

trueIT.

JÜRIG KOLLER

PARTNER /  
CONSULTANT

[Juerg.Koller@trueit.ch](mailto:Juerg.Koller@trueit.ch) Twitter: @juergkoller

[www.trueit.ch](http://www.trueit.ch)



# Hardware Inventory Erweiterungen

Secunia  
Stay Secure



Configuration Manager Community Event 2014  
CU1

# Agenda



- Grundlagen
- Welche Daten können inventarisiert werden
- Anpassungen an Default Inventory Klassen
- WMI Klassen im Inventar ergänzen
- Registry Keys inventarisieren
- Hardware Inventar Datenfluss
- Wie kann ich die Daten verwenden

# Grundlagen



- Zwei grundsätzliche Arten von Inventory Konfigurationen:
  - **Hardware Inventory**
  - **Software Inventory**
- Vor Änderungen an der Inventory Konfiguration Site **Backup** und Kopie von configuration.mof erstellen
- Grundsätzlich können alle gezeigten Erweiterung auch für SCCM 2007 gemacht werden ([Ablauf Mainstream Support am 8.7.2014!](#))

# Hardware Inventory für Windows



- Inventarisiert Informationen aus Windows Management Instrumentation WMI wie zum Beispiel:
  - Hardware Model, Serien Nummer, Diskgrößen
  - Installierte Applikationen aus Add/Remove Programs
  - Netzwerk Konfiguration, Services
  - Konsolen Benutzer Information, letzter angemeldeter Benutzer, häufigster Benutzer
  - usw.

# Software Inventory



- Namen ist irreführend. Sollte besser Datei Information Inventar heissen
- Liefert für definierte Dateitypen Informationen zu:
  - Datei Pfad, Datei Version, Datei Grösse, Datei Änderungsdatum
- Inventarprozess kann je nach System und Konfiguration sehr lange (mehrere Stunden) dauern. Während dieser Zeit wird das HW Inventory nicht aktualisiert

# Software Inventory



- Komponente aktiviert per Default
- Keine Datei Typen definiert
- Schedule alle 7 Tage
- **Falls keine Dateitypen konfiguriert werden, Komponente deaktivieren**
- Mit versteckter Datei **Skpswi.dat** kann ein Verzeichnis inklusive Unterverzeichnisse vom Scan Prozess ausgenommen werden

# Datei Collection



- Standardmässig nicht aktiviert
- Sollte nur für sehr spezifische Fälle und kleine Files verwendet werden
- Dateien werden auf den Clients kopiert und auf dem Site Server unter `<ConfigMgr InstallDir>\Inboxes\Sinv.box\Filecol` gespeichert
- Dateien können im Ressource Explorer geöffnet werden
- Bandbreiten- und Speicherbedarf beachten

# Asset Intelligence Daten



- Durch Asset Intelligence kann Software Inventar Information verbessert werden
- Kategorisierung von Software Produkten
- Eindeutige Identifikation von Software
- Zusätzliche Inventar Klassen und mehr als 50 zusätzliche Reports zur Datenauswertung
- Aktualisierung durch Asset Intelligence Synchronisation Point



# HW Inventar Erweiterungen



- Bestehende Inventar Klassen können im Default Client Setting wie auch in eigenen Client Settings aktiviert oder deaktiviert werden
- Neue Inventar Klassen können nur im Default Client Setting ergänzt werden

# Configuration.mof



- Definiert die Datenklassen welche die Inventory Agent Komponente verwendet
- Configuration.mof Inhalt wird der Client Policy angehängt und auf dem Client verarbeitet.
- Falls Syntax Fehler vorhanden sind, kann dadurch die Computer Policy korrupt werden
- **Sicherungskopie machen vor Änderungen**



# Demo

## Default Hardware Inventory



# WMI Klassen im Inventar ergänzen

# BitLocker Status



- Liefert Information zum BitLocker Verschlüsselungsstatus der Disks
- WMI Klasse [Win32\\_EncryptableVolume](#)  
Attribute:
  - Device ID
  - Drive Letter
  - Protection Status

# Computer System Product



- Bei einigen HW Vendors (z.B. Lenovo) wird die Hardware Modell Information in dieser Klasse gespeichert
- WMI Klasse [Win32\\_ComputersystemProduct](#)  
Attribute:
  - Name
  - Vendor
  - Version

# TPM Version



- Information über TPM Spezifikation vom Computer
  - WMI Klasse [Win32 TPM](#)
- Attribute:
- Manufacturer Version
  - Spec Version

# Harddisk Information



- Information über freier Harddisk Platz. Zusätzliches Attribut selektieren
- WMI Klasse SMS\_LogicalDisk  
Attribute:
  - Free Space (MB)



# Hotfix Information



- Information über installierte Windows Updates und Hotfixes
- WMI Klasse [Win32\\_QuickfixEngineering](#)  
Attribute:
  - Hotfix ID
  - Caption
  - Description
  - InstalledBy
  - InstalledOn

# WMI Klassen ergänzen



- Default Client Settings > Hardware Inventory > Set Classes
- Add > Connect to Namespace
- Namespace und Klasse auswählen und gewünschte Attribute selektieren

# Was bietet WMI für Infos



- Abhängig von der Windows Version und installierter Software hunderte verschiedene Klassen mit Informationen rund um OS, installierter Software, Hardware, BIOS, usw.
- Tools um WMI Informationen zu browsen gibt es diverse:
  - [Scriptomatic](#)
  - [Powershell WMI Explorer](#)
  - [Coretech WMI and Powershell Explorer](#)

# Interessante WMI Klassen



- `ROOT\ccm\softmgmtagent`  
Liefert Informationen zu ConfigMgr Cache Grösse und Inhalt

# Computer BIOS Information



- Die meisten Hardware Hersteller bieten über WMI Informationen zu den BIOS Settings
- HP
  - ROOT\HP\InstrumentedBIOS
- DELL
  - [Installation von DELL OMCI](#)
  - ROOT\dcim\sysman
- Lenovo
  - ROOT\wmi Class Lenovo\_BiosSetting

# Weitere WMI Information



- Daten können auch per Script abgefragt und in eine eigene WMI Klasse geschrieben und von dort inventarisiert werden
  - z.B. Mitglieder von lokaler Administrator Gruppe
  - ODBC Einstellungen
  - Scheduled Tasks
  - Größen von bestimmten Ordner oder Dateitypen
  - Eingabewerte von Userdialogen z.B. Abfrage Computerstandort, usw.



# Demo

**WMI Informationen zum Inventory ergänzen**



# Registry Keys inventarisieren



# RegKeytoMof



- Mit [RegKeytoMof Tool](#) kann Registry gebrowst und gewünschte configuration.mof Erweiterung direkt generiert und kopiert werden
- ClassGroup und ClassName müssen eindeutig sein
- Erweiterung am Ende von configuration.mof Datei in `<ConfigMgr InstallDir>\inboxes\clifiles.src\hinv` einfügen
- Vorgängig Sicherungskopie von configuration.mof erstellen

# Registry Key inventarisieren



- Vor dem Einfügen von configuration.mof snippet Syntax Check mit `mofcomp.exe -check <snippet.mof>`
- Nach dem Speichern der configuration.mof das `<ConfigMgr InstallDir>\logs\dataldr.log` in cmtrace öffnen und nach MOFCOMP suchen. Policy Konvertierung muss mit **returning 0x0** enden
- Automatische Sicherungskopie vom alten configuration.mof unter `<ConfigMgr InstallDir>\data\hinvarchive`
- Nach dem nächsten Machine Policy Refresh, wird die neue WMI Klasse auf den Clients erstellt
- Anschliessend neue WMI Klasse zu den Inventory Settings hinzufügen



# Demo

## Inventarisierung von Registry Informationen



# Datenfluss



# Hardware Inventory Prozess - Client Side

- Client erhält Hardware Inventory Konfiguration per Machine Policy
- Hardware Inventory Cycle
- InventoryAgent.log

Log Text	Component	Date/Time	Thread
Inventory: ***** Start of message processing. *****	InventoryAgent	11.06.2014 11:35:50	632 (0x278)
Inventory: Message type is InventoryAction	InventoryAgent	11.06.2014 11:35:50	632 (0x278)
Inventory: Temp directory = C:\WINDOWS\CCM\Inventory\Temp\	InventoryAgent	11.06.2014 11:35:50	632 (0x278)
Inventory: Clearing old collected files.	InventoryAgent	11.06.2014 11:35:50	632 (0x278)
Inventory: Opening store for action {00000000-0000-0000-0000-000000000001} ...	InventoryAgent	11.06.2014 11:35:50	632 (0x278)
Inventory: Action=Hardware, ReportType=Delta, MajorVersion=1, MinorVersion=11	InventoryAgent	11.06.2014 11:36:25	632 (0x278)
Inventory: Initialization completed in 34.735 seconds	InventoryAgent	11.06.2014 11:36:25	632 (0x278)
Collection: Namespace = root\SmsDm; Query = SELECT __CLASS, __PATH, __RELPATH, DeviceOEMInfo, ...	InventoryAgent	11.06.2014 11:36:25	632 (0x278)
Collection: Class "SMS_ActiveSyncConnectedDevice" does not exist.	InventoryAgent	11.06.2014 11:36:25	632 (0x278)
Collection: Namespace = \\.\root\cimv2; Query = SELECT __CLASS, __PATH, __RELPATH, IdentifyingNum...	InventoryAgent	11.06.2014 11:36:25	632 (0x278)
Collection: Namespace = \\.\root\cimv2; Query = SELECT __CLASS, __PATH, __RELPATH, Availability, Des...	InventoryAgent	11.06.2014 11:36:25	632 (0x278)
<b>Date/Time:</b> 11.06.2014 11:15:53 <b>Component:</b> InventoryAgent			
<b>Thread:</b> 2712 (0xA98) <b>Source:</b> agentendpoint.cpp:409			
Inventory: ***** End of message processing. *****			



# Hardware Inventory Prozess – Client Side

- WMI Klassen und Registry werden gemäss der Policy inventarisiert
- Temporäre XML Datei erstellt und zum Management Point geschickt

Configuration Manager Trace Log Tool - [C:\Windows\CCM\Logs\InventoryAgent.log]

Log Text	Component	Date/Time	Thread
Inventory: Collection Task completed in 30.094 seconds	InventoryAgent	11.06.2014 11:36:55	632 (0x278)
<b>Inventory: 12 Collection Task(s) failed.</b>	<b>InventoryAgent</b>	<b>11.06.2014 11:36:55</b>	<b>632 (0x278)</b>
Inventory: Temp report = C:\WINDOWS\CCM\Inventory\Temp\53ead481-8af8-4c87-8134-61985488ba53....	InventoryAgent	11.06.2014 11:36:55	632 (0x278)
Inventory: Starting reporting task.	InventoryAgent	11.06.2014 11:36:55	632 (0x278)
Reporting: 30 report entries created.	InventoryAgent	11.06.2014 11:36:56	632 (0x278)
Inventory: Reporting Task completed in 0.468 seconds	InventoryAgent	11.06.2014 11:36:56	632 (0x278)
Inventory: Successfully sent report. Destination:mp:MP_HinvEndpoint, ID: {551321D4-EE28-448B-810F-AF...}	InventoryAgent	11.06.2014 11:36:56	632 (0x278)
Inventory: Cycle completed in 94.110 seconds	InventoryAgent	11.06.2014 11:37:24	632 (0x278)
Inventory: Action completed.	InventoryAgent	11.06.2014 11:37:24	632 (0x278)
Inventory: ***** End of message processing. *****	InventoryAgent	11.06.2014 11:37:24	632 (0x278)

**Date/Time:** 11.06.2014 11:15:53    **Component:** InventoryAgent  
**Thread:** 2712 (0xA98)    **Source:** agentendpoint.cpp:409

Inventory: \*\*\*\*\* End of message processing. \*\*\*\*\*



# Hardware Inventory Prozess – Client Side

- Damit zu Troubleshooting Zwecken die temporäre XML Datei nicht gelöscht wird unter %systemroot%\ccm\inventory\temp Datei **archive\_reports.sms** erstellen

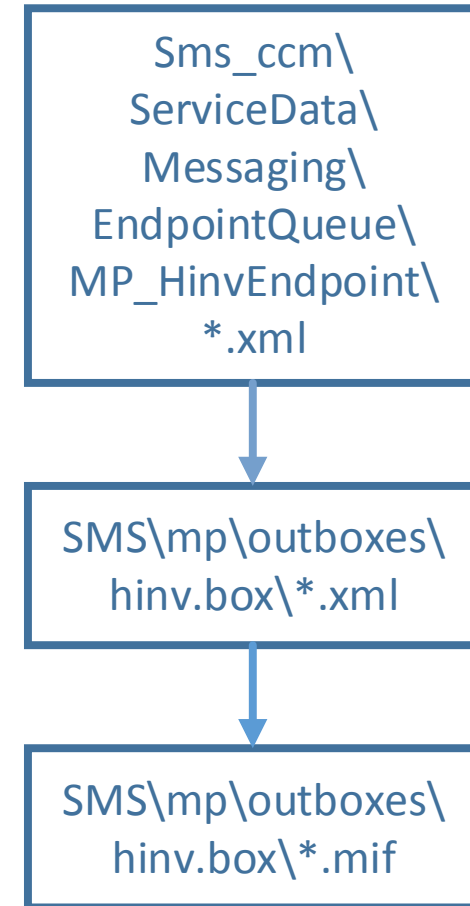
```
C:\Windows\CCM\Inventory\Temp\53ead481-8af8-4c87-8134-61985488ba53.xml
C:\Windows\CCM\Invento... x

<?xml version="1.0" encoding="UTF-16"?>
- <Report>
  - <ReportHeader>
    - <Identification>
      - <Machine>
        <ClientInstalled>1</ClientInstalled>
        <ClientType>1</ClientType>
        <ClientID>GUID:9C5D1E24-FBD8-4A2E-955E-F5AD823A1696</ClientID>
        <ClientVersion>5.00.7958.1203</ClientVersion>
        <NetBIOSName>W7-X64-001</NetBIOSName>
        <CodePage>850</CodePage>
        <SystemDefaultLCID>2055</SystemDefaultLCID>
      </Machine>
    </Identification>
  - <ReportDetails>
    <ReportContent>Inventory\x0020Data</ReportContent>
    <ReportType>Delta</ReportType>
    <Date>20140611113655.000000+120</Date>
    <Version>1.11</Version>
    <Format>1.1</Format>
  </ReportDetails>
  <InventoryAction ActionType="Predefined">
    <InventoryActionID>{00000000-0000-0000-0000-000000000001}</InventoryActionID>
    <Description>Hardware</Description>
    <InventoryActionLastUpdateTime>20140522143010.000000+000</InventoryActionLastUpdateTime>
  </InventoryAction>
  </ReportHeader>
  - <ReportBody>
    - <Instance Content="Updated" Namespace="\\W7-X64-001\root\cimv2\sms" Class="CCM_RecentlyUsedApps" ParentClass="CCM_RecentlyUsedApps">
      - <CCM_RecentlyUsedApps>
        <AdditionalProductCodes/>
        <CompanyName>Microsoft\x0020Corporation</CompanyName>
        <ExplorerFileName>slui.exe</ExplorerFileName>
      </CCM_RecentlyUsedApps>
    </Instance>
  </ReportBody>
</Report>
```



# Hardware Inventory Prozess – Management Point

- sms\_ccm\logs\MP\_Hinv.log
- XML Datei kommt in der EndpointQueue an und wird in .mif Datei konvertiert
- Datei wird auf MP in Outbox verschoben und dort vom File Dispatch Manager aufgenommen und auf den Site Server in die dataldr Inbox verschoben







# Hardware Inventory Prozess – Site Server

- <ConfigMgrInstallDir>\logs\dataldr.log

The screenshot shows the Configuration Manager Trace Log Tool interface. The main window displays a log table with columns for Log Text, Component, and Date/Time. A specific log entry is highlighted in blue, showing the process of moving a MIF file. Below the table, a summary box provides details for the selected entry, including the Date/Time, Component, and Thread. At the bottom of the window, the elapsed time is displayed as 1860h 53m 45s 70ms (6699225.070 seconds).

Log Text	Component	Date/Time
Checking inbox for any MIFs to process...	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:06
>> Add 1 files to process directory ...	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:06
Moving MIF file D:\Program Files\Microsoft Configuration Manager\inbox\auth\dataldr.box\HOUFRPGQ.MIF to D:\Program Files\Microsoft C...	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:06
Started the machine MIF processing thread, thread ID = 1128	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:06
Worker thread 3132 starting execution.	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:06
Done with job queuing.	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:07
Blocking until completion.	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:07
Thread: 0 is using GUID	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:07
Thread: 3132 will use GUID GUID:9C5D1E24-FBD8-4A2E-955E-F5AD823A1696	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:07
Processing Inventory for Machine: W7-X64-001 Version 1.11 Generated: 06/11/2014 11:36:55	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:07
Begin transaction: Machine=W7-X64-001(GUID:9C5D1E24-FBD8-4A2E-955E-F5AD823A1696)	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:07
Commit transaction: Machine=W7-X64-001(GUID:9C5D1E24-FBD8-4A2E-955E-F5AD823A1696)	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:07
Done: Machine=W7-X64-001(GUID:9C5D1E24-FBD8-4A2E-955E-F5AD823A1696) code=0 (32 stored procs in XHOUFRPGQ.MIF)	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:07
Done blocking until completion.	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:08
No more machine MIFs to be processed, terminating thread	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:08
Shutting down Machine Writer.	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:08
Worker thread 3132 halting execution.	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:08
Finished processing 1 MIFs	SMS_INVENTORY_DATA_LOADER	11.06.2014 11:37:13
Looking for inventory schema changes...	SMS_INVENTORY_DATA_LOADER	11.06.2014 12:21:12

**Date/Time:** 11.06.2014 11:37:06    **Component:** SMS\_INVENTORY\_DATA\_LOADER  
**Thread:** 3632 (0xE30)    **Source:**

Moving MIF file D:\Program Files\Microsoft Configuration Manager\inbox\auth\dataldr.box\HOUFRPGQ.MIF to D:\Program Files\Microsoft Configuration Manager\inbox\auth\dataldr.box\process\HOUFRPGQ.MIF

Elapsed time is 1860h 53m 45s 70ms (6699225.070 seconds)



# Event ID 2719 SMS\_INVENTORY\_DATA\_LOADER Error

Configuration Manager Status Message Viewer for <PS1> <Techlab Primary Site>

Severity	Type	Site code	Date / Time	System	Component	Message ID	Description
Information	Milestone	PS1	11.06.2014 15:13:30	CM12R2-001....	SMS_CLIENT...	501	This
Information	Milestone	PS1	11.06.2014 15:13:30	CM12R2-001....	SMS_CERTIFI...	501	This
Information	Milestone	PS1	11.06.2014 15:13:30	CM12R2-001....	SMS_AWEBS...	501	This
Information	Milestone	PS1	11.06.2014 15:13:30	CM12R2-001....	SMS_AMT_P...	501	This
Information	Milestone	PS1	11.06.2014 15:13:30	CM12R2-001....	SMS_ALERT...	501	This
Warning	Milestone	PS1	11.06.2014 15:12:50	CM12R2-001....	SMS_INVENT...	2703	Inven
Error	Milestone	PS1	11.06.2014 15:12:50	CM12R2-001....	SMS_INVENT...	2719	Inven
Information	Milestone	PS1	11.06.2014 15:10:32	CM12R2-001....	SMS_INVENT...	4610	Com
Information	Milestone	PS1	11.06.2014 15:10:32	CM12R2-001....	SMS_COMP...	4610	Com
Information	Milestone	PS1	11.06.2014 15:10:32	CM12R2-001....	SMS_INVENT...	4608	Com
Information	Milestone	PS1	11.06.2014 15:10:32	CM12R2-001....	SMS_COMP...	4608	Com

All Status Messages : 586 of 586 messages displayed. 1 selected.

### Status Message Details

Date: 11.06.2014 Type: Milestone  
Time: 15:12:50.180 Severity: Error  
Site code: PS1 Message ID: 2719  
System: CM12R2-001.techlab.n Process ID: 2888  
Source: SMS Server Thread ID: 7016  
Component: SMS\_INVENTORY\_DATA\_LOADER

Description:  
Inventory Data Loader failed to process the file D:\Program Files\Microsoft Configuration Manager\inboxes\auth\dataldr.box\Process\HGCJJZZI.MIF because it is larger than the defined maximum allowable size of 500000.

D:\Program Files\Microsoft Configuration Manager\inboxes\auth\dataldr.box

Name	Date modified	Type	Size
bad	25.03.2014 22:44	File folder	
BADMIFS	11.06.2014 15:10	File folder	
process	11.06.2014 15:16	File folder	
retry	25.03.2014 22:44	File folder	

D:\Program Files\Microsoft Configuration Manager\inboxes\auth\dataldr.box\BADMIFS

Name	Date modified	Type	Size
ExceedSizeLimit	11.06.2014 15:12	File folder	



# Max MIF Size anpassen

- Kann auf dem Site Server in der Registry angepasst werden
- SMS\_Inventory\_Data\_Loader neu starten
- Grösse darf maximal 50 MB betragen, Default Wert ist 5 MB
- [Alte Files bleiben im BADMIF Ordner liegen](#)

The screenshot shows the Windows Registry Editor with the following registry path selected: `Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SMS\Components\SMS_INVENTORY_DATA_LOADER`. The registry value `Max MIF Size` is highlighted, and its current value is `0x004c4b40 (5000000)`. A dialog box titled "Edit DWORD (32-bit) Value" is open, showing the value name "Max MIF Size" and the value data "5000000" in decimal format. The "Base" is set to "Decimal".

Name	Type	Data
(Default)	REG_SZ	(value not set)
Backlog Threading Threshold	REG_DWORD	0x00000100 (256)
Configuration.MOF Crc	REG_DWORD	0x632fc6ba (1664075450)
Configuration.MOF Time (High)	REG_DWORD	0x01cf75ca (30373322)
Configuration.MOF Time (Low)	REG_DWORD	0x4911a3fc (1225892860)
Current MIF Count	REG_DWORD	0x00000000 (0)
Delete Bad MIFs Older Than (Days)	REG_DWORD	0x0000000e (14)
Delete Orphan MIFs Older Than (Days)	REG_DWORD	0x0000000e (14)
Max MIF Size	REG_DWORD	0x004c4b40 (5000000)
Max worker threads	REG_DWORD	0x00000004 (4)
Polling Interval	REG_DWORD	0x0000003c (60)
Processing Flag	REG_DWORD	0x00000000 (0)
Resynchronize Clients	REG_DWORD	0x00000001 (1)
SCF Max Mif Size	REG_DWORD	0x00001388 (5000)



# Daten verwenden

# Daten verwenden



- Die gesammelten Inventardaten können verwendet für:
  - Resource Explorer in der ConfigMgr Konsole
  - Collection Queries
  - Konsolen Queries
  - Reports
  - SQL Abfragen
  - Daten Export in CMDB wie z.B. System Center Service Manager

# Datenbank Informationen



- [Creating Custom Reports By Using Configuration Manager 2007 SQL Views](#)
- [SQL Server Views in System Center 2012 Configuration Manager](#)



# Demo

**Daten verwenden**



# Zusammenfassung



# Zusammenfassung



- Hardware Inventar ist sehr flexibel
- Sehr einfach WMI basierte Daten zu inventarisieren
- Vor Änderungen Backup sicherstellen
- Eigene .mof Erweiterungen Syntaxcheck
- Nicht mehr verwendete Daten wieder löschen
- Inventar Definition muss innerhalb der ConfigMgr Hierarchie konsistent sein

# Danke



Herzlichen Dank

Jürg Koller @juergkoller

[www.trueit.ch](http://www.trueit.ch)

trueIT.

Bewertung der Session: [Configmgr.ch](http://www.configmgr.ch)

- Xing: <https://www.xing.com/net/cmce>
- Facebook: <https://www.facebook.com/groups/411231535670608/>
- LinkedIn: <http://www.linkedin.com>
- Twitter: [https://twitter.com/configmgr\\_ch](https://twitter.com/configmgr_ch) oder Tweet mit Hashtag #CMCE\_CH

Nächster Event: Donnerstag 2. Oktober Digicomp Zürich  
Tagesthema OS-Deployment mit Johan Arwidmark und Mikael Nyström