fokus liegt hier in der Überwachung des Betriebssystems von Oracle aus und nicht in der Übertragung von Oracle Objekten (Tabellen, ...) in die Powershell**

# Aufruf von Powershell Skripten

## ◆ Powershell Fenster öffnen

- ▶ Start/Ausführen/powershell

Programme (5)

- ▶ powershell.exe
- ▶ Windows PowerShell (x86)
- ▶ Windows PowerShell ISE
- ▶ Windows PowerShell ISE (x86)
- ▶ Windows PowerShell Modules

## ◆ Aufruf aus dem Oracle Scheduler:

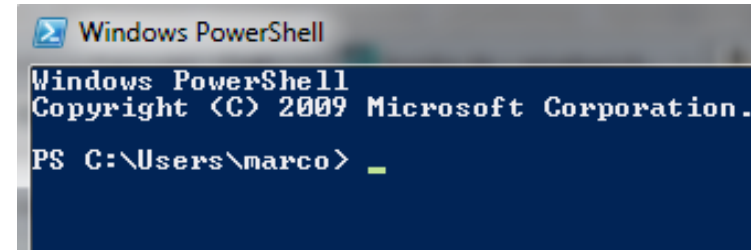
- ▶ `cmd /c powershell <cmd>`

## ◆ Aufruf aus der Dos-Box

- ▶ `powershell <cmd>`

## ◆ Endung der Powershell Dateien:

- ▶ `.ps1`



```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation.
PS C:\Users\marco> _
```

# Hilfe für Commandlets anzeigen

## ◆ Kurzhilfe anzeigen:

- ▶ `get-help <cmd>`

## ◆ Komplette Hilfe anzeigen:

- ▶ `get-help <cmd> -full`

## ◆ Beispiele für das Commandlet anzeigen:

- ▶ `get-help <cmd> -examples`

## ◆ Detaillierte Hilfe anzeigen:

- ▶ `get-help <cmd> -detailed`

## ◆ Hilfe für alle Befehle anzeigen:

- ▶ `get-help *`

# Powershell Pipes

- ◆ Pipes sind ein mächtiges Werkzeug zur Datenweitergabe oder Datenfilterung
- ◆ Ganze Arrays können via Pipes weitergegeben werden
- ◆ Beispiel:

- ▶ `$file="D:\Oracle\diag\rdbms\${ORACLE_SID}\${ORACLE_SID}\trace>alert_${ORACLE_SID}.log"`



- ▶ `$l = get-content $file | Select-String -pattern "ORA-|Checkpoint not"`

# Erforschen der Rückgabemenge

- ◆ Viele Commandlets besitzen eine umfangreiche Ergebnismenge mit diversen Spalten
- ◆ Zum Herausfinden, was von einem Objekt wie verwendet werden kann, benutzt man das Commandlet `get-member`

- ◆ Beispiel:

- ▶ `get-date | get-member`

- ◆ Ausgabe (gekürzt)

- ▶ `AddDays`      `Method`      `System.DateTime AddDays(double value)`
  - ▶ `Day`            `Property`    `System.Int32 Day {get;}`
  - ▶ `ToString`     `Method`      `string ToString(), string ...)`



# Erforschen der Rückgabemenge(f)

## ◆ Damit können Sie also verwenden:

- ▶ `(get-date) . adddays (1)`

```
Samstag, 14. Januar 2012 13:21:44
```

- ▶ `(get-date) . day`

```
13
```

- ▶ `(get-date) . tostring('HH:mm:ss')`

```
13:37:31
```

# Datumsformate

Datumsformat (get-date).tostring(...)	Beispielausgabe
dd.MM.yyyy	18.1.2012
dddd, dd MMMM yyyy	Dienstag, 18 Januar 2012
dddd, dd MMMM yyyy HH:mm	Dienstag, 18 Januar 2012 18:22
dddd, dd MMMM yyyy H:mm	Dienstag, 18 Januar 2012 18:22
dddd, dd MMMM yyyy h:mm:ss	Dienstag, 18 Januar 2012 6:22:30
dd.MM.yyyy HH:mm	18.01.2012 18:22
dd.MM.yyyy H:mm	18.1.2012 18:22
dd.MM.yyyy hh:mm:ss	18.1.2012 06:22:07
MMMM dd	Januar 18
yyyy'-'MM'-'dd'T'HH': 'mm': 'ss.fffffffK	2012-01-18T18:22:47.2640185+01:00
ddd, dd MMM yyyy HH': 'mm': 'ss 'GMT'	Mi, 18 Jan 2012 18:22:59 GMT
HH:mm	18:22
yyyy MMMM	2012 Januar
dddd, dd MMMM yyyy HH:mm:ss	Mitwoch, 18 Januar 2012 18:22:07

# Kleine Datumsarithmetik

## ◆ Aktuelles Datum:

▶ `Get-Date`

## ◆ Vor 5 Minuten

▶ `(Get-Date) .addminutes (-5)`

## ◆ In 12 Stunden

▶ `(Get-Date) .addHours (12)`

## ◆ Vor einem Tag

▶ `(Get-Date) .addDays (-1)`

## Weitere Optionen

`.AddDays`

`.AddHours`

`.AddMilliseconds`

`.AddMinutes`

`.AddMonths`

`.AddSeconds`

`.AddYears`

# Powershell Operatoren

## ◆ Wenn \$x=5 Ausgabe auf Terminal:

```
▶ if ($x -eq 5)
    {Write-Host "x=5"}
```

## ◆ Wenn \$x>1 dann Befehl ausführen

```
▶ If ($x -gt 1) { ...}
```

## ◆ Wenn \$x>1 und \$y<1 dann ...

```
▶ If ($x -gt 1 -AND $y -lt 1)
    { ...}
```

CMD	Operatoren
-eq	Equal to (=)
-lt	Less than (<)
-gt	Greater than (>)
-ge	Greater Equal (>=)
-le	Less than / equal (<=)
-ne	Not equal to (!=)
-not	Negation
!	Negation
-and	Und Verknüpfung
-or	Oder Verknüpfung

# Powershell String Operatoren

## ◆ Prüfen ob Variable i mit Tagesnamen beginnt

```
▶ if ($i -match  
"^[Mon|Tue|Wed|Thu|  
Fri|Sat|Sun] ")
```

## ◆ Prüfen ob i nicht mit Text ORA- beginnt

```
▶ if ($i -notmatch "^[ORA-]*")
```

## ◆ Strings verknüpfen:

```
▶ $string= "Hallo $name" + "Willkommen zur DOAG"
```

### String Operatoren

- match enthält
- notmatch enthält nicht
- like "a\*" beginnt mit
- contains Sucht in Array

# Powershell Sonderzeichen

- ◆ Wenn Sie in einem String einen Zeilenumbruch einfügen möchten:

- ▶ `$str="Übersicht: `n1.Start `n2.Ende"`
- ▶ `$str # String ausgeben`

```
Übersicht :  
1.Start  
2.Ende
```

- ◆ Soll ein PS-Kommando über mehrere Zeilen gehen, wird das Zeichen ``` verwendet:

- ▶ `get-process ``  
`| sort-Object cpu`

## Sonderzeichen

- ``0` Null
- ``a` Benachrichtigung
- ``b` Rückschritt
- ``f` Seitenvorschub
- ``n` Zeilenwechsel
- ``r` Wagenrücklauf
- ``t` Horizontaler Tabstopp
- ``v` Vertikaler Tabstopp

# Registrywerte lesen/schreiben

◆ \$OH="hk1m:\Software\Oracle\KEY\_OraDb11g\_home1"

◆ Werte des Keys ORACLE\_SID auslesen

▶ `get-itemproperty -path $OH -name ORACLE_SID |  
select ORACLE_SID`

◆ Neuen Key eintragen (Nur als Administrator)

▶ `new-itemproperty -path $OH -name NLS_DATE_FORMAT  
-value "DD.MM.YYYY HH24:MI:SS" -propertyType string`

◆ Key ändern

▶ `set-itemproperty -path $OH -name NLS_DATE_FORMAT  
-value "DD.MM.YYYY HH24:MI" -type string`

# Fehlerbehandlung

- ◆ Mit dem Trap Befehl können Fehler abgefangen werden
- ◆ Beispiel: Fehler ausgeben und weitermachen:

- ▶ 

```
trap {  
    write-host $_.Exception.Message  
    continue}
```

- ▶ Oder etwas ausführlicher

- ▶ 

```
trap {  
    write-host  
    write-error $("Fehler aufgetreten:" +  
    $_.Exception.GetType().FullName);  
    write-error $("Fehler aufgetreten:" +  
    $_.Exception.Message);  
    continue; }
```



# Beispiel: Oracle Dienste starten/prüfen

## ◆ Listener neu starten, falls er gestoppt wurde

```
▶ get-service -displayName  
"OracleOraDb11g_home1TNSListener" |  
where-object {$_.Status -eq "Stopped"} |  
restart-service
```

## ◆ Welche Oracle Dienste sind gestartet ?

```
▶ get-service |  
where-object {$_.name -match "ORACLE"}
```

Status	Name	DisplayName
Running	OracleDBConsole...	OracleDBConsoleo11g
Stopped	OracleJobSchedu...	OracleJobSchedulerO11G
Stopped	OracleJobSchedu...	OracleJobSchedulerXE
Stopped	OracleMTSRecover...	OracleMTSRecoveryService
Stopped	OracleOraDb11g_...	OracleOraDb11g_home1ClrAgent
Running	OracleOraDb11g_...	OracleOraDb11g_home1TNSListener
Running	OracleServiceO11G	OracleServiceO11G
Stopped	OracleServiceXE	OracleServiceXE
Stopped	OracleUsrWriter...	Oracle O11G USS Writer Service
Stopped	OracleXEClrAgent	OracleXEClrAgent
Stopped	OracleXETNSList...	OracleXETNSListener

# Prüfen, ob Instanz läuft

- ◆ Hier wird ein gesperrter Benutzer (z.B. OUTLN) verwendet, um festzustellen, ob die Instanz läuft:
- ◆ # ohne Listener
- ◆ `$o_check = sqlplus -L outln/outln | select-string "ORA-"`
- ◆ # mit Listener
- ◆ `$o_check = sqlplus -L outln/outln@prod | select-string "ORA-"`
- ◆ `if ($o_check -notmatch "ORA-28000: the account is locked") {write-host "Fehler" $ora_check} else {write-host "Anmeldung OK"}`

# Platzverbrauch auf den Platten

```
◆ gwmi win32_logicaldisk | select
  @{Name="DevID";Expression="{0:N1}" -f ($_.DeviceID) }},
  VolumeName,
  @{Name="Size";Expression="{0:N1}" -f ($_.Size/1gb)}},
  @{Name="FreeSpace;";Expression="{0:N1}" -f
  ($_.FreeSpace/1gb)}} | ft -auto
```

DevID	VolumeName	Size	FreeSpace;
B:	Tera	1.187,0	656,0
C:	Win7_SSD64GB	59,6	12,3
D:	Stripe	630,8	603,5
E:	Backup1	385,9	384,6
F:	Backup2	255,5	255,4
J:		0,0	0,0
K:	USB1.5T	1.397,3	1.019,9

# ALERT.LOG durchsuchen

## ◆ Suche nach ORA- in der Alert-Datei und zeige 3 Zeilen davor und 1 Zeile danach an:

- ▶ `$s="o11g"`
- ▶ `$al="C:\Oracle\diag\rdbms\${s}\${s}\trace>alert_${s}.log"`
- ▶ `$aa=Select-String $al -pattern "ORA-" -context 3,1`
- ▶ `$al= $al -replace "\\","\""`
- ▶ `$aa -replace $al,""`

```
> :3189:ORA-12012: error on auto execute of job "EXFSYS"."RLM$$SCHDNEGACTION"  
> :3190:ORA-12578: TNS:wallet open failed  
:3191:Thu Nov 17 16:47:44 2011  
:3192:Time drift detected. Please check UKTM trace file for more details.  
:3193:Thu Nov 17 16:47:44 2011  
:3194:Errors in file C:\ORACLE\diag\rdbms\o11g\o11g\trace\o11g_j000_5892.trc:
```

# Beispiel: Prozesse anzeigen

- ◆ Prozesse nach CPU sortiert ausgeben (mit mehr als 1 Sekunden):
- ◆ `Get-Process | where-object {$_.cpu -gt 1} | sort-object cpu`

Handles	NPM(K)	PM(K)	WS(K)	UM(M)	CPU(s)	Id	ProcessName
61	7	3720	7700	58	1,61	896	conhost
553	26	62172	63760	578	3,51	5616	powershell
677	73	69276	14240	639	5,01	1648	CCC
169	19	10920	19164	98	6,19	5528	notepad++
177	20	17664	21108	144	10,28	5648	plugin-container
476	72	44672	50796	153	11,12	2808	Dropbox
1029	84	54628	75164	310	15,05	4492	explorer
220	23	17664	23540	127	18,30	5480	picpick
452	48	22876	4572	156	19,97	3848	avp
127	17	40304	48280	233	32,96	4960	dwm
664	33	31676	39928	152	43,85	860	mstsc
848	86	46948	7576	341	146,22	2600	POWERPNT
552	157	306224	322496	513	228,23	4892	firefox

# Beispiel: Prozesse beenden

- ◆ **Über Prozessnummer (3512)**
  - ▶ `Stop-Process 3512`
- ◆ **Über Prozessname (oracle.exe)**
  - ▶ `Stop-Process -processname oracle.exe`
- ◆ **Mit Platzhalter (Alle Prozesse, die mit orac beginnen)**
  - ▶ `Stop-Process -processname orac*`

# Beispiel: Email Versand

- ◆ `$emailFrom = "m.patzwahl@muniqsoft.de"`
- ◆ `$emailTo = "m.patzwahl@muniqsoft.de"`
- ◆ `$subject = "Powershell Test-Email"`
- ◆ `$body = "Hier könnte Ihr Text stehen"`
- ◆ `$smtpServer = "email.muniqsoft.de"`
- ◆ `$smtp = new-object  
Net.Mail.SmtpClient($smtpServer)`
- ◆ `$smtp.Send($emailFrom, $emailTo, $subject,  
$body)`

# Spalten formatieren

- ◆ Bei den Prozessen die Spalten `cpu,threads` ausgeben:
  - ▶ `Get-Process | select cpu,threads | ft -auto`
- ◆ Breite der Spalte `THREADS` auf 60 Zeichen formatieren:
  - ▶ `Get-Process | Format-Table  
@{Expression={$_.threads};  
Label="Threads";  
width=60}`
- ◆ Ergebnis der Spalte `Size` auf 2 Stellen runden:
  - ▶ `gwmi win32_logicaldisk |  
select @{Name="Size";  
Expression={"{0:N2}" -f ($_.Size/1gb)}}`



# Ausgaben im CSV Format speichern

- ◆ Sie können eine Ausgabe direkt im CSV Format speichern:
- ◆ Beispiel: Alle Fehlermeldungen des Event-Logs in Datei `c:\temp\eventlog.csv` speichern
  - ▶ `Get-EventLog -LogName System -EntryType Error | export-csv c:\temp\eventlog.csv`

# Beispiel: Eventlog auslesen

## ◆ Nur Fehler anzeigen

▶ `Get-EventLog -LogName System -EntryType Error`

## ◆ Fehler anzeigen, als CSV Datei speichern und wieder anzeigen

▶ `Get-EventLog -LogName System -EntryType Error  
| export-csv c:\temp\tmp.csv ; Get-content  
c:\temp\tmp.csv`

## ◆ Fehler anzeigen und die Spalten Timewritten, eventid, message und Breite automatisch formatieren

▶ `Get-EventLog -LogName System -EntryType Error `  
-newest 10 | ft Timewritten,eventid,message `  
-autosize`

# Wie kommen die Daten in die DB?

- ◆ **Oracle bietet seit Version 11.2 die Möglichkeit, ein Preprocessing für External Tables durchzuführen**
- ◆ **Man kann damit z.B. eine Datei zuerst mit einem Unzip entpacken und dann in DB einlesen**
- ◆ **Jedoch lässt sich auch eine Batch-Datei starten und das Ergebnis einlesen**

# Wie kommen die Daten in die DB? (f)

UTL File schreibt Powershell  
Befehl in eine Batchdatei



Batch Datei wird als  
Preprocessing für External  
Table verwendet



SQL\*Loader Treiber lädt  
Ausgabe des PS  
Kommandos in die DB

```
UTL_FILE.PUT_LINE('...cmd /c  
powershell "Get-EventLog  
-LogName System
```



```
CREATE TABLE "MSTA_OS_EVL" (...)  
ORGANIZATION EXTERNAL (...  
PREPROCESSOR os_scripts_dir:  
'read_ev1.bat'
```



```
SELECT * FROM  
table(read_eventlog);
```

# Ladeskript

## ◆ Typen für Pipelined Function

```
▶ CREATE TYPE evl_obj AS OBJECT  
  ( ev_date VARCHAR2(32),  
    ev_id   VARCHAR2(10),  
    ev_message VARCHAR2(1000));  
/
```

```
▶ CREATE TYPE evl_tab AS TABLE OF msty_evl_obj;  
/
```

## ◆ Directory anlegen und Rechte an Benutzer vergeben

```
▶ CREATE DIRECTORY os_scripts_dir as  
  'C:\Oracle\admin\o11g\scripts';
```

```
▶ GRANT read,write,execute ON DIRECTORY os_scripts_dir  
  TO scott;
```

# Ladeskript (f)

```
◆ CREATE TABLE "MSTA_OS_EVL" (  
    "EV_DATE"          VARCHAR2(32 BYTE) ,  
    "EV_ID"            VARCHAR2(10 BYTE) ,  
    "EV_MESSAGE"       VARCHAR2(256 BYTE)  
    ORGANIZATION EXTERNAL  
    (TYPE ORACLE_LOADER DEFAULT DIRECTORY "OS_SCRIPTS_DIR"  
    ACCESS PARAMETERS (  
    RECORDS DELIMITED BY NEWLINE NOLOGFILE NOBADFILE NODISCARDFILE  
    PREPROCESSOR os_scripts_dir: 'read_ev1.bat'  
    SKIP 5  
    LOAD WHEN (  
        ev_date          != BLANKS AND  
        ev_id            != BLANKS AND  
        ev_message       != BLANKS )  
    FIELDS (  
        ev_date          POSITION(1 :19)   CHAR(20) ,  
        ev_id            POSITION(21 :8)   CHAR(8) ,  
        ev_message       POSITION(29 :128) CHAR(128) )  
    LOCATION ( 'read_ev1.bat' )  
    )  
    REJECT LIMIT UNLIMITED ;
```

Verzeichnis für Skript

Batch File

Skript Ausgabeformat

# Ladeskript (ff)

```
◆ CREATE OR REPLACE FUNCTION read_eventlog
RETURN msty_evl_tab
AUTHID CURRENT_USER PIPELINED IS
    v_file UTL_FILE.FILE_TYPE;
BEGIN
    v_file := UTL_FILE.FOPEN('OS_SCRIPTS_DIR', 'read_evl.bat', 'w');

    UTL_FILE.PUT_LINE(v_file,
q'!cmd /c powershell "Get-EventLog -LogName System -EntryType Error -newest 10 |
ft Timewritten,eventid, @{Label='Message';Expression={$_.message} ;width=200} -
autosize"!');
    UTL_FILE.FCLOSE(v_file);
    FOR r IN (SELECT * FROM msta_os_evl) LOOP
        PIPE ROW (
            msty_evl_obj(
                r.ev_date,
                r.ev_id,
                r.ev_message));
    END LOOP;
RETURN;
END ;
/
```

Batch Skript

Powershell Skript

External Table auslesen

# Alternativen

- ◆ Wenn mehrere Powershell-Skripte mit verschiedener Spaltenanzahl verwendet werden sollen:
- ◆ Sie können zwischen den Spalten ein Trennzeichen ausgeben:
  - ▶ `get-process | format-table ProcessName, @ {Label="Trenner"; Expression="{ "; } ,StartTime -autosize -HideTableHeaders`
  - ▶ Als Return Type wird dann nur VARCHAR2(4000) oder CLOB verwendet
  - ▶ Im Statement muss dann die Zeile nach dem Trennzeichen durchsucht und in die Spalten aufgeteilt werden



# Fehler

- ◆ Wenn Sie folgende Fehlermeldung bekommen, haben Sie das PS Fenster nicht als Administrator geöffnet:
- ◆ Restart-Service : Der Dienst "OracleOraDb11g\_home1TNSListener (OracleOraDb11g\_home1TNSListener)" kann aufgrund des folgenden Fehlers nicht beendet werden: Der Dienst OracleOraDb11g\_home1TNSListener kann nicht auf dem Computer . geöffnet werden.

# Powershell Sicherheit

## Get-ExecutionPolicy und Set-ExecutionPolicy

Einstellung	Beschreibung
Restricted	Es werden keine Konfigurationsdateien geladen und keine Scripts ausgeführt (Default)
AllSigned	Signierte Scripts und Konfigurationsdateien von einem vertrauenswürdigen Herausgeber werden ausgeführt. Lokal erstellte Scripts müssen signiert sein.
RemoteSigned	Aus Internet heruntergeladenen Scripts und Konfigurationsdateien müssen von einem vertrauenswürdigen Herausgeber signiert sein.
Unrestricted	Alle Konfigurationsdateien & Scripts werden ausgeführt. Nicht signierte Scripts vom Internet müssen vor Ausführung mit Prompt bestätigen.
Bypass	Keinerlei Blockade, keine Warnungen oder Prompts.
Undefined	Entfernt die gerade zugewiesene Richtlinie (nur für lokal zugewiesene Richtlinien, nicht für GPO-applizierte)

# Beispiel und Dokumentationsquellen

## ◆ Beispiele + Erklärung in Deutsch

▶ <http://www.msxfaq.de/code/powershell.htm>

## ◆ Microsoft Doku

▶ <http://technet.microsoft.com/de-de/library/bb978525.aspx>

## ◆ Windows PowerShell: Learn It Now Before It's an Emergency

▶ <http://technet.microsoft.com/en-us/scriptcenter/dd742419>

# MuniQSoft GmbH

- ◆ **Gegründet 1998**
- ◆ **Tätigkeitsbereiche:**
  - ▶ **Oracle Support**  
Hotline: Mo-Fr 8.00 – 18.00 Uhr  
Erweiterung um eine Rufbereitschaft auch am Wochenende möglich
  - ▶ **Oracle IT Consulting & Services**
  - ▶ **Oracle Schulungen (SQL, PL/SQL, DBA, APEX, B&R, ...)**
  - ▶ **Software-Lösungen**
  - ▶ **Oracle Lizenzen**

MuniQSoft GmbH  
Grünwalder Weg 13 a  
D-82008 Unterhaching  
[www.munisoft.de](http://www.munisoft.de)  
+49 89 6228 6789-0



# Aliasübersicht

# Aliasübersicht