

weise (klug); Weise, Weisheit  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
weisen (↑ R 10 ff.)



**Netz-Weise**  
Lernen von den Besten.

# Windows Powershell 3.0

Der nächste Schritt in eine  
gescriptete Zukunft

weise (klug); Weisheit, Weisheit  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
Weisen (↑ R 10 ff.)



Holger Voges

CCA, MCSE, MCDBA, MCT, MCITP DB Administrator /  
DB Developer, MCTIP Enterprise Administrator,  
MCSA Windows Server 2012

Netz-Weise  
Freundallee 13a  
30173 Hannover  
[www.netz-weise.de](http://www.netz-weise.de)

# Agenda

- Sprachfeatures
- Powershell-ISE
- Integration in den Task-Scheduler
- Powershell Web Access
- Netzwerk- und Systemmanagement mit CIM
- Workflows

## Features im Überblick

- > 2000 neue Commandlets
- Auch in Server Core und Windows PE!
- Dynamic Language Runtime
- Vereinfachte Syntax
- Deutlich verbesserte ISE (Powershell Editor)
- Robuste Remote-Sessions
- Remote-Gateway: „Powershell Überall“
- Zeitgesteuerte Powershell-Jobs
- WMI-Events
- Eingeschränkte Endpunkte
- Powershell-Workflows



# > 2000 neue Commandlets

<b>Add-Computer (I)</b> ComputerName (+) Force (+) LocalCredential (+) NewName (+) Options (+) Restart (+) UnjoinDomainCredential (+) <b>Add-Content (I)</b> Stream (+) <b>Add-Member (I)</b> NotePropertyMembers (+) NotePropertyName (+) NotePropertyValue (+) TypeName (+) <b>Add-Type (I)</b> LiteralPath (+) <b>Clear-Content (I)</b> Stream (+) <b>Export-Alias (I)</b> LiteralPath (+) <b>Export-Clixml (I)</b> LiteralPath (+) <b>Export-Csv (I)</b> Append (+) LiteralPath (+) <b>Export-FormatData (I)</b> LiteralPath (+) <b>Export-PSSession (I)</b> Certificate (+) <b>ForEach-Object (I)</b> ArgumentList (+) Confirm (+) MemberName (+) RemainingScripts (+) Whatif (+)	<b>Get-Acl (I)</b> AllCentralAccessPolicies (+) InputObject (+) LiteralPath (+) <b>Get-AuthenticodeSignature (I)</b> LiteralPath (+) <b>Get-ChildItem (I)</b> Attributes (+) Directory (+) File (+) Hidden (+) ReadOnly (+) System (+) <b>Get-Command (I)</b> All (+) <b>Get-Content (I)</b> Raw (+) Stream (+) Tail (+) <b>Get-Date (I)</b> Millisecond (+) <b>Get-Item (I)</b> Stream (+) <b>Get-Job (I)</b> After (+) Before (+) Filter (+) Newest (+) <b>Get-Module (I)</b> Authentication (+) CimSession (+) ComputerName (+) Credential (+) PSSession (+) <b>Get-PfxCertificate (I)</b> LiteralPath (+)	<b>Get-PSSession (I)</b> Authentication (+) CertificateThumbprint (+) Credential (+) Port (+) State (+) ThrottleLimit (+) UseSSL (+) <b>Import-Alias (I)</b> LiteralPath (+) <b>Import-Clixml (I)</b> First (+) IncludeTotalCount (+) LiteralPath (+) Skip (+) <b>Import-Csv (I)</b> Encoding (+) LiteralPath (+) <b>Import-LocalizedData (I)</b> PassThru (+) <b>Import-Module (I)</b> AsWorkflow (+) Authentication (+) CimSession (+) ComputerName (+) Credential (+) MinimumVersion (+) NoClobber (+) PSSession (+) RequiredVersion (+) Scope (+) Version (-) <b>Import-PSSession (I)</b> Certificate (+) <b>Invoke-Command (I)</b> Disconnected (+) SessionName (+)	<b>New-ModuleManifest (I)</b> DefaultCommandPrefix (+) DependentWorkflows (+) ExportAsWorkflow (+) HelpInfoUri (+) ModuleToProcess (-) RootModule (+) WorkflowsToProcess (+) <b>New-PSSessionOption (I)</b> OutputBufferingMode (+) <b>Out-File (I)</b> LiteralPath (+) <b>Out-GridView (I)</b> Block (+) OutputMode (+) PassThru (+) <b>Receive-Job (I)</b> Wait (+) WriteEvents (+) WriteJobInResults (+) <b>Register-PSSessionConfiguration (I)</b> AccessMode (+) ModulePath (+) Path (+) PSVersion (+) RunAsCredential (+) SessionType (+) SessionTypeOption (+) TransportOption (+) UseSharedProcess (+) <b>Remove-Computer (I)</b> ComputerName (+) Credential (-) LocalCredential (+) Restart (+) UnjoinDomainCredential (+) Workgroup (+)	<b>Remove-Item (I)</b> Stream (+) <b>Remove-Job (I)</b> Filter (+) <b>Rename-Item (I)</b> LiteralPath (+) <b>Reset-ComputerMachinePassword (I)</b> Credential (+) <b>Restart-Computer (I)</b> Delay (+) For (+) Timeout (+) Wait (+) <b>Select-Object (I)</b> Wait (+) <b>Select-String (I)</b> LiteralPath (+) <b>Select-Xml (I)</b> LiteralPath (+) <b>Set-Acl (I)</b> CentralAccessPolicy (+) ClearCentralAccessPolicy (+) InputObject (+) LiteralPath (+) SecurityDescriptor (+) <b>Set-AuthenticodeSignature (I)</b> LiteralPath (+) <b>Set-Content (I)</b> Stream (+) <b>Set-PSSessionConfiguration (I)</b> AccessMode (+) ModulePath (+) PSVersion (+) RunAsCredential (+) SessionType (+) SessionTypeOption (+) TransportOption (+) UseSharedProcess (+)	<b>Set-WSManQuickConfig (I)</b> SkipNetworkProfileCheck (+) <b>Start-Job (I)</b> DefinitionName (+) DefinitionPath (+) DefinitionType (+) LiteralPath (+) PSVersion (+) WriteToStore (+) <b>Start-Transcript (I)</b> LiteralPath (+) <b>Stop-Job (I)</b> Filter (+) <b>Tee-Object (I)</b> Append (+) LiteralPath (+) <b>Test-Path (I)</b> NewerThan (+) OlderThan (+) <b>Update-TypeData (I)</b> DefaultDisplayProperty (+) DefaultDisplayPropertySet (+) Force (+) InheritPropertySerialization (+) MemberName (+) MemberType (+) PropertySerializationSet (+) SecondValue (+) SerializationDepth (+) SerializationMethod (+) StringSerializationSource (+) TargetTypeForDeserialization (+) TypeAdapter (+) TypeConverter (+) TypeData (+) Value (+)	<b>Wait-Job (I)</b> Filter (+) <b>Where-Object (I)</b> CContains (+) CEQ (+) CGE (+) CGT (+) CIn (+) CLE (+) CLike (+) CLT (+) CMatch (+) CNE (+) CNotContains (+) CNotIn (+) CNotLike (+) CNotMatch (+) Contains (+) EQ (+) GE (+) GT (+) In (+) Is (+) IsNot (+) LE (+) Like (+) LT (+) Match (+) NE (+) NotContains (+) NotIn (+) NotLike (+) NotMatch (+) Property (+) Value (+) <b>Connect-PSSession (+)</b> <b>ConvertFrom-Json (+)</b> <b>ConvertTo-Json (+)</b>	<b>Disable-PSRemoting (+)</b> <b>Disconnect-PSSession (+)</b> <b>Export-CimCommand (+)</b> <b>Get-ControlPanelItem (+)</b> <b>Get-TypeData (+)</b> <b>Invoke-RestMethod (+)</b> <b>Invoke-WebRequest (+)</b> <b>New-PSSessionConfigurationFile (+)</b> <b>New-PSTransportOption (+)</b> <b>New-WinEvent (+)</b> <b>Receive-PSSession (+)</b> <b>Register-JobEvent (+)</b> <b>Remove-TypeData (+)</b> <b>Rename-Computer (+)</b> <b>Resume-Job (+)</b> <b>Save-Help (+)</b> <b>Show-Command (+)</b> <b>Show-ControlPanelItem (+)</b> <b>Suspend-Job (+)</b> <b>Test-PSSessionConfigurationFile (+)</b> <b>Unblock-File (+)</b> <b>Update-Help (+)</b>
---	--	--	--	---	---	--	---

weise (klug); Weise, Weisheit  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
weisen (↑ R 10 ff.)



**Netz-Weise**  
Lernen von den Besten.

- Updatefähige Hilfe (get-command -update)
- Online-Hilfe (get-command -online)
- Show-Command, die GUI für Commandlets
- Attribut-Validierung für Variablen
- Mandatory-parameter standardmässig \$true

## Vereinfachte Sytax ?!?

- (Dir).fullname
- Get-Process | foreach name
- Get-process | where CPU -GT 100
- Get-Process | foreach {\$PSItem.Name}



- ~~• (Dir).length~~
- ~~• Get-Process | foreach name.ToUpper()~~
- ~~• Get-process | where CPU -GT 100 -and -LT 10~~



# Neue Operatoren

```
PS:> 1..10 -contains 1  
ist gleich  
PS:> 1 -in 1..10
```

Operator	Wirkung
-shl	Shift Bits left
-shr	Shift Bits right
-in	Wie Contains mit umgekehrten Operanden
-notin	Wie notcontains mit umgekehrten Operanden





## Neue Umleitungs-Operatoren

- Do-Something 3> warnung.txt #Warnungen an Warnung.txt
- Do-Something 4>> Verbose.txt #Verbose-Ausgabe in Verbose.txt
- Do-Something 5>&1 #Debug-Output an Standard-Ausgabe
- Do-Something \*> out.txt #Alle Ausgaben in out.txt

weise (klug); Weise, Weisheit  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
weisen (↑ R 10 ff.)



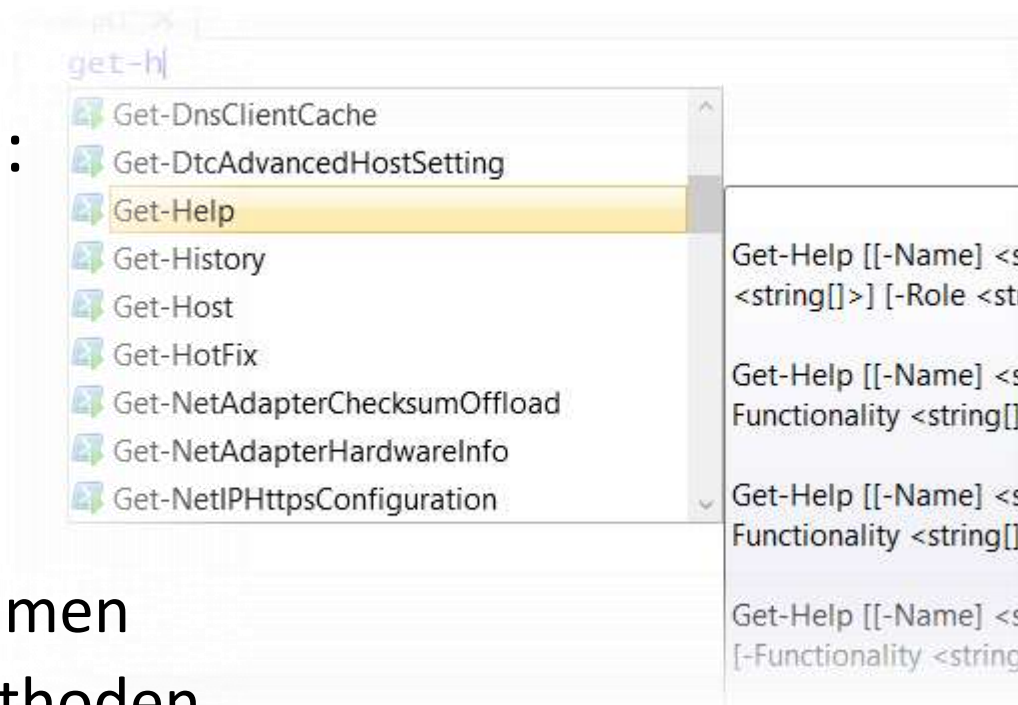
**Netz-Weise**  
Lernen von den Besten.

## ISE wird Entwicklungsumgebung

- Intellisense
- Snippets
- Plug-Ins
- Befehle Add-on
- Zeilenweises Ausführen
- Debugging

# Intellisense

- Vervollständigung von:
  - Commandlets
  - Parametern
  - Variablen
  - Pfaden
  - Typen und Namensräumen
  - Eigenschaften und Methoden
- Erweitern von Objekteigenschaften
- # Strg + Space zeigt die Kommando-Historie
- Suchvorschläge in Bibliotheken: [net + TAB



weise (klug); Weise, Weisheit  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
weisen (↑ R 10 ff.)

## Snippets

- Script-Schnipsel zum Einfügen
- CTRL-J öffnet die Snippet-Liste
- New-IseSnippet legt eigene Snippets an\*
- Module können ebenfalls Snippets ausliefern

\* Mit Strg+F1 ruft man direkt Show-Command auf

we|se (klug); 'Weise,  
-n, -n; ↑ R 5 ff. (kluger Mann)  
Weisen (↑ R 10 ff.)



**Netz-Weise**  
Lernen von den Besten.



**Demo**

## Erweiterte Modul-Funktionalität

- Module können Ereignisse im Eventlog speichern
- Das Logging kann in der Konsole oder per Group Policy konfiguriert werden
- Auto-Import von Modulen
- Die neue ExportedCommands-Eigenschaft zeigt alle Befehle eines Moduls an
- DefaultCommandPrefix im Manifest legt einen Standard-Namensraum fest
- Angepasste Parameter für minimale bzw. Benötigte Versionen (MinimumVersion / RequiredVersion)

weise (klug); Weisheit, Weisheit  
 -n, -n; ↑ R 5 ff. (kluger Mensch)  
 Weisen (↑ R 10 ff.)

# Event-Logging

**Turn on Module Logging**

Turn on Module Logging Previous Setting Next Setting

Not Configured  
 Enabled  
 Disabled

Comment:

Supported on: At least Microsoft Windows 7 or Windows Server 2008 family

Options: Help:

To turn on logging for one or more modules, click Show, and then type the module names in the list. Wildcards are supported.

Module Names:

To turn on logging for the Windows PowerShell core modules, type the following module names in the list:

Microsoft.PowerShell.\*  
 Microsoft.WSMan.Management

This policy setting allows you to turn on logging for Windows PowerShell modules.

If you enable this policy setting, pipeline execution events for members of the specified modules are recorded in the Windows PowerShell log in Event Viewer. Enabling this policy setting for a module is equivalent to setting the LogPipelineExecutionDetails property of the module to True.

If you disable this policy setting, logging of execution events is disabled for all Windows PowerShell modules. Disabling this policy setting for a module is equivalent to setting the LogPipelineExecutionDetails property of the module to False.

If this policy setting is not configured, the LogPipelineExecutionDetails property of a module or snap-in determines whether the execution events of a module or snap-in are logged. By default, the LogPipelineExecutionDetails property of all modules and snap-ins is set to False.

In der folgenden Tabelle sind die Einstellungsvariablen für das Ereignisprotokoll aufgeführt.

Variable	Beschreibung
\$LogEngineLifeCycleEvent	Protokolliert Start- und Beendigungsvorgänge von Windows PowerShell.
\$LogEngineHealthEvent	Protokolliert Windows PowerShell-Programmfehler.
\$LogProviderLifeCycleEvent	Protokolliert Start- und Beendigungsvorgänge von Windows PowerShell-Anbietern.
\$LogProviderHealthEvent	Protokolliert Windows PowerShell-Anbieterfehler.
\$LogCommandLifecycleEvent	Protokolliert den Start und den Abschluss von Befehlen.
\$LogCommandHealthEvent	Protokolliert Befehlsfehler.

(Weitere Informationen zu Windows PowerShell-Anbietern erhalten Sie mit folgendem Befehl: "get-help about\_providers".)

Standardmäßig sind nur die folgenden Ereignistypen aktiviert:

```
$LogEngineLifeCycleEvent
$LogEngineHealthEvent
$LogProviderLifeCycleEvent
$LogProviderHealthEvent
```

weise (klug); Weisheit, Weisheit  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
weisen (↑ R 10 ff.)

## Robuste und Disconnected Sessions

- Sitzungen laufen auch nach Sitzungstrennung weiter
- Automatischer Reconnect
- Wiederverbinden von beliebigem Client möglich
  - New-PSSession
  - Disconnect-PSSession
  - Get-PSSession
  - Receive-PSSession





weise (klug); Weisheit, Weisheit  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
weisen (↑ R 10 ff.)



**Netz-Weise**  
Lernen von den Besten.

## Zeitgesteuerte Powershell-Jobs

- Task-Scheduler per Powershell steuern
- Powershell-Jobs zeitgesteuert starten
  - > New-Jobtrigger
  - > Register-ScheduledJob
  - > Receive-Job



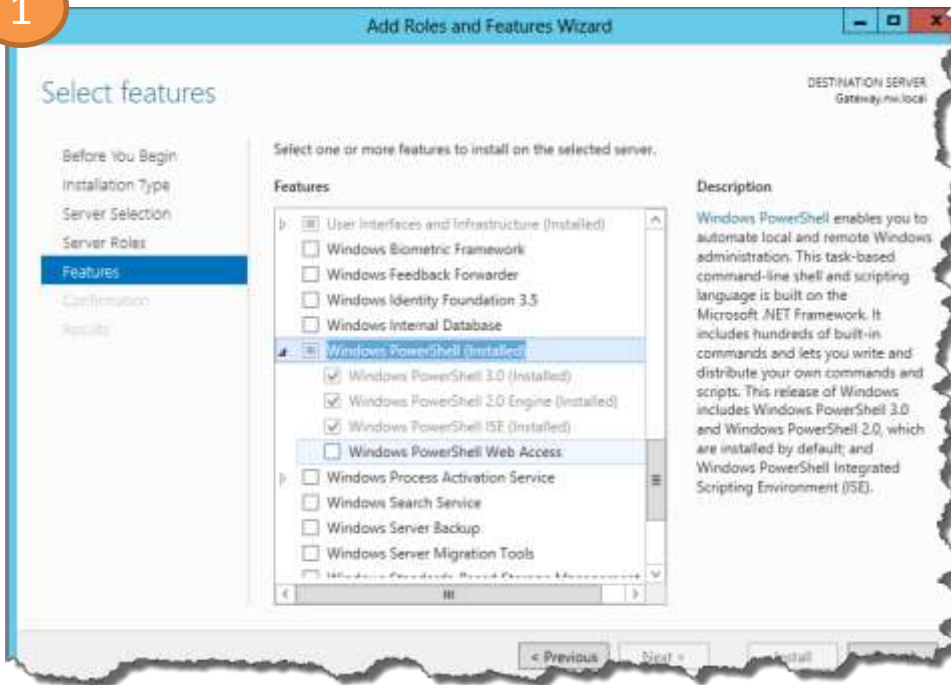
# Sitzungskonfigurationsdateien

- Einrichten von eingeschränkten Remotesitzungen per Konfigurationsdatei
- Konfigurationsdatei bestimmt:
  - Module
  - Scripte
  - Formatdateien
  - Commandlets
  - Sprachelemente
  - Sichtbare Variablen
- Deutlich vereinfachte Einrichtung zu V 2.0
- New-PSSessionConfigurationFile cmdlet, Register-PSSessionConfiguration, Set-PSSessionConfiguration cmdlets.

wei|se (klug); 'Weise,  
-n, -n; ↑ R 5 ff. (kluger Men  
Weisen (↑ R 10)

# Powershell Web Access

1



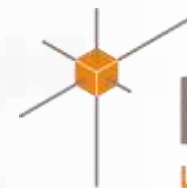
2

Install-PswaWebApplication  
-UseTestCertificate

3

Add-PswaAuthorizationRule -  
UserName \* -ComputerName  
\* -ConfigurationName \*

wei|se (klug); 'Weise, ...  
-n, -n; ↑ R 5 ff. (kluger Mann)  
weisen (↑ R 108)



**Netz-Weise**  
Lernen von den Besten.

Windows Server 2012 Microsoft

## Windows PowerShell Web Access ?

Enter your credentials and connection settings

User name:

Password:

Connection type:

Computer name:

Optional connection settings

© 2012 Microsoft Corporation. All rights reserved.

Windows PowerShell  
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

```
PS C:\Users\Administrator\Documents>
```

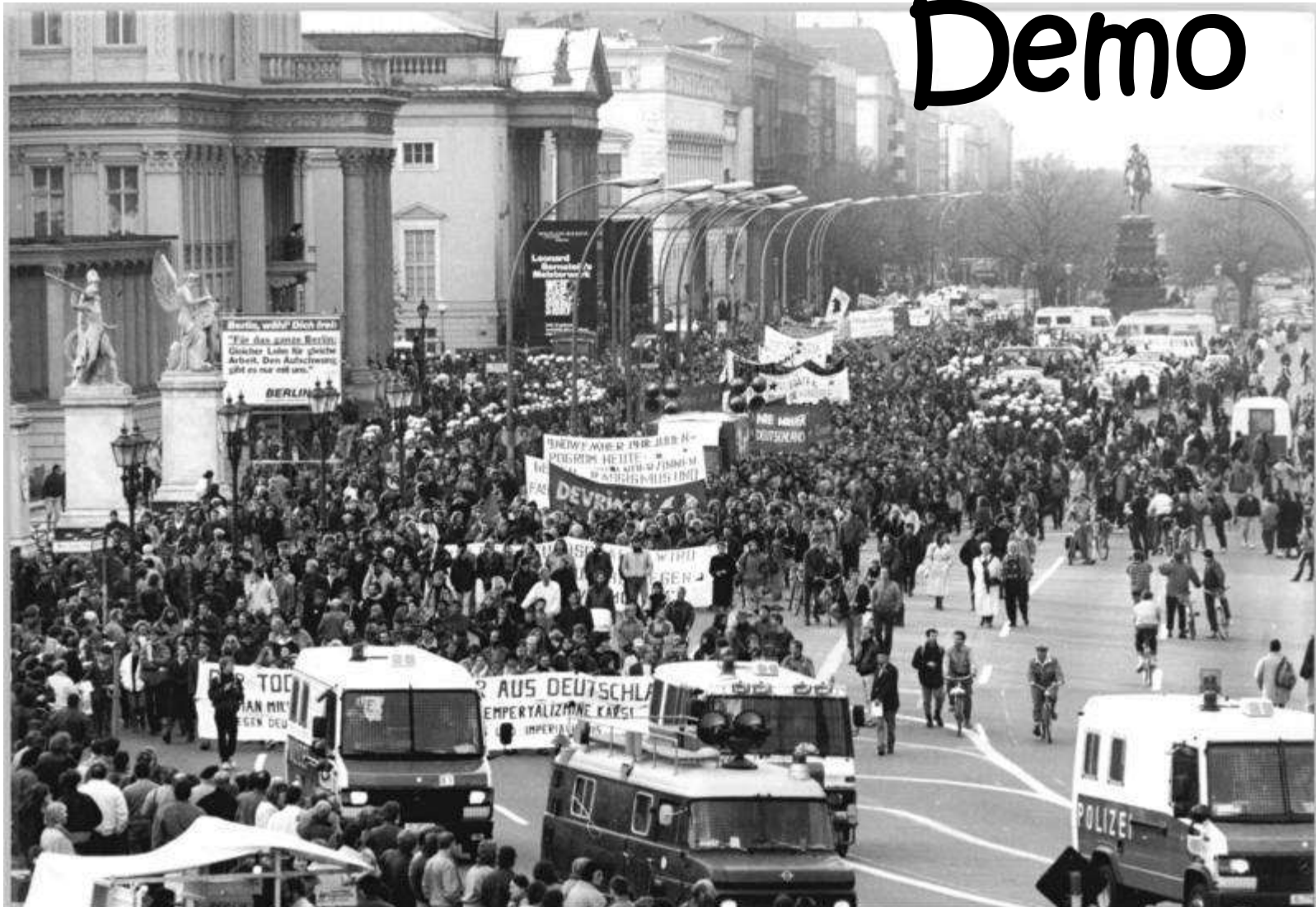
Connected to: nwdc

weisse (klug); Weisse,  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
Weisen (↑ R 19)



**Netz-Weise**  
Lernen von den Besten.

# Demo



wei|se (klug); Weisheit, Weisheit  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
Weisen (↑ R 10 ff.)

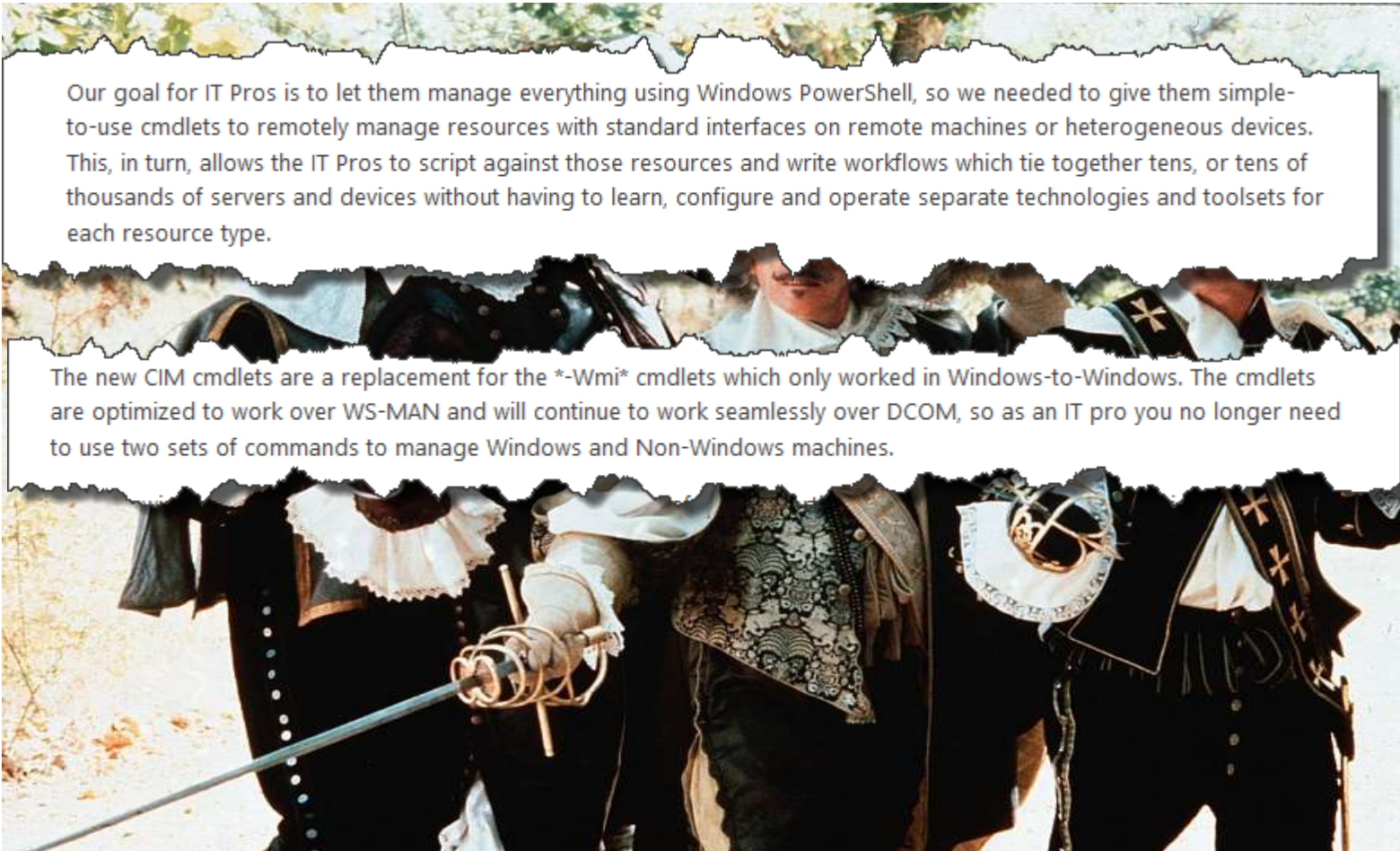


**Netz-Weise**  
Lernen von den Besten.

## CIM – Einer für Alle

Our goal for IT Pros is to let them manage everything using Windows PowerShell, so we needed to give them simple-to-use cmdlets to remotely manage resources with standard interfaces on remote machines or heterogeneous devices. This, in turn, allows the IT Pros to script against those resources and write workflows which tie together tens, or tens of thousands of servers and devices without having to learn, configure and operate separate technologies and toolsets for each resource type.

The new CIM cmdlets are a replacement for the \*-Wmi\* cmdlets which only worked in Windows-to-Windows. The cmdlets are optimized to work over WS-MAN and will continue to work seamlessly over DCOM, so as an IT pro you no longer need to use two sets of commands to manage Windows and Non-Windows machines.



# CIM-Integration

Distributed Management Task Force	Zusammenschluss div. Unternehmen mit dem Ziel, IT-Management zu vereinfachen
Common Infrastructure Model	DMTF-Standard für die Verwaltung von IT-Komponenten (Server, Clients, Software...)
Windows Management Instrumentation	Microsofts Implementierung eines CIM-Servers für Windows
Web-Services-Management Protocol	SOAP-basiertes Protokoll zur Kommunikation zwischen CIM-kompatiblen Geräten
Windows Remote Management	Microsofts Implementierung von WS-Man
CIM Object Manager	Zentrale Software, die alle Klassen und deren Verknüpfungen kennt (CIM-Agent)

weise (klug); Weise, Weisheit  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
weisen (↑ R 10 ff.)

# Web Based Enterprise Management

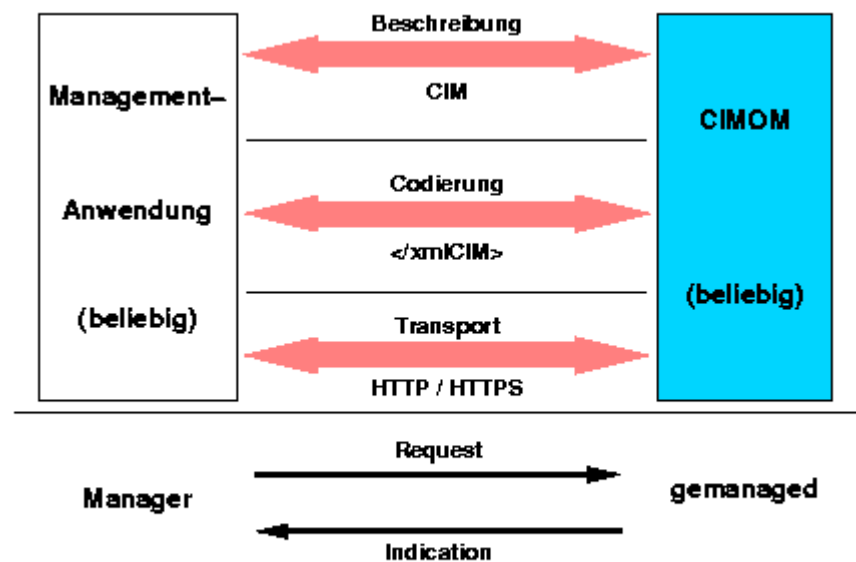


Abbildung 5.15: Architektur von WBEM



# Commandlet-Vergleich

Altes WMI-Commandlet	Neues CIM-Commandlet
Get-WmiObject	Get-CimInstance
Get-WmiObject -list	Get-CimClass
Set-WmiInstance	Set-CimInstance
Set-WmiInstance -PutType CreateOnly	New-CimInstance
Remove-WmiObject	Remove-CimInstance
Invoke-WmiMethod	Invoke-CimMethod

**# OLD:**

```
PS:> Invoke-WMIMethod -class Win32_Process -Name create -ArgumentList 'calc.exe'
```

**# NEW:**

```
PS:> Invoke-CimMethod Win32_Process -MethodName create -Arguments @{CommandLine='calc.exe'}
```

New Cmdlet takes a hash table or ordered dictionary as input not an Object.

weisse (klug); Weisse, ...  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
Weisen (↑ R 10 ff.)



**Netz-Weise**  
Lernen von den Besten.



## Management OData

- Odata (Open Data Protocol) ist ein Web-Protokoll, um Daten standardisiert in Tabellenähnlicher Form zu übertragen
- Mit Odata können Powershell-Funktionen über Webservices zur Verfügung gestellt werden.

Finally, for the Web developers that want to manage Windows from non-Windows platforms we have developed the Management OData IIS Extension. This contains tools and components that simplify building REST APIs (OData Service endpoints).

OData is a set of URI conventions, tools, components and libraries for building REST APIs. What makes the OData services

wei|se (klug); 'Weise, ...  
-n, -n; ↑ R 5 ff. (kluger Man...  
weisen (↑ R 10)



**Netz-Weise**  
Lernen von den Besten.

## Powershell Workflows Definition

In one sentence, Windows Powershell Workflow is all about:

*„Reliably executing long-running tasks across multiple computers, devices or IT processes“*

Mir Rosenberg [MSFT]  
Senior Program Manager  
Windows PowerShell

Windows PowerShell Workflow enables IT professionals and developers to author sequences of activities that are long-running, repeatable, frequent, parallelizable, interruptible, suspendable, and/or restartable as workflows.

A workflow is a set of activities. An activity is an individual step within a workflow that performs a defined task, such as getting the list of virtual machines in a host computer.

weise (klug); Weise, Weisheit  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
weisen (↑ R 10 ff.)



**Netz-Weise**  
Lernen von den Besten.

- Workflows bestehen aus den Elementen
  - workflow
  - inlinescript {}
  - foreach –parallel
  - parallel {}
  - sequence {}

weise (klug); 'Weise, der  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
weisen (↑ R 10 ff.)



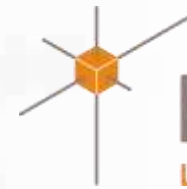
**Netz-Weise**  
Lernen von den Besten.

## workflow

Definiert einen neuen Workflow:

```
workflow hallo-welt  
{  
    „Hallo welt“  
}
```

weise (klug); Weisheit, Weisheit  
-n, -n; ↑ R 5 ff. (kluger Mensch)  
weisen (↑ R 10 ff.)



**Netz-Weise**  
Lernen von den Besten.

## Inlinescript {}

Alle Kommandos innerhalb eines Workflows laufen isoliert. Inlinescript erstellt einen gemeinsamen Arbeitsbereich:

```
workflow invoke-inlinescript
{
    inlinescript {
        $a = 2
        $b = $a+2
        $b
    }
}
```

## parallel {} / sequence {}

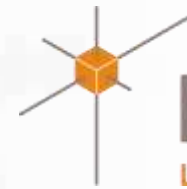
Parallel: Alle Prozesse in beliebiger Reihenfolge /  
Sequence: definiert nacheinander

```
workflow Invoke-ParallelWorkflow
{
    parallel
    {
        Get-Process -Name PowerShell

        sequence
        {
            "In sequence 1 of 2"
            "In sequence 2 of 2"
        }

        "In parallel 1 of 2"
        "In parallel 2 of 2"
        Get-Service -Name WinRM
    }
}
```





## foreach-parallel {}

Die Befehle in der Schleife werden gleichzeitig gestartet

```
workflow Invoke-ForEachParallel
{
    param([string[]]$computerName)

    foreach -parallel($computer in $computerName)
    {
        "Executing on $computer"
    }
}
```

## Anpassung an die PSH-Version

- #requires legt Mindestanforderungen an ein Script fest
- „Powershell.exe –version 2“ startet Powershell 2.0
- \$PSversionTable zeigt die benutzte Version an
  - #requires –PsSnapIn Microsoft.PowerShell.Security
  - #requires –Version 2
  - #requires –PsSnapIn Microsoft.PowerShell.Security –  
Version 2

# Andere Quellen

- [www.powershell.com](http://www.powershell.com)
- [Powershell Workflow Getting Started](http://download.microsoft.com/download/A/F/1/AF1D6992-54EC-47CD-B2BF-38A68C171641/WMF3%20CTP2%20Windows%20PowerShell%20Workflow.pdf)  
<http://download.microsoft.com/download/A/F/1/AF1D6992-54EC-47CD-B2BF-38A68C171641/WMF3%20CTP2%20Windows%20PowerShell%20Workflow.pdf>
- [Technet : Writing a Script Workflow](http://technet.microsoft.com/en-us/library/jj574157.aspx)  
<http://technet.microsoft.com/en-us/library/jj574157.aspx>
- [Don Jones – PowerShell Workflow: When should you use it?](http://powershell.org/wp/2012/08/30/powershell-workflow-when-should-you-use-it/)  
<http://powershell.org/wp/2012/08/30/powershell-workflow-when-should-you-use-it/>
- [Technet: What's New in Windows Powershell 3.0](http://technet.microsoft.com/library/hh857339)  
<http://technet.microsoft.com/library/hh857339>
- [WS 2012 Server Management and Automation White Paper](http://download.microsoft.com/download/D/4/3/D43132A6-DE21-4B10-8D2E-B49A7CFEEA89/WS%202012%20White%20Paper_Server%20Management%20and%20Automation.pdf)  
[http://download.microsoft.com/download/D/4/3/D43132A6-DE21-4B10-8D2E-B49A7CFEEA89/WS%202012%20White%20Paper\\_Server%20Management%20and%20Automation.pdf](http://download.microsoft.com/download/D/4/3/D43132A6-DE21-4B10-8D2E-B49A7CFEEA89/WS%202012%20White%20Paper_Server%20Management%20and%20Automation.pdf)
- [Hey Scripting Guy Blog](http://blogs.technet.com/b/heyscriptingguy)  
<http://blogs.technet.com/b/heyscriptingguy>
- [Windows Powershell Blog](http://blogs.msdn.com/b/powershell/)  
<http://blogs.msdn.com/b/powershell/>
- [Keith Hills Blog](http://rkeithhill.wordpress.com/)  
<http://rkeithhill.wordpress.com/>